Vol. 32, No. 2, 2023

Machine
Graphics & Vision

International Journal

Published by
The Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-776 Warsaw, Poland
in cooperation with
The Association for Image Processing, Poland – TPO

An Efficient Pedestrian Attribute Recognition System under Challenging Conditions

Ha X. Nguyen[1,3,*], Dong N. Hoang[3], Tuan A. Tran[2,3], and Tuan M. Dang[3,4,5]

[1] *Research Group Intelligent Robots, Hanoi University of Science and Technology,*
1 Dai Co Viet, Hanoi, Vietnam

[2] *School of Applied Mathematics and Informatics, Hanoi University of Science and Technology,*
1 Dai Co Viet, Hanoi, Vietnam

[3] *CMC Applied Technology Institute, CMC Corporation, 11 Duy Tan, Hanoi, Vietnam*

[4] *CMC University, CMC Corporation, 11 Duy Tan, Hanoi, Vietnam*

[5] *Posts and Telecommunication Institute of Technology,*
KM 10 Nguyen Trai, Ha Dong, Hanoi, Vietnam

[*] *Corresponding author: Ha X. Nguyen (ha.nguyenxuan@hust.edu.vn)*

Abstract. In this work, an efficient pedestrian attribute recognition system is introduced. The system is based on a novel processing pipeline that combines the best-performing attribute extraction model with an efficient attribute filtering algorithm using keypoints of human pose. The attribute extraction models are developed based on several state-of-the-art deep networks via transfer learning techniques, including ResNet50, Swin-transformer, and ConvNeXt. Pre-trained models of these networks are fine-tuned using the Ensemble Pedestrian Attribute Recognition (EPAR) dataset. Several optimization techniques, including the advanced optimizer Adam with Decoupled Weight Decay Regularization (AdamW), Random Erasing (RE), and weighted loss functions, are adopted to solve issues of data unbalancing or challenging conditions like partial and occluded bodies. Experimental evaluations are performed via EPAR that contains 26 993 images of 1477 person IDs, most of which are in challenging conditions. The results show that the ConvNeXt-v2-B outperforms other networks; mean accuracy (mA) reaches 85.57%, and other indices are also the highest. The addition of AdamW or RE can improve accuracy by 1-2%. The use of new loss functions can solve the issue of data unbalancing, in which the accuracy of data-less attributes improves by a maximum of 14% in the best case. Significantly, when the attribute filtering algorithm is applied, the results are dramatically improved, and mA reaches an excellent value of 94.85%. Utilizing the state-of-the-art attribute extraction model with optimization techniques on the large-scale and diverse dataset and attribute filtering has shown a good approach and thus has a high potential for practical applications.

Key words: pedestrian attribute recognition, Deep Learning, vision transformer, security surveillance.

1. Introduction

Pedestrian Attribute Recognition (PAR) is an area of computer vision that tries to assess and comprehend the characteristics of people shown in still images and moving videos. The purpose of the PAR system is to automatically extract and classify features such as clothing style, things that a pedestrian is carrying, and physical factors such as age, gender, and ethnicity from photos and videos of pedestrians. The information PAR

Machine GRAPHICS & VISION 32(2):3–18, 2023. DOI: 10.22630/MGV.2023.32.2.1 .

gleans may be used in various applications, including image retrieval, human-computer interaction, and video surveillance [2].

The variable looks of humans, the existence of occlusions, and the constantly shifting lighting conditions all contribute to the difficulty of measuring PAR. There have been many advances made in PAR [1,6,8,11,15,17,19,34,35,38]. However, there are still a lot of obstacles to overcome, such as dealing with complex and complicated circumstances, increasing recognition accuracy, and lowering the computational processing requirement of PAR systems. In recent years, deep learning strategies have seen widespread use in PAR due to their encouraging results in learning complicated and hierarchical representations of human characteristics. Deep neural networks are incredibly effective in PAR and are now being used by several cutting-edge computer systems [31].

Most of the existing works regarding PAR models have been adopted via off-the-shelf pre-trained deep networks as their backbone network architecture. The pre-trained deep networks are often developed based on large-scale datasets such as ImageNet [4]. Most techniques exploit the ResNet50 [9] as the backbone. Recently, some novel deep networks have been proposed, such as Swin-transformer [21] and ConvNeXt [22, 33]. Although these networks are based on the structure of modern vision transformers with many advancements, they need to be adapted to match the unique characteristics of PAR systems. Thus, novel deep networks should be further developed. For example, in [3], Cheng *et al.* proposed a new model architecture to achieve the best accuracy on the RAP and PA-100K datasets. However, scaling the introduced model up with extensive backbones depends a lot on the size of the embedding words of the textual module. Therefore, in the development of novel backbones there is still much work to do.

Most of the published works use well-known datasets like PA100K [20], PETA [5], RAPv2 [16], MSMT17 [32], and Market1501 [18] as benchmarks for both training and testing phase. Recently, a unified dataset named UPAR to allow generalization experiments for 40 attributes across four PAR datasets PA100K [20], PETA [5], RAPv2 [16], and Market1501 [18] was proposed [28]. These studies concentrated on one particular setting, such as an indoor or outdoor environment, and use relatively large-scale datasets. Nevertheless, there are many restrictions surrounding the generation of real-world surveillance datasets. Challenging conditions including multi-view, unbalanced data distribution, occlusion, low resolution, and poor or varying illumination, are the main issues influencing the final recognized performance. Therefore, augmentation techniques for large-scale and diverse datasets and training optimization facing these issues are current research trends.

Besides the dataset strategies, there have been several adaptions for the transfer learning techniques of the state-of-the-art deep networks to overcome challenging conditions. The issues of data unbalancing can be solved by using suitable loss functions. Many new loss functions, such as (Weighted) Cross Entropy Loss, Contrastive Loss, Center Loss, Triplet Loss, and Focal Loss, have been suggested for the optimization of deep neural networks. Recently, several novel advanced loss functions for PAR have been reported [14, 29, 36]. Also, the Random Erasing (RE) technique [37] is introduced to overcome the issue of the occluded or partial body. The use of novel optimizers, like Adam [12], or Adam with Decoupled Weight Decay Regularization (AdamW) [23], can also improve the model's accuracy. It has been known that the use of advanced techniques for the transfer learning process can improve quite a small amount of recognition accuracy.

Our primary goal in this research is to develop an efficient pedestrian attributes recognition system in challenging conditions scenarios. The proposed system will have not only a remarkably high accuracy but also the robustness against challenging recognition conditions of practical applications.

The main contributions of this work are as follows:

- Proposing a novel processing pipeline combining feature extractions models with keypoints-based attributes filtering for recognizing pedestrians' attributes from challenging condition scenarios.
- Generating an ensemble person attribute recognition dataset based on several state-of-the-art datasets, considering the diverse and challenging scenarios.
- Applying the transfer learning technique selected from among various deep network architectures, including ResNet50, Swin-transformer, and ConvNeXt, to obtain the best-performing model.
- Tuning the best-performing model by adapting training optimization techniques like AdamW, Random Erasing, and advanced loss functions to overcome issues of data unbalancing and partial or occluded body.
- Systematically evaluating experimental results on a challenging dataset and producing system design instructions for practical applications.

In the following section, detailed descriptions of the proposed system are presented.

2. Proposed system

The overall processing pipeline of the proposed system is shown in Fig. 1. Unlike in the most of the published works [7, 27], which have concentrated only on models for feature extraction to achieve higher accuracy on typical datasets, our system is adapted by using the keypoint concept as an additional information channel for the prediction. There are three main modules, each responsible for a specific task. Image frames from camera streams are pushed into the human body, pose detection and tracking module. As a result, the bounding box and corresponding pose of detected persons are achieved and consequently tracked. Images and poses of tracked persons are then pushed into the attributes recognition module for attributes prediction and the color recognition module for body color prediction. The use of keypoints of the pose has the advantage

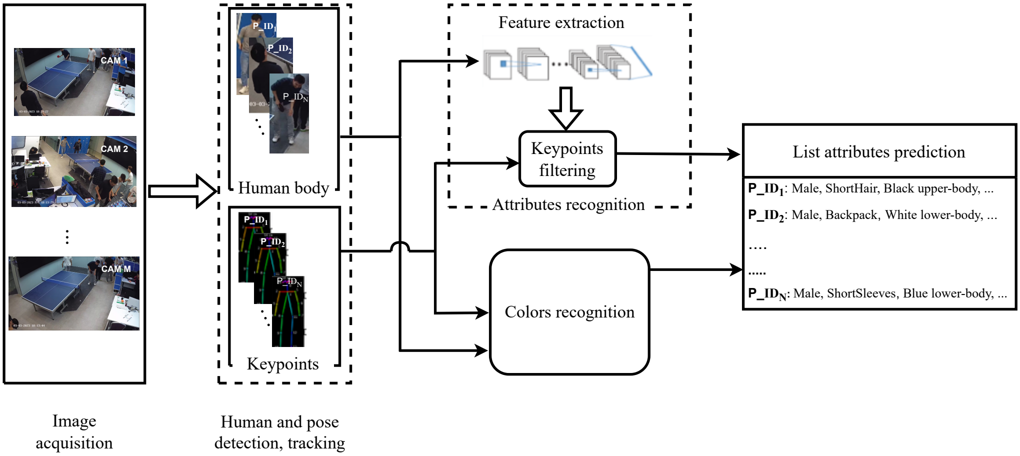

CAM 1
CAM 2
CAM M
P_ID1
P_ID2
P_IDN
Human body
Keypoints
Feature extraction
Keypoints filtering
Attributes recognition
Colors recognition
List attributes prediction
P_ID1: Male, ShortHair, Black upper-body, ...
P_ID2: Male, Backpack, White lower-body, ...
....
.....
P_IDN: Male, ShortSleeves, Blue lower-body, ...
Image acquisition
Human and pose detection, tracking

Fig. 1. Overview of our processing pipeline.

of improving the accuracy and robustness of the system in challenging conditions such as occluded or partial detecting bodies and viewpoint-variant bodies. In the following, details of each module are described.

2.1. Ensemble Pedestrian Attribute Recognition dataset

The ensemble dataset (EPAR) was created by collecting and processing image data from the four publicly accessible datasets, including PA100K [20], MSMT17 [32], RAPv2 [16], and Market1501 [18]. We have labeled the data in the test set of the datasets mentioned above using the Pseudo-Label approach [13] in addition to the data in the training set. We have set the classification threshold to 0.9 and utilized the data labeling tool to accurately align the labels using pre-trained weights previously trained on the RAPv2 dataset. We then have got the EPAR dataset by joining them all. The ensemble of these datasets has the benefit that the EPAR has a massive variation in situations, races, and qualities. Images of EPAR are very diverse and cover many challenging conditions, including significant variations in pose, lighting conditions, background, and occlusions. For practical applications in security surveillance scenarios, 13 typical attributes of the human body outfits are chosen to label, including “Male”, “Female”, “Adult”, “Children”, “Long-Hair”, “ShortHair”, “Hat”, “Long-Sleeves”, “Short-Sleeves”, “Trouser-Jeans”, “Skirt”, “Short”, and “Backpack”. Since images of PA100k, MSMT17, and Market1510 are not assigned to person IDs, the re-identification method from [26] was used for the person

Tab. 1. Statistics portion of each dataset contributing to the EPAR.

Dataset from	No. IDs	No. Imgs	Properties
RAPv2 [16]	2589	30 315	annotated with viewpoints, occlusions, and body parts information by multiple cameras in real-world environments
PA100K [20]	3556	36 920	images are blurry due to the relatively low resolution and the positive ratio of each binary attribute is low
Market1501 [18]	700	12 668	images in this dataset exhibit significant variations in pose, lighting conditions, background, and clothing
MSMT17 [32]	600	10 101	images are captured in morning, noon, and afternoon in campus
EPAR (**ours**)	7445	90 004	all of above properties

ID register. The EPAR dataset contains 90 004 images of 7445 person IDs, with annotations for 13 binary attributes. Tab. 1 lists the portion and properties of each referring dataset contributing to the EPAR. EPAR is divided into 60%, 20%, and 20% for the test, train, and validation set. Fig. 2 shows the frequency of appearance of each attribute in all 90 004 images of EPAR. It is clear that the dataset EPAR has a big issue of data unbalancing, where some attributes, for example, “Children” and “Short” have a low frequency of appearance, making EPAR more challenging.

2.2. Human body and pose detection and tracking

Video frames from cameras are processed using Yolov7-Pose [24, 30]. The outputs are the bounding boxes and keypoints of the pose of detected persons. The use of Yolov7-Pose has the benefit that this model can simultaneously detect persons and estimate their poses, allowing us to save computational hardware resources. Also, this model outperforms others in accuracy and robustness in challenging detection scenarios. Since the images of a detected person can appear in several video frames of different cameras, the re-identification method [26] is used to match these images together and assign them to a tracking ID. Consequently, images and poses of each tracking ID with different viewpoints from different cameras are continued processing in the following steps. Since attributes of a person can be predicted from viewpoint-different images, the loss of information regarding the viewpoint can be maximally reduced.

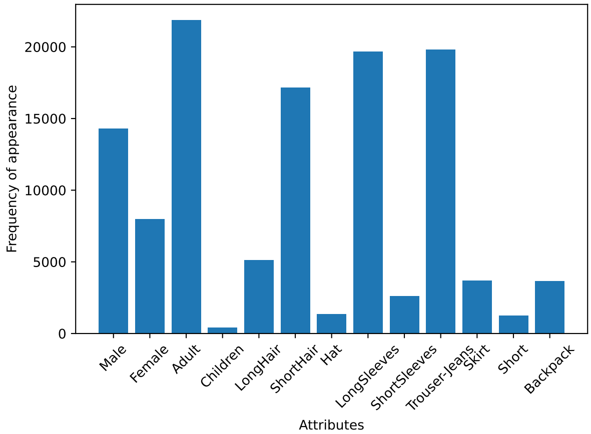


Fig. 2. Statistics of frequency of appearance of each attribute in EPAR.

2.3. Human attributes extraction models

Images of tracked persons are continued processing to extract corresponding attributes via an attributes extraction model. There are several approaches using convolution neural networks [9], visual transformer [21], or a hybrid of those like ConvNeXt [22, 33]. The ConvNeXt is considered to be state-of-the-art in accuracy. Thus, in this work, transfer learning techniques are applied to the ConvNeXt-v2-B [33] to achieve the attributes extraction model. Also, to evaluate the performance of the ConvNeXt-v2-B, other backbones including ResNet50 [9], Swin-Transformer [21], and ConvNeXt-v1-B [22], are fine-tuned. Consequently, the performance and accuracy of these models are compared and analyzed. All fine-tuning processes are performed using the EPAR dataset.

For the best performance of the developed model, the transfer learning technique is adapted to make several improvements. First, instead of using the conventional optimizer like Adam [12], the newly introduced one, namely the AdamW [23], was used. AdamW brings an improvement by incorporating weight decay into the Adam algorithm. In the original Adam algorithm, weight decay is usually applied by adding a term to the gradient of the parameters. However, this can lead to the loss of certain important properties of the Adam algorithm. In AdamW, weight decay is computed differently by directly applying it to the weights instead of modifying the gradient. This helps preserve the invariance properties of gradient scale and enhances the stability of the training process. Second, besides typical data augmentation techniques of the baseline, like random flipping, random gray-scale, and cropping, the Random Erasing [37] was additionally used. This helped the training process to stay balanced and to ensure the diversity of the training data. The third improvement was regarding the loss function. A suitable loss function is expected to solve data unbalancing, as shown in Fig. 2. The loss function of the baseline training method is based on the cross-entropy formulated as

$$\mathcal{L} = -\sum_{j=1}^{M} w_j \left(y_{ij} \log\left(p_{ij}\right) + \left(1 - y_{ij}\right)\left(1 - \log\left(p_{ij}\right)\right)\right) , \tag{1}$$

where w_j is the attribute weight function of j^{th} attribute; p_{ij} is the output of the classifier layer. The three most popular weight function methods called $L1$ [14], $L2$ [29], and $L3$ [36], were experimented and evaluated. The description of each weight function is detailed in Tab. 2

2.4. Keypoints-Based Attributes Filtering

In fact, many recognition situations exist where images of the human body are occluded or partial, caused by obstacles or view-point variances. If only the attributes extraction model is used, the model always produces predictions for all attributes, even in cases where some of these attributes are occluded or do not appear in images. This leads to false predictions and thus reduces the system's accuracy. This issue is solved by using the keypoints as additional information for the attributes prediction. The pose's keypoints and confidence scores will let us know which body part in the images is occluded or

Tab. 2. Types of weight function used in the transfer learning process.

Method	Weight function
$L1$ by Li *et. al.* [14]	$w_j = \begin{cases} e^{1-r_j} & \text{when } y_{ij} = 1\,, \\ e^{r_j} & \text{when } y_{ij} = 0\,; \end{cases}$
$L2$ by Tan *et. al.* [29]	$w_j = \begin{cases} \sqrt{\dfrac{1}{2r_j}} & \text{when } y_{ij} = 1\,, \\ \sqrt{\dfrac{1}{2(1-r_j)}} & \text{when } y_{ij} = 0\,; \end{cases}$
$L3$ by Zhang *et. al.* [36]	$w_j = \begin{cases} \dfrac{\frac{1}{r_j^{\alpha}}}{\frac{1}{r_j^{\alpha}} + \frac{1}{(1-r_j)^{\alpha}}} & \text{when } y_{ij} = 1\,, \\ \dfrac{\frac{1}{(1-r_j)^{\alpha}}}{\frac{1}{r_j^{\alpha}} + \frac{1}{(1-r_j)^{\alpha}}} & \text{when } y_{ij} = 0\,; \end{cases}$

where α is a hyper-parameter to adjust the weight between positive ratio and negative ratio, and r_j is the positive sample ratio of j-th attribute in the training set partial. The attributes belonging to the occluded or partial parts will not be predicted. This will help the system avoid false predictions. The pose inferred by the Yolov7-Pose has 17 keypoints arranged to three parts of the human body, including the head, upper body, and lower body. An algorithm for attributes filtering was developed as illustrated in Algorithm 1. For each part of the human body, if more than half of the number of keypoints in part have a confidence score smaller than a threshold, this part will be concluded to be occluded or partial.

2.5. Keypoints-based body's color recognition

The keypoints were also used for predicting the color of the upper and lower body parts. An algorithm was proposed for this prediction. For the upper-body parts, a pair of keypoints, including the "right shoulder" and "left hip", is used to calculate the center of the upper-body part, which is the middle point of this pair of keypoints. Similarly, the "left hip" and "left ankle" are used for the lower-body part. The color of each interesting body part is calculated from the average of the color value of all pixels in a square with dimensions of 20 × 20 at the center of the part. The color of each pixel is computed using functions of OpenCV library [25] in the RGB mode.

3. Results and discussion

An evaluation dataset was created to evaluate the proposed system's accuracy and robustness. The dataset covered 17 677 images of the test dataset of EPAR and 9252 augmented images. The augmented images in the evaluating dataset were to add challenging conditions for the evaluation. As illustrated in Fig. 3, challenging situations in real-life applications, including partial or occluded bodies, images with low resolution, or inadequate lighting conditions, were considered. The evaluation dataset covered 26 933

Algorithm 1: Attributes Filtering.

```
Input: logit keypoints $p$, keypoints confidence score $k$, threshold $T$.
Output: filter logit $p^*$.
parts = $k[a : b]$
count = 0
for part in parts do
    if $part < T$ then
        count = count + 1
end
if count > (number of keypoints in parts)/2 then
    $p[c : d] = 0$
$p^* = p$
```

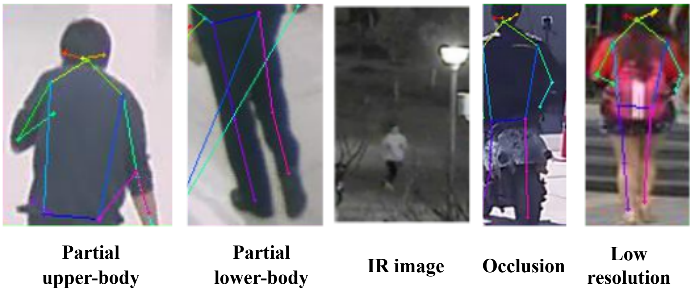


Fig. 3. Illustration of images in challenging conditions in the evaluating dataset.

Tab. 3. Evaluation results of the proposed system.

Method	Backbone	mA	Acc	Prec	Recall	$F1$
Baseline [10]	ResNet50 [9]	82.05 ± 0.14	77.97 ± 0.23	80.78 ± 0.14	93.75 ± 0.28	86.78 ± 0.20
	Swin-S [21]	84.61 ± 0.19	79.52 ± 0.92	81.59 ± 0.69	95.33 ± 0.43	87.92 ± 0.59
	ConvNeXt-v1-B [22]	85.04 ± 0.39	79.85 ± 0.57	81.95 ± 0.47	95.41 ± 0.33	88.08 ± 0.35
	ConvNeXt-v2-B [33]	85.57 ± 0.28	80.04 ± 0.54	82.19 ± 0.38	95.88 ± 0.19	88.08 ± 0.29
+ AdamW [23]	ConvNeXt-v2-B	85.67 ± 0.02	81.43 ± 0.07	83.09 ± 0.07	96.06 ± 0.06	89.11 ± 0.07
+ Random Erasing [37]	ConvNeXt-v2-B	85.76 ± 0.05	81.63 ± 0.06	83.99 ± 0.08	$\mathbf{96.15 \pm 0.13}$	89.15 ± 0.05
+ Keypoints (**ours**)	ConvNeXt-v2-B	$\mathbf{94.28 \pm 0.01}$	$\mathbf{91.57 \pm 0.07}$	$\mathbf{94.43 \pm 0.07}$	93.65 ± 0.07	$\mathbf{93.61 \pm 0.06}$

images of 1477 person IDs, including 30% partial, 15% occlusion, 23% outdoor, 7.5% indoor, 10% gray-scale, and 15% normal-quality images. For metrics, four instance-level metrics and one attribute-level (label-level) measure were used to evaluate the model's performance based on the literature [20]. Accuracy (Acc), Precision (Prec), Recall (Recall), and $F1$ are used as metrics at the instance level. Mean accuracy (mA) is used as an attribute-level statistic since it focuses on the recognition accuracy of a particular attribute. The instance-level metric is used when we want to evaluate the model's overall performance in terms of its ability to predict entire instances. This approach is particularly relevant in regression tasks or when the primary concern is the quality of instance-level predictions rather than individual label predictions. The attribute-based criteria are used when we want to assess the model's performance with a focus on individual labels or classes. These criteria are helpful for understanding the model's precision, recall, and accuracy for each category, which can be especially useful when dealing with imbalanced datasets.

The evaluation results of the system are shown in Tab. 3. In the second-row cluster, results with the baseline method with backbone ResNet50, Swin-S, ConvNeXt-v1-B, and ConvNeXt-v2-B are compared. The third-row cluster shows the results of our improvements based on the ConvNeXt-v2-B with several adaptions using: i) AdamW; ii) AdamW and RE, and iii) AdamW and RE and keypoints, respectively. Each result in the table is an average of ten repeated tests accompanied by the deviation. It is seen that, with the baseline method, if different backbones are used, the mA and other parameters are slightly changed. The ConvNext-v2-B, in which the mA reaches 85.57%, outperforms the other networks. When other optimizations, like AdamW or AdamW and RE, are used, the accuracy is also increased, but the increase is tiny at 0.1-0.2%. With our proposed method using keypoints as the filter, the mA increases significantly from 85.76% to 94.28%. Similarly, the Acc, Prec, and $F1$ also increase. Only the Recall decreases from 96.15% to 93.65%. This issue can be explained by that the pose detection model used in this work is just a pre-trained version of Yolov7-Pose without any modification. Thus, this still has failures in some detection scenarios. Consequently, keypoints-based filtering can eliminate some attributes which appear in images leading to a reduction of the True Positive Rate and the Recall.

Influences of the weight functions on the system's accuracy and data unbalancing are also evaluated. Tab. 4 shows the evaluation results of the system on different weight functions listed in Tab. 2. It can be seen that the weight function method L3 [36] slightly outperforms others in mA and Prec. This improvement is explained that, as stated by Zhang *et al.* [36], the $L3$ weight function (with $\alpha = 1$) helps re-weight for attributes with low frequency of appearance. As a result, $L3$ increases the True Negative Rate and thus reduces the incorrect recognition probability for attributes. That means the mA is improved. As Zhang *et al.* [36] mentioned in their paper, when setting $\alpha = 1$, $L3$ can help attributes-balanced re-weight, which increases mA. With $\alpha = 0$, the loss function transforms into conventional CE loss to help instance-balanced re-weight, which increases instance-level indexes such as Accuracy, Precision, Recall, and $F1$. It is inferred that the weight function can help improve the system's accuracy, especially in unbalanced data cases. However, the improvement is not much in the 0.1% (94.85% vs. 94.04%).

Detailed analysis of the True Positive Rate results for attributes with a low appearance frequency in the EPAR is shown in Fig. 4. The results of considering weight functions are presented. It is seen that the use of new weight functions improves the True Positive Rate significantly. For example, with the "Children" attribute, when the $L3$ is used, the True Positive Rate increases to 14%, compared to the conventional loss function. Similar trends for other considered attributes are also confirmed. These results have proven the efficient role of weight functions in solving the issue of data unbalancing.

Tab. 5 shows evaluated results of the computational requirement of the system. This work utilized a server with NVIDIA Tesla V100 32GB GPUs equipped with PyTorch 1.7.1 and CUDA 10.1 for the training and evaluation processes. It can be seen that although the ConvNeXt and Swin-S back-bone have more parameters as well as computing complexity requirements, the processing time is three times less than that of ResNet50 (210ms vs. 70ms). This can be clarified that the ConvNeXt or Swin-S have a new inverted bottlenecks design which helps reduce the computing complexity and required

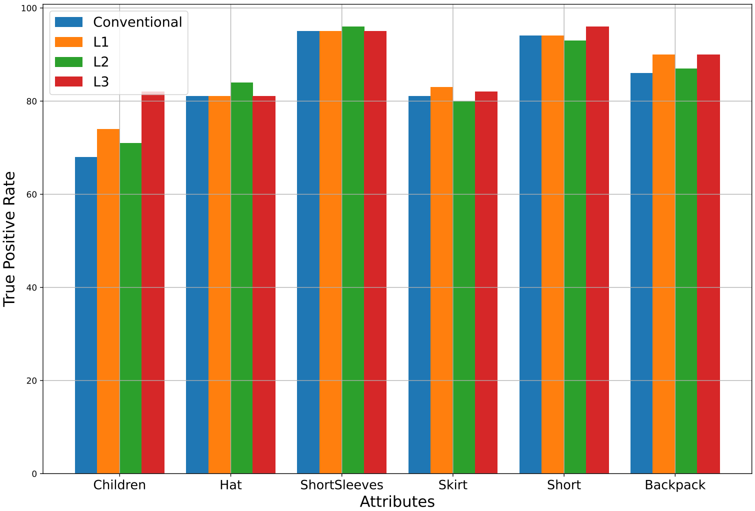


Fig. 4. True positive rate of attributes with unbalanced data.

Tab. 4. Evaluation results of the system on different weight functions.

Method	mA	Acc	Prec	Recall	$F1$
Conventional	94.41 ± 0.04	$\mathbf{91.64 \pm 0.05}$	$\mathbf{94.55 \pm 0.04}$	93.62 ± 0.11	$\mathbf{93.66 \pm 0.04}$
L1 [14]	94.28 ± 0.01	91.57 ± 0.07	94.43 ± 0.07	$\mathbf{93.65 \pm 0.07}$	93.61 ± 0.06
L2 [29]	94.04 ± 0.11	91.47 ± 0.03	94.12 ± 0.04	93.54 ± 0.06	93.39 ± 0.06
L3 ($\alpha = 1$) [36]	$\mathbf{94.85 \pm 0.04}$	91.22 ± 0.09	94.51 ± 0.03	93.36 ± 0.03	93.53 ± 0.02

inference time. In addition, the ConvNeXt architecture uses fewer action functions and normalization layers than ResNet, reducing the computational requirement.

4. Conclusions and outlook

An efficient human attributes recognition system has been successfully presented in this work. Efforts to use a state-of-the-art backbone like Swin-S or ConvNeXt can improve the system's accuracy. However, this improvement is still slight in the range of 1-2%. There is quite a similar trend when optimization techniques like AdamW or RE are used.

Tab. 5. Evaluation results of computational requirements for different models.

Backbone	FLOPs [G]	No. of params [M]	Inference time [ms]
ResNet50 [9]	4.12	23.53	210
Swin-S [21]	8.52	48.85	70
ConvNeXt-v1-B [22]	15.36	87.58	70
ConvNeXt-v2-B [33]	15.36	87.71	80

The use of advanced weight functions can also solve the issue of data unbalancing and thus significantly improve the accuracy. The ConvNeXt-v2-B should be used since it has shown the best accuracy and computational efficiency. Any approaches trying to improve the attribute recognition model are hard to produce remarkable results, especially for practical applications with many challenging conditions. The post-processing technique using keypoints of the pose for filtering attributes proposed in this work is efficient, which improves the system's accuracy significantly. Also, the use of keypoints can make the system robust against the challenging conditions of real-life applications such as images containing occluded or partially visible human body parts. For the best practice system, we should combine two directions. On the one hand, we improve the attribute recognition model with a state-of-the-art backbone, a diverse and large dataset, and optimization training techniques, as shown in this paper. On the other hand, the post-processing technique presented in this work should be added.

Although the keypoints concept has proven to have significant results for the system, it should be further improved. First, the accuracy of the pose detection model can be improved by fine-tuning the Yolov7-Pose on a more diverse and challenging dataset. Second, the algorithm that uses keypoints to filter the attributes must be thoroughly evaluated, especially in real-life and in challenging scenarios. For practical applications, a lightweight model based on the processing pipeline proposed in this work is planned to be developed so that it can be deployed on limited hardware resources of edge computing devices.

References

[1] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *Proc. 2011 Int. Conf. Computer Vision (ICCV)*, pages 1543–1550, Barcelona, Spain, 6-13 Nov 2011. IEEE. doi:10.1109/ICCV.2011.6126413.

[2] W.-C. Chen, X.-Y. Yu, and L.-L. Ou. Pedestrian attribute recognition in video surveillance scenarios based on view-attribute attention localization. *Machine Intelligence Research*, 19(2):153–168, 2022. doi:10.1007/s11633-022-1321-8.

[3] X. Cheng, M. Jia, Q. Wang, and J. Zhang. A simple visual-textual baseline for pedestrian attribute recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):6994–7004, 2022. doi:10.1109/TCSVT.2022.3178144.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. 2009 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL, USA, 20-25 Jun 2009. doi:10.1109/CVPR.2009.5206848.

[5] Y. Deng, P. Luo, C. C. Loy, and X. Tang. Pedestrian attribute recognition at far distance. In *Proc. 22nd ACM Int. Conf. Multimedia (MM'14)*, ACM Conferences, pages 789–792, Orlando, FL, USA, 3-7 Nov 2014. doi:10.1145/2647868.2654966.

[6] A. Diba, A. M. Pazandeh, H. Pirsiavash, and L. Van Gool. Deepcamp: Deep convolutional action & attribute mid-level patterns. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3557–3565, Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.387.

[7] H. Galiyawala, M. S. Raval, and M. Patel. Person retrieval in surveillance videos using attribute recognition. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2022. doi:10.1007/s12652-022-03891-0.

[8] G. Gkioxari, R. Girshick, and J. Malik. Actions and attributes from wholes and parts. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 2470–2478, Santiago, Chile, 13-16 Dec 2015. doi:10.1109/ICCV.2015.284.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.90.

[10] J. Jia, H. Huang, X. Chen, and K. Huang. Rethinking of pedestrian attribute recognition: A reliable evaluation under zero-shot pedestrian identity setting. *arXiv*, 2021. arXiv:2107.03576. doi:10.48550/arXiv.2107.03576.

[11] J. Joo, S. Wang, and S.-C. Zhu. Human attribute recognition by rich appearance dictionary. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 721–728, Sydney, Australia, 1-8 Dec 2013. doi:10.1109/ICCV.2013.95.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. arXiv:1412.6980. doi:10.48550/arXiv.1412.6980.

[13] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proc. Workshop on Challenges in Representation Learning (WREPL), part of Int. Conf. Machine Learning (ICML)*, page 896. Atlanta, GE, USA, 16-21 Jun 2013.

[14] D. Li, X. Chen, and K. Huang. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In *Proc. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 111–115, Kuala Lumpur, Malaysia, 3-6 Nov 2015. IEEE. doi:10.1109/ACPR.2015.7486476.

[15] D. Li, X. Chen, Z. Zhang, and K. Huang. Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. In *Proc. 2018 IEEE Int. Conf. Multimedia and Expo (ICME)*, pages 1–6, San Diego, CA, USA, 23-27 Jul 2018. doi:10.1109/ICME.2018.8486604.

[16] D. Li, Z. Zhang, X. Chen, and K. Huang. A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. *IEEE Transactions on Image Processing*, 28(4):1575–1590, 2018. doi:10.1109/TIP.2018.2878349.

[17] Y. Li, C. Huang, C. C. Loy, and X. Tang. Human attribute recognition by deep hierarchical contexts. In *Computer Vision, Proc. 14th European Conf. Computer Vision (ECCV 2016)*, volume 9910 Part VI of *Lecture Notes in Computer Science*, pages 684–700, Amsterdam, The Netherlands, 11-14 Oct 2016. Springer. doi:10.1007/978-3-319-46466-4_41.

[18] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, and Y. Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, 2019. doi:10.1016/j.patcog.2019.06.006.

[19] P. Liu, X. Liu, J. Yan, and J. Shao. Localization guided learning for pedestrian attribute recognition. In *Proc. British Machine Vision Conference (BMVC 2018)*, Northumbria, UK, 3-6 Sep 2018. BMVA Press. Accessible also as arXiv:1808.09102. `https://bmva-archive.org.uk/bmvc/2018/contents/papers/0573.pdf`.

[20] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 350–359, Venice, Italy, 22-29 Oct 2017. doi:10.1109/ICCV.2017.46.

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, pages 10012–10022, Montreal, QC, Canada, 10-17 Oct 2021. doi:10.1109/ICCV48922.2021.00986.

[22] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A ConvNet for the 2020s. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, New Orleans, LA, USA, 18-24 Jun 2022. doi:10.1109/CVPR52688.2022.01167.

[23] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proc. 7th Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, 6-9 May 2019. `https://openreview.net/forum?id=Bkg6RiCqY7`.

[24] D. Maji, S. Nagori, M. Mathew, and D. Poddar. YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2636–2645, New Orleans, LA, USA, 19-20 Jun 2022. doi:10.1109/CVPRW56347.2022.00297.

[25] OpenCV Team. OpenCV, 2022. `https://opencv.org`. [Accessed 15 Jan 2022].

[26] H. X. Nguyen, D. N. Hoang, T. V. Nguyen, T. M. Dang, A. D. Pham, and D.-T. Nguyen. Person re-identification from multiple surveillance cameras combining face and body feature matching. *Modern Physics Letters B*, 37(19):2340031, 2023. doi:10.1142/S0217984923400316.

[27] S. Sakib, K. Deb, P. K. Dhar, and O.-J. Kwon. A framework for pedestrian attribute recognition using deep learning. *Applied Sciences*, 12(2):622, 2022. doi:10.3390/app12020622.

[28] A. Specker, M. Cormier, and J. Beyerer. UPAR: Unified Pedestrian Attribute Recognition and person retrieval. In *Proc. 2023 IEEE/CVF Winter Conf. Applications of Computer Vision (WACV)*, pages 981–990, Los Alamitos, CA, USA, 3-7 Jan 2023. doi:10.1109/WACV56688.2023.00104.

[29] Z. Tan, Y. Yang, J. Wan, G. Guo, and S. Z. Li. Relation-aware pedestrian attribute recognition with graph convolutional networks. In *Proc. AAAI Conf. Artificial Intelligence*, volume 34 of *AAAI-20 Technical Tracks 7*, pages 12055–12062, New York, NY, USA, 7-12 Feb 2020. AAAI Press. doi:10.1609/aaai.v34i07.6883.

[30] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475, Vancouver, Canada, 18-22 Jun 2023. Accessible also as arXiv:2207.02696. `https://openaccess.thecvf.com/content/CVPR2023/html/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.html`.

[31] X. Wang, S. Zheng, R. Yang, A. Zheng, Z. Chen, J. Tang, and B. Luo. Pedestrian attribute recognition: A survey. *Pattern Recognition*, 121:108220, 2022. doi:10.1016/j.patcog.2021.108220.

[32] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer GAN to bridge domain gap for person re-identification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 79–88, Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00016.

[33] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Vancouver, Canada, 18-22 Jun 2023. Accessible also as arXiv:2301.00808. `https://openaccess.thecvf.com/content/CVPR2023/html/Woo_ConvNeXt_V2_Co-Designing_and_Scaling_ConvNets_With_Masked_Autoencoders_CVPR_2023_paper.html`.

[34] L. Yang, L. Zhu, Y. Wei, S. Liang, and P. Tan. Attribute recognition from adaptive parts. *arXiv*, 2016. arXiv:1607.01437. doi:10.48550/arXiv.1607.01437.

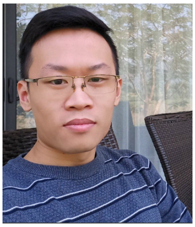
[35] N. Zhang, M. Paluri, M'A. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose Aligned Networks for Deep Attribute modeling. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1644, Columbus, OH, USA, 23-28 Jun 2014. doi:10.1109/CVPR.2014.212.

[36] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2361–2370, Nashville, TN, USA, 20-25 Jun 2021. doi:10.1109/CVPR46437.2021.00239.

[37] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proc. AAAI Conf. Artificial Intelligence*, volume 34 of *AAAI-20 Technical Tracks 7*, pages 13001–13008, New York, NY, USA, 7-12 Feb 2020. AAAI Press. doi:10.1609/aaai.v34i07.7000.

[38] J. Zhu, S. Liao, D. Yi, Z. Lei, and S. Z. Li. Multi-label CNN based pedestrian attribute learning for soft biometrics. In *Proc. 2015 Int. Conf. Biometrics (ICB)*, pages 535–540, Phuket, Thailand, 19-22 May 2015. IEEE. doi:10.1109/ICB.2015.7139070.

Ha X. Nguyen received his Ph.D. degree in computing science and microrobotics from the University of Oldenburg, Germany, in 2014. He is now working as a lecturer for intelligent robotics at Hanoi University of Science and Technology. He also serves as a consultant expert at the IoT/Smart-Devices Laboratory at the CMC Applied Technology Institute, CMC Corporation. His research interests cover intelligent robots, micro-robotics, and computer vision.



Dong N. Hoang has five years of research experience in machine learning, computer vision, and building security surveillance systems. His research fields include facial, human, vehicle, and building scalable pipeline architectures. He received his B.Sc. degree in 2015 from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam. Now he is working as the deputy head of the IoT/Smart Device Laboratory at the CMC Applied Technology Institute, CMC Corporation, Hanoi, Vietnam.

Tuan A. Tran has three years of research experience in machine learning, computer vision, and optimization. He received his B.Sc. degree in 2022 from School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, Vietnam. Now, he is working as researcher at CMC Applied Technology Institute, CMC Corporation, Hanoi, Vietnam.

Tuan M. Dang received his Ph.D. degree in mathematics at Academy of Military Science and Technology, Ministry of Defense, Vietnam. He is now working as a lecturer for computer science at Posts and Telecommunication Institute of Technology, Vietnam, and researcher at CMC Applied Technology Institute, CMC Corporation, Hanoi, Vietnam. His research interests cover cryptography, blockchain and artificial intelligence.

Use of Virtual Reality to Facilitate Engineer Training in the Aerospace Industry

Andrzej Paszkiewicz[1,*], Mateusz Salach[1], Dawid Wydrzyński[2],
Joanna Woźniak[3], Grzegorz Budzik[2], Marek Bolanowski[1],
Maria Ganzha[4], Marcin Paprzycki[4], Norbert Cierpicki[5]

[1]*Department of Complex Systems, Rzeszow University of Technology, Rzeszów, Poland*

[2]*Department of Machine Design, Rzeszow University of Technology, Rzeszów, Poland*

[3]*Department of Management Systems and Logistics,
Rzeszow University of Technology, Rzeszów, Poland*

[4]*Systems Research Institute, Polish Academy of Sciences, Warszawa, Poland*

[5]*The Faculty of Electrical and Computer Engineering,
Rzeszow University of Technology, Rzeszów, Poland*

[*]*Corresponding author: Andrzej Paszkiewicz (andrzejp@prz.edu.pl)*

Abstract. This work concerns automation of the training process, using modern information technologies, including virtual reality (VR). The starting point is an observation that automotive and aerospace industries require effective methods of preparation of engineering personnel. In this context, the technological process of preparing operations of a CNC numerical machine has been extracted. On this basis, a dedicated virtual reality environment, simulating manufacturing of a selected aircraft landing gear component, was created. For a comprehensive analysis of the pros and cons of the proposed approach, four forms of training, involving a physical CNC machine, a physical simulator, a software simulator, and the developed VR environment were instantiated. The features of each training form were analysed in terms of their potential for industrial applications. A survey, using the Net Promoter Score method, was also conducted among a target group of engineers, regarding the potential of use of each training form. As a result, the advantages and disadvantages of all four training forms were captured. They can be used as criteria for selecting the most effective training form.

Key words: virtual reality, industry 4.0, CNC machines, aviation industry, quality management.

1. Introduction

Virtual reality (VR) technology is relatively young, but it is rapidly developing. It involves creation of multimedia representations of objects, spaces, and events. Thus far, it has been used mostly in the entertainment industry [8,16]. However, it is finding its way also to education [13,14], culture [11,17], or industry [1,5], among others. With the increasing availability of VR devices, i.e. goggles/glasses, the space of its potential applicability is systematically increasing [19]. In this work, of particular interest is the use of VR in education, in the manufacturing industry in particular [15]. This mainly refers to employee training, to support familiarization with the tasks performed on the job.

Traditional training consists of two parts: theoretical and practical. In the industrial settings, the practical training can be carried on special stations, called trainers,

Machine GRAPHICS & VISION 32(2):19–44, 2023. DOI: 10.22630/MGV.2023.32.2.2 .

which are (usually, not fully functional) production stations. They are used to emulate processes occurring on production lines. However, the final (advanced) training occurs using the actual equipment. Only then, it is possible to realistically verify the level of acquired skills, and familiarize the trainee with the specific details of a given process. Such traditional approach, in addition to obvious benefits, has significant drawbacks and limitations. First, the number of employees that can be trained simultaneously is restricted. Moreover, very often, such training involves removal of a device from the manufacturing process. Obviously, even when a partially functional machine is used as a trainer, this has associated costs. Furthermore, high costs are associated with consumption of materials used during training. Finally, use of actual machines, prevents testing critical scenarios, e.g. equipment malfunction, damages in the work-station area, operator errors, etc.

Therefore, use of VR technology in training seems very attractive. Specifically, VR (1) allows multiple repetition of test scenarios, without additional costs, (2) supports checking and verifying behaviour of the operator in various, also extreme, non-standard situations, and (3) allows simulating processes using diverse materials and processing tools. In this context, the advantages of VR are explored in the actual implementation of training of operation and programming of CNC machine tools, with the use case based on the needs of the aerospace industry. Here, it should be noted that in this industry, the precision manufacturing, using highly specialized tools, is a norm. Consequently, the equipment, and the tooling that comes with it, are particularly expensive. Moreover, training of CNC machine operators very often requires taking equipment out of production for an extended period (as it is too costly to purchase machine(s) just for teaching). Finally, observe also that, currently, employee mobility is high and will be higher in the coming years. As a result, it will be necessary to increase the frequency of dedicated training.

In this context it is easy to realize that VR can be, relatively easily, adapted to represent required training simulations. This makes it easy to train any person – new to CNC, or already skilled – to practice (repeatedly) the designed training path. However, in some areas, e.g. in dealing with highly accurate tools, materials, and products, VR can pose challenges. Since VR facilitates operations on 3D models, accuracy may be difficult to achieve for objects with complex structures, with edges and angles. Hence, the question can be asked, whether it is reasonable to apply VR to every field and process?

The need to address this question for aerospace industry is of particular importance in the Subcarpathian region of Poland, where an aerospace industry hub is located. Moreover, Rzeszow University of Technology trains many specialists for this industry. In this context, work aimed at developing and testing VR solutions for CNC machine operation training was undertaken. Accordingly, the main purpose of this contribution is to analyse usefulness of modern approaches to training CNC machine operators. In this context, an aircraft landing gear beam was chosen, as an example of a machined part.

Moreover, in addition to VR, training on an actual machine, training on machine simulator and using software simulator have been instantiated, to compare their effectiveness. To assess pros and cons of each approach, the NPS indicator was used. Finally, a set of recommendations concerning use of each training environment has been formulated.

2. Related work

VR technology is widely used in aircraft crew training and in ground handling. The scope and number of training sessions, directly affect the level of safety, and this aspect of air transport is the most important. Therefore, any innovation to streamline and raise efficiency and quality of training is in demand, and use of VR technology opens up new possibilities. However, because it is a relatively new technology, it still requires a lot of research, and testing, to achieve the needed level of efficiency.

In the aviation field, the first simulation systems were used to train pilots in the military. Studies showed that simulators reduced the flight training time, needed to achieve the required skills and competencies [15]. One of the important benefits of simulators is a very rapid feedback that allows analysis and elimination of undesirable behaviours. Here, a simple VR-based flight simulator that gives the perspective of flying in the air and allows interaction with the computer-generated environment, was presented in [18]. However, note that as early as 1993, potential role of VR-based flight simulators in training of civilian pilots was considered [20]. Moreover, the key features of virtual reality for industrial simulations were discussed. Interesting research, concerning training aircraft pilots, was presented in [3]. This work points to the need for appropriate adaptation of teaching materials and content to the technical requirements. Analysing literature related to pilot training, can be concluded that solutions based on VR not only reduce costs for flight schools but also provide a faster and more efficient learning process.

Another area of VR application in the aviation industry is the training of aircraft maintenance workers. Research reported in [9] discusses examples of VR and AR applications in the aviation industry, such as a VR training system for a Boeing 737. Specifically, VR was used to practice thrust reversal. Another example is a system capturing troubleshooting procedures for Airbus A320/A330 aircraft components [7]. Separately, VR is used to train employees of manufacturing companies. For instance, the Boeing Company, used VR and reduced employee preparation to just few weeks [4].

Of course, VR training of a workforce is gaining traction among manufacturers, and there are multiple dedicated solutions addressing needs of individual professions [6,12,21]. However, today there is no universal training platform for multiple industries, factories, or positions. Only narrow in scope systems with high potential to of cost and/or training time minimization are developed. Following this pathway, and recognizing the need for aviation industry workers training, a VR environment, for operating CNC equipment, was developed. Next its use and usability was compared with three other, popular,

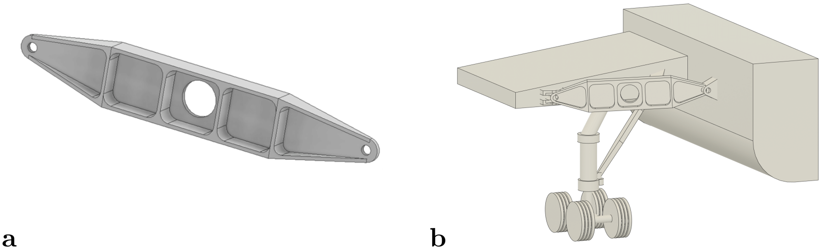


Fig. 1. (**a**) Example of an aircraft landing gear beam (simplified model). (**b**) A beam in full view of the aircraft landing gear.

approaches to worker training. Before proceeding to report gathered experiences, let us describe main assumptions guiding development of the VR system.

3. Preliminary assumptions

3.1. Technological process

Because of local interest in the aerospace industry, the aircraft landing gear beam was chosen as an example of a machined part. This beam is a structural part connecting the aircraft wing to the landing gear assembly. It works together with other landing gear components, e.g. a shock absorber. It is located in the wing, above the landing gear.

The landing gear beam, along with the landing gear itself, are subjected to considerable stress during aircraft landing. These parts are typically manufactured from aluminum and titanium alloys, due to the desired ratio of weight and strength. Figure 1 shows an example of a landing gear beam and its location in the landing gear, respectively. For VR representation of the training process, it was simplified to include only key machining operations needed to deliver the part. However, in the future, the "missing steps of the process" will be added to the application.

To make the part, it is necessary to consider the technological process, taking place during the manufacturing process (Fig. 2). Conceptualizing it, is the starting point for studying use of various forms of training for CNC machine operators. The captured information includes: change in shape, dimensions, physical and chemical properties, and surface quality of the workpiece, which take place as a result of operations to which it is subjected. Moreover, details concerning all necessary workshop aids, and steps needed to be completed is represented. The technological process was formulated on the basis of a detailed drawing of the part and in collaboration with the factory that actually produces such parts. The resulting process can be summarized as follows:

1. roughing and finishing milling of external surfaces,
2. roughing and finishing milling of internal surfaces,

3. spot drilling,
4. drilling,
5. threading.

In the next step, such factors as the workpiece material, the technological capabilities of the machine tool, and the production volume were also taken into account, when conceptualizing the technological process. The tool selection was carried out on the basis of the Gühring tool catalogues. Figure 3 shows the selected tools that have been used in the manufacturing process, represented in VR.

A block of material, measuring 302 × 52 × 23 mm, made of aluminum alloy PA9/7075 (ISO AlZn5.5MgCu), was used (in the VR system) as a semi-finished part (element that is to be processed). The object was subjected to milling, drilling, drilling, and threading. Due to the prior preparation of the semi-finished product, to fit within tolerance dimensions, a single workpiece fixture was used. The semi-finished part was clamped in a vice (Fig. 4). The selection of tools and of the semi-finished product was

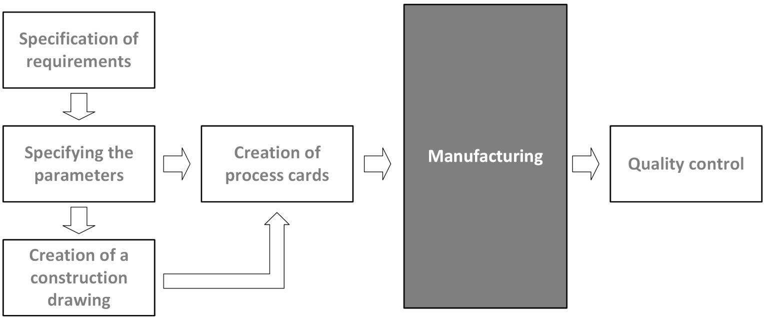

Specification of requirements
Specifying the parameters
Creation of a construction drawing
Creation of process cards
Manufacturing
Quality control

Fig. 2. The technological process.

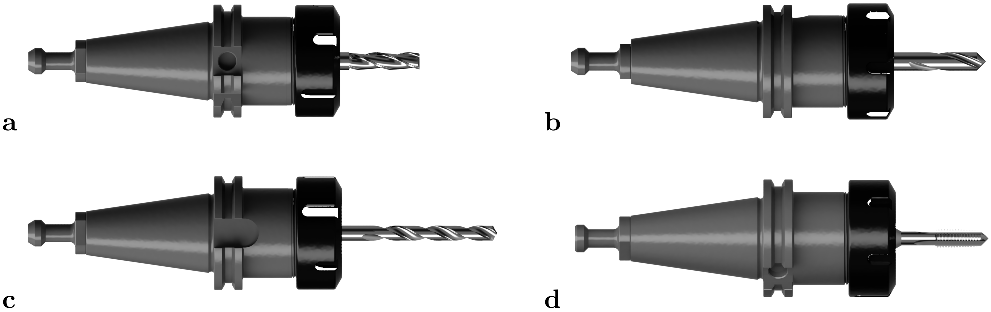


Fig. 3. Tools used in the VR environment: (**a**) endmill $\phi10$, (**b**) NC spot drill $\phi8$, (**c**) twist drill $\phi8.5$, and (**d**) thread tap M10.

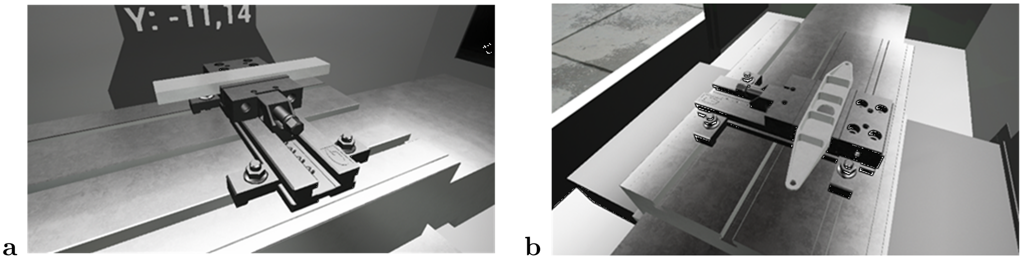


Fig. 4. Fabrication of aircraft landing gear beam in the VR environment: (**a**) mounting of the semi-finished product in a vice; (**b**) finished beam after the BV-represented fabrication process.

appropriately represented in the VR implementation, and all other forms, of training. In this way their usability could have been properly compared.

On the basis of the manufacturing drawing, and of the guidelines contained therein, and the information gathered from the manufacturer; process framework, process card, machining instruction sheets, machining sketch sheets, and tool setting sheets were prepared. According to the documentation, the machining of the part was programmed in the NX Siemens system. The resulting NC code, contains commands, needed to prepare the tool and to selects the cutting parameters. The NC code contains also coordinates of points along which setting and cutting movements are to be performed.

In the industrial practice, documentation prepared in this way, combined into a guide, becomes the input for the operator-programmer, producing the part on the machine tool. Specifically, to actually make the part, steps shown in Fig. 5 must be completed. Note that, each part, prepared (for instance, using the NX Siemens system) a similar documentation has to be prepared. This, in turn, can be used to instantiate any of the four training environments considered in this contribution.

Operations shown in Fig. 5 require the operator to know machine tool operation and programming. Specifically, execution of the correct tasks by the machine tool involves calling appropriate functions on the machine tool's control panel. Note that this can be correctly performed only by the trained operator.

3.2. Training forms

Four different forms of training, realizing the technological process described above, were considered: (a) physical Haas VF2 device, (b) physical simulator of the Haas device, (c) software simulator, and (d) dedicated VR environment (Fig. 6).

Available forms of training differ, depending on the tools used. (1) The first type of training is the most common form of on-site training. Moreover, it best reflects the actual conditions. However, most often, it requires taking the machine out of the production process. (2) In the case of the physical simulator, it is possible to learn the

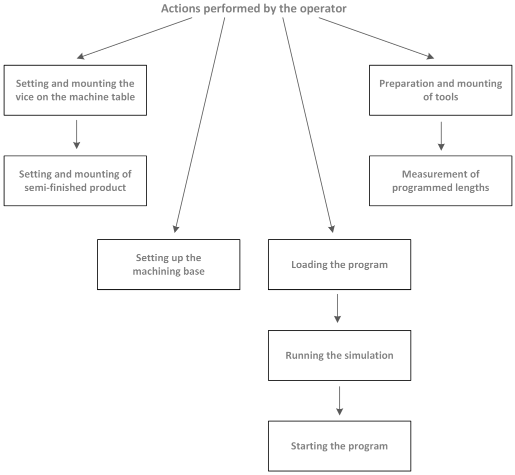
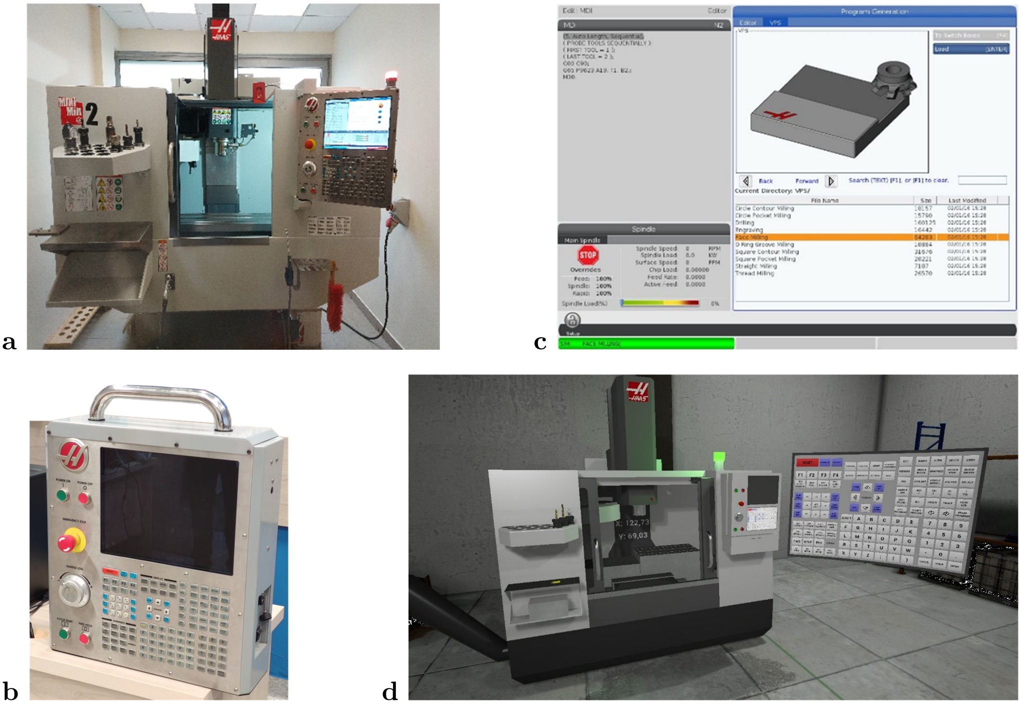

Actions performed by the operator
Setting and mounting the vice on the machine table
Setting and mounting of semi-finished product
Preparation and mounting of tools
Measurement of programmed lengths
Setting up the machining base
Loading the program
Running the simulation
Starting the program

Fig. 5. Procedure for the preparation of the machining process on a numerical machine.

configuration and location of individual functions on a desktop that is identical to the machine tool. The disadvantage is the inability to observe commands being executed, due to lack of viewing devices, such as a monitor. (3) Software simulators are most popular in self-education. Note that, because of the relatively low prices (possibly existence of free simulators), self-training at home is also possible. Current software simulators, in addition to control panels, faithfully reproduce the body of the machine and simulate the movements and actions of the operator's command. (4) Since the last form of training is the focus of this contribution, it will be analysed in more details.

To compare the four forms of training, differentiated features were selected to provide a broader view of their potential. The selection of features is based on multiple years of experience, gathered by the research team in development and use of VR environments in didactic and engineering education. Table 1 summarizes the features that have been selected in the case of each form of training.

In the case of the first three forms of education, the time required to prepare the exercise tasks is relatively short. These activities usually involve preparation of a set of instructions, and technical documents that can be reused multiple times. The process of implementing training, based on simulation tools is the shortest, due to the flexibility of the simulators. The training performed directly on the CNC machine needs more time and attention for the teacher/instructor to prepare the machine and the raw materials. Moreover, instructor must carefully oversee what a student does. CNC machines are relatively "fragile" and this requires more commitment from the trainer, to avoid damages caused by student errors. Here, an adequate room, power supply, and safety



```
(5. Auto Length, Sequential)
( PROBE TOOLS SEQUENTIALLY )
( FIRST TOOL = 1 );
( LAST TOOL = 2 );
G00 G90;
G65 P9023 A19. T1. B2.;
M30
```

```
Spindle Speed   0        RPM
Spindle Load    0.0      KW
Surface Speed   0        FPM
Chip Load       0.00000
Feed Rate       0.0000
Active Feed     0.0000

Feed:    100%
Spindle: 100%
Rapid:   100%
```

Current Directory: VPS/

File Name	Size	Last Modified
Circle Contour Milling	18157	02/01/16 15:28
Circle Pocket Milling	15790	02/01/16 15:28
Drilling	160125	02/01/16 15:28
Engraving	16442	02/01/16 15:28
Face Milling	54283	02/01/16 15:28
O-Ring Groove Milling	18884	02/01/16 15:28
Square Contour Milling	31676	02/01/16 15:28
Square Pocket Milling	20221	02/01/16 15:28
Straight Milling	7187	02/01/16 15:28
Thread Milling	26570	02/01/16 15:28

Fig. 6. Four potential training options: (**a**) a physical Haas machine; (**b**) a Haas simulator; (**c**) a computerized Haas simulator, and (**d**) a VR environment mimicking the Haas machine.

systems, need to be ensured during the instruction. In comparison, the hardest and the most time-consuming is the process for creating a VR training system. The virtual environment not only requires labour-intensive programming, but also (usually) includes 3D model design and implementation, as well as testing activities. This process involves also appropriate rooms, VR equipment and computers. However, most importantly, training of instructors who, by definition, are not IT specialists, is required. The aforementioned factors directly affect the cost of implementing each form of training.

Another important aspect, is the repeatability of exercises, which for organizational and cost reasons, can be limited for physical machines. The situation is different for simulators, which naturally facilitate repetition of specific exercises, without generating additional costs. Here, the VR environment, in which once-programmed training procedures can be repeated indefinitely, also compares very well. Note also that simulator and VR systems do not require consumable materials, further reducing costs of exercises.

The trainees' concentration during the class is also an important aspect of different

Tab. 1. A summary of the features of each form of education.

Feature/Potential	Physical machine CNC	Physical simulator	Software simulator	VR environment
Time-consuming task/exercise preparation	Low	Low	Low	High
The implementation process	Medium	Low	Low	High
Repetition of exercises	Low	Medium	High	High
Implementation costs	High	Medium	Low	Medium
Unit cost of exercise	High	Low	Low	Low
Concentrations during class	Low	Low	Medium	High
Hint system	Medium	Medium	Medium	High
Evaluation automation (grading)	Low	Low	Medium	High
Possibility of remote learning	Low	Low	High	High
Opportunities for self-education	Low	Low	High	Medium
Learning with a teacher	High	High	Medium	Medium
Need for additional training with a physical machine	Low	Medium	High	Medium
Risk of damage to equipment	High	Medium	Low	Low
Stress during training	High	Medium	Low	Medium
Training close to real conditions	High	Medium	Low	Medium
Training time	High	Low	Low	Low

forms of training. For the physical machine work, concentration is at a low level, because the operator starts the job and does not need to remain focused, when the physical fabrication can take a long time. Additionally, here, many factors can distract her, e.g. noise, other people, etc. Working with physical simulators is also not conducive to concentration. On the other hand, use of computer software forces a higher level of concentration due to the need to perform frequent, and deliberate, configuration activities, needed to achieve desired results. However, the highest level of concentration takes place in VR environments, and is caused by the immersion. This contributes to better learning results.

When analysing the prompting system, note that for physical machines, there is always an instructor next to the student. With simulators, there is one instructor for several students, while in the simulator and the VR environment, prompts are displayed by the systems when needed, while help from the instructor may be an extra option.

Another important aspect is that in the case of the physical machine and of the simulator, it is difficult, or even impossible, to automate the evaluation. Obviously, the instructor can look at the trainees' work during the exercises, but usually, the evaluation of work performance occurs at the end of the work, or after the completion of major milestones. In the case of the computer simulator, progress at various stages of the task can be evaluated, while in the VR environment, performance can be evaluated in real-time.

Separately, let us note that modern teaching processes largely take into account the possibility of implementing remote learning. This is due to the dispersion of trainees, cost reduction, time flexibility, and external factors such as pandemics or weather conditions. Recent years in particular (prompted by the COVID-19 pandemics) made it clear that remote learning can be implemented effectively, even on a large scale. Therefore, it is also necessary to look at this aspect of education. Of course, due to the way the physical machine is operated, it is not possible to use it directly for remote education, including general instructional courses. The situation is similar to physical simulators, as they were designed for direct access by trainees. Computer simulators, on the other hand, can be used anywhere and anytime, on local and remote computers. In this context, it might seem that VR systems are dedicated to desktop learning. However, nowadays, the cost of purchasing simple VR systems continues to decrease, making them accessible to a large audience, who, thanks to the Internet, can access repositories of VR applications. Furthermore, VR allows one to work remotely with an instructor. This brings about possibility of self-learning. However, it should be admitted that as of today, use of VR technology may require cooperation of the instructor.

The nature of solutions, based on physical equipment, is that there is always a risk of damage to the machinery through carelessness, misuse, or simple failure. In contrast, computer software reduces such risks to almost zero. However, any mistakes, made by the user, can result in the need to start work from the scratch or, in the worst situation, to restart the software, or the system. Nevertheless, devices modelled in the VR environment cannot be physically damaged by the user. However, the potential risk is the possibility of damaging the VR goggles and/or the manipulators. Another aspect of use of particular forms of training, also related to the possibility of damage to the machinery, is the possible stress on trainees. From this perspective, working on the physical equipment can involve much higher level of stress than use of computer software. The awareness of the consequences of mistakes, made during exercises, in terms of the equipment, consumables, or one's own health, may result in stress and lack of comfort in some trainees. However, VR technology can create also a certain level of discomfort, this time caused by working in a virtual environment.

Analysing various forms of training, it can be concluded that solutions based on hardware platforms, by definition, provide an environment that is closer to real working conditions than computer software. On the other hand, in the VR environment it is possible to relatively well reproduce the real working conditions. Nevertheless, as of today, the technology still has some limitations related to the reproduction of physical phenomena, and involves simplifications and generalizations, necessary when creating a VR environment. However, an advantage of working with computer systems and software is the ability to speed up certain processes that, in reality, take substantial amount of time. Therefore, such forms of training allow more efficient use of available training time.

Usually, after machine operators have finished training, there is a need for additional bench training at the target manufacturing machine. However, when training with a physical CNC machine, it is often identical to the actual workstation (or from the same series). This minimizes the need for additional training. In the case of a real simulator, use of an actual control system is involved. However, the specifics of manufacturing of the parts, their attachment, tool wear, etc., cannot covered. Here, additional training is needed, with particular emphasis on manual handling of manufacturing elements of the device. In computer simulators, trainees learn the process of numerical preparation of parts for manufacturing, but the visualization of the machining control process, on the device's manipulator, may be limited (e.g. a universal manipulator may be used). Here, the training on the actual equipment is needed, and must take into account the specific CNC equipment that is to be used. Finally, VR training combines elements from the remaining approaches. Here, one can accurately replicate all work scenarios that operators may encounter in the real world. VR training is followed by training in a real environment, focused primarily on improving manual skills and taking into account, for example, the weight of individual components and the specifics of their assembly (e.g., the torque of individual components).

In summary, it can be concluded that each form of training has its pros and cons. Each solution requires a repetitive approach, as engineers learn through practice. Here, the VR-based solution can be naturally adjusted to any training at any facilities. What is needed is a team of programmers/graphics designers and testers to create an environment ready for the company needs. Therefore, it was deemed important to compare side-by-side actual realizations of each of the four training approaches and verify the, above stated, expectations, with reactions of actual trainees.

4. Survey of participants' opinions on selected forms of education

As mentioned above, a survey has been performed on the group of random participants with and without knowledge of using the CNC machine. To analyse results the NPS indicator has been used with additional A Mann–Whitney U test.

4.1. The research process

In order to verify and analyse the working hypothesis, an anonymous survey was administered to a group of 25 respondents in 2022. A survey questionnaire was made available through the CAVI (Computer Assisted Web Interview) and PAPI (PAper and Pencil Interview) methods [22]. During the survey, no personal data was collected. Moreover, both CAVI and PAPI were set in such a way that the collected information has been fully anonymized. The questionnaire targeted a group of engineers after they tested each of the four methods. Then the respondents were asked to answer questions related to:

1. working and learning with a physical CNC machine,
2. working and learning with a physical simulator,
3. working and learning with the use of a software simulator,
4. working and learning with the VR system.

The questionnaire included closed and open-ended questions. The questionnaire also used a metric that asked respondents to specify gender information, as well as experience with VR systems, CNC machines, and simulators. The NPS indicator was also used to examine the extent to which respondents are willing to promote the selected forms of education in their environment. STATISTICA 13 was used to analyse the data. The statistical analysis primarily used count tables, which showed both numerical and percentage summaries of individual responses, and used the arithmetic mean and coefficient of variation (denoted as v). The coefficient of variation is the ratio of the standard deviation to the mean. The following assumptions were made: when $v < 25\%$ – there is low variability, $25\% \leq v < 45\%$- there is average variability, $45\% \leq v \leq 100\%$ – there is strong variability, $v > 100\%$ – there is compelling variability. Responses to open-ended questions on the main advantages and disadvantages of using selected forms of education were categorized and presented graphically in the order of the most frequent responses. The Mann–Whitney U test was used to determine the relationship between qualitative and quantitative characteristics. The study was conducted at a significance level of $\alpha = 0.05$. According to the literature, it was assumed that: when $p < .05$ – there is a statistically significant relationship; $p < .01$ – there is a highly significant relationship; $p < .001$ – there is a very high statistically significant relationship [2, 10].

4.2. Analysis of survey results

The structure of the set of respondents by gender is shown in Tab. 2. The data indicates that the survey group is predominantly male – 88%. In contrast, 12% of the respondents were women. Another question in the survey-concerned respondents' experience with VR environments. Analysing the data, it can be concluded that 92% of respondents have used such solutions in the past. In contrast, 8% of the respondents had no exposure to VR systems. Respondents' experience with physical CNC machines was further considered in the survey. 76% of the respondents confirmed their experience. Nearly one in the four respondents had not operated CNC machines in the past. Another question in the survey concerned the experience of the use of simulation applications. As in the case of CNC machines, 76% of the respondents were familiar with such solutions. In contrast, 24% of respondents to this question answered negatively. Table 2 also presents detailed data on respondents' experience with selected forms of training. According to the survey, 68% of respondents have experience with CNC machines, simulators, and VR environments.

Tab. 2. Characteristics of the respondents.

Variable	N	%
Gender		
Male	22	88%
Female	3	12%
Declaration of experience with the VR environment		
Yes	23	92%
No	2	8%
Experience of work with physical CNC machines		
Yes	19	76%
No	6	24%
Experience of use of simulation applications		
Yes	19	76%
No	6	24%
Form(s) of education		
Experience with a physical CNC machine, simulators and VR environment	17	68%
Experience in operating in a VR environment	6	24%
Experience in operating a physical CNC machine and simulators	2	8%

On the other hand, 24% of respondents have only used VR systems before. Experience in operating a physical CNC machine and simulators was declared by 8% of respondents.

Based on the collected research material, an evaluation of selected education forms was conducted (Tab. 3). According to the survey, respondents for each type of training mostly believe that additional training is needed. However, the largest group of trainees (72%) believe that additional training should be provided in the case of using VR systems. In contrast, 28% of trainees believe that further training is not necessary in this case. Considering the results obtained in the area of the computer simulator, 68% of respondents gave an affirmative answer, and one in three respondents believed that additional training is not necessary. More than half (56%) of the respondents declare the need for additional classes related to the operation of the physical simulator, and 44% of respondents have no opinion. The need to organize additional classes on the operation of the physical CNC machine is stated by 48% of the trainees, 12% believe it is not necessary, and 40% of the respondents have no opinion on the subject.

When asked about the clarity and comprehensibility of the defined tasks, respondents in each case mostly answered affirmatively. The VR environment received 84% positive responses, working with a physical CNC machine and a physical simulator each received 80% affirmative responses, and the use of a software simulator received 76% affirmative responses. Among the respondents, there was no person who negatively evaluated this aspect of learning.

According to the respondents, each listed training course supports application of the acquired knowledge in practice, with the greatest opportunities provided by exposure to a physical CNC machine (88% of positive responses), the VR environment (72% of respondents answered affirmatively), and use of a physical simulator (68% of respondents). The use of a software simulator received 64% of the positive responses.

Understanding practical aspects of operating a CNC machine tool is possible by using a physical machine (100% of positive responses), as well as the use of a VR environment (80% of affirmative responses). In the case of the use of a software simulator, this was confirmed by 52% of respondents, and 68% of respondents in this aspect positively evaluate the use of a physical simulator, 8% expressed themselves negatively, and 24% have no opinion on the subject.

In the next part of the survey, the possibility of achieving the expected learning outcomes during the implementation of the selected training courses was verified. The data indicate that, according to the respondents, the greatest opportunities in this aspect are provided by working with a physical CNC machine (92% of affirmative responses), as well as the VR environment (88% of positive responses). The realization of the established learning objectives when using a software simulator is declared by 60% of the trainees. On the other hand, 44% of respondents believe that it is possible in the case of using a physical simulator, and 4% of respondents believe that it is impossible to achieve the assumed learning outcomes during this form of training.

According to respondents, the availability of the necessary functionalities needed to perform tasks to the greatest extent is possible by using working with a physical CNC machine (100% confirm such possibilities), as well as in a VR environment (72% of respondents gave an affirmative answer). 44% of respondents declare the usefulness of a physical simulator in this area, and only 28% of respondents positively evaluated the use of a software simulator in this aspect.

Another question in the survey concerned the ability to replicate real manufacturing processes. According to respondents, the greatest potential in this regard is working with a physical CNC machine (88% of respondents gave an affirmative answer, 12% have no opinion), as well as working in a VR environment (72% of respondents confirm such possibilities, 28% have no opinion). The use of a software simulator is viewed positively in this aspect by 44% of respondents, but 40% gave a negative answer. Nearly one in five respondents believe that working with a physical simulator does not allow them o reflect real manufacturing processes, 16% confirm such possibilities, and 60% of respondents have no opinion on this issue.

The data allow one to conclude that the possibility of cooperation and exchange of experience occurs to the greatest extent when participating in training using VR solutions (96% of affirmative answers, 4% have no opinion), as well as when working with a physical CNC machine (80% of affirmative answers, 20% of negative answers). Analysing the data for software simulator users, the possibility of cooperation is declared by 60% of respondents, 36% gave a negative answer, and 4% of trainees had no opinion. In the case of working with a physical simulator, 44% of respondents believe that this form of training gives the possibility of exchanging knowledge and experience with others, 8% believe that this is impossible, and 48% of respondents have no opinion on the subject.

Based on the cited data, it can be concluded that the necessity of repetition is most prevalent in the case of training with a physical simulator (76% of respondents gave an affirmative answer, 24% are negative answers), as well as a computer simulator (76% of respondents confirmed this necessity, 24% of trainees have no opinion). In the case of working with a physical CNC machine, 48% of users believe that repetitive exercises are necessary, whereas 52% of respondents have no opinion. Analysis of the data for the VR environment showed that 40% of respondents confirm the necessity of additional classes, whereas 60% of respondents have no opinion on the subject.

According to respondents, the most time-consuming form of learning is training with a physical CNC machine (80% of respondents answered in the affirmative, and 12% of respondents gave a negative answer). 56% of trainees declared that learning with a physical simulator requires a considerable amount of time, while 16% of respondents believe that it is not necessary. On the other hand, similar results in the evaluations of respondents who received training using a software simulator and VR environment, where 64% of respondents believe that these forms of training are not time-consuming.

The data allow us to conclude that the most activating and engaging for the trainee during the class is learning using the VR environment – 100% of respondents declare involvement during this form of training. Slightly more than half of the respondents (52%) confirmed active participation while working with a physical CNC machine, while 48% had no opinion on the subject. In the case of using a software simulator, 44% of respondents stated that they are active during classes, whereas 40% gave a negative answer. The high involvement during classes with the physical simulator is confirmed by only 28% of respondents, 44% have no opinion, and 28% of trainees answered in the negative.

The survey further verified the need for other forms of training in the area of CNC machine operation. Among the respondents, the greatest need for other additional training is in the case of the use of the VR environment (76% of respondents declare the need to organize other forms of learning, 4% of respondents answered in the negative) and software simulation (72% of respondents answered in the affirmative, and 4% of trainees think it is not necessary). The necessity of other forms of learning in the case of training with the use of a physical CNC machine is declared by 32% of respondents, 60% have no opinion, and 8% of respondents answered in the negative. Analysing the results obtained

Tab. 3. Evaluation of selected forms of education.

Question	Response	CNC machines	Physical simulator	Computer simulator	VR system
Does this form of education require additional training?	Definitely yes	28%	0%	0%	12%
	Yes	20%	56%	68%	60%
	Difficult to say	40%	44%	0%	0%
	No	12%	0%	32%	28%
Were the tasks to be carried out defined in a clear and understandable manner?	Definitely yes	20%	8%	0%	40%
	Yes	60%	72%	76%	44%
	Difficult to say	20%	20%	20%	12%
	No	0%	0%	4%	4%
Does this form of training allow you to use the acquired theoretical knowledge?	Definitely yes	64%	8%	0%	24%
	Yes	24%	60%	64%	48%
	Difficult to say	12%	28%	32%	20%
	No	0%	4%	4%	8%
Does this form of education allow you to understand the practical aspects of operating CNC machine tools?	Definitely yes	80%	4%	0%	32%
	Yes	20%	64%	52%	48%
	Difficult to say	0%	24%	44%	12%
	No	0%	8%	4%	8%
In your opinion, does this form of education achieve the desired learning outcomes?	Definitely yes	80%	0%	0%	40%
	Yes	12%	44%	60%	48%
	Difficult to say	8%	52%	40%	8%
	No	0%	4%	0%	4%
In your opinion, did this form of education provide access to all the necessary functionalities needed to perform the tasks?	Definitely yes	60%	4%	0%	12%
	Yes	40%	64%	28%	60%
	Difficult to say	0%	24%	64%	20%
	No	0%	8%	8%	8%
Does this form of education fully replicate the actual manufacturing processes?	Definitely yes	40%	0%	0%	12%
	Yes	48%	16%	44%	60%
	Difficult to say	12%	60%	16%	28%
	No	0%	24%	40%	0%
Does this form of education enable collaboration and exchange of knowledge and experience with others?	Definitely yes	12%	0%	16%	28%
	Yes	68%	44%	44%	68%
	Difficult to say	0%	48%	4%	4%
	No	20%	8%	36%	0%
Does this form of education require repeated exercises to acquire the required?	Definitely yes	0%	0%	28%	12%
	Yes	48%	76%	48%	28%
	Difficult to say	52%	0%	24%	60%
	No	0%	24%	0%	0%
In your opinion, is this form of education time-consuming?	Definitely yes	8%	0%	0%	0%
	Yes	72%	56%	36%	28%
	Difficult to say	8%	28%	0%	8%
	No	12%	16%	64%	64%
Does this form of education most activate and engage during classes?	Definitely yes	20%	0%	16%	84%
	Yes	32%	28%	28%	16%
	Difficult to say	48%	44%	16%	0%
	No	0%	28%	40%	0%
Does this form of training also require the use of another form of training in the area of CNC machine operation?	Definitely yes	20%	0%	28%	12%
	Yes	12%	28%	44%	64%
	Difficult to say	60%	72%	24%	20%
	No	8%	0%	4%	4%
Is this form of education adequate for training a team of engineers for the aerospace industry?	Definitely yes	68%	0%	0%	36%
	Yes	24%	36%	44%	44%
	Difficult to say	8%	44%	56%	16%
	No	0%	20%	0%	4%
Was the level of support from the system/software/device enough?	Definitely yes	0%	12%	20%	44%
	Yes	72%	36%	40%	56%
	Difficult to say	20%	28%	36%	0%
	No	8%	24%	4%	0%
Did you feel comfortable completing tasks using this form of education?	Definitely yes	0%	0%	16%	76%
	Yes	60%	52%	76%	24%
	Difficult to say	40%	48%	8%	0%
	No	0%	0%	0%	0%

in the area of training with the use of a physical simulator, it can be concluded that the majority of respondents (72%) have no opinion on the subject, and 28% believe that there is a need to organize additional forms of learning.

Another question in the survey concerned the appropriateness of using selected forms of training to teach engineering personnel for the aerospace industry. Thus, analysing the data, it can be concluded that in this case, the trainees most prefer training using a physical CNC machine (92% of affirmative responses), as well as VR (80% of positive prompts). 44% of respondents consider the appropriateness of using a software simulator. 36% of respondents recommend the use of physical simulators, while one in five trainees do not recommend this form of training.

The survey further analysed the level of support from the system/software/device. As can be seen from the data in Tab. 3 in this aspect, the highest rating was given to the VR environment (100% of positive responses), followed in order by the use of a physical CNC machine (72% of affirmative responses, 8% of negative responses), working a software simulator (60% of positive responses, 4% of negative responses), and the use of a physical simulator (48% of affirmative responses, 24% of negative responses).

The level of comfort during training is highest when working with VR systems (100% of positive responses). High comfort was also declared when using a software simulator (92% of positive responses). 60% of respondents confirmed comfort when using a physical CNC machine. On the other hand, 52% of trainees confirmed comfort when training using a physical simulator.

4.3. NPS indicator survey

In the following part of the questionnaire, respondents were asked to answer the following question, "How likely are you to recommend this form of training for learning to operate CNC machine tools?" The question was constructed based on an 11-point scale, where the number 0 – I would definitely not recommend this form of training, while the number 10 – I would definitely recommend this form of training. Table 4 shows the aggregate results of the analysis.

Analysing the results, the following conclusions can be drawn:

- The highest NPS rates are found for training with the use of a physical CNC machine (84) and VR systems (76), indicating a high degree of satisfaction with participation in this type of training.
- The NPS rate for software simulator training was 24, which is below the satisfaction level.
- Considering the results for physical simulator training, it turns out that among the respondents there were no people who could confidently recommend this type of training. Thus, after determining the difference between the percentage of promoters (0%) and detractors (56%), it turns out that the NPS rate, in this case, was −56.

Tab. 4. Summary results of the NPS analysis.

Feature/Parameters	Physical CNC machine	Physical simulator	Software simulator	VR environment
Promoters	84%	0%	44%	76%
Passives	16%	44%	36%	24%
Detractors	0%	56%	20%	0%
NPS	84	−56	24	76
Average	9.12	6.44	7.60	9.08
Coef. of variation	9.66	18.53	20.45	8.36

- The highest arithmetic average is in the case of training with the use of a physical CNC machine – 9.12, followed, in turn, by training using VR systems – 9.08; training using a software simulator – 7.6; training using a physical simulator – 6.44.
- The coefficient of variation for the study variables was as follows: training with a physical CNC device – 9.66; training with a physical simulator – 18.53; training with a software simulator – 20.45; training with the use of a VR system – 8.36. These results indicate low variability in the responses to the question asked.

4.4. Verification of the stated hypotheses

Taking into account the results of the surveys presented in section IV.B and C, this article attempts to verify the following research hypotheses:

- it is assumed that experience with the VR environment does not affect the propensity to promote selected forms of education in the environment,
- it is assumed that experience with a physical CNC machine does not affect the propensity to promote selected forms of education in the environment,
- it is assumed that experience with simulation applications does not affect the propensity to promote selected forms of education in the environment,
- it is assumed that gender does not affect the propensity to promote selected forms of education in the environment.

A Mann–Whitney U test was used to assess the relationship between quantitative and qualitative characteristics. The first part of the study tested whether experience with a VR environment influenced the propensity to promote selected forms of education in the environment. The same tests were conducted for the experience with physical CNC machines and the experience with simulation applications. In each case, the test probability p was higher than the significance-level $\alpha = 0$, adopted for the study (Tab. 5, Tab. 6 and Tab. 7). This means that this relationship does not exist.

In the next part of the study, it was decided to test whether gender impacts the

Tab. 5. Results of research on the impact of experience with VR systems on the propensity to promote selected forms of education.

	p
Willingness to promote training with a physical CNC device	0.1763
A tendency to promote training using a physical simulator	1.0000
The tendency to promote training using a software simulator	0.8023
Willingness to promote training with the use of the VR system	0.0639

Tab. 6. Results of research on the impact of experience with physical CNC machines on the propensity to promote selected forms of education.

	p
Willingness to promote training with a physical CNC device	0.5455
A tendency to promote training using a physical simulator	0.3227
The tendency to promote training using a software simulator	0.2391
Willingness to promote training with the use of the VR system	0.7991

propensity of respondents to promote selected forms of education in their environment (Tab. 8). The described results with a test probability-level p and the adopted significance-level $\alpha = 0.05$ allow us to accept the null hypothesis. Thus, the value of the Mann – Whitney U tests allows us to conclude that the propensity to promote selected forms of education does not depend on gender. Additionally, the data in Fig. 7 allow us to conclude that the average grade scores obtained in the various groups studied differ slightly.

4.5. Advantages and disadvantages of using selected forms of education

In open-ended questions, respondents were asked to provide their opinions on the benefits as well as drawbacks of using the selected forms of training. When analysing the results,

Tab. 7. Results of research on the impact of experience with simulation applications on the propensity to promote selected forms of education.

	p
Willingness to promote training with a physical CNC device	0.5455
A tendency to promote training using a physical simulator	0.3399
The tendency to promote training using a software simulator	0.2391
Willingness to promote training with the use of the VR system	0.7991

Tab. 8. Results of research on the effect of gender on the propensity to promote selected forms of education.

	p
Willingness to promote training with a physical CNC device	0.8344
A tendency to promote training using a physical simulator	0.7442
The tendency to promote training using a software simulator	0.8671
Willingness to promote training with the use of the VR system	0.1949

the respondents' answers were categorized and presented in the order of the most frequent responses. Table 9 shows the summary results of the analysis regarding the benefits of using selected forms of training.

According to the survey, the most common answer regarding the benefits of training with a physical CNC machine is the ability to replicate real manufacturing processes,

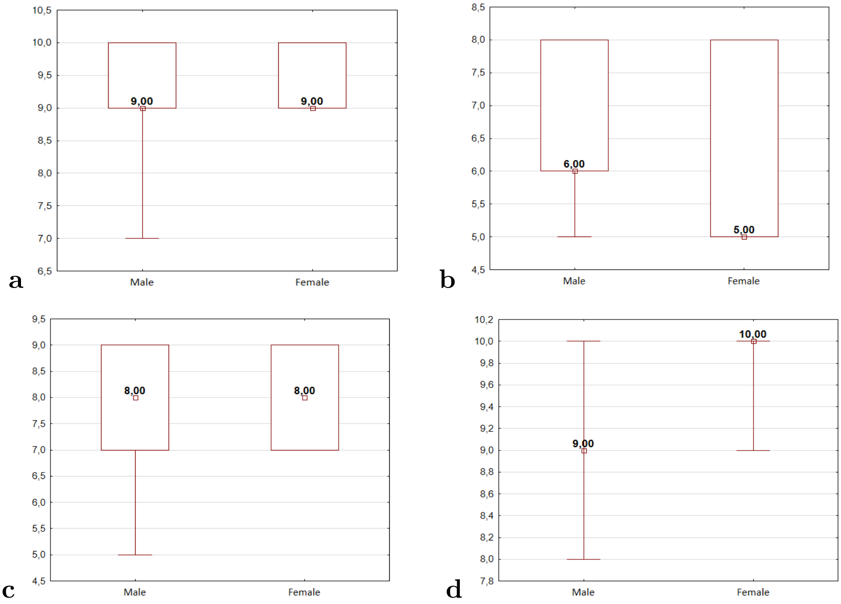


Fig. 7. Evaluation of selected forms of education by gender. Questions for which the graph were made: (**a**) How likely is it that you would recommend training with a physical CNC machine as a form of learning how to use CNC machine tools? (**b**) How likely is it that you would recommend a physical simulator to learn how to use CNC machine tools? (**c**) How likely is it that you would recommend computer simulators to learn how to use CNC machine tools? (**d**) How likely is it that you would recommend VR systems for learning how to use CNC machine tools?

Tab. 9. Respondents' opinions on the benefits of using selected forms of education.

Variable	N	%
CNC machines		
The ability to replicate real manufacturing processes	19	76%
The ability to recognize the behaviour and functionality of the CNC machine	15	60%
Physical simulator		
Reduction in costs associated with the purchase of a CNC machine	16	64%
No possibility of damaging the CNC machine	14	56%
Introduction to working with a CNC machine	9	36%
Possibility to train many people at the same time	7	28%
Software Simulator		
Possibility of remote learning	15	60%
The ability to run the simulator on any computer at any time	8	32%
VR System		
High interactivity	21	84%
An immersive approach to the manufacturing process and the use of the CNC machine	18	72%
The modernity of the applied technologies	12	48%
Low training costs	10	40%

which allows the recognition of machine tool behaviour and functionality. Respondents, therefore, pointed to the opportunity for the trainee to acquire practical skills and qualifications to perform the tasks of the operator's position.

In the case of using a physical simulator, respondents listed the reduction in costs associated with the purchase of a CNC machine among the main advantages. Another benefit is that there is no possibility of damaging the machine, which is especially important for trainees who are just starting out, thus significantly increasing their comfort level. Training with the use of a physical simulator also allows for the possibility of teaching many people simultaneously, which was also mentioned by respondents.

According to the respondents, the main advantage of training using a software simulator is the possibility of remote learning, which allows the exercises to be carried out at any time and place. Note that until recently, the form of remote learning was used mainly as a solution for further training and improving professional skills. Currently, it is a way of education, which is steadily gaining in popularity due to its benefits. This is because it allows individualization of the learning process, as well as reducing time and costs associated with travel or temporary accommodation.

Among the benefits of using VR systems, respondents mentioned, first, high interactivity, which allows active participation in the training, and an immersive approach, thanks to which the trainee assimilates knowledge more effectively and remembers the information directed to him. Respondents also pointed out that the VR environment is counted among the modern technologies included in Industry 4.0, which makes this training very attractive to development-oriented engineers. Respondents also point to the low cost of implementing training in a VR scenario, as through this type of training companies reduce investments related to the purchase of machinery. This perception of VR technology may be appropriate from the viewpoint of those being trained. However, preparing VR simulations, machine tool models, and scenarios is a time- and cost-intensive process. Once a full training scenario is prepared in a VR environment, it scales very well, however, because the preparation of each successive training station comes down to the purchase of the actual computer and VR glasses, which is many times cheaper than buying a new CNC machine. Therefore, it can be said that in the case of preparing a single CNC and VR workstation, the costs are similar but as the number of workstations increases, the CNC costs increase linearly, while in the case of VR, they only involve the purchase of further relatively inexpensive VR sets. In this model, leasing models of VR infrastructure along with models of VR environments for training implementation are also increasingly popular, which can further reduce the cost of VR training.

The last question in the survey concerned the disadvantages associated with the use of the selected forms of education. The results of the analysis are shown in Tab. 10.

The analysis of the collected material shows that in the case of training with a physical CNC machine, respondents listed the need for physical access to the machine and the possibility of damage to the machine tool among the biggest risks. Thus, there is a concern among respondents that their inexperience could affect the creation of errors, and thus incur costs to the company for repairs.

According to the respondents, the main drawbacks of using physical simulators are the inability to replicate real manufacturing processes, which do not allow to recognize the behaviour and functionality of the CNC machine. As a result the lack of contact with physical equipment and an immersive approach are the main drawbacks of using software simulators, according to respondents.

Respondents' concerns about the use of a VR environment again primarily included the lack of contact with a physical machine. Thus, it can be concluded that for the trainees, it is critical to replicate real manufacturing processes and to be able to interact with the machine. Another threat that respondents presented is the need for a large amount of space. Respondents also pointed out that the use of the VR environment can cause side effects such as eye pain. However, this aspect of using VR technology is very subjective, and some users may experience negative effects from being in VR, while others may not.

Tab. 10. Respondents' opinions on the disadvantages of using selected forms of education.

Disadvantages	N	%
CNC machines		
The need for physical access to the machine	15	60%
Possibility of damaging the machine	13	52%
Long waiting time for delivery/access to the machine	9	36%
The need for continuous supervision of trainers (physical presence)	7	28%
The need to have a base of premises where the machines will be located, including adequate strength of ceilings and noise	5	20%
Cost of consumables and artefact materials	4	16%
Physical simulator		
Lack of the ability to replicate actual manufacturing processes	19	76%
Inability to recognize the behaviour and functionality of the CNC machine	12	48%
Need to own/purchase specialized simulation dashboards – long waiting time for delivery	6	24%
The versatility of dashboards – differences between real dashboards and those in physical simulators	4	16%
Software Simulator		
No contact with physical equipment	19	76%
Not a very immersive form of learning	12	48%
Lack of possibility to develop manual skills to operate a particular machine	4	16%
VR System		
No contact with the physical machine	20	80%
Necessary to have a large amount of space	9	36%
Discomfort in using VR glasses related to eye pain and weight of glasses, pressure on the head	7	28%
Dizziness and vagal problems	4	16%
No physical sensation of the weight of the parts in question, the pressure force when inserting the tool, etc.	3	12%

5. Conclusion

The rapid development of industry, especially in the fields of aviation and automotive industry, requires the implementation of modern methods and means to support manufacturing processes. One of the basic elements of the manufacturing process is the preparation of operators of digital machines, including CNC. Classic training methods tend to be time-consuming and costly. Moreover, in the case of extraordinary situations such as epidemics, natural disasters, etc., personnel preparation cycles can be interrupted, resulting in a significant disruption in the schedule of preparation and implementation of production processes. In particular, the SARS-CoV-2 epidemic period has made everyone aware of the importance of alternative solutions that were not commonly used before. One such solution is the possibility of using a virtual reality environment to automate various processes, including the training.

In this paper four approaches to training engineering personnel in the operation of CNC equipment are compared and analysed. One of them is a novel solution based on VR technology. Given the complexity of manufacturing components, for the aerospace industry, in particular, the process of manufacturing a specific component of an aircraft landing gear is presented. Due to the high-quality requirements of such components, the process of preparing a team of operators must guarantee a high level of quality. This, in turn, involves ensuring the repetition of training procedures and activities. As a result, their cost and the time required to conduct them increase. The results show that using an appropriate VR environment can be very useful for training specialists in the area of for example operating the CNC machine. Based on results many participants chose VR over other simulations as a good solution for training.

Studies conducted indicate great potential for the use of VR technology in the industry. Of course, this work refers only to one specific application. As it was mentioned before a practical approach is better then theoretical in many ways. By using physical machine a user can learn all required processes and steps to operate it. However, shifting a very expensive machine from the production process to the training process can be cost inefficient for the industry. The production must be halted to perform training exercises. This solution is very good only in case of machine not used at a defined period of time. The simulation by a computer is efficient for "work from home", although a user is missing the practical experience of using and maintaining a CNC machine. What is learnt is only software.

The usage of the CNC device simulator is good for basic learning when starting training of a CNC operator. A user can learn how to load and run programs; however, training of the usage of a physical machine is also missing and a device simulator is needed. The VR solution comes with many pros. It can be used at any place without special high-cost devices, a user can learn all required process of CNC operation, all based on programmed training, and what an employer needs from a potential employee. Although the contact is simulated, the nowadays VR solutions, including the headsets and controllers, can bring reliable experience for everyone.

With all this, it should be emphasized that the VR environment can guarantee the automation of personnel preparation processes for the industry while maintaining the reproducibility and accuracy of procedures. Critical is the fact of obtaining a high satisfaction rate of survey participants with regard to VR technology.

Acknowledgement

This research paper was developed under the project financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

References

[1] A. Burghardt, D. Szybicki, P. Gierlak, K. Kurc, P. Pietruś, et al. Programming of industrial robots using virtual reality and digital twins. *Applied Sciences*, 10(2):486, Jan 2020. doi:10.3390/app10020486.

[2] G. W. Corder and D. I. Foreman. *Nonparametric statistics: A step-by-step approach.* John Wiley & Sons, Hoboken, New Jersey, second edn., 2014.

[3] P. Dymora, B. Kowal, M. Mazurek, and R. Śliwa. The effects of virtual reality technology application in the aircraft pilot training process. *IOP Conference Series: Materials Science and Engineering*, 1024(1):012099, Jan 2021. doi:10.1088/1757-899X/1024/1/012099.

[4] L. Fade. Virtual reality training being used to cut maintenance time by 75%. VR Vision Group, Jan 2023. `https://vrvisiongroup.com/virtual-reality-training-being-used-to-cut-maintenance-time-by-75/`.

[5] A. C. Firu, A. I. Tapîrdea, A. I. Feier, and G. Drăghici. Virtual reality in the automotive field in industry 4.0. *Materials Today: Proceedings*, 45:4177–4182, 2021. doi:10.1016/j.matpr.2020.12.037.

[6] C. A. Garcia, J. E. Naranjo, A. Ortiz, and M. V. Garcia. An approach of virtual reality environment for technicians training in upstream sector. *IFAC-PapersOnLine*, 52(9):285–291, 2019. doi:10.1016/j.ifacol.2019.08.222.

[7] L3Harris. Virtual maintenance trainer. L3Harris® Fast. Forward., Sep 2014. `https://www.l3harris.com/all-capabilities/virtual-maintenance-trainer`.

[8] E. Lemle, K. Bomkamp, M. K. Williams, and E. Cutbirth. Virtual Reality and the Future of Entertainment. In: *Two Bit Circus and the Future of Entertainment*, pp. 25–37, SpringerBriefs in Computer Science. Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-25793-8_4.

[9] S. Lozé. Beyond the manual: VR training on aircraft maintenance. Unreal Engine, 15 Oct 2019. `https://www.unrealengine.com/en-US/spotlights/beyond-the-manual-vr-training-on-aircraft-maintenance`.

[10] Lærd Statistics. Mann-Whitney U Test using SPSS Statistics, Apr 2016. `https://statistics.laerd.com/spss-tutorials/mann-whitney-u-test-using-spss-statistics.php`.

[11] A. Marto, A. Gonçalves, M. Melo, and M. Bessa. A survey of multisensory VR and AR applications for cultural heritage. *Computers & Graphics*, 102:426–440, Feb 2022. doi:10.1016/j.cag.2021.10.001.

[12] A. Paszkiewicz, M. Salach, P. Dymora, M. Bolanowski, G. Budzik, et al. Methodology of implementing virtual reality in education for Industry 4.0. *Sustainability*, 13(9):5049, Apr 2021. doi:10.3390/su13095049.

[13] A. Paszkiewicz, M. Salach, D. Strzałka, G. Budzik, A. Nikodem, et al. VR education support system—A case study of digital circuits design. *Energies*, 15(1):277, Dec 2021. doi:10.3390/en15010277.

[14] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, Apr 2020. doi:10.1016/j.compedu.2019.103778.

[15] E. M. Rantanen and D. A. Talleur. Incremental transfer and cost effectiveness of groundbased flight trainers in university aviation programs. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49(7):764–768, Sep 2005. doi:10.1177/154193120504900705.

[16] F. Reer, L.-O. Wehden, R. Janzik, W. Y. Tang, and T. Quandt. Virtual reality technology and game enjoyment: The contributions of natural mapping and need satisfaction. *Computers in Human Behavior*, 132:107242, Jul 2022. doi:10.1016/j.chb.2022.107242.

[17] P. Tennent, S. Martindale, S. Benford, D. Darzentas, P. Brundell, et al. Thresholds: Embedding Virtual Reality in the museum. *Journal on Computing and Cultural Heritage*, 13(2):1–35, Jun 2020. doi:10.1145/3369394.

[18] K. Valentino, K. Christian, and E. Joelianto. Virtual reality flight simulator. *Internetworking Indonesia Journal*, 9:21–25, Jan 2017. `https://www.internetworkingindonesia.org/Issues/Vol9-No1-2017/iij-vol9-no1-2017.html`.

[19] Future of virtual reality – market trends and challenges. In: Vijay, R. Desyatnikov, and Swati, eds., *Software Testing Help (STH) blog*, 11 Oct 2023. `https://www.softwaretestinghelp.com/future-of-virtual-reality/`.

[20] J. Vince. Virtual reality techniques in flight simulation. In: *Virtual Reality Systems*, pp. 135–141. Elsevier, 1993. doi:10.1016/B978-0-12-227748-1.50018-4.

[21] B. Xie, H. Liu, R. Alghofaili, Y. Zhang, Y. Jiang, et al. A review on virtual reality skill training applications. *Frontiers in Virtual Reality*, 2:645153, Apr 2021. doi:10.3389/frvir.2021.645153.

[22] J. Zmud, M. Lee-Gosselin, M. Munizaga, and J. A. Carrasco, eds. *Transport Survey Methods: Best Practice for Decision Making.* Emerald Group Publishing Limited, Jan 2013. doi:10.1108/9781781902882.

Generating Layout for Complex Cave-like Levels with Schematic Maps and Cellular Automata

Izabella Antoniuk
Department of Artificial Intelligence, Institute of Information Technology
Warsaw University of Life Sciences – SGGW, Warsaw, Poland.
izabella_antoniuk@sggw.edu.pl

Abstract. In this paper an algorithm for creating cave-like, user-guided layout is presented. In applications such as computer games, underground structures offer unique challenges and interesting space for player actions. Preparation of such areas can be time consuming and tiresome, especially during the design process, when many ideas are often scrapped. Presented approach aims at improving this process. Schematic input is used so the user can quickly define the general layout. Cave system is divided into levels and tiles – easily-parallelizable modules for the following method stages. Cellular automata are used to extend initial system sketch with interesting shapes while the diamond-square algorithm spreads the final terrain heights. Each stage uses the results of the previously performed operations as input, providing space for alterations. Input maps can be reused to obtain different variations of the same system. The final structure is represented as a 3D point cloud. Chosen representation supports multilevel systems and can be used either as a base for further algorithms, or as a final mesh. The presented approach can be easily incorporated into game design process, while visualizing initial layouts and speeding up preparation of unique, interesting and challenging game spaces for the players to traverse.

Key words: Cellular Automata, computer games, Diamond-Square, procedural cave generation, procedural level generation, schematic maps.

1. Introduction

Computer games are growing increasingly more robust, both visually and in terms of the overall complexity. Creating such content can take significant amount of time. While preparing in-game objects manually brings precise and visually appealing results, it can also lead to repeatable content. This is especially the case, when the designer needs to prepare large amounts of similar elements. While tiring for human, appropriately defined algorithms can easily generate such content, achieving required visual complexity without losing the diversity. It is in such applications, that procedural content generation shines the most, speeding up the modelling process, or creating complete elements.

Procedural content generation is not a new area of research. There are quite a few algorithms and approaches addressing this problem. Different solutions can vary in complexity, focusing on single elements, such as plants [36], rivers [26] or roads [19]; generating specific terrain fragments [15, 20, 28], cities [22] or entire worlds [34, 42, 46]. Another division concerns type of usage, focusing either on generating complete content according to user requirements, or on speeding up the modelling process, giving designers another tool to use. Especially in recent years procedural algorithms gained recognition in widely used applications, such as Blender Geometry Nodes [9] or newly announced

Unreal Engine 5.2 Procedural Content Generation Framework [51]. This clearly shows the need for procedural methods in such applications.

Dungeons or caves are very common in computer games. Defining them in terms of algorithm-related parameters can be tricky, due to the existence of multilevel structure with overlapping elements. At the same time, such areas can be the most interesting for players to explore. They are an integral part of numerous games, from classic dungeon crawlers such as Legend of Grimrock [23] and Dungeon Master [13], to more recent productions like The Elder Scrolls V: Skyrim [41], Witcher and Dragon Age series [16, 54] or Elden Ring [17]. Defining layout usually requires additional constraints and rules, which makes them more difficult to generate accurately. At the same time, existing solutions usually either focus on 2D maps of such areas [27, 52], or in case of 3D approaches do not provide user with enough control. Generation often requires large amounts of data, producing output that is not easy to modify [10, 11, 12, 14, 31, 39].

Research presented in this paper focuses on developing an algorithm for procedural generation of cave-like structures, that are both visually interesting and complex. Presented method takes into account the multilevel property of the chosen terrain type, and can represent it accurately. User can additionally define the layout of the entire system, using simplified sketches, ensuring that it will follow the initial design, while the algorithm will add detail to it. Finally the method allows both for simplified visualization using point-cloud, as well as further edition of generated content (either using modelling applications such as Blender [9], or by incorporating them into game engines, like Unity [49] or Unreal Engine [50]).

2. Related works

Procedural content generation (PCG) especially in recent years is developing quickly. New approaches are prepared and increasing number of them takes into account specific requirement, coming from fields such as computer games and simulations. At the same time, there are still some drawback that can be seen in those methods. Most of existing solutions can be roughly divided into two main categories: complex generation or methods focused on speeding up the modelling process.

Both approaches can be interesting for applications such as computer games, assuming certain requirements are met. Designers would usually want to transfer their vision into the final outcome of the algorithm. Because of that it is important to include some way to define content properties. At the same time numerous, unintuitive parameters can have exactly opposite effect, over-complicating the process and making it to tiresome. Existing methods use various types of input files, as well as ways to evaluate generated elements. Most interesting from the point of view of research performed in this paper, are solutions that:

- use different types of maps as algorithm input,

- focus on content meant for computer games, or
- generate cave-like and dungeon-like structures.

All of the above groups represent some properties of the presented approach. At the same time no method was found meeting all of them from the chosen field.

For a comprehensive survey of various PCG methods used to create virtual world elements see [44, 53, 56].

2.1. Generating world elements using input maps

One of the major areas, where procedural content generation shines, is creating world elements. Producing such areas (similar in structure, but not repeatable) using algorithms is more than justified. At the same time defining terrain with input maps makes it easier for the user to make sure that the results meet his requirements.

A complex approach with multiple maps representing information about terrain details is presented in [47]. Authors use separate files to define general terrain height, bodies of water, vegetation, roads and buildings, and later combine them to represent full scene. The method can model interactions between different layers, i.e. creating bridge if a road crosses over river. This approach was further developed in [43, 45, 46], adding detail to the method and improving the generation process and procedures used to connect different elements.

Slightly different approach, instead of using schematic maps to outline terrain, applies them as modifiers [55]. In that case provided sketch map defines key points, such as canyons, rivers or mountain ranges. This map is then used to generated 3D scene, using USGS DEM information for additional details.

A series of solutions focuses on generating different virtual world fragments using simplified inputs [2, 3, 4, 5, 6, 7]. Similarly to research presented in this paper, terrain is divided into tiles, and – in case of underground structures – levels, while the generation process is defined by user-set properties. The approach also takes into account various constraints related to the computer games in general. Final results are represented as editable, 3D models.

As can be seen in different approaches, even simple input maps can be used to generate complicated and engaging content, with some of them using just a single map for that purpose [1, 38]. Such input is easier for a designer to understand, than sets of numeric parameters. High level of control with intuitive file structure and its influence over final result are the key elements that decided the types of maps used in the approach presented in this paper.

2.2. Procedural generation for games

When it comes to computer games, any content meant for such application needs to have specific properties. Preparing algorithms that take various requirements into account is another vast area of research. A comprehensive survey of PCG methods used in computer games was presented in [25].

As was noted in [21], there are quite a few areas, where repeatability of different elements can be noted. Authors point out, that since repetition is not a natural occurrence, with real wold containing infinite numbers of unique patterns, it can often result in player losing the immersion because of this. Repetitive elements can tire the player. When recognized patterns can additionally be transferred to gameplay strategies, the difficulty will also decrease, resulting in boredom. Making sure it is not the case for newly created content is an excellent task for procedural generation algorithms.

In [8] a fitness function is used to evolve mazes that meet specified requirements. Layouts are produced by adjusting parameters, ensuring that both ends of the maze are always connected. Different evolution-based approach focuses on generating 3D terrain fragments, that the user can adjust [37]. In this case, the terrain is constructed based on selected patches, creating a seamless crossover, in theory closer to the user requirements.

Another set of examples instead of defining properties of the created world, takes into account the story that will happen in it. In [24] authors use story, to later generate world supporting its key elements. This solution is capable of creating complex terrains, that are well adapted to given input. While the maps are two-dimensional, they could be used as a base for further work in 3D space. In [32] authors incorporate user-defined key points, and relationships between them. This allows the creation of a map with strategically placed towns and cities. Content created in such a way aligns with specifically set constraints, consistent with the game story.

In case of computer games, level of control is equally important as interesting results. In [30] the user can choose a set of actions, that will later be represented in the resulting map. Used constraints are all gameplay-related and need to be specified by the designers. The operation of the algorithm was presented on the example of Dwarf Quest game, and resulted in complicated layouts.

One interesting method describes procedural level generation using snappable meshes [40]. Authors use set of predefined assets with established connection points, a set of constraints describing how they can be connected and general way the level will be constructed. Different level types can be created and implementation in the Unity game engine was also prepared. Although the main focus of the algorithm is to “avoid size and layout limitations”, it is heavily dependent on the quality of the prepared assets. The map generated can contain multiple levels, but this again depends from the structure of initial assets.

2.3. Creating cave and dungeon structures

When it comes to procedural generation of underground structures, a series of additional constraints need to be considered. The layout tends to be more complex than in case of the surface areas. Additionally such structures in real world tend to have multiple levels with overlapping areas, that cannot be represented by a simple height map.

One of many approaches to cave generation in particular considers the problem of creating natural-looking structures, with main focus put on the karst caves [18]. The method is implemented in Unity, and prepares both the layout of the cave system, along with individual shape of passages, textures and cave features (such as speleothems). The generation is heavily based on the natural process of cave formation, although it is simplified to expedite the computation. Unfortunately it does not provide any way to define or adjust the layout of the cave system. It also doesn't take into account any computer game requirements apart from generation time. At the same time it is noted, that current version of the method is not adjusted to such application.

Another intersecting example uses genetic algorithms to evolve dungeons according to user specifications [29]. Authors use two maps for this process. High-level sketch of the dungeon is used to define overall connectivity of different fragments and the content of those which are passable. Second map is a low-level, high resolution representation of individually generated segments. In the second evolution step individual segments are generated. Authors take into account many game-specific requirements. Generated shapes can be complex and visually interesting. The resulting map is mostly two-dimensional though, without any vertical transitions.

Algorithm presented in [31] focuses on generating 3D caves for application in computer games. Method consists of two main steps. First one uses L-system to define the structural points – the general layout of the created level. Tunnels and caves are generated after that, by wrapping a meta-ball along paths defined in the first step. While initially voxel representation is used, final terrain is obtained by converting resulting scene to mesh with assigned textures and shading. The method can create complex, multilevel structures with various features. Unfortunately, user has very little control over resulting layout, making it difficult to apply in computer games where specific terrain properties are required.

Overall, while there are quite a few interesting approaches breaching the subjects presented in this paper, none of them meets all of the defined requirements. Prepared algorithm builds on those drawbacks, using input maps to ensure user control and allowing definition of multilevel structures with precise layout. Generation process can be influenced during various stages, with final results stored in an easily editable manner.

3. Input maps

One of the more important aspects of procedural generation for computer games, is the definition of initial input files. Using only generation parameters poses some problems. While such approach can be sufficient to simple applications, computer designers usually will require more significant way to define final content. Another problem is that parameters in general tend to be harder to understand in terms of their influence over the final object. At the same time, input which is to specific will reduce procedural generation to the slightly faster manual modelling (as is the case with SpeedTree application [48]). While using such tools tends to speed up the level creation it also doesn't work well with large amounts of repeatable elements. The need to create large quantities of similar content often results in areas that are visually similar to each other. Procedural content generation can shine especially in such areas, assuming it can at the same time sufficiently include input from the human designer.

To achieve high level of control without reducing the generation process to tedious, manual modelling, set of schematic input maps is used. In presented approach user needs to prepare total of two maps, to represent the overall structure of underground system: placement map, and system layout map.

3.1. Placement map

First used map defines placement of individual tiles inside each level vertically. Standard approach for the height maps is used, with values represented in grey-scale in range (0, 255). The main difference lies in how those values are applied. In that case each pixel represents single region in final terrain (tile). Height value represents placement of tile in single level. Actual height values for tiles are scaled according to each level spread (with single step value resulting from dividing overall level height by 256). Each value in the placement map corresponds directly to basic height of the tile, used for further operations. Since placement map does not include tile size, once defined it can be reused for systems with the same structure, but different sizes of the individual elements, making it easy to experiment with various levels of complexity for the final system. Example visualization of tile spread with corresponding height map is presented in Figure 1.

3.2. System layout map

Second map represents general system layout. Since the designer might want to define key features (i.e. large room in chosen location, or some crossings that must occur), this map is used to represent such elements. There are few key aspects that need to be represented:

- definition of the general layout for the underground system,
- indication of existing connections between tiles,
- exclusion of tile connections when necessary,
- indication of passages leading to lower levels.

In [5, 6] similar approach was used, with one map defining tile placement, and two additional ones denoting connections and type of terrain in each region. While specific, such definition is not intuitive enough, especially in terms of defining connections.

To avoid such problems, in the presented approach all key elements are represented by using different coloured annotations on single image. User can sketch a simple map, where white colour represents the general system layout, that will be later used during

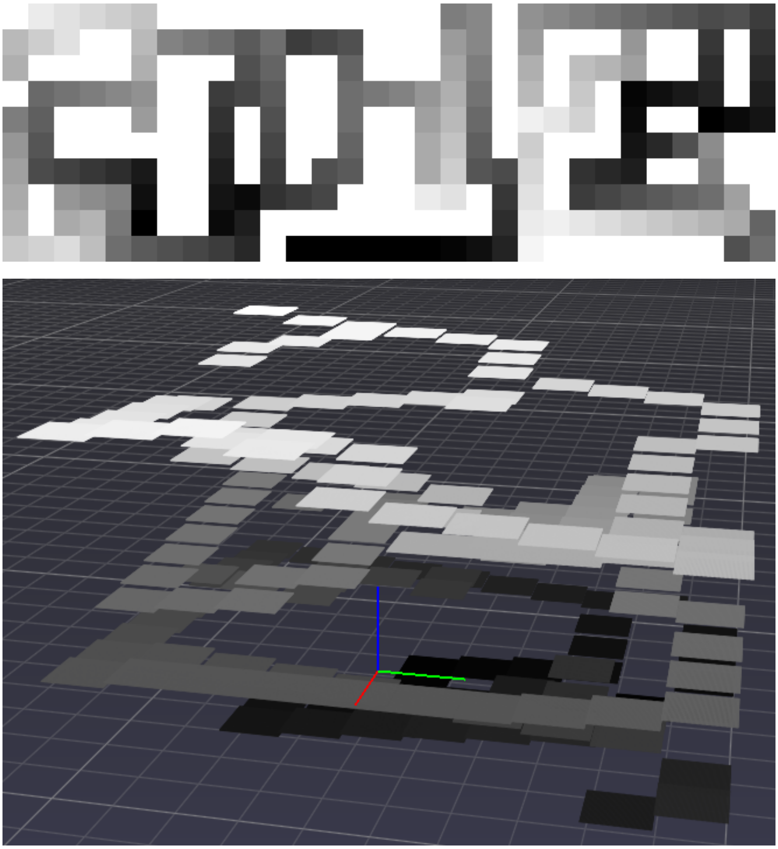


Fig. 1. Placement map used for deciding initial tile location (top), with corresponding point-cloud visualization representing a region spread in 3D space (bottom). The placement map is prepared for 3-level dungeon, where each level contains 10×10 separate tiles, with tile size equal to 51. The tiles were marked in grayscale, to represent the overall system structure, with lighter values representing higher regions.

generation. At the tile edges, connections to lower levels can be specified using green colour, and passage exclusions can be defined with blue. Tile connections are obtained automatically from general system shape (transitions for each region are named according to the tile they lead to: either Top, Bottom, Right or Left). There are four cases that need to be considered:

- Tile edge has green mark, indicating connection to lower level. The connection will be saved with ‘L_’ prefix and will be used in level connection stage during generation process.

- Tile edge has blue mark, indicating excluded connection. This direction will be saved with ‘E_’ prefix and even if the system shape will reach the edge of the tile, it will not be considered as connection in future operations.
- Tile edge has no marks and system sketch (white) reaches edge of the tile. The connection will be saved in basic form (Top, Bottom, Right or Left), and will be considered during generation process.
- Tile edge has no marks, and system sketch does not reach edge of the tile. The connection will be saved as ‘None’, and will not be considered during generation process.

Example system map is presented in Figure 2.

Additional problem with underground systems concerns overall space representation. For the created structure to be interesting in terms of exploration it should contain multilevel and complex structures. Standard 2D files are not able to directly represent such areas. To address that problem, the space in generated caves is divided into vertically aligned levels, with structures that are not overlapping with each other. Each level has size and spread. The size of level defines number of tiles along each side. Spread decides single level height and is a base for division used while placing individual regions. Vertical transitions can be realised by connecting tiles between different levels, allowing creation of more complex structures. All input maps contain combined information for all levels in created system, with highest level data placed on the left side, and succeeding, lower levels placed next to it (see Figures 1 and 2).

4. Layout generation

After user prepares the input files, initial data is derived from them and used to generate the underground system. The process is divided into three, separate stages: preparing initial system layout, generating heights for cave shape in each tile, and combining data.

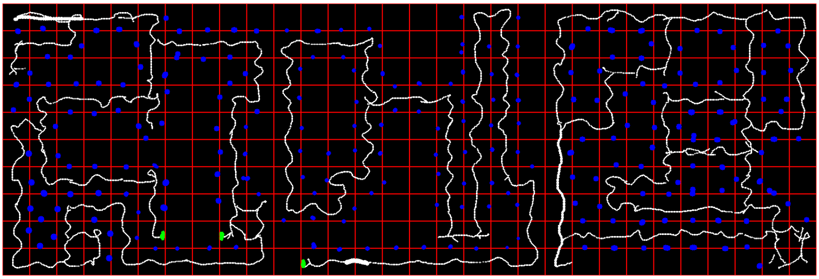


Fig. 2. System layout map used for shape generation. Presented sketch is prepared for 3-level dungeon, where each level contains 10x10 separate tiles. Single tile size is set at 51x51 pixels. Red-coloured grid marks borders between individual tiles.

First stage generates the overall system shape in each tile, generating cave-like shapes using cellular automata algorithm, and connecting it to user-prepared system sketch. This map can be later extended using individual operations, either on entire system or on individual tiles. After the initial map is prepared, heights for each pixel representing system shape are generated using the diamond-square algorithm, to add 3D details. The data from both stages is than combined, to fully reprsent the final system shape. Each pixel in the original map correscponds directly to single vertex in 3D space. Results are visualized as a point-cloud. Pseudocode outlining the entire procedure is presented in Algorithm 1.

4.1. Initial system shapes with cellular automata

First step during the generation process concerns creating the initial system shape. Taking into account the connections between tiles, the system sketch created by the user will be expanded using cellular automata algorithm.

Such approach was chosen for several reasons. Firstly, cellular automata can produce realistic results, that resemble the real-world structures. In [27] this algorithm produced visually interesting shapes. At the same time, it is hard to control the layout of the system without some modifications. One huge drawback is the connectivity of the generated structures. Especially in applications such as computer games, existence of areas that

Algorithm 1 Cave system layout generation from schematic input maps.

```
Input: height map, system sketch
for tiles in system do
    Generate cellular automata shapes
    Connect shapes to sketch (system sketch)
end for
System map = generated map
for User input action do
    updated system map = Perform user action (System map)
    System map = updated system map
end for
for tiles in system do
    Check user options for height generation
    Generate heights in tiles (System map, user options)
end for
Final system = Combine data (System map, height map, system sketch)
Calculate point coordinates for export (System map, height map, system sketch,
    Final system)
```

are not connected to the main system can be problematic. It is even more important if additional method is used to place key objects or the player himself as it can create a scenario, where either the player is stuck in small area, or the key elements required for progression are unreachable. Both situations are extremely undesirable. Additionally, designer would usually want the cave system to follow certain layout. By expanding the sketch, instead of generating the entire structure, this requirements will be easily met, while producing visually interesting and complex shapes.

The chosen approach uses cellular automata algorithm, to generate cave-like shapes of various sizes across the tile. Those shapes are then attached, firstly to the initial sketch presented by the user, and later to the overall structure created during generation. With multiple operations used, the final system can grow without the danger of disconnected elements being created along the way.

During this stage the initial map of the system is created. While complex and interesting, in order to provide greater level of control over system shape, additional operations were required. After the initial generation, user is provided with additional methods for further edition of the cave system shape (either on single tile, or on the system as a whole). The resulting image can also be modified in external application.

For the edition stage (apart from manually drawing over created system), few basic operations are available, so the final shape will better meet the user requirements. Individual methods work as follows.

- Run CA – performs single run of cellular automata (CA) algorithm for the current system in the classical form.
- Combine with sketch – removes all disconnected elements, leaving only those, that are connected to the main system; method was added because methods “Remove CA shapes” and “Thin system edge” can produce unattached shapes.
- Add CA shapes – generates new CA shapes and connects them to the existing system.
- Remove CA shapes – generates new CA shapes and removes them from the existing system.
- Fill system edge – runs along the edge of existing system (edge is defined, when a cell has at least one neighbour, that is not classified as system shape or white), and expands the shape by drawing circle at each edge cell.
- Thin system edge – runs along the edge of existing system and thins it by changing the cell values to wall (black).

Results for single run of each method are presented in Figure 3.

Shape generation using cellular automata are the longest part of the presented approach. Because of that, all of the operations can be done either on all the tiles in the system or on individual regions. This will also address the situation, when user likes the general look of the structure, but wants to improve its individual fragments, without regenerating everything. Example of full system shape obtained during this stage of generation is presented in Figure 4.

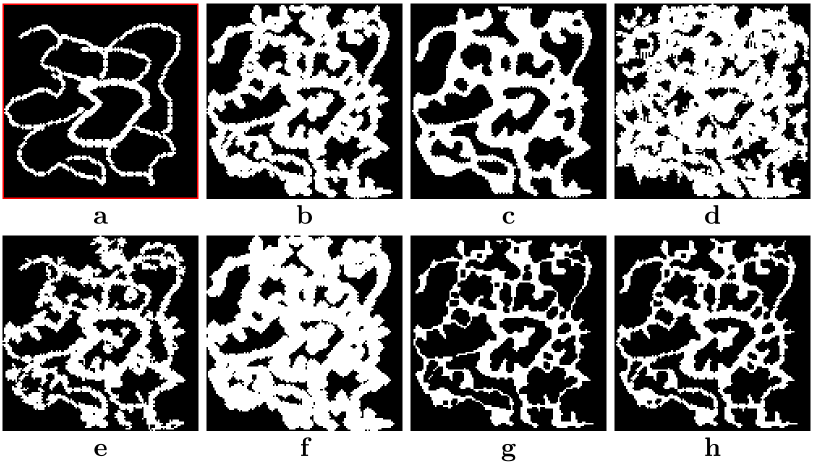


Fig. 3. Results of individual operations on single tile. (**a**) First two images show original system sketch, (**b**) additionally outlined in red and initially generated shape. The following images present results of each operation on (b); respectively: (**c**) Run CA, (**d**) Add CA shapes, (**e**) Remove CA shapes), (**f**) Fill system edge and (**g**) Thin system edge. Final image (**h**) shows result of “Combine with sketch operation”, performed on the shape presented in (g).

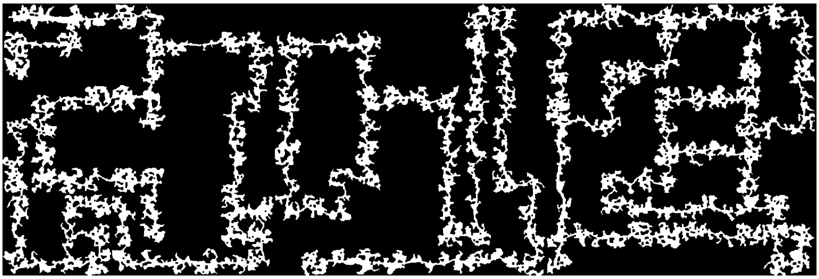


Fig. 4. Initial system shape generated from input maps presented in Figures 1 (top) and 2. Single tile size is set at 51×51 pixels.

4.2. Diamond-square height generation

After the initial generation, each tile contains flat representation of the system. Next step involves generation of height map, to include in that shape. Initial placement map will spread the tiles vertically, while this step will add required details to each area. The diamond-square algorithm is used in that aspect. This method was chosen, since the general direction of the height map can be steered, by changing the initial values placed in the corners.

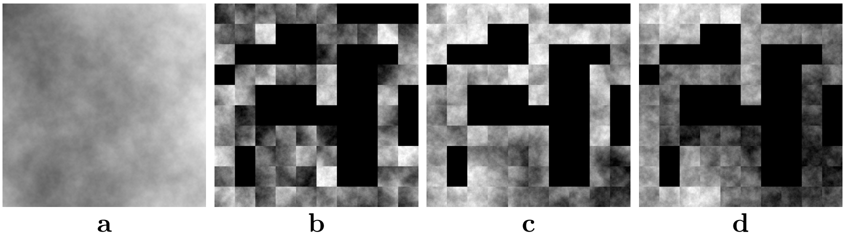


Fig. 5. Height maps generated using the diamond-square algorithm with different corner settings: (**a**) for the entire level without tile division; for single tiles, with initial corner values: (**b**) generated randomly, (**c**) calculated from the initial tile placement for three tiles adjacent to each corner, and (**d**) obtained from the two tiles adjacent to the corner for the edges with connections.

Since depending from the final application, different properties of the terrain might be necessary, few variations of the height map generation algorithm are considered:

1. for the entire level, for each of the defined levels;
2. separately for each tile;
3. for each tile, with initial values for the corners calculated from the placement of the neighbouring tiles;
4. for each tile, with initial values for the corners calculated from placement of neighbouring tiles, along connected edges.

For the first two cases the initial heights of the corners are generated randomly, the only difference being the map size. Diamond-square method requires the size of image equal to $2^n + 1$. Firstly the appropriate size is obtained, by finding smallest possible value, that will allow creation of either single tile, or entire level. The corner values are then randomly generated and the algorithm proceeds, with final map selected from subset of the generated one.

In the third case, initial values are calculated by averaging height of neighbouring tiles connected to each corner. The method takes initial tile heights obtained from placement map, and average is calculated according to the number of elements in the final set. For the corners, that have no neighbours, random height value will be taken.

Final method takes into account connections between tiles. In that case, the method first checks if the connection exists along each of the two edges connected to the current corner. Only if such connection exists, the neighbouring tile height is added to the average. This method was used to better reflect the initial structure of the cave system obtained from the input maps. Example height maps generated for each version of the diamond-square algorithm are presented in Figure 5.

4.3. Point-cloud visualization

Final step of the generation process is the visualization in 3D space, using Python Point Processing Toolkit library (pptk) [35].

During first step of this procedure, heights obtained from the diamond-square algorithm are used to update placement of each point according to those values. Single point in the final representation corresponds directly to individual pixels in tiles. Point Processing Toolkit requires a list containing 3D coordinates, so firstly the algorithm makes sure, that the value from height map is transferred to tile points accordingly. With this initial operation done, the algorithm then focuses on connecting the tiles and preparing final list of points for visualization.

Since the tiles are initially placed in 3D space using the data from placement map, transferring the position from the generated height map would result in gaps between system fragments. Therefore, before visualization, the point height values along the edges are updated in each tile to correct that problem. At the same time values are transferred to the format required by the used library. The points at the edges of the tile are first tilted and then connected vertically, making sure that each transition will be included into final system. Since number of connections to the lower levels usually is much lower, than the total number of connections, those transitions are extracted into separate list and iterated through after the initial process is finished. The outline of the entire operation is presented in Algorithm 2.

Last step takes all the final heights obtained so far and visualizes them. Firstly, all points that are classified as wall (black value on the images) are excluded. Resulting data structure will contain all points organized in a list, including final placement of each point in 3D coordinates. Points are then visualized in pptk tool, using 'gray' colour map – as a result higher points have whiter values in the gray-scale range, better showing the structure of the entire system.

Depending from the generation type chosen for the diamond-square algorithm, the effects of the final visualization will vary, since the heights obtained from that algorithm are directly transferred to the final point set. Overall height is scaled to the appropriate range, obtained from level spread.

Algorithm 2 Point coordinates preparation for the visualization process

```
for tiles in system do
    Check tile connections
    for connections in tile do
        Tilt tile along connected edge.
        Connect tile to the neighbour
    end for
end for
for connection in level connections do
    Connect tile to the neighbour in lower level.
end for
```

5. Results and discussion

In order to evaluate algorithms presented in this paper, method implementation was prepared using Python programming language. All tests were performed on a personal computer with Intel® Core$^{\text{TM}}$ i7-9750H CPU 2.60GHz, with Nvidia GeForce RTX 2070 graphics card and 32GB of RAM.

First part of evaluation was mainly visual in nature, comparing generated shapes to natural caves, and evaluating potential game requirements in that aspect. Cave structures in real world usually will contain spaces with characteristic shapes, big enough for a person to enter. Layout can be quite complicated, often with multi-layer structure and differently shaped passages of various length, height and general structure. It is caused by characteristics of the formation process, where acidic water first dissolves the rock, to wash it out later on [33]. Depending on the types of rocks and their strengths in different places, large structures can be placed next to the tight passages with various formations. All of those elements are important from gameplay point of view, providing additional challenge for the player. Initial structure containing such elements can be achieved with the system sketch. Larger areas can be easily defined by the designer, as well as any type of desirable layout. At the same time, while the generated shapes follow the sketch, they are not repetitive, and provide interesting areas to traverse, similar in their structure to the natural caves (see Figure 2, and the resulting map in Figure 4).

Another set of parameters concerned the cave-like structures presented in various computer games. In general few key features of used cave systems can be defined:

- system shape should be controllable;
- the system must contain only connected areas;
- system should contain side-spaces where potential enemies or in-game objects can be hidden;
- system should contain narrow passages, that can be blocked by enemies;
- system layout should be complex enough, that it poses challenge for the player, but at the same time is possible to represent on 2D map.

Underground structures generated by the presented method have all of the above features. Thanks to the used input maps, even very complex layouts can be represented, ensuring that the designers idea will transfer to the resulting structure. At the same time it can be edited, both during the generation, as well as after it, in other applications. Algorithms incorporated in the process can create natural-looking systems, with numerous niches and narrow passages placed along user-defined sketch. It is also ensured, that any areas in the system will be connected to the main passage, so there is no problem with inaccessible spaces. Examples of multiple maps obtained from single sketch are presented in Figure 6. As can be seen, depending from initial user outline, resulting system can contain complex structures with various features, while retaining

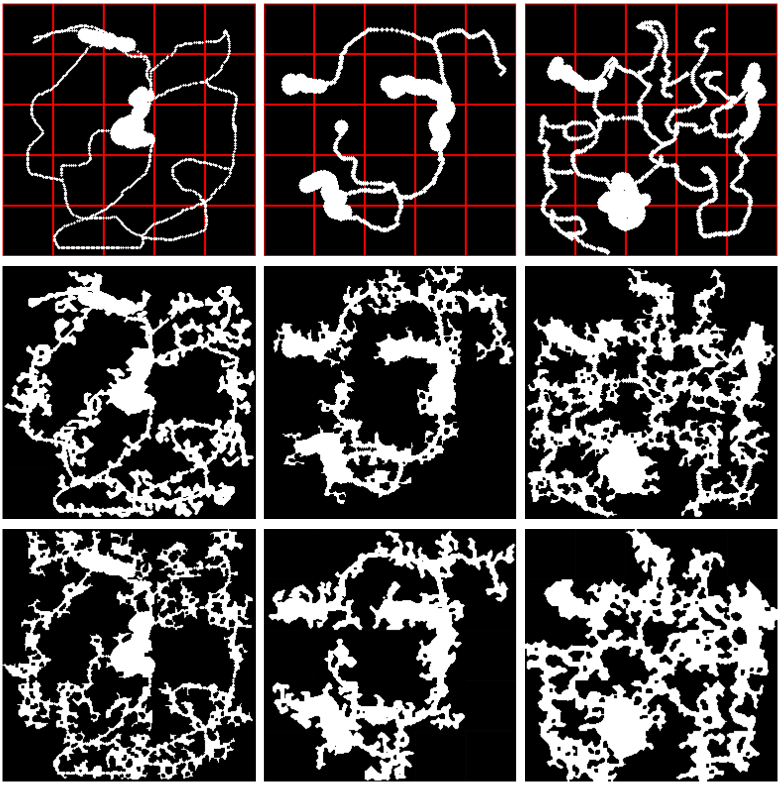


Fig. 6. Examples of used maps (top row) and two variations of resulting cave systems (remaining rows). The map consisted of single level 5×5 tiles, with tile size set at 51 pixels. The generation of each variation, including additional operations (like thinning/filling system, adding CA to tiles, etc.) took under one minute.

key design elements. At the same time, different variations of the same system can be easily produced.

Remaining tests concerned the overall method performance. Since presented approach is mainly focused on speeding up the designers workflow it needs to compute relatively fast. While some waiting time is acceptable, it should also be short enough to provide user with mostly real-time interaction. If the designer needs to wait long hours for the result, only to decide that it is not satisfactory, such approach will not be sufficient. Figure 7 presents generation times for consecutive steps of used method, under various conditions.

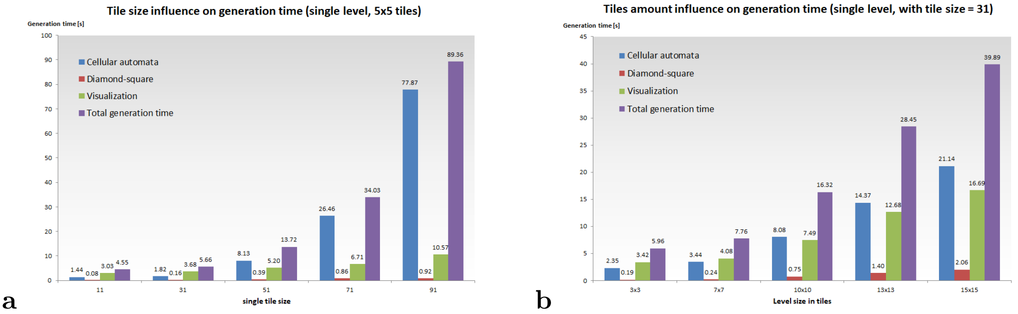


Fig. 7. Generation times obtained for prepared methods. (**a**) Key steps of prepared algorithm are evaluated for increasing tile size, and (**b**) number of tiles in level. Each result was averaged from data obtained during 100 iterations.

In general, increasing tile size has greater impact on the generation time than higher number of tiles. This is mostly due to used multiprocessing method, where each region is generated separately. It can be seen, that for larger tiles, the main factor during generation was the cellular automata algorithm, used to obtain system shape. This changes with larger number of tiles, where CA algorithm and visualization have closer values. Since the visualization part can be omitted, i.e. when creating data for other applications, this might be a better solution in those cases.

Final tests checked the influence the type of height-map in the diamond-square algorithm has on generation time. This test was done for single level, 5×5 tiles, with tile size set at 91 pixels. Results for the tile based methods were very similar, reaching 1.01, 1.01, and 0.97 s, for height generation methods 2, 3 and 4, respectively (see Section 4.2). Since in all cases only the corner values are changed, it is only natural, that the differences are minimal. The only real influence lies with the method that generates single height-map for the entire level (method 1), and this is again, due to the used multiprocessing method, and larger size of single item in this case.

Overall, obtained times are more than satisfactory. The method works close to real time, and even larger levels can be generated relatively quickly, depending from their structure (i.e. single level, 15×15 tiles with size equal to 31 takes less than 40 seconds to generate and visualize). While not fully real-time, it still provides user with possibility to obtain different variations for the chosen layout at reasonable intervals. At the same time, the generation is fast enough, that the system can be edited and improved close to real time – especially when user wants to change few tiles, instead of the entire system. Example visualization of fully generated, three-level cave system, with a close-up of one of its fragments is presented in Figure 8.

While presented algorithm in its current form has large potential, there are still few areas for improvements, that can be addressed in future work. Firstly, visualization in applications such as Blender, Unity or Unreal Engine might bring additional insight in

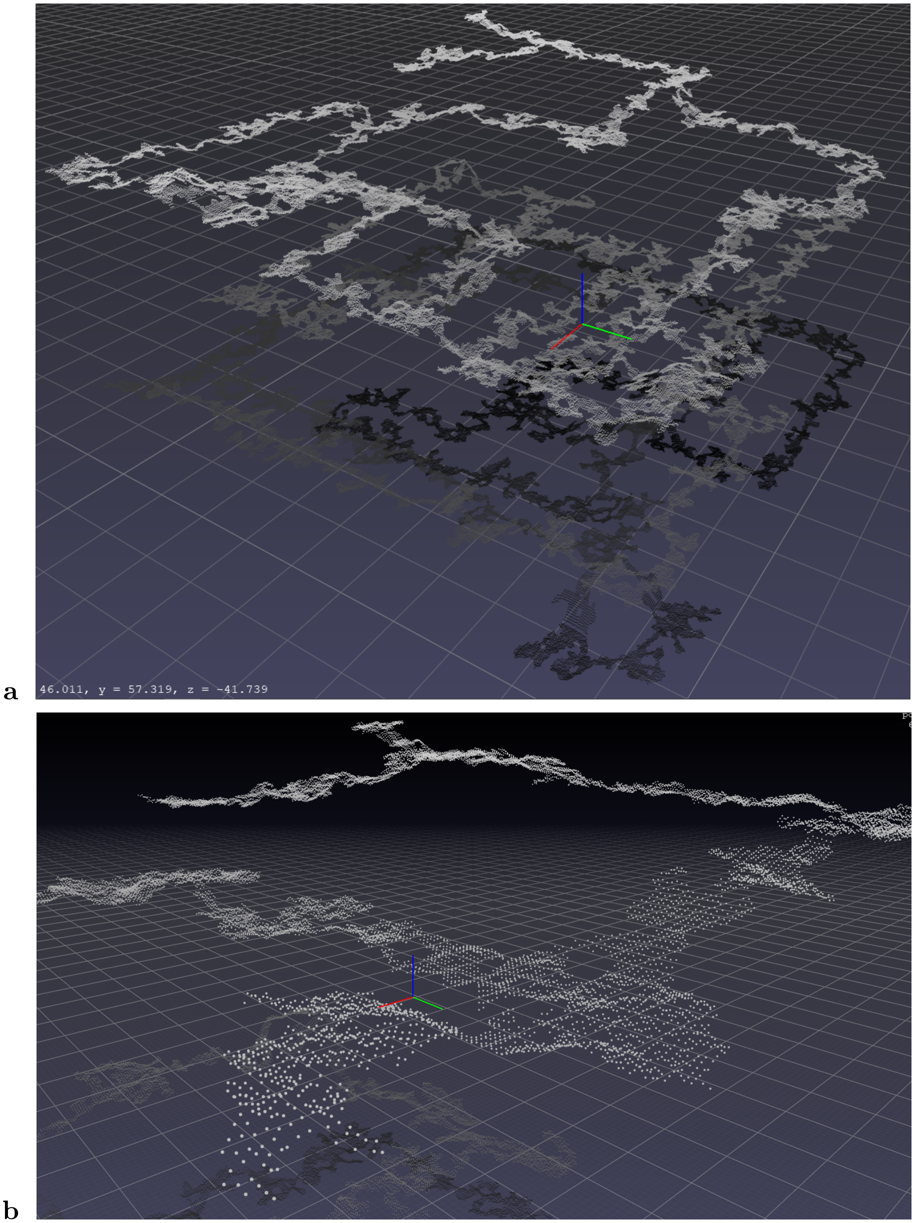


Fig. 8. Example point-cloud visualization of the system (**a**) after the full generation process, (**b**) with close-up of system fragment. The system was generated from input maps from Figs. 1 and 2.

terms of general requirements for used methods. 3D modelling environments and game engines have different specificity, and might require additional changes to the algorithm stages. Creating a method for more interactive visualization might also be interesting, i.e. in the Unreal Engine. Another potential area concerns time improvement for larger tiles. While even very complex systems can be represented with smaller elements, some user might still want to use larger areas. Finally, further edition of point cloud generated by the method would be required. Creating mesh, or using prefabs to build a game level based on generated points is an interesting, and most logical next step in that aspect.

Overall, the presented approach is very promising, with various possibilities for development. It can be adapted to different games, and speeds up the design process. At the same time, with minimal additions it can generate full terrain for a simple game.

6. Conclusion

In this paper a procedural layout generation algorithm for multi-level, complex, cave-like systems was presented. Described method uses two schematic maps, one for placing tiles inside each level, and one for defining general layout and system transitions. Results can be represented and visualized in 3D space.

Since the main method application is focused on computer games, different properties were taken into account. The designer can not only define the layout of the entire system, but also correct obtained results at different stages of the generation. The method works at an interactive rate, producing complex layouts in reasonable time (i.e. generating and visualizing system with single level 15×15 tiles with size equal to 31×31 takes 39.89 s). With all those elements taken into account, it can be used either for terrain generation in simple game, or during design process to quickly visualize different layouts. Use of schematic maps not only provides designer with a method to quickly and intuitively define desired results, but also allows fast creation of variations that share the same transitions. Since the results are presented as a point-cloud, it can be further edited and modified, either by transferring those points into 3D modelling environment, or creating levels using other methods, i.e., with predefined assets.

References

[1] D. B. Adams. *Feature-based Interactive Terrain Sketching.* Master's thesis, Brigham Young University, 2009. https://www.proquest.com/openview/ccd517e9097577329a8faf0d38327518.

[2] I. Antoniuk, P. Hoser, and D. Strzęciwilk. L-system application to procedural generation of room shapes for 3D dungeon creation in computer games. In: *Proc. Int. Multi-Conf. Advanced Computer Systems*, pp. 375–386. Springer, 2018. doi:10.1007/978-3-030-03314-9_32.

[3] I. Antoniuk and P. Rokita. Feature-based procedural generation of adjustable game content. *Challenges of Modern Technology*, 5(4):21–26, 2014. https://www.infona.pl/resource/bwmeta1.element.baztech-bd463454-b467-4eed-b4a7-a4f9b74a9d99.

[4] I. Antoniuk and P. Rokita. Procedural generation of adjustable terrain for application in computer games using 2D maps. In: *Pattern Recognition and Machine Intelligence: Proc. 6th Int. Conf. PReMI 2015*, vol. 9124 of *Lecture Notes in Computer Science*, pp. 75–84. Springer, Warsaw, Poland, Jun 30-Jul 2015. Article No. 10. doi:10.1007/978-3-319-19941-2_8.

[5] I. Antoniuk and P. Rokita. Generation of complex underground systems for application in computer games with schematic maps and L-systems. In: *Proc. Int. Conf. Computer Vision and Graphics*, pp. 3–16. Springer, 2016. doi:10.1007/978-3-319-46418-3_1.

[6] I. Antoniuk and P. Rokita. Procedural generation of underground systems with terrain features using schematic maps and L-systems. *Challenges of Modern Technology*, 7(3):8–15, 2016. doi:10.5604/01.3001.0009.5443.

[7] I. Antoniuk and P. Rokita. Procedural generation of multiclevel dungeons for application in computer games using schematic maps and L-system. In: *Intelligent Methods and Big Data in Industrial Applications*, pp. 261–275. Springer, 2019. doi:10.1007/978-3-319-77604-0_19.

[8] D. Ashlock, C. Lee, and C. McGuinness. Search-based procedural generation of maze-like levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):260–273, 2011. doi:10.1109/TCIAIG.2011.2138707.

[9] Blender Reference Manual, version 3.5. https://docs.blender.org/manual/en/latest/index.html, Accessed: 22.04.2023.

[10] M. Boggus and R. Crawfis. Explicit generation of 3D models of solution caves for virtual environments. In: *Proc. 2009 Int. Conf. Computer Graphics & Virtual Reality (CGVR)*, pp. 85–90. Las Vegas, Nevada, USA, 2009. http://web.cse.ohio-state.edu/tech-report/2009/TR18.pdf.

[11] M. Boggus and R. Crawfis. Procedural creation of 3D solution cave models. In: *Proc. 20th IASTED Int. Conf. Modelling and Simulation*, pp. 180–186. 6-8 Jul, Banff, Alberta, Canada 2009. https://www.actapress.com/Abstract.aspx?paperId=35085.

[12] M. Boggus and R. Crawfis. Prismfields: A framework for interactive modeling of three dimensional caves. In: *Proc. Int. Symp. Visual Comput.*, pp. 213–221, 2010. doi:10.1007/978-3-642-17274-8_21.

[13] J. Calderon. Dungeon Master. https://ka-plus.pl/en/dungeon-master/, Accessed: 18.12.2023.

[14] J. Cui, Y.-W. Chow, and M. Zhang. Procedural generation of 3D cave models with stalactites and stalagmites. *IJCSNS*, 11(8):94, 2011. https://ro.uow.edu.au/infopapers/3625/.

[15] D. M. De Carli, C. T. Pozzer, F. Bevilacqua, and V. Schetinger. Procedural generation of 3D canyons. In: *Proc. 2014 27th SIBGRAPI Conf. Graphics, Patterns and Images*, pp. 103–110. IEEE, Rio de Janeiro, Brazil, 26-30 Aug 2014. doi:10.1109/SIBGRAPI.2014.41.

[16] Dragon Age game series webpage. https://www.ea.com/en-gb/games/dragon-age, Accessed: 18.12.2023.

[17] Elden Ring game webpage. https://en.bandainamcoent.eu/elden-ring/elden-ring, Accessed: 18.12.2023.

[18] K. Franke and H. Müller. Procedural generation of 3D karst caves with speleothems. *Computers & Graphics*, 102:533–545, 2022. doi:10.1016/j.cag.2021.10.002.

[19] E. Galin, A. Peytavie, N. Maréchal, and E. Guérin. Procedural generation of roads. *Computer Graphics Forum*, 29(2):429–438, 2010. doi:10.1111/j.1467-8659.2009.01612.x.

[20] M. N. Gamito and F. K. Musgrave. Procedural landscapes with overhangs. In: *Proc. 10th Portuguese Computer Graphics Meeting*, vol. 2. Lisbon, Porugal, 2001.

[21] S. Greuter and A. Nash. Game asset repetition. In: *Proc. 2014 Conf. Interactive Entertainment*, pp. 1–5, 2014. doi:10.1145/2677758.2677782.

[22] S. Greuter, J. Parker, N. Stewart, and G. Leach. Real-time procedural generation of ‘pseudo infinite’ cities. In: *GRAPHITE ’03: Proc. 1st Int. Conf. Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 87–ff. ACM, Melbourne, Australia, 11-14 Feb 2003. doi:10.1145/604471.604490.

[23] Legend of Grimrock game webpage. https://www.grimrock.net/, Accessed: 18.12.2023.

[24] K. Hartsook, A. Zook, S. Das, and M. O. Riedl. Toward supporting stories with procedurally generated game worlds. In: *Proc. 2011 IEEE Conf. Computational Intelligence and Games (CIG’11)*, pp. 297–304. IEEE, 2011. doi:10.1109/CIG.2011.6032020.

[25] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013. doi:10.1145/2422956.2422957.

[26] R. Huijser, J. Dobbe, W. F. Bronsvoort, and R. Bidarra. Procedural natural systems for game level design. In: *Proc. 2010 Brazilian Symp. Games and Digital Entertainment*, pp. 189–198. IEEE, Florianopolis, Brazil, 08-10 Nov 2010. doi:10.1109/SBGAMES.2010.31.

[27] L. Johnson, G. N. Yannakakis, and J. Togelius. Cellular automata for real-time generation of infinite cave levels. In: *PCGames '10: Proc. 2010 Workshop on Procedural Content Generation in Games*, pp. 1–4. Monterey, California, 18 Jun 2010. Article No. 10. doi:10.1145/1814256.1814266.

[28] K. R. Kamal and M. Kaykobad. Generation of mountain ranges by modifying a controlled terrain generation approach. In: *Proc. 2008 11th Int. Conf. Computer and Information Technology*, pp. 527–532. IEEE, Khulna, Bangladesh, 24-27 Dec 2008. doi:10.1109/ICCITECHN.2008.4803058.

[29] A. Liapis. Multi-segment evolution of dungeon game levels. In: *Proc. Genetic and Evolutionary Computation Conference*, pp. 203–210, 2017. doi:10.1145/3071178.3071180.

[30] R. Van der Linden, R. Lopes, and R. Bidarra. Designing procedurally generated levels. In: *Proc. Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013. doi:10.1609/aiide.v9i3.12592.

[31] B. Mark, T. Berechet, T. Mahlmann, and J. Togelius. Procedural generation of 3D caves for games on the GPU, 2015. `https://portal.research.lu.se/en/publications/procedural-generation-of-3d-caves-for-games-on-the-gpu`, Paper presented at *Foundations of Digital Games*.

[32] E. A. Matthews and B. A. Malloy. Procedural generation of story-driven maps. In: *Proc. 2011 16th Int. Conf. Computer Games (CGAMES)*, pp. 107–112. IEEE, 2011. doi:10.1109/CGAMES.2011.6000324.

[33] A. N. Palmer. Origin and morphology of limestone caves. *Geological Society of America Bulletin*, 103(1):1–21, 1991. doi:10.1130/0016-7606(1991)103<0001:OAMOLC>2.3.CO;2.

[34] A. Peytavie, E. Galin, J. Grosjean, and S. Mérillou. Arches: A framework for modeling complex terrains. *Computer Graphics Forum*, 28(2):457–467, 2009. doi:10.1111/j.1467-8659.2009.01385.x.

[35] Documentation for Point Processing Toolkit (pptk) Python library. `https://heremaps.github.io/pptk/index.html`, Accessed: 22.04.2023.

[36] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer Science & Business Media, 2012. doi:10.1007/978-1-4613-8476-2.

[37] W. L. Raffe, F. Zambetta, and X. Li. Evolving patch-based terrains for use in video games. In: *Proc. 13th Annual Conf. Genetic and Evolutionary Computation*, pp. 363–370, 2011. doi:10.1145/2001576.2001627.

[38] T. Roden and I. Parberry. From artistry to automation: A structured methodology for procedural content creation. In: *Proc. Int. Conf. Entertainment Computing*, pp. 151–156. Springer, 2004. doi:10.1007/978-3-540-28643-1_19.

[39] A. Santamaría-Ibirika, X. Cantero, S. Huerta, I. Santos, and P. G. Bringas. Procedural playable cave systems based on Voronoi diagram and Delaunay triangulation. In: *Proc. 2014 Int. Conf. Cyberworlds*, pp. 15–22. IEEE, 2014. doi:10.1109/CW.2014.11.

[40] R. C. Silva, N. Fachada, D. De Andrade, and N. Códices. Procedural generation of 3D maps with snappable meshes. *IEEE Access*, 10:43093–43111, 2022. doi:10.1109/ACCESS.2022.3168832.

[41] The Elder Scrolls V: Skyrim, game webpage. `https://elderscrolls.bethesda.net/pl/skyrim10`, Accessed: 18.12.2023.

[42] R. Smelik, K. Galka, K. J. De Kraker, F. Kuijper, and R. Bidarra. Semantic constraints for procedural generation of virtual worlds. In: *PCGames '11: Proc. 2nd Int. Workshop on Procedural Content Generation in Games*, pp. 1–4, 2011. Article No. 9. doi:10.1145/2000919.2000928.

[43] R. Smelik, T. Tutenel, K. J. De Kraker, and R. Bidarra. Integrating procedural generation and manual editing of virtual worlds. In: *Proc. 2010 Workshop on Procedural Content Generation in Games*, pp. 1–8, 2010. doi:10.1145/1814256.1814258.

[44] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes. A survey on procedural modelling for virtual worlds. *Computer Graphics Forum*, 33(6):31–50, 2014. doi:10.1111/cgf.12276.

[45] R. M. Smelik, T. Tutenel, K. J. De Kraker, and R. Bidarra. Declarative terrain modeling for military training games. *International journal of computer games technology*, 2010, 2010. doi:10.1155/2010/360458.

[46] R. M. Smelik, T. Tutenel, K. J. de Kraker, and R. Bidarra. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 35(2):352–363, 2011. doi:10.1016/j.cag.2010.11.011.

[47] R. M. Smelik, T. Tutenel, K. J. de Kraker, R. Bidarra, et al. A proposal for a procedural terrain modelling framework. In: *EGVE (Posters)*, 2008.

[48] Speedtree documentation. `https://docs9.speedtree.com/`, Accessed: 22.04.2023.

[49] Home of Unity Technologies project. `https://unity.com/`, Accessed: 22.04.2023.

[50] Unreal Engine webpage. `https://www.unrealengine.com/en-US/`, Accessed: 22.04.2023.

[51] Unreal Engine 5.2 Roadmap. `https://portal.productboard.com/epicgames/1-unreal-engine-public-roadmap/tabs/85-unreal-engine-5-2`, Accessed: 22.04.2023.

[52] V. Valtchanov and J. A. Brown. Evolving dungeon crawler levels with relative placement. In: *C3S2E '12: Proc. Fifth International C* Conf. Computer Science and Software Engineering*, pp. 27–35. Montreal, Quebec, Canada, 27-29 Jun 2012. doi:10.1145/2347583.2347587.

[53] B. M. F. Viana and S. R. dos Santos. Procedural dungeon generation: A survey. *Journal on Interactive Systems*, 12(1):83–101, 2021. doi:10.5753/jis.2021.999.

[54] The Witcher game series webpage. `https://www.thewitcher.com/pl/en/`, Accessed: 18.12.2023.

[55] H. Yin and C. Zheng. A parametrically controlled terrain generation method. In: *Proc. 2012 7th Int. Conf. Computer Science & Education (ICCSE)*, pp. 775–778. IEEE, 2012. doi:10.1109/ICCSE.2012.6295187.

[56] Y. Zhang, G. Zhang, and X. Huang. A survey of procedural content generation for games. In: *Proc. 2022 Int. Conf. Culture-Oriented Science and Technology (CoST)*, pp. 186–190. IEEE, 2022. doi:10.1109/CoST57098.2022.00046.

Izabella Antoniuk received her MS from the Faculty of Applied Informatics and Mathematics at the Warsaw University of Life Sciences in the field of Computer Science, specializing in Multimedia Applications. In the same year began doctoral studies at the Faculty of Electronics and Information Technology of the Warsaw University of Technology. She obtained PhD in the field of: Technical Sciences and in the discipline: Computer Science in December 2018, with thesis entitled "Procedural generation of an editable three-dimensional terrain model based on selected features and schematic two-dimensional maps". In addition to working at the Warsaw University of Life Sciences she cooperates with other universities, as well as with business, implementing the EU applications. Her research interests include procedural content generation and advanced algorithms for use in computer games, artificial intelligence, machine learning, and data analysis.

Assessment of the Possibility of Imitating Experts' Aesthetic Judgments about the Impact of Knots on the Beauty of Furniture Fronts Made of Pine Wood

Krzysztof Gajowniczek[1], Marcin Bator[1,*],
Katarzyna Śmietańska[2], and Jarosław Górski[2]
[1] *Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*
[2] *Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*
[*] *Corresponding author: M. Bator (marcin_ bator@sggw.edu.pl)*

Abstract. Our research aims to reconstruct expert preferences regarding the visual attractiveness of furniture fronts made of pine wood using machine learning algorithms. A numerical experiment was performed using five machine learning algorithms of various paradigms. To find the answer to the question of what determines the expert's decision, we determined the importance of variables for some machine learning models. For random forest and classification trees, it involves the overall reduction in node impurities resulting from variable splitting, while for neural networks it uses the Garson algorithm. Based on the numerical experiments we can conclude that the best results of expert decision reconstruction are provided by a neural network model. The expert's decision is better reconstructed for more beautiful images. The decision for nice images is made based on the best 4 or 5 variables, while for ugly images many more features are important. Prettier images and those for which the expert's decision is better reconstructed have fewer knots.

Key words: image processing, knots on the fronts, machine learning, preference learning, solid wood furniture, quality control, importance of variables.

1. Introduction

Computer aided control of product quality in furniture manufacturing has been a relevant research problem for a long time. It turned out that image processing can be an effective way of automatically ensuring whether different types of wood products conform to set specifications [37]. However, without a doubt, the quality of wooden furniture has not only objective, but also subjective aspects – especially aesthetic value, in which case, current computer technology is not enough. Wood, being a product of nature, is not homogeneous. It contains natural features, often formally recognized as defects (knots, cracks, defects in shape, color, even mechanical damage), for many people can be an advantage, thanks to which this material gains its unique, individual character. Unlike objective characteristics that can be defined by standards (for example, dimension, strength properties), the only reliable measure of product quality in this case seems to be subjective human assessment. Therefore, a key issue in the case of wood furniture is to know the human preferences, which are additionally subject to constant change due to

Machine GRAPHICS & VISION 32(2):67–88, 2023. DOI: 10.22630/MGV.2023.32.2.4 .

product innovations and changing lifestyles [31]. Such knowledge should be a fundamental step towards improving the production process in furniture factories, consequently striving for automation in those features identification and classification [4,15,17]. Evaluation and quantification of furniture aesthetic value played an important role in the furniture design, and it is one of the difficult issues in this research field [5,34]. For example, nowadays you can easily use image processing algorithms to control the number of knots visible on a furniture front made of solid wood, but first you should know what the effect of this number on the relative, aesthetic attractiveness of this front is. Firstly, in order to know human preferences for wooden furniture, an adequate group of people should be examined [13].

In research where aesthetic values are concerned, it is common practice to use experts in a given field [1,6,14]. Unfortunately, we do not find many scientific studies presenting data on the real specific aesthetic preferences of furniture made of wood. Research conducted so far is mostly of marketing nature and does not focus on specific aesthetic features. However, they clearly emphasize that one of the most important factors influencing the purchase decision is the aesthetic value, including, among others, design or color [12,16,17,32]. The aesthetic functions (e.g. visual sensation, emotions), determinative forms and fashionable style play a very important role in furniture design and production [1,14].

The previous research [36] proves that there is a relationship between the occurrence of knots, their size and location, and aesthetic impressions. These studies used sets of questionnaires (based on standard 5 point Likert scale) for various groups of experts, and the analysis of the results included the proposed features describing the characteristics of each image. This study raised a question about the adequacy of these features for assessing aesthetics. Hence, numerous experiments in the field of machine learning were carried out to verify whether, using the features defined there and their values, it is possible to recreate the classification method (assessment method) performed by individual people or groups of people.

This research is part of the development of an automatic furniture front evaluation system. Its elements are image acquisition, image analysis (calculation of features), classification (evaluation). As in many tasks, a good approach is to complete the modules from the final module to the initial one, rather than from the initial one to the final one. In other words, we first answer the question of what features are important, then how to calculate them in the image, when we know what quality of the image is needed, only then the image acquisition begin.

Preference modeling can be done in three ways [19], treating each opinion independently as a separate class without maintaining order relations between ratings, as an ordinal regression/classification problem or using learning to rank approach [7,29]. Unfortunately, the second approach has produced only a few extensions to existing machine learning algorithms. The third approach assumes that we have all the images together and rank them among themselves [8, 25, 30], but in our study we evaluate each image separately on an ordinal scale, so this approach is also not applicable to us. Due to that, in our paper, we will use the first approach because the results provided by the second approach are only slightly better. However, what is more important, the first approach allows us to use and compare machine learning algorithms of various paradigms and use methods deriving the importance of variables.

To name the subject of this paper, we will use the terms reconstruction, reproduction, mapping, and classification interchangeably. Therefore, our research aims to reconstruct expert preferences regarding the visual attractiveness of furniture fronts made of pine wood using machine learning algorithms. The numerical experiment will be performed using five machine learning algorithms, i.e. classification trees [3], artificial neural networks [22], k-nearest neighbors [28], random forest [2] and support vector machines [23]. From this point on, we will understand the label l as the true rating R (based on the Likert scale) given by an expert to a given image and its predicted value by a predictive model or benchmarking method. To achieve the intended research goal, we defined the following research questions:

1. Which machine learning-based model delivers the best results of reconstruction of the expert's preferences?
2. Is there a relationship between the quality of reconstruction of the expert's preferences and the beauty of the image?
3. Which group of experts is the best and worst reproducible?
4. Which features matter in the structure of a given machine-learning model?
5. What feature values characterize the best and the worst reproduced images?

The remainder of this paper is organized as follows. Section 2 presents the data used in this study. In Section 3, a detailed description of the methodology of the numerical experiment is presented. Section 4 provides the results of the numerical experiment. The paper ends with a discussion regarding the results and concluding remarks in Section 5.

2. Data characteristics

The furniture elements analyzed in the research was a furniture the front (600 mm × 600 mm) of the single-door cabinet made of solid wood. Each front was divided into 5 zones with surface area, but different location and shape (i.e. upper left, upper right, bottom left, bottom right and central). All of them were prepared by an expert based on visual analysis of knots presented on images. A subset of them was previously presented in [36]. A key question is "is it possible to reconstruct an expert preferences using those features based on knots position and size, or some other information from an image is needed?" In this study the condition question: "what kind of information could be also needed?" is not set. The focus was on arranging a selected set of knots in various, precisely defined configurations. This allowed us to eliminate the influence of unnecessary features that are irrelevant to the research, such as the shape or color of knots. The aim of the experiment was to examine the impact of features of furniture fronts on their aesthetic quality in the opinion of the judges. Sample images are presented in Fig. 1 and Fig. 2. The list below presents the features with their description (with abbreviations used later on):

1. The number of knots (`qty1`).
2. Evenness of the number of knots (`qty2`): 0=odd; 1=even.
3. Four classes of the number of knots (`qty3`): 0=none, 1=small, i.e. from 1 to 2 knots; 2=medium, i.e. from 3 to 5 knots; 3=many, i.e. from 6 to 8 knots).
4. Dispersion (`disp1`): 0=concentrated in a specific zone; 1=dispersed, i.e. located in as many zones as possible.
5. Presence of at least one knot in the central zone (`pos1`): 0=not present; 1=present.
6. Presence of at least one knot in the upper left zone (`pos2`): 0=not present; 1=present.
7. Presence of at least one knot in the upper right zone (`pos3`): 0=not present; 1=present.
8. Presence of at least one knot in the bottom left zone (`pos4`): 0=not present; 1=present.
9. Presence of at least one knot in the bottom right zone (`pos5`): 0=not present; 1=present.
10. Presence of at least one knot in the bottom zone (`pos6`): 0=not present; 1=present.
11. Presence of at least one knot in the upper zone (`pos7`): 0=not present; 1=present.
12. Presence of at least one knot in the left zone (`pos8`): 0=not present; 1=present.
13. Presence of at least one knot in the right zone (`pos9`): 0=not present; 1=present.
14. Knot size overall (`siz2`): 0=no knots; 1=only small knots; 2=both small and large; 3=only large knots.
15. Size of knots in detail (`siz3`): 0=no knots; 1=only small ones; 2=both small and large but more small ones; 4=both small and large but more large ones; 5=only large ones.
16. Symmetry (`sym1`): 0=not present; 1=present.

The study employed the standard consensus-based assessment (CBA) technique. Four expert groups, totaling 50 participants, were invited to take part in the experiment. The selection of experts is extremely important. Based on the literature review (in chapter 1), it can be considered a sufficient or relatively large number. Below list presents group of experts with their description (with later used abbreviation):

1. The first group (Art), consisted of 12 interior design professionals, with ages ranging from 28 to 61 from the Academy of Fine Arts (Faculty of Interior Design) in Warsaw and the University of the Arts in Poznan (Department of Furniture Design and Department of Interior Design).
2. The second group of experts (WTD), comprised 12 individuals specializing in furniture design. They were members of the research and teaching faculty at the Institute of Wood Sciences and Furniture within the Warsaw University of Life Sciences (WULS). Their ages ranged from 30 to 56 years old.

3. Group 3 (Std) comprised 14 students from the Department of Furniture Manufacturing (WULS, Faculty of Wood Technology). These individuals were considered semi-professionals and fell within the age range of 20 to 21 years old.
4. Additionally, for comparative purposes, an entirely non-professional group of experts (WWa) was included. This group comprised 12 individuals, aged between 22 and 82, randomly selected from Warsaw residents. They were chosen based on their belief in their knowledge of furniture and their strong motivation to participate in scientific research.

The special designers method such as Questionnaire research was used in the study. All groups of judges (50 people in total) were asked to fill out an online questionnaire based on standard 5 point Likert scale (1-strongly disagree, 2-disagree, 3-neutral, 4-agree, 5-strongly agree) which contained 99 questions about each image. Each of them was the same: "Do you agree that furniture front shown in the photo above is more attractive than others?".

3. Research methodology

3.1. Machine learning based models

All simulations were prepared using R software [20] (version 4.3) and corresponding libraries implementing certain machine learning algorithms. The main infrastructure was the *caret* package [18] (short for *C*lassification *A*nd *RE*gression *T*raining) which is a set of functions aimed at improving the process of creating various predictive models. All algorithms were trained in the standard state-of-the-art cross-validation regime using a leave-one-out approach and checking various combinations of the hyper-parameters. To gain better numerical stability and have variables on comparable scales (which is required for some algorithms), the data were normalized using standardization. Each model was trained for the classification problem. The potential explanatory variables $\mathbf{x}$ were the variables described in the list 2 and the target variable was the expert's rating, $y \in \{1, 2, 3, 4, 5\}$.

The rpart package, implementing the CART (Classification and Regression Trees) algorithm [3], was utilized to train classification trees. Throughout the process of partitioning a multi-dimensional space, the criterion focused on minimizing the Gini impurity of the dependent variable for observations within the same leaf node. The node had a minimum requirement of 20 observations, and a leaf needed at least 6 observations to avoid further splitting. Rather than pruning the tree at the algorithm's conclusion, we employed a pruning technique during the tree's growth phase. This method halted the creation of new splits when prior splits only marginally improved predictive accuracy. The complexity parameter *cp* was tested using the following values 0, 0.001, 0.005, 0.01, 0.05, 0.1, and 0.2. The tree was constructed up to a depth of 30 levels.

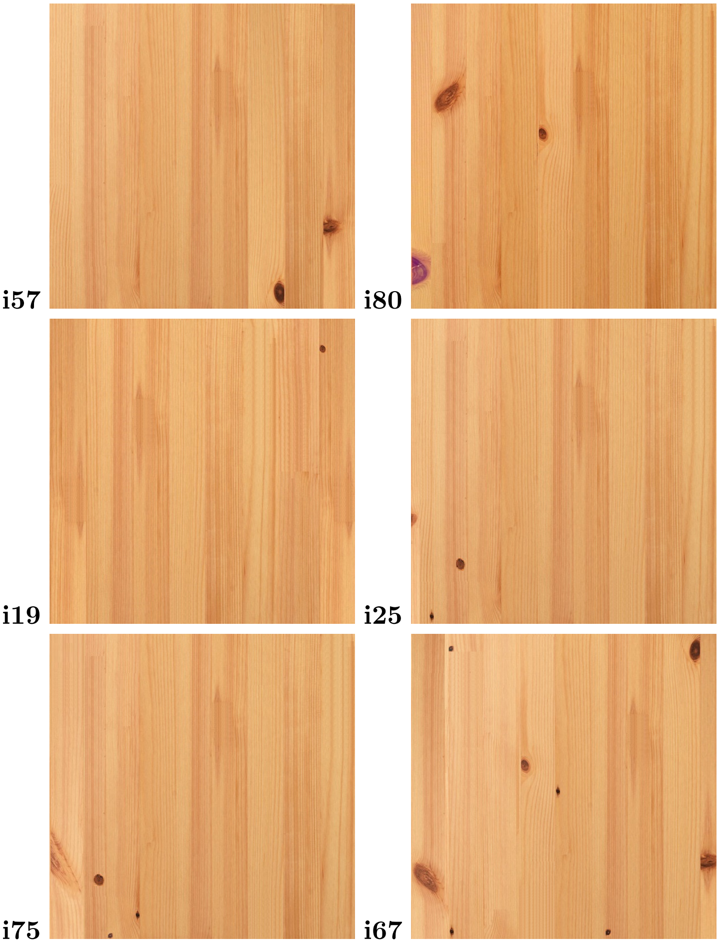


Fig. 1. Examples of images with the best (left-hand side) and the worst (right-hand side) classification results.

To train the neural networks, we used the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm [9,10,22], which belongs to the broad family of quasi-Newton optimization methods (available in the nnet library). Each neural network consisted of one input layer with 16 neurons (one for each feature), one hidden layer (a different number of neurons was tested, i.e. 1, 5, 10, 15, 20; *size* parameter) and one output layer with five neurons (one neuron for one class). The target feature was decoded using one-hot encoding, i.e. a matrix with five columns in which the number one indicates the true label, keeping

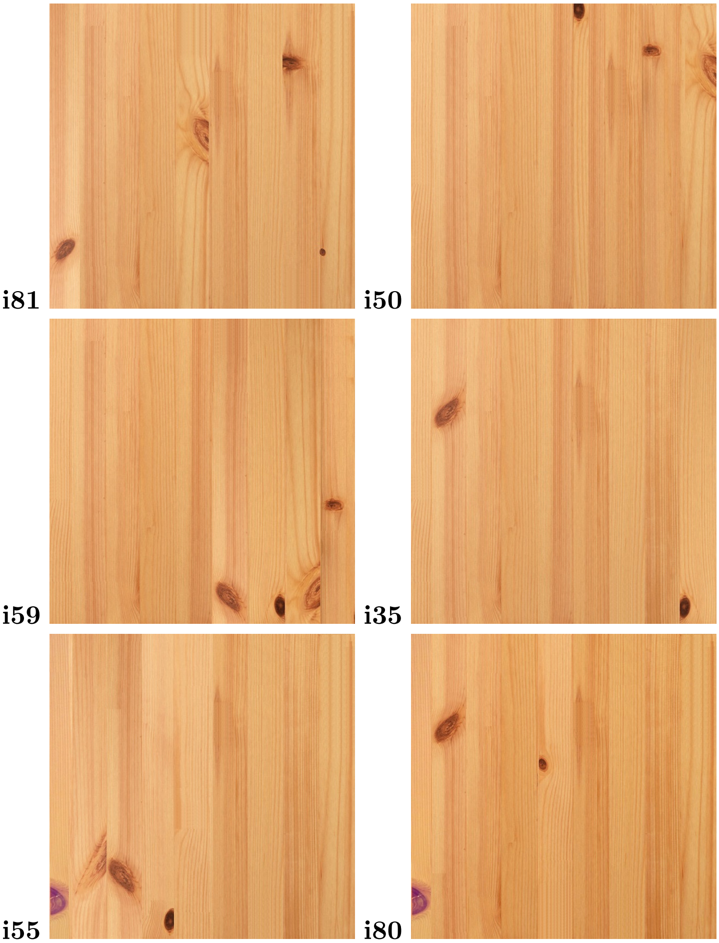


Fig. 2. Examples of the prettiest (left-hand side) and the ugliest (right-hand side) images.

zero for all other labels. A sigmoid function was used to activate all of the neurons in the network. To prevent overfitting the regularization term (weight decay) is employed which uses as the penalty the sum of squares of the weights was set at 0, 0.01, 0.5, and 0.1 (decay parameter). The maximum number of iterations was set at 200 with the stopping criterion set at $1.0e - 4$.

Classification using k-nearest neighbors algorithm [28] was performed using knn3 function from the caret library. Different numbers of the k-values (k parameter) were proposed in the experiments including the following: 1, 5, 10, 15, 25, 50, 75.

The random forest [2] was trained using an algorithm sourced from the randomForest library. Preceding each training session, samples comprising n elements were drawn with replacement, representing around 63% of the population. These samples were employed to create $CART$ trees, each tree being constructed to its full size without any pruning, ensuring that no leaf contained 5 or fewer observations. The count of variables selected randomly as potential candidates at each split varied from 4 to 16 by 4 ($mtry$ parameter). The total number of trees in the forest was 500.

For building the support vector machine [23], the `ksvm` function from the kernlab library was employed, utilizing its Sequential Minimal Optimization algorithm. This algorithm was utilized to address the quadratic programming problem involved in the process. The radial basis was used as a kernel function, with `sigma` set at 0.01, 0.05, 0.1, 0.5 and 1. The regularization parameter `C` which controls the over-fitting, was arbitrarily set, and the simulations were run for the following values: 0.1, 0.5, 1, 2, and 5.

Importance of variables was calculated using `VarImp` which is a generic model-specific method. For the random forest and classification trees it involves the overall reduction in node impurities resulting from variable splitting (for rf averaged across all trees). The node impurity was assessed using the Gini index. For neural networks, it uses the Garson algorithm [11] which delineates the relative significance of features by dissecting the model weights. Assessing the relative importance (or strength of association) of a particular feature for the response variable involves identifying all weighted connections between relevant nodes. This encompasses pinpointing all weights that link the specific input node, traverse through the hidden layer, and culminate at the response variable.

3.2. Benchmarking methods

To examine the quality of the developed models, we proposed two benchmarking methods. Both indicate a baseline obtained through a kind of naive forecast. The first method uses a median value of the true labels for all 99 images for a given expert. Usually, it is a label of 3, but for some experts, it is a 2 or 4 because some experts have not given any image a label of 3 or other labels. The second approach employs random values from the empirical distribution for a given expert. In other words, let's assume that for the event space $\Omega = \{1, 2, 3, 4, 5\}$ the frequency (across all 99 images) of an expert selecting a given label (l) is $P(l = 1) = 0.10$, $P(l = 2) = 0.30$, $P(l = 3) = 0.20$, $P(l = 4) = 0.35$ and $P(l = 5) = 0.05$. Then as a predicted label for a given image, we take random value taking into account the aforementioned distribution.

3.3. Prediction quality measure

Instead of a standard accuracy measure where only exact matching (perfect prediction) increases accuracy, we have used Absolute Accuracy Error (AE). It provides robust information about how far the model predictions are from the original labels. This kind of measure is crucial for our analysis because it provides deep insight into the distribution of errors. Our intuition was that importance of variables in the model depends on the error magnitude taking into account the true label. For example, we have assumed that different variables are important when: 1) the true label is 5 and the error is 2; 2) the true label is 5 and the error is 0; 3) the true label is 2 and the error is 1; etc. The measure is defined as follows:

$$\mathrm{AE} = \sum_{i=1}^{n} |t_i - p_i| \,, \tag{1}$$

where n is the number of images, t is the true label and p is the predicted label.

3.4. Additional configuration and notation

To be able to reproduce the simulation study, the initial value (`set.seed` function) of the pseudo-random number generator (Mersenne-Twister algorithm) was set for both machine learning models and benchmarking methods.

To make the message of the next part concise, we use the following notations and abbreviations: rpart – decision tree algorithm, nnet – neural network algorithm, knn – k-nearest neighbors algorithm, rf – random forest algorithm, svm – support vector machine algorithm, median – prediction based on the median value, and random – prediction based on the random value from the empirical distribution.

4. Results of the numerical experiment

4.1. Expert and image reconstruction

To perform the analysis presented in this section, it was first necessary to prepare a table with the true labels and the predicted labels for each model or benchmarking method. Results for each model present results for the best-tuned model (i.e. minimal error defined in 1) based on the aforementioned combinations of the hyper-parameters. Each table was of the size #`expert` × #`images` with the true or the predicted label for the expert for a particular image taken for a leave-on-out iteration.

On the Table 1 and Table 2 we present classification results for each image and each expert, respectively. Each value presents an accuracy measure defined in (1) and in the rounded brackets its standard error. Both tables are sorted (from the best to the worst result) based on the Avg. column (7th column) which is derived as the average accuracy measure for five machine learning-based models. We apply a sort of majority voting to be independent of the quality of the individual model and learning paradigm. In both tables first column presents the image number or expert number. As mentioned earlier, the best (e.g. i57 and i19) and the worst (e.g. i25 and i67) reproduced decisions for images are presented in the Figure 1.

Tab. 1. Classification results for each image. Each value presents distance defined in (1) and its standard error (the best result for each row is indicated in bold).

Image	rpart	nnet	knn	rf	svm	Avg.	Median	Random
i57	23(±0.76)	**18(±0.63)**	24(±0.68)	19(±0.60)	20(±0.64)	20.8(±0.66)	54(±0.80)	78(±1.05)
i19	20(±0.70)	**18(±0.69)**	31(±0.83)	25(±0.76)	20(±0.70)	22.8(±0.74)	53(±0.74)	63(±1.12)
i75	23(±0.73)	**20(±0.61)**	30(±0.86)	21(±0.64)	20(±0.67)	22.8(±0.70)	60(±0.76)	54(±1.16)
i54	28(±0.67)	18(±0.56)	35(±0.93)	18(±0.78)	**16(±0.62)**	23.0(±0.71)	97(±0.79)	81(±1.38)
i32	**19(±0.64)**	22(±0.73)	28(±0.81)	22(±0.70)	26(±0.71)	23.4(±0.72)	51(±0.80)	55(±1.13)
i90	**19(±0.64)**	29(±0.81)	19(±0.73)	29(±0.76)	22(±0.73)	23.6(±0.73)	48(±0.70)	55(±0.93)
i81	29(±0.76)	20(±0.73)	36(±1.01)	20(±0.81)	**19(±0.83)**	24.8(±0.83)	102(±0.7)	101(±1.25)
i63	28(±0.81)	24(±0.81)	25(±0.79)	**21(±0.73)**	28(±0.76)	25.2(±0.78)	52(±0.73)	51(±1.02)
i92	24(±0.81)	**20(±0.64)**	36(±0.76)	24(±0.71)	22(±0.50)	25.2(±0.68)	86(±0.67)	87(±1.43)
i89	31(±0.75)	23(±0.89)	31(±0.75)	26(±0.93)	**19(±0.67)**	26.0(±0.80)	84(±0.82)	87(±1.38)
i17	31(±0.97)	23(±0.79)	29(±0.88)	26(±0.91)	**22(±0.79)**	26.2(±0.87)	48(±0.70)	52(±0.99)
i6	**23(±0.84)**	28(±0.86)	25(±0.84)	29(±0.84)	27(±0.84)	26.4(±0.84)	58(±0.65)	67(±1.10)
i47	27(±0.86)	**24(±0.68)**	29(±0.76)	26(±0.68)	26(±0.71)	26.4(±0.74)	48(±0.78)	54(±1.14)
i33	29(±0.81)	27(±0.81)	**24(±0.79)**	27(±0.84)	26(±0.79)	26.6(±0.81)	55(±0.76)	48(±0.97)
i46	29(±0.84)	**23(±0.73)**	24(±0.76)	31(±0.78)	26(±0.76)	26.6(±0.77)	49(±0.77)	47(±1.02)
i66	30(±0.93)	27(±0.68)	27(±0.76)	**25(±0.76)**	29(±0.84)	27.6(±0.79)	57(±0.76)	58(±1.15)
i62	28(±0.91)	27(±0.79)	35(±0.71)	**20(±0.57)**	29(±0.70)	27.8(±0.74)	78(±0.76)	74(±1.33)
i61	**24(±0.79)**	26(±0.79)	34(±0.82)	27(±0.81)	29(±0.76)	28.0(±0.79)	46(±0.75)	48(±0.95)
i94	27(±0.81)	35(±0.89)	26(±0.71)	30(±0.76)	**25(±0.71)**	28.6(±0.78)	50(±0.76)	60(±1.01)
i21	28(±0.88)	30(±0.83)	**24(±0.76)**	37(±0.90)	25(±0.79)	28.8(±0.83)	50(±0.78)	56(±1.02)
i40	27(±0.86)	38(±1.06)	32(±0.90)	27(±0.89)	**20(±0.67)**	28.8(±0.88)	53(±0.82)	59(±1.12)
i26	33(±0.89)	**24(±0.71)**	30(±0.78)	28(±0.70)	30(±0.83)	29.0(±0.78)	56(±0.72)	68(±1.03)
i20	36(±0.83)	**22(±0.67)**	28(±0.79)	37(±0.88)	23(±0.71)	29.2(±0.78)	45(±0.65)	57(±0.99)
i36	**22(±0.67)**	29(±0.78)	39(±0.89)	31(±0.78)	27(±0.79)	29.6(±0.78)	48(±0.75)	71(±1.07)
i69	27(±0.76)	39(±0.97)	**21(±0.61)**	37(±0.78)	24(±0.65)	29.6(±0.75)	50(±0.78)	59(±1.12)
i59	33(±0.85)	25(±0.71)	39(±0.93)	30(±1.03)	**24(±0.74)**	30.2(±0.85)	103(±0.71)	91(±1.32)
i95	**26(±0.81)**	29(±0.95)	33(±1.00)	33(±0.85)	30(±0.81)	30.2(±0.88)	51(±0.82)	64(±1.20)
i88	36(±0.99)	24(±0.84)	42(±1.11)	**16(±0.65)**	34(±1.04)	30.4(±0.93)	60(±0.78)	73(±1.39)
i96	24(±0.74)	**22(±0.61)**	44(±1.00)	25(±0.71)	37(±0.90)	30.4(±0.79)	49(±0.77)	63(±1.17)
i31	39(±0.91)	30(±0.70)	**24(±0.71)**	30(±0.86)	30(±0.86)	30.6(±0.81)	50(±0.78)	45(±0.95)
i72	27(±0.73)	**24(±0.68)**	37(±0.80)	32(±0.80)	33(±0.80)	30.6(±0.76)	36(±0.70)	59(±1.02)
i77	29(±0.84)	**26(±0.84)**	37(±0.88)	28(±0.88)	34(±0.89)	30.8(±0.87)	60(±0.83)	69(±1.19)
i24	37(±0.96)	31(±1.05)	36(±1.03)	**25(±1.05)**	26(±1.01)	31.0(±1.02)	86(±0.81)	100(±1.26)
i30	36(±0.88)	**27(±0.73)**	34(±0.79)	28(±0.76)	30(±0.76)	31.0(±0.78)	70(±0.81)	57(±1.13)
i83	27(±0.76)	34(±0.96)	45(±0.95)	**19(±0.73)**	31(±0.92)	31.2(±0.86)	45(±0.76)	52(±1.07)
i43	33(±0.87)	34(±0.82)	36(±0.95)	**28(±0.79)**	30(±0.83)	32.2(±0.85)	59(±0.75)	54(±1.10)
i65	40(±0.95)	31(±0.81)	30(±0.76)	37(±0.92)	**24(±0.74)**	32.4(±0.84)	63(±0.75)	65(±1.22)
i52	26(±0.65)	**25(±0.76)**	46(±0.94)	31(±0.73)	39(±0.86)	33.4(±0.79)	54(±0.80)	63(±1.17)
i53	36(±1.01)	34(±0.96)	**31(±0.92)**	34(±0.96)	32(±0.90)	33.4(±0.95)	49(±0.77)	52(±1.12)
i18	30(±0.81)	34(±0.89)	**28(±0.84)**	41(±0.98)	35(±0.93)	33.6(±0.89)	58(±0.74)	60(±1.20)
i58	25(±0.76)	**25(±0.71)**	45(±0.91)	31(±0.78)	42(±0.98)	33.6(±0.83)	76(±0.68)	68(±1.10)
i76	32(±0.92)	**28(±0.79)**	38(±0.92)	38(±0.96)	32(±0.90)	33.6(±0.90)	58(±0.62)	70(±1.21)
i56	**29(±0.81)**	36(±0.93)	38(±0.87)	32(±0.75)	34(±0.91)	33.8(±0.85)	53(±0.68)	82(±1.27)
i14	41(±0.96)	36(±0.88)	**27(±0.84)**	38(±0.94)	28(±0.84)	34.0(±0.89)	54(±0.70)	51(±1.13)
i70	36(±0.86)	33(±0.92)	38(±0.87)	**26(±0.81)**	37(±0.88)	34.0(±0.87)	46(±0.75)	57(±0.99)
i71	34(±0.91)	**30(±0.81)**	38(±0.96)	35(±0.91)	34(±0.94)	34.2(±0.91)	47(±0.79)	66(±1.13)
i3	31(±0.90)	34(±0.94)	38(±0.92)	42(±0.96)	**27(±0.79)**	34.4(±0.90)	52(±0.67)	68(±1.12)
i86	33(±0.87)	37(±0.85)	39(±0.86)	**31(±0.78)**	32(±0.78)	34.4(±0.83)	52(±0.78)	65(±1.07)
i73	34(±0.94)	**32(±0.96)**	39(±0.97)	35(±0.95)	33(±0.92)	34.6(±0.95)	45(±0.76)	61(±1.17)
i49	34(±0.82)	**30(±0.88)**	39(±1.04)	35(±0.89)	36(±0.95)	34.8(±0.92)	56(±0.69)	60(±1.14)
i41	33(±0.92)	**31(±0.88)**	33(±0.98)	41(±1.04)	37(±0.90)	35.0(±0.94)	56(±0.82)	78(±1.16)
i51	**29(±0.81)**	35(±0.97)	43(±0.97)	33(±0.87)	35(±0.93)	35.0(±0.91)	54(±0.80)	56(±1.02)
i27	38(±0.89)	**29(±0.73)**	40(±0.83)	33(±0.89)	36(±0.73)	35.2(±0.81)	58(±0.91)	57(±0.99)
i37	35(±0.81)	40(±0.90)	41(±0.98)	**27(±0.73)**	33(±0.85)	35.2(±0.85)	55(±0.81)	70(±1.11)
i13	47(±1.00)	**26(±0.74)**	35(±0.89)	40(±0.95)	30(±0.81)	35.6(±0.88)	47(±0.71)	66(±0.89)
i99	40(±0.95)	37(±1.03)	39(±0.97)	**31(±0.85)**	32(±0.96)	35.8(±0.95)	56(±0.69)	56(±1.22)
i82	35(±0.95)	39(±1.02)	37(±0.96)	**34(±0.87)**	35(±0.93)	36.0(±0.95)	52(±0.81)	56(±1.06)
i85	**30(±0.81)**	39(±0.82)	38(±0.98)	41(±0.77)	32(±0.94)	36.0(±0.86)	63(±0.75)	63(±1.23)
i97	**29(±0.88)**	35(±0.81)	48(±1.09)	32(±0.78)	37(±0.88)	36.2(±0.89)	49(±0.71)	57(±1.23)
i74	32(±0.88)	39(±0.93)	40(±0.88)	39(±0.82)	**32(±0.83)**	36.4(±0.87)	51(±0.74)	65(±1.07)
i68	41(±1.00)	34(±0.89)	34(±0.84)	47(±1.00)	**29(±0.84)**	37.0(±0.91)	42(±0.71)	57(±1.11)
i87	39(±0.93)	40(±0.95)	34(±0.89)	42(±0.98)	**30(±0.88)**	37.0(±0.93)	67(±0.72)	69(±1.18)
i11	37(±0.80)	41(±0.87)	38(±0.72)	**34(±0.87)**	38(±0.94)	37.6(±0.84)	53(±0.87)	69(±1.18)
i4	**34(±0.96)**	38(±1.00)	47(±1.17)	34(±1.00)	36(±1.01)	37.8(±1.03)	42(±0.68)	74(±1.18)
i78	39(±0.93)	42(±0.93)	36(±0.83)	43(±0.90)	**31(±0.81)**	38.2(±0.88)	41(±0.69)	60(±1.03)
i44	42(±0.93)	**33(±0.94)**	40(±1.05)	38(±0.96)	40(±0.99)	38.6(±0.97)	69(±0.73)	62(±1.20)
i48	39(±0.82)	42(±0.79)	40(±0.83)	**36(±0.73)**	38(±0.82)	39.0(±0.80)	59(±0.94)	60(±1.16)
i79	**31(±0.83)**	46(±0.90)	43(±0.88)	36(±0.81)	41(±0.90)	39.4(±0.86)	49(±0.71)	72(±0.97)
i28	51(±0.98)	35(±0.84)	**34(±0.79)**	39(±0.89)	39(±0.89)	39.6(±0.88)	46(±0.90)	62(±1.08)
i45	44(±1.02)	39(±0.91)	39(±0.91)	42(±1.00)	**34(±0.96)**	39.6(±0.96)	64(±0.78)	63(±1.07)
i9	42(±1.13)	**33(±0.89)**	41(±1.02)	37(±0.99)	46(±1.03)	39.8(±1.01)	55(±0.74)	67(±1.21)
i1	39(±0.97)	36(±0.83)	40(±0.97)	51(±0.96)	**34(±0.84)**	40.0(±0.91)	62(±0.80)	73(±1.05)
i39	44(±1.08)	**37(±1.01)**	41(±1.02)	39(±0.93)	41(±1.02)	40.4(±1.01)	57(±0.88)	63(±1.27)
i38	**34(±0.87)**	45(±1.07)	45(±1.16)	42(±1.06)	38(±0.94)	40.8(±1.02)	60(±0.86)	70(±1.28)

i42	46(±0.97)	**34(±0.82)**	44(±1.02)	41(±0.87)	39(±0.97)	40.8(±0.93)	68(±0.83)	64(±1.09)
i10	43(±0.86)	**36(±0.93)**	39(±0.91)	45(±0.99)	43(±0.88)	41.2(±0.91)	66(±0.65)	68(±1.35)
i5	41(±0.87)	42(±1.13)	43(±1.01)	**38(±0.96)**	46(±0.99)	42.0(±0.99)	67(±0.82)	74(±1.11)
i55	44(±0.96)	**30(±0.70)**	54(±0.92)	43(±0.83)	40(±0.81)	42.2(±0.84)	81(±0.83)	74(±1.31)
i60	35(±0.79)	42(±1.06)	52(±0.97)	**34(±0.74)**	50(±0.88)	42.6(±0.89)	78(±0.84)	75(±1.16)
i34	45(±0.91)	44(±0.94)	43(±0.95)	43(±0.97)	**39(±0.95)**	42.8(±0.94)	52(±0.86)	70(±1.05)
i91	45(±0.99)	42(±1.04)	42(±0.96)	48(±1.18)	**37(±0.92)**	42.8(±1.02)	68(±0.72)	67(±1.29)
i84	49(±1.02)	39(±0.86)	49(±0.87)	44(±0.92)	**35(±0.79)**	43.2(±0.89)	61(±0.91)	73(±1.18)
i8	44(±0.98)	39(±0.86)	54(±0.92)	50(±1.11)	**37(±0.92)**	44.8(±0.96)	52(±0.81)	64(±1.03)
i93	50(±1.18)	43(±1.03)	**43(±0.97)**	44(±1.08)	44(±1.19)	44.8(±1.09)	58(±0.71)	83(±1.32)
i7	36(±0.97)	46(±1.03)	58(±1.02)	**33(±0.92)**	53(±0.96)	45.2(±0.98)	89(±0.76)	79(±1.18)
i98	48(±1.09)	44(±1.06)	46(±0.88)	48(±1.14)	**42(±1.02)**	45.6(±1.04)	70(±0.76)	81(±1.24)
i2	41(±1.02)	42(±0.91)	58(±1.08)	51(±1.00)	**39(±1.04)**	46.2(±1.01)	64(±0.76)	69(±1.21)
i35	49(±1.06)	40(±0.90)	64(±1.13)	**29(±0.78)**	51(±1.06)	46.6(±0.99)	62(±0.85)	64(±1.26)
i12	48(±1.09)	44(±1.02)	48(±1.11)	**43(±1.01)**	51(±1.08)	46.8(±1.06)	55(±0.89)	56(±1.06)
i22	44(±1.04)	**35(±0.89)**	50(±1.11)	56(±1.10)	49(±1.13)	46.8(±1.05)	56(±0.77)	66(±1.10)
i64	**38(±0.87)**	48(±1.23)	49(±1.06)	56(±1.24)	44(±1.10)	47.0(±1.10)	66(±0.79)	67(±1.17)
i15	51(±0.91)	40(±1.01)	59(±0.98)	**39(±0.89)**	55(±0.89)	48.8(±0.94)	62(±0.85)	70(±1.14)
i23	44(±1.04)	**43(±0.86)**	61(±1.04)	47(±0.98)	53(±0.98)	49.6(±0.98)	85(±0.76)	86(±1.20)
i16	**43(±0.93)**	52(±1.07)	51(±1.10)	56(±1.04)	51(±1.08)	50.6(±1.04)	47(±0.68)	63(±1.14)
i80	48(±0.95)	**41(±0.90)**	64(±1.13)	46(±1.08)	59(±1.06)	51.6(±1.02)	75(±0.71)	88(±1.36)
i25	65(±0.95)	49(±0.98)	**43(±0.95)**	59(±0.96)	46(±0.97)	52.4(±0.96)	62(±0.74)	50(±1.01)
i67	78(±1.13)	50(±1.05)	44(±1.04)	52(±1.09)	**38(±0.92)**	52.4(±1.05)	55(±0.68)	61(±1.09)
i29	**40(±0.95)**	58(±1.11)	65(±0.97)	52(±1.11)	56(±0.98)	54.2(±1.02)	76(±0.81)	68(±1.12)
i50	**54(±1.10)**	57(±1.11)	78(±1.09)	63(±1.17)	76(±1.18)	65.6(±1.13)	69(±0.75)	90(±1.23)

We can see that results provided by each model when aggregating predictions, on both, expert and image levels, are better than two baselines (Median and Random columns). This means that the models are able to reconstruct the expert's preferences (to some extent and with a certain degree of accuracy).

Spearman correlation coefficient between the beauty ranking of an image (average image beauty for all 50 experts) and the quality of reproduction of the expert's decision (order based on the Avg. column) about the image is 0.21. This value, of course, indicates a relationship between the rankings, but not as great as was initially assumed before the study began. This prompted us to look for hidden multidimensional relationships invisible at first glance, which depend on the prediction error and image beauty.

Let us now answer the 1-st research question. When creating a quality ranking (Table 3 is based on the Tables 1 and 2) of a given machine learning-based model (when aggregating results for an image), it can be noticed that the best classifier is the nnet. It appears most often in the first place (31.3%) and very often in the second place (26.3%). The second place belongs to svm, the third to rf, and the fourth to rpart. The worst model is knn, which takes last place 43%. For the expert level, the best model is rf getting the best result 40%, however, it also has a large share of last place, as much as 30%. The nnet is in second place, gaining as much as 76% on the podium. It seems that svm is in third place, rpart is in fourth place, and again knn closes the entire rank.

When it comes to the 2-nd research question it can be concluded that the answer is positive. The results are presented in Table 4. By aggregating the results for the 5 or 10 best and worst reproducible images, it can be seen that the average or median expert rating differs. For the entire population, the average expert rating for the top 5 is 3.67, while for the low 5, it is 2.72. When aggregating the results into individual expert groups, this trend is maintained. The smallest difference is for the Art group and the

Tab. 2. Classification results for each expert. Each value presents distance defined in (1) and its standard error (the best result for each row is indicated in bold).

Expert	rpart	nnet	knn	rf	svm	Avg.	Median	Random
WTD10	**38(±0.58)**	46(±0.66)	50(±0.75)	57(±0.72)	44(±0.69)	47.0(±0.68)	122(±0.70)	139(±1.06)
Art5	45(±0.56)	57(±0.67)	52(±0.56)	**42(±0.70)**	47(±0.54)	48.6(±0.61)	52(±0.56)	80(±0.72)
WWa1	57(±0.80)	**31(±0.62)**	57(±0.92)	68(±0.74)	33(±0.67)	49.2(±0.75)	87(±1.00)	141(±1.21)
WTD5	54(±0.67)	56(±0.61)	61(±0.70)	**27(±0.57)**	53(±0.72)	50.2(±0.65)	70(±0.67)	99(±0.81)
WTD4	**40(±0.60)**	47(±0.59)	55(±0.61)	63(±0.66)	47(±0.59)	50.4(±0.61)	96(±0.71)	88(±0.78)
WTD9	61(±0.68)	48(±0.63)	55(±0.56)	**39(±0.57)**	49(±0.54)	50.4(±0.60)	88(±0.64)	101(±0.93)
WTD12	61(±0.90)	60(±0.89)	54(±0.84)	**40(±0.64)**	53(±0.82)	53.6(±0.82)	136(±0.56)	142(±1.24)
Std2	40(±0.70)	**38(±0.68)**	79(±0.91)	75(±1.04)	50(±0.76)	56.4(±0.82)	148(±1.06)	145(±1.19)
WTD2	68(±0.99)	**48(±0.86)**	54(±0.94)	63(±0.79)	49(±0.92)	56.4(±0.90)	106(±0.29)	116(±1.09)
WTD6	**45(±0.73)**	46(±0.75)	83(±0.93)	48(±0.56)	64(±0.88)	57.2(±0.77)	127(±1.01)	135(±1.16)
WWa8	**35(±0.58)**	49(±0.68)	76(±0.83)	85(±1.11)	46(±0.67)	58.2(±0.77)	102(±0.54)	115(±1.01)
Std5	50(±0.69)	53(±0.70)	52(±0.66)	95(±1.10)	**47(±0.63)**	59.4(±0.76)	60(±0.75)	76(±0.81)
WWa9	78(±0.99)	**49(±0.87)**	56(±0.91)	60(±0.91)	54(±0.90)	59.4(±0.92)	83(±1.07)	98(±1.09)
Std1	55(±0.79)	**49(±0.75)**	49(±0.75)	95(±1.05)	52(±0.79)	60.0(±0.83)	104(±0.44)	123(±1.01)
Art11	74(±0.81)	**48(±0.60)**	59(±0.68)	70(±1.11)	51(±0.61)	60.4(±0.76)	93(±0.60)	107(±0.95)
Art6	58(±0.77)	61(±0.98)	**55(±0.94)**	80(±0.80)	58(±0.77)	62.4(±0.85)	220(±0.75)	77(±1.01)
WWa11	76(±0.89)	**46(±0.66)**	70(±0.80)	67(±0.90)	59(±0.71)	63.6(±0.79)	111(±0.61)	154(±1.07)
Art2	**51(±1.00)**	58(±0.99)	76(±1.09)	64(±0.88)	72(±1.11)	64.2(±1.01)	112(±1.01)	128(±1.11)
Std13	64(±0.88)	**56(±0.80)**	75(±0.94)	73(±0.94)	60(±0.83)	65.6(±0.88)	118(±0.42)	134(±1.17)
WWa4	67(±0.93)	63(±0.94)	60(±0.91)	83(±0.93)	**57(±0.88)**	66.0(±0.92)	72(±1.02)	102(±1.14)
Art3	**59(±0.79)**	68(±0.75)	77(±0.84)	67(±1.09)	64(±0.80)	67.0(±0.85)	131(±0.65)	133(±1.13)
Std4	76(±0.97)	73(±0.94)	73(±0.95)	**54(±0.64)**	60(±0.88)	67.2(±0.88)	114(±1.06)	104(±1.02)
WWa7	75(±0.94)	69(±0.92)	73(±0.93)	**59(±0.70)**	66(±0.88)	68.4(±0.87)	112(±0.42)	123(±1.03)
Std3	68(±0.86)	67(±0.89)	73(±0.91)	74(±0.95)	**63(±0.83)**	69.0(±0.89)	117(±0.41)	118(±1.11)
Std9	83(±0.87)	**51(±0.83)**	75(±0.92)	69(±0.85)	71(±0.88)	69.8(±0.87)	178(±1.15)	153(±1.37)
Art10	69(±0.83)	62(±0.78)	73(±0.86)	98(±1.01)	**52(±0.68)**	70.8(±0.83)	107(±0.55)	122(±1.04)
WTD7	**53(±0.75)**	58(±0.73)	91(±0.99)	88(±0.82)	66(±0.88)	71.2(±0.83)	136(±0.60)	140(±1.25)
Art4	74(±1.04)	60(±0.98)	94(±1.23)	**57(±0.67)**	75(±0.99)	72.0(±0.98)	157(±0.53)	148(±1.47)
Std11	77(±0.88)	**61(±0.79)**	70(±0.82)	93(±0.97)	63(±0.83)	72.8(±0.86)	107(±0.49)	149(±1.05)
WTD8	**55(±0.76)**	66(±0.83)	74(±0.87)	96(±1.22)	73(±0.86)	72.8(±0.91)	95(±0.60)	121(±1.03)
Art8	63(±0.97)	66(±0.98)	97(±1.20)	**60(±0.71)**	79(±1.07)	73.0(±0.99)	153(±0.58)	178(±1.46)
WTD11	82(±0.86)	72(±0.79)	90(±0.86)	**50(±0.92)**	75(±0.77)	73.8(±0.84)	103(±0.62)	138(±1.03)
Std14	84(±0.92)	73(±0.95)	93(±0.90)	**42(±0.73)**	79(±0.99)	74.2(±0.90)	110(±0.51)	139(±1.13)
WWa2	77(±1.06)	68(±0.95)	106(±1.05)	**54(±0.64)**	75(±1.03)	76.0(±0.95)	150(±1.14)	177(±1.51)
WWa12	88(±0.81)	**63(±0.72)**	88(±0.88)	81(±0.94)	72(±0.75)	78.4(±0.82)	111(±0.64)	143(±0.98)
Std8	70(±1.05)	86(±1.03)	95(±1.11)	**67(±0.82)**	81(±1.03)	79.8(±1.01)	102(±1.07)	152(±1.19)
Art1	**69(±0.99)**	77(±1.04)	79(±1.04)	98(±1.01)	79(±1.02)	80.4(±1.02)	119(±1.02)	148(±1.15)
Art9	85(±0.83)	78(±0.77)	89(±0.80)	77(±1.01)	**73(±0.79)**	80.4(±0.84)	101(±0.64)	119(±1.03)
WTD1	90(±0.99)	92(±0.91)	87(±0.97)	**59(±0.83)**	82(±0.98)	82.0(±0.94)	138(±0.53)	139(±1.21)
WWa6	91(±0.95)	86(±0.85)	97(±0.98)	**62(±0.93)**	91(±0.94)	85.4(±0.93)	118(±0.77)	149(±1.15)
Std6	81(±1.03)	95(±1.03)	95(±0.98)	**65(±0.85)**	95(±0.95)	86.2(±0.97)	139(±0.55)	147(±1.36)
WTD3	87(±1.11)	98(±1.22)	101(±1.14)	**55(±0.79)**	90(±1.20)	86.2(±1.09)	126(±0.51)	163(±1.15)
WWa3	96(±1.16)	83(±1.09)	**83(±1.03)**	84(±0.87)	89(±1.07)	87.0(±1.04)	150(±0.54)	131(±1.34)
Art7	102(±1.12)	94(±1.03)	108(±1.12)	**55(±0.63)**	85(±1.02)	88.8(±0.98)	139(±1.13)	166(±1.41)
WWa10	99(±0.94)	81(±0.95)	97(±0.98)	**77(±0.91)**	97(±0.97)	90.2(±0.95)	152(±1.12)	136(±1.23)
Art12	87(±1.03)	109(±1.01)	101(±0.91)	**56(±0.67)**	101(±0.91)	90.8(±0.91)	120(±0.54)	137(±1.10)
WWa5	110(±1.06)	**83(±0.97)**	92(±0.98)	86(±0.99)	86(±0.99)	92.7(±1.00)	93(±0.64)	129(±0.96)
Std12	85(±0.96)	**83(±1.00)**	93(±1.08)	120(±1.15)	92(±1.06)	94.6(±1.05)	133(±0.52)	125(±1.29)
Std7	98(±1.05)	117(±1.18)	105(±1.07)	**68(±0.89)**	116(±1.21)	100.8(±1.08)	122(±1.09)	150(±1.38)
Std10	96(±1.02)	108(±1.09)	**93(±1.00)**	145(±1.28)	96(±1.02)	107.6(±1.08)	121(±0.56)	143(±1.19)

largest for the WWa group. The difference also persists for the top and low 10. This means that nice pictures are reproduced better.

The answers to the question 3-rd are included in the Table 5. Taking into account the voting results of all models (Overall row), the best-reconstructed group is WTD in both the top 5 and 10, i.e. the mentioned sets contain 17% and 33% of all experts from this group. In the low 5 or 10, there are only 8% of experts from this group. The Std group is next in the ranking, while the Art group has the worst reproducibility (0% in the top 5). It should be noted that both the best groups come from the same environment, i.e.

Tab. 3. The frequency of occurrence of a given model on a given position in the classification quality ranking for a given image or expert.

Level	Rank	rpart	nnet	knn	rf	svm
Image	1	19.2%	31.3%	11.1%	21.2%	23.2%
	2	20.2%	26.3%	15.2%	21.2%	29.3%
	3	21.2%	19.2%	11.2%	22.2%	22.2%
	4	18.2%	16.1%	19.2%	18.2%	22.2%
	5	21.2%	7.1%	43.3%	17.2%	3.1%
Expert	1	18.0%	28.0%	8.0%	40.0%	11.0%
	2	24.0%	30.0%	2.0%	6.0%	42.0%
	3	14.0%	18.0%	28.0%	12.0%	36.0%
	4	20.0%	16.0%	28.0%	12.0%	12.0%
	5	24.0%	8.0%	34.0%	30.0%	0.0%

Tab. 4. Statistics (average and median) of true labels for the best (top 5 and 10) and for the worst (low 5 and 10) classified images. Presented results are for the entire population and for each group of the experts.

Statistic	Top 5	Low 5	Top 10	Low 10
Avg. Label	3.67(±0.51)	2.72(±1.09)	3.77(±0.61)	2.96(±0.95)
Med. Label	4.20(±0.45)	2.60(±1.52)	4.15(±0.88)	2.90(±1.29)
Avg. Label Art	3.48(±0.48)	2.77(±0.93)	3.53(±0.58)	2.94(±0.91)
Avg. Label Std	3.84(±0.59)	2.91(±1.02)	3.91(±0.67)	3.14(±0.87)
Avg. Label WTD	3.60(±0.60)	2.62(±1.28)	3.87(±0.72)	2.92(±1.07)
Avg. Label WWa	3.72(±0.50)	2.57(±1.19)	3.75(±0.68)	2.82(±1.04)
Med. Label Art	3.70(±0.57)	2.50(±1.41)	3.75(±0.82)	2.95(±1.40)
Med. Label Std	4.20(±0.45)	3.00(±1.41)	4.25(±0.59)	3.10(±1.20)
Med. Label WTD	3.90(±0.74)	2.50(±1.50)	4.10(±0.91)	2.90(±1.26)
Med. Label Wwa	4.10(±0.55)	2.60(±1.52)	4.05(±0.93)	2.85(±1.25)

students or teaching members at the Faculty of Wood Technology or Institute of Wood Sciences and Furniture, respectively.

4.2. Importance of variables

The answer to the question 4-th will be presented at two levels of data aggregation. First for all images in total and then broken down into the error made by a given model. In the Figure 3 we present the frequency of occurrence of a given variable in the model structure for each image. The results were obtained via the VarImp function (described in 3.1) only for nnet and rpart. Due to the fact that each tree in rf is built to the

Tab. 5. Frequencies showing the best and worst reproducible expert groups.

Model	Expert	Top 5	Low 5	Top 10	Low 10
rpart	Art	8%	8%	17%	17%
	Std	7%	7%	14%	14%
	WTD	17%	0%	42%	42%
	WWa	8%	25%	8%	8%
nnet	Art	0%	8%	8%	8%
	Std	7%	21%	14%	14%
	WTD	17%	8%	42%	42%
	WWa	17%	0%	17%	17%
knn	Art	8%	17%	17%	17%
	Std	14%	7%	14%	14%
	WTD	17%	8%	42%	42%
	WWa	0%	8%	8%	8%
rf	Art	0%	17%	8%	8%
	Std	14%	14%	21%	21%
	WTD	17%	8%	33%	33%
	WWa	8%	0%	17%	17%
svm	Art	8%	8%	17%	17%
	Std	7%	21%	14%	14%
	WTD	8%	0%	33%	33%
	WWa	17%	8%	17%	17%
Overall	Art	0%	17%	18%	17%
	Std	14%	14%	21%	29%
	WTD	17%	8%	33%	8%
	WWa	8%	0%	17%	25%

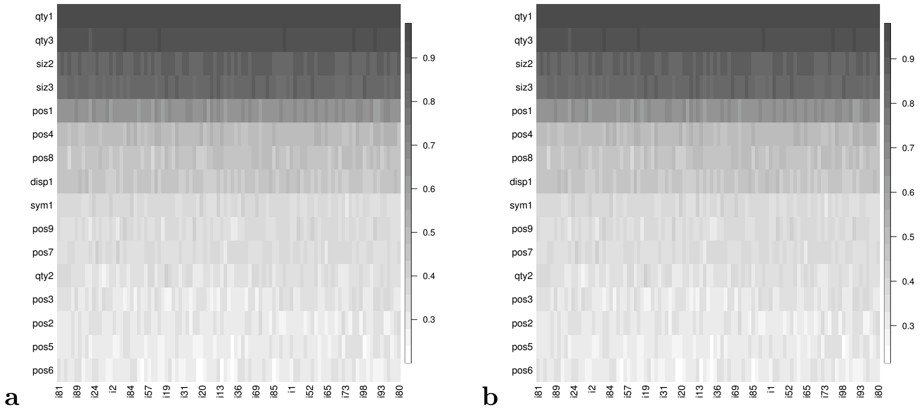


Fig. 3. Graphics showing the frequency of occurrence of a given variable in the model for each image: (**a**) rpart; (**b**) nnet. Each chart is sorted by columns (from the prettiest to the ugliest image) and by rows (from the most frequent to the least frequent variable). Due to the resolution and readability of the chart, labels of only some images are shown.

maximum level and the tree is very extensive, all variables are always included in the structure of a given tree. This results in the fact that each variable would receive the value 100% on this graph and there would be no cognitive value. For knn and svm there are no methods to derive importance of variables. The most common variables for nnet are `qty1`, `qty3`, `siz2`, and `siz3`, with a frequency of occurrence exceeding 80%. The `pos1` variable occurs in approximately 65% of images, the remaining variables appear less frequently than 45%. The results and conclusions for rpart are almost identical.

The next Figure 4 shows the average position in the importance ranking of a given variable for a given expert and a given image. This time results for rf are included as well. It is important to note that if a given variable did not appear in the model (i.e. the importance value was 0), when determining the ranking, the given variable received the lowest possible position, i.e. 16. In the case of rpart, one can see that 4 variables dominate (`qty1`, `qty3`, `siz2`, and `siz3`). It is the same set but with a slightly different order than in the case of frequency of occurrence. Additionally, unlike the previous analysis, all positional variables (`pos1-pos9`) occupy the last positions. In the case of nnet, the distribution of values is less polarized. The same group of variables as for the rpart leads the entire ranking. However, the remaining variables are no longer so far apart in the ranking and it can be seen that the values are similar. This means that the position in the ranking varied and depended on the expert and the image. In the case of rf, three more variables (`sym1`, `qty2`, and `pos1`) are added to the previously mentioned set. The variable `disp1` was ranked the lowest.

The next Figure 5 shows again the frequency of occurrence of a given variable in the

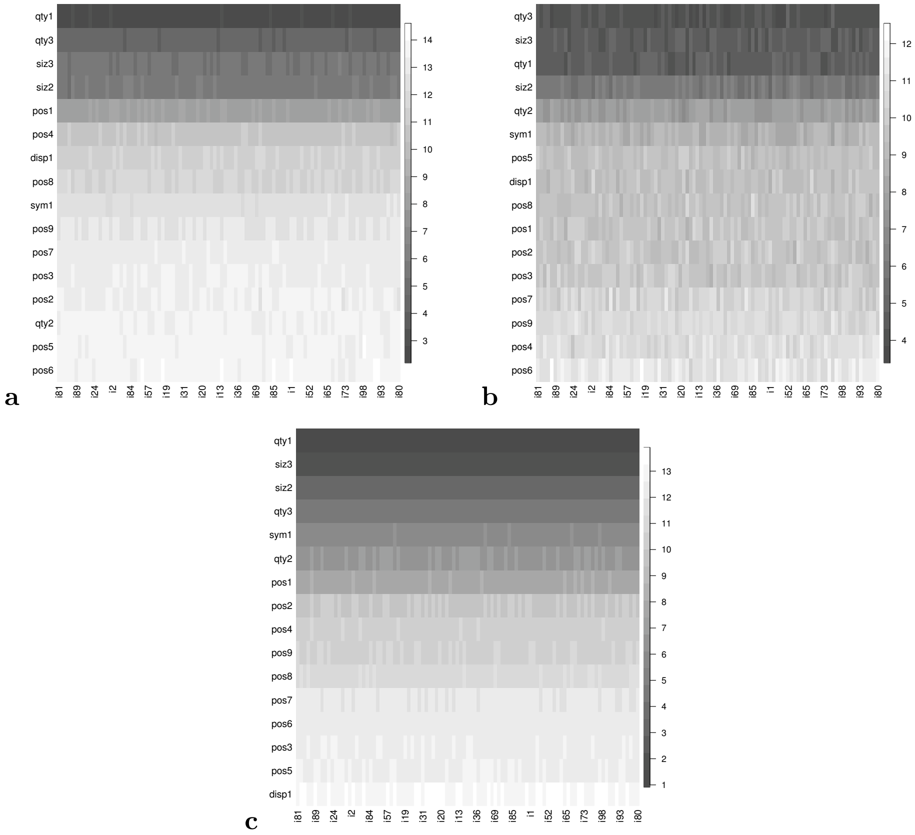


Fig. 4. Graphics showing the average importance ranking of a given variable in the model for each image: (**a**) rpart; (**b**) nnet; (**c**) rf. Each chart is sorted by columns (from the prettiest to the ugliest picture) and by rows (from the most relevant to the least relevant variable). Due to the resolution and readability of the chart, labels of only some images are shown.

model, but this time broken down into the error made by a given model for a given image. There are 9 possible errors from -4 (the predicted label is 5 while the true label is 1) to 4 (the predicted label is 1 while the true label is 5). Due to the length of the paper and the clarity of the message, we show the figure only for nnet, while the conclusions for the remaining models are quite similar.

Errors -4 and -3 usually occur with ugly images on the right side of the graph, this tendency, but to a lesser extent, is also visible for errors -2 and -1. This means that the rating was rather low and the model predicted value was rather higher. The better reproducible images (on the left-hand side) show a more clustered distribution of

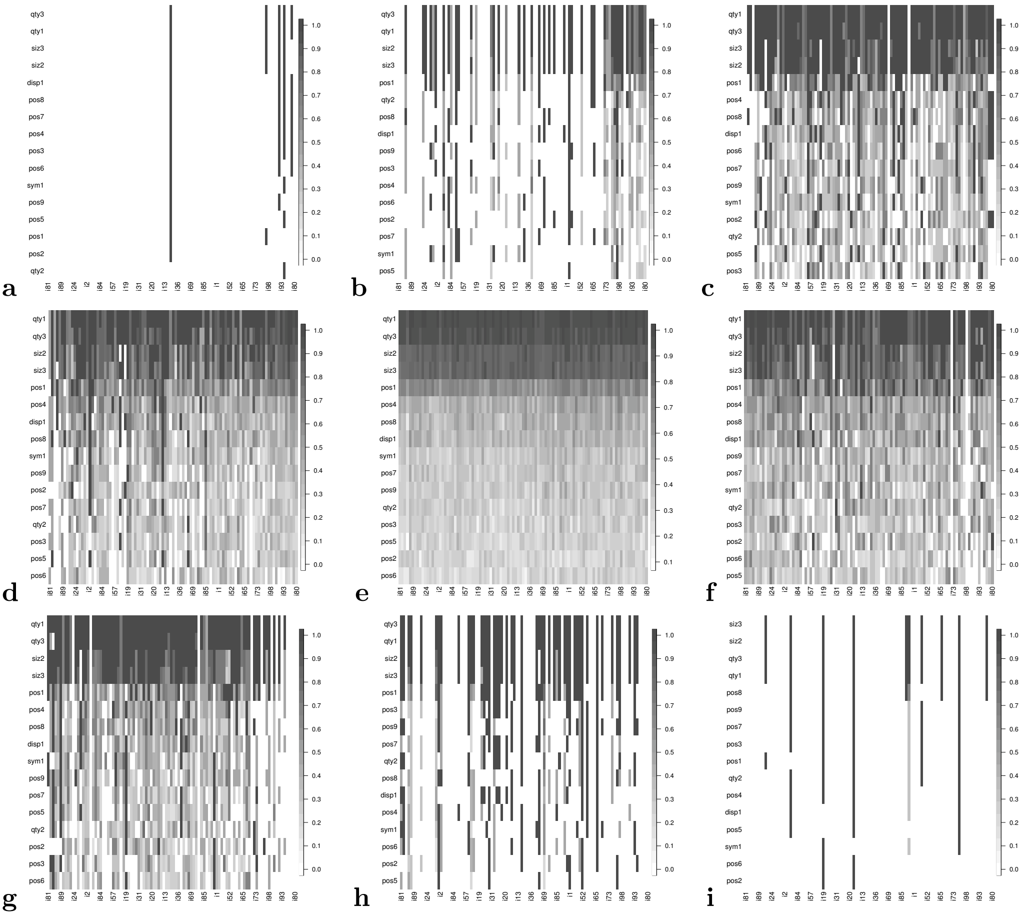


Fig. 5. Graphics showing the frequency of occurrence of a given variable in the nnet model for each image where prediction error equals (i.e. true label – prediction): (**a**) -4; (**b**) -3; (**c**) -2; (**d**) -1; (**e**) 0; (**f**) 1; (**g**) 2; (**h**) 3; (**i**) 4. Each chart is sorted by columns (from the prettiest to the ugliest image) and by rows (from the most frequent to the least frequent variable). Due to the resolution and readability of the chart, labels of only some images are shown.

frequencies to the best 4 or 5 variables (darker top right corner and lighter bottom left corner), while the images on the right-hand side have a more even distribution (longer vertical stripes of similar color). This proves that the decision for nice images is made based on the best 4 or 5 variables, while for images on the right, many more features are important. For the error equal to zero, the graph is very similar to the graph obtained for the entire population. In the case of errors directed in the opposite direction (positive

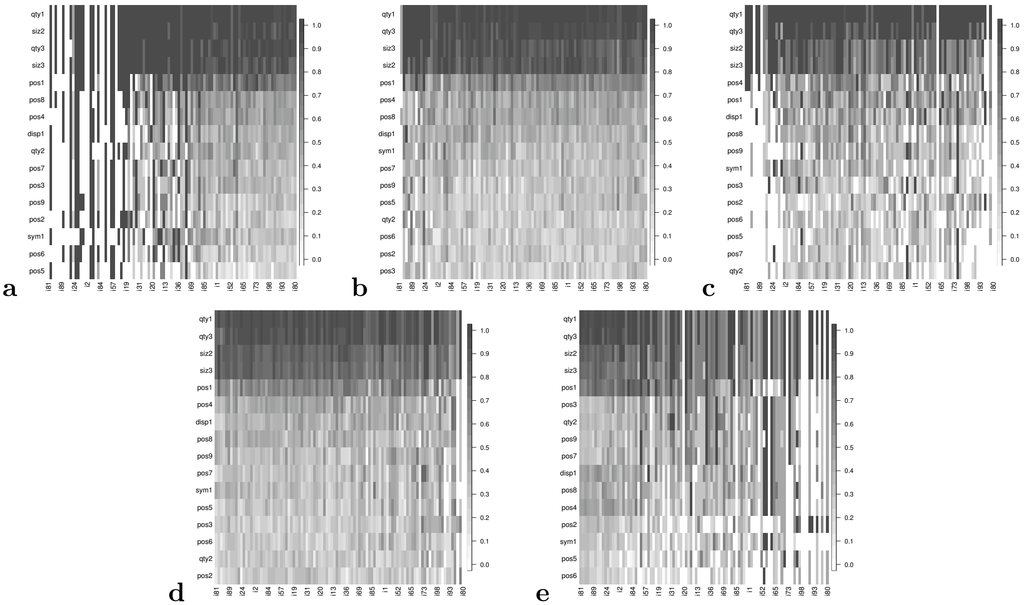


Fig. 6. Graphics showing the frequency of occurrence of a given variable in the neural network model for each true label of the image: (**a**) 1; (**b**) 2; (**c**) 3; (**d**) 4; (**e**) 5. Each chart is sorted by columns (from the prettiest to the ugliest image) and by rows (from the most frequent to the least frequent variable). Due to the resolution and readability of the chart, labels of only some images are shown.

errors 1 and 2), a slightly opposite tendency is visible, i.e. these errors occur less often for ugly images (white vertical stripes on the right). Finally, if an extreme error of 4 occurs, the models have more variables in their structure. Analyzing the order of occurrence of the variables, one can notice slightly opposite behavior of the variables `qty2` and `sym1`. For an error in the range of -2 to 2, the more important variable is `sym1` (however both are in the 3rd or 4th quadrant). For extreme negative and positive values, the order changes, and the `qty2` variable is more important.

The last set of charts (Figure 6), like the previous ones, shows the frequency of occurrence of a given variable in the nnet model, but this time divided into the true value of the label, from 1 to 5.

One can observe white stripes on the left-hand side for label 1 and on the right-hand side for label 5, this is a direct result of the fact that there are obviously no images from a given class here. The most important variables that always top the ranking regardless of the rating are `qty1`, `qt3`, `siz2`, and `siz3`. The `qty2` variable gains importance in extreme ratings (1 and 5), e.g. for 3 it is in last place. The `sym1` variable is more

Tab. 6. Average values (and their standard deviations) of the features (top 5 and 10; low 5 and 10) for the prettiest and ugliest images and the best and the worst reconstructive images.

	Beauty				Reconstruction			
Feature	Top 5	Low 5	Top 10	Low 10	Top 5	Low 5	Top 10	Low 10
`qty1`	2.6(±1.14)	4.0(±2.92)	2.3(±1.25)	3.0(±2.36)	3.6(±1.52)	5.4(±1.95)	2.9(±1.52)	3.9(±2.23)
`qty2`	0.4(±0.55)	0.6(±0.55)	0.3(±0.48)	0.6(±0.52)	0.6(±0.55)	0.8(±0.45)	0.5(±0.53)	0.5(±0.53)
`qty3`	1.6(±0.55)	2.0(±1.00)	1.5(±0.71)	1.6(±0.97)	2.0(±0.71)	2.6(±0.55)	1.7(±0.67)	2.1(±0.74)
`disp1`	0.2(±0.45)	0.6(±0.55)	0.3(±0.48)	0.6(±0.52)	0.4(±0.55)	0.8(±0.45)	0.3(±0.48)	0.4(±0.52)
`pos1`	0.2(±0.45)	0.6(±0.55)	0.2(±0.42)	0.4(±0.52)	0.6(±0.55)	0.8(±0.45)	0.4(±0.52)	0.4(±0.52)
`pos2`	0.4(±0.55)	0.4(±0.55)	0.3(±0.48)	0.5(±0.53)	0.4(±0.55)	0.6(±0.55)	0.4(±0.52)	0.5(±0.53)
`pos3`	0.0(±0.00)	0.6(±0.55)	0.1(±0.32)	0.4(±0.52)	0.4(±0.55)	1.0(±0.00)	0.2(±0.42)	0.6(±0.52)
`pos4`	0.4(±0.55)	0.6(±0.55)	0.3(±0.48)	0.5(±0.53)	0.6(±0.55)	0.8(±0.45)	0.5(±0.53)	0.6(±0.52)
`pos5`	0.2(±0.45)	0.6(±0.55)	0.3(±0.48)	0.3(±0.48)	0.4(±0.55)	0.8(±0.45)	0.4(±0.52)	0.4(±0.52)
`pos6`	0.4(±0.55)	0.4(±0.55)	0.4(±0.52)	0.5(±0.53)	0.4(±0.55)	0.2(±0.45)	0.3(±0.48)	0.4(±0.52)
`pos7`	0.6(±0.55)	0.4(±0.55)	0.6(±0.52)	0.4(±0.52)	0.4(±0.55)	0.0(±0.00)	0.5(±0.53)	0.2(±0.42)
`pos8`	0.2(±0.45)	0.4(±0.55)	0.5(±0.53)	0.3(±0.48)	0.2(±0.45)	0.2(±0.45)	0.3(±0.48)	0.2(±0.42)
`pos9`	0.8(±0.45)	0.2(±0.45)	0.6(±0.52)	0.5(±0.53)	0.6(±0.55)	0.0(±0.00)	0.6(±0.52)	0.4(±0.52)
`siz2`	2.0(±1.00)	1.8(±0.84)	1.8(±1.14)	1.7(±0.95)	2.0(±0.71)	2.0(±0.71)	2.3(±0.82)	2.1(±0.74)
`siz3`	2.8(±2.05)	3.0(±1.87)	2.6(±2.12)	2.9(±1.91)	2.8(±1.64)	3.2(±1.64)	3.5(±1.78)	3.3(±1.64)
`sym1`	0.2(±0.45)	0.4(±0.55)	0.2(±0.42)	0.4(±0.52)	0.2(±0.45)	0.0(±0.00)	0.1(±0.32)	0.0(±0.00)

important in the middle ratings (2-4) and less important for the extreme ones. The relation between `qty2` and `sym1` is quite similar as broken down into the errors. The `disp1` variable is always in the middle of the rank at the same level for all ratings.

4.3. Statistics of variables

To answer the question 5-th the Table 6 is prepared. It shows the actual average feature values divided into image beauty and reproducibility. As said before, the best variables are `qty1`, `qty3`, `siz2` and `siz3`. The mean values for the `qty1` variable for both divisions differ in the groups. For beauty in the top 5, one can see that prettier images have fewer knots (2.6) and uglier have 4. For reconstruction, these values are greater and the difference is bigger (3.6 vs. 5.4). A similar relationship exists for the `qty3` variable but not with the same explanatory power. For `siz2` and `siz3` variables, it cannot be concluded that there is a difference in the average values of these features.

Prettier images are less symmetrical (0.2 vs. 0.4), on the other hand, better reproducible images are more symmetrical (0.2 vs. 0). For both divisions, more beautiful and better reproducible images are less dispersed (`disp1`).

5. Conclusion

The findings from the questionnaire survey indicate that all five expert groups found the furniture featuring solid wood fronts sufficiently appealing to consider using them in both their personal residences and in a public building. Based on the research, it can be concluded that specific features describing the image influence the expert's perception of attractiveness. The findings based on the numerical experiments can be summarized as follows:

1. The best results of expert decision reconstruction are provided by a neural network model.
2. The expert's decision is better reconstructed for more beautiful images.
3. The best reproducible groups of experts are groups with a similar background, i.e. WTD and Std.
4. The most significant features are `qty1`, `qty2`, `siz2`, `siz3` and `pos1`.
5. Adequate value of 4 or 5 features is enough to score as nice image, while to score as ugly more features are needed.
6. There is a slightly opposite behavior of the variables `qty2` and `sym1`. The `qty2` variable gains importance in extreme ratings, while `sym1` is more important for middle ratings.
7. Prettier images and those for which the expert's decision is better reconstructed have fewer knots. In other words, more knots, worst reconstruction what is also usual score as less attractive.

Our future research focuses on the development of an automatic scoring system for fronts made of pine wood. Since we roughly know what features are important for the expert, the next stage will be to calculate them automatically in the image. The calculation of these features can be done using pattern recognition and image analysis methods or using deep learning (DL) algorithms. DL algorithms also allow us to find other features not identified here that may improve the quality of reconstruction of the expert's decisions [21, 26, 33]. Future research is therefore part of the growing trend of explainable artificial intelligence [24, 27, 35].

References

[1] M. R. Antal, D. Domljan, and P. G. Horváth. Functionality and aesthetics of furniture – Numerical expression of subjective value. *Drvna industrija*, 67(4):323–332, 2017. doi:10.5552/drind.2016.1544.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/a:1010933404324.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification And Regression Trees*. Routledge, Oct 2017. doi:10.1201/9781315139470.

[4] I. Cetiner, A. Ali Var, and H. Cetiner. Classification of knot defect types using wavelets and KNN. *Elektronika ir Elektrotechnika*, 22(6), 2016. doi:10.5755/j01.eie.22.6.17227.

[5] M. Chen and J. H. Lyu. Aesthetic evaluation of furniture design based on anp method. *Applied Mechanics and Materials*, 574:318–323, 2014. doi:10.4028/www.scientific.net/amm.574.318.

[6] L. Deng and G. Wang. Quantitative evaluation of visual aesthetics of human-machine interaction interface layout. *Computational Intelligence and Neuroscience*, 2020:1–14, 2020. doi:10.1155/2020/9815937.

[7] J. Fürnkranz and E. Hüllermeier. Preference learning. In: C. Sammut and G. I. Webb, eds., *Encyclopedia of Machine Learning*, p. 789–795. Springer US, Boston, MA, 2011. doi:10.1007/978-0-387-30164-8_662.

[8] M. Gagolewski and J. Lasek. Learning experts' preferences from informetric data. In: *Advances in Intelligent Systems Research*, ifsa-eusflat-15. Atlantis Press, 2015. doi:10.2991/ifsa-eusflat-15.2015.70.

[9] K. Gajowniczek, Y. Liang, T. Friedman, T. Ząbkowski, and G. Van den Broeck. Semantic and generalized entropy loss functions for semi-supervised deep learning. *Entropy*, 22(3):334, 2020. doi:10.3390/e22030334.

[10] K. Gajowniczek, A. Orłowski, and T. Ząbkowski. Simulation study on the application of the generalized entropy concept in artificial neural networks. *Entropy*, 20(4):249, 2018. doi:10.3390/e20040249.

[11] G. D. Garson. Interpreting neural-network connection weights. *AI Expert*, 6(4):46–51, 1991. `https://dl.acm.org/doi/10.5555/129449.129452`.

[12] S. Gold and F. Rubik. Consumer attitudes towards timber as a construction material and towards timber frame houses – selected findings of a representative survey among the german population. *Journal of Cleaner Production*, 17(2):303–309, 2009. doi:10.1016/j.jclepro.2008.07.001.

[13] T. A. Guzel. Consumer attitudes toward preference and use of wood, woodenware, and furniture: A sample from kayseri, turkey. *BioResources*, 15(1):28–37, 2019. doi:10.15376/biores.15.1.28-37.

[14] J. Han, H. Forbes, and D. Schaefer. An exploration of how creativity, functionality, and aesthetics are related in design. *Research in Engineering Design*, 32(3):289–307, 2021. doi:10.1007/s00163-021-00366-9.

[15] U. R. Hashim, S. Z. Hashim, and A. K. Muda. Automated vision inspection of timber surface defect: A review. *Jurnal Teknologi*, 77(20), 2015. doi:10.11113/jt.v77.6562.

[16] S. Kizito, A. Y. Banana, M. Buyinza, J. R. S. Kabogozza, R. K. Kambugu, et al. Consumer satisfaction with wooden furniture: an empirical study of household products produced by small and medium scale enterprises in uganda. *Journal of the Indian Academy of Wood Science*, 9(1):1–13, 2012. doi:10.1007/s13196-012-0068-1.

[17] A. Krähenbühl, B. Kerautret, I. Debled-Rennesson, F. Longuetaud, and F. Mothe. *Knot Detection in X-Ray CT Images of Wood*, p. 209–218. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-33191-6_21.

[18] M. Kuhn. Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5), 2008. doi:10.18637/jss.v028.i05.

[19] J. Parmar, S. Chouhan, V. Raychoudhury, and S. Rathore. Open-world machine learning: Applications, challenges, and opportunities. *ACM Computing Surveys*, 55(10):1–37, 2023. doi:10.1145/3561381.

[20] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2023. `https://www.R-project.org/`.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In: *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, KDD '16, p. 1135–1144. ACM, New York, NY, USA, 13-17 Aug 2016. doi:10.1145/2939672.2939778.

[22] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, Jan 1996. doi:10.1017/cbo9780511812651.

[23] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000. doi:10.1162/089976600300015565.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019. doi:10.1007/s11263-019-01228-7.

[25] J. Y. Shin, C. Kim, and H. J. Hwang. Prior preference learning from experts: Designing a reward with active inference. *Neurocomputing*, 492:508–515, 2022. doi:10.1016/j.neucom.2021.12.042.

[26] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv*, 2015. ArXiv.1412.6806. doi:10.48550/arXiv.1412.6806.

[27] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In: *Proc. 34th Int. Conf. Machine Learning*, vol. 70 of *ICML'17*, p. 3319–3328. JMLR.org, 6-11 Aug 2017. `https://dl.acm.org/doi/abs/10.5555/3305890.3306024`.

[28] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer New York, 2002. doi:10.1007/978-0-387-21706-2.

[29] M. N. Volkovs, H. Larochelle, and R. S. Zemel. Learning to rank by aggregating expert preferences. In: *Proc. 21st ACM Int. Conf. Information and Knowledge Management*, CIKM'12. ACM, Maui, Hawaii, USA, 29 Oct – 2 Nov 2012. doi:10.1145/2396761.2396868.

[30] C. van Winkelen and R. McDermott. Learning expert thinking processes: using KM to structure the development of expertise. *Journal of Knowledge Management*, 14(4):557–572, 2010. doi:10.1108/13673271011059527.

[31] C. Xiaolei, S. Jun, and L. Bing. Customer preferences for kitchen cabinets in China using conjoint analysis. *Journal of Chemical and Pharmaceutical Research*, 6(2):14–22, 2014. `https://www.jocpr.com/articles/customer-preferences-for-kitchen-cabinets-in-china-using-conjoint-analysis.pdf`.

[32] S. Yoon, H. Oh, and J. Y. Cho. Understanding furniture design choices using a 3D virtual showroom. *Journal of Interior Design*, 35(3):33–50, 2010. doi:10.1111/j.1939-1668.2010.01041.x.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In: D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., *Computer Vision – Proc. ECCV 2014*, pp. 818–833. Springer International Publishing, Cham, Zurich, Switzerland, 6-12 Sep 2014. doi:10.1007/978-3-319-10590-1_53.

[34] L. Zeng and D. Liu. A study on the model of furniture aesthetic value based on fuzzy AHP comprehensive evaluation. In: *Proc. 2010 7th Int. Conf. Fuzzy Systems and Knowledge Discovery.* IEEE, Yantai, China, 10-12 Aug 2010. doi:10.1109/fskd.2010.5569152.

[35] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv*, 2017. ArXiv.1702.04595. doi:10.48550/arXiv.1702.04595.

[36] K. Śmietańska and J. Górski. Impact of visible knots on relative visual attractiveness of furniture fronts made of pine wood (pinus sylvestris l.). *Wood Material Science & Engineering*, 18(5):1749–1754, 2023. doi:10.1080/17480272.2023.2186263.

[37] K. Śmietańska, P. Podziewski, M. Bator, and J. Górski. Automated monitoring of delamination factor during up (conventional) and down (climb) milling of melamine-faced MDF using image processing methods. *European Journal of Wood and Wood Products*, 78(3):613–615, 2020. doi:10.1007/s00107-020-01518-9.

Advancing Chipboard Milling Process Monitoring through Spectrogram-Based Time Series Analysis with Convolutional Neural Network using Pretrained Networks

Jarosław Kurek[1,*], Karol Szymanowski[2],
Leszek J. Chmielewski[1], and Arkadiusz Orłowski[1]
[1] *Institute of Information Technology, Warsaw University of Life Sciences – SGGW*
[2] *Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*
[*] *Corresponding author: Jarosław Kurek (jaroslaw_kurek@sggw.edu.pl)*

Abstract. This paper presents a novel approach to enhance chipboard milling process monitoring in the furniture manufacturing sector using Convolutional Neural Networks (CNNs) with pretrained architectures like VGG16, VGG19, and RESNET34. The study leverages spectrogram representations of time-series data obtained during the milling process, providing a unique perspective on tool condition monitoring. The efficiency of the CNN models in accurately classifying tool conditions into distinct states ('Green', 'Yellow', and 'Red') based on wear levels is thoroughly evaluated. Experimental results demonstrate that VGG16 and VGG19 achieve high accuracy, however with longer training times, while RESNET34 offers faster training at the cost of reduced precision. This research not only highlights the potential of pretrained CNNs in industrial applications but also opens new avenues for predictive maintenance and quality control in manufacturing, underscoring the broader applicability of AI in industrial automation and monitoring systems.

Key words: convolutional neural networks, CNN, vgg16, vgg19, resnet34, tool state monitoring, chipboard milling.

1. Introduction

In the realm of furniture production, integrating sensor technology at various manufacturing stages is becoming a prevalent theme in automation research. This area is inherently intricate, involving numerous steps that demand precision. Adjustments are often necessary, especially when minor components are altered or added. The infusion of sophisticated technologies in these processes marks a significant leap forward, especially in tool condition monitoring. Mistimed or incorrect decisions regarding tool replacements could diminish product quality, potentially leading to losses for the manufacturing entity [3, 4, 5, 6, 8, 10, 11, 14, 15].

This paper primarily delves into the milling process, where precision in decision-making is crucial. Utilizing sensor-based technologies to monitor tool conditions offers a novel angle to address these challenges [7, 18, 21]. While tool state assessment can be done manually, it's a laborious process that interrupts production. Hence, automating this task is a substantial step forward in the industry.

Machine GRAPHICS & VISION 32(2):89–108, 2023. DOI: 10.22630/MGV.2023.32.2.5 .

Tool monitoring, a subject of extensive discussion and analysis, revolves around the progressive wear of the cutting edge, impacting product quality. An ideal automatic solution should prevent unnecessary production halts while ensuring timely tool replacement to avoid subpar output. The solution should be precise, offering real-time, automated feedback. Applying a dedicated array of sensors for capturing specific production line signals and analyzing this data appears to be an effective strategy.

A notable innovation in this study is the application of sensor data in addressing the complex challenges of tool condition monitoring. Although furniture manufacturing utilizes various materials, wood-based products are the most common. The data-driven approach for tool condition monitoring introduced here presents new opportunities for enhancing industry processes. Different signals are analyzed to determine their efficacy in detecting tool conditions during the machining process. Despite thorough documentation of these problems, there remains a demand for an automatic and precise solution that is easy to integrate into production environments.

In the realm of manufacturing and material processing, the milling of chipboard represents a significant domain, demanding consistent monitoring and analysis for quality control and process optimization. Recent advancements in machine learning and neural networks have opened up innovative pathways for enhancing these monitoring systems. This research delves into the intersection of these advanced technologies, focusing on the utilization of Convolutional Neural Networks (CNNs) for interpreting time series data derived from milling processes.

The core of this study lies in the exploration of spectrogram-based time series analysis, a technique which converts time series data into a more visually interpretable format, capturing both frequency and time information simultaneously. This approach leverages the intrinsic power of CNNs, particularly the renowned VGG16, VGG19, RESNET34 architectures, to analyze these spectrograms for insightful patterns and anomalies that are indicative of the chipboard milling process's condition and performance.

Applying a pretrained network [1, 8, 9, 12] as part of a CNN architecture presents a methodologically sound approach. Pretrained networks, having already learned rich feature representations from large datasets, offer a robust foundation for the model. When applied to the context of milling process monitoring, this model can detect subtle nuances and changes in the data, which are pivotal for predicting tool wear, identifying inefficiencies, and ensuring product quality.

Furthermore, this research is grounded in the analysis of real-world data, ensuring its relevance and applicability. Data collected during the milling process – such as noise, vibrations, and other pertinent physical parameters – are utilized to construct a comprehensive dataset. The application of the Short-Time Fourier Transform (STFT) and Discrete Wavelet Transform (DWT) to this dataset facilitates the generation of spectrograms. These spectrograms, in turn, serve as the input for the CNN model, allowing for a detailed and nuanced analysis of the data.

This study aims to demonstrate the effectiveness of using CNNs, particularly with the VGG16, VGG19, RESNET34 architectures, in monitoring and analyzing the milling process of chipboards. By advancing the methodological approaches to process monitoring and incorporating cutting-edge neural network technologies, this research contributes to the enhancement of quality control, process efficiency, and predictive maintenance in the manufacturing sector.

As the field of artificial intelligence continues to evolve, the implications of this research extend beyond its immediate application. The methodologies and findings presented here could inspire similar approaches in various industrial and manufacturing processes, paving the way for smarter, more efficient, and data-driven operations.

2. Dataset

This study aims to develop a diagnostic system that can accurately assess the wear level of tools during ongoing production activities. Data for this purpose were gathered through various means. The primary data collection was performed using a Jet 130 CNC machining center (Busellato, Italy), utilizing a 40 mm cutter head with a replaceable carbide cutting edge, supplied by Faba SA, Poland.

For the experimental setup, a chipboard panel measuring 300 × 150 mm was applied. The panel was secured onto a measurement platform, where a 6mm deep groove was milled at a spindle speed of 18 000 rpm and a feed rate of 0.15 mm per tooth. These parameters were selected based on extensive literature review and practical experience in milling chipboard. The chosen rotational speed and feedrate are commonly used in the industry for optimal surface finish and minimal tool wear. The 6 mm cutting depth is a standard practice for milling chipboards of similar dimensions.

The condition of the tool was categorized into three distinct states: 'Green', 'Yellow', and 'Red'. The 'Green' state indicates a new or well-conditioned tool, 'Yellow' signifies a tool in usable but worn condition, and 'Red' represents a tool that requires replacement due to significant wear. The classification was determined using the VBmax parameter.

The Figure 1 provides a microscopic view of the wear on a drill. This image illustrates the VBmax parameter, which is a key factor in classifying the tool's condition. The VBmax parameter is used to categorize the wear level of the tool into different classes such as 'Green', 'Yellow', and 'Red', corresponding to new, moderately worn, and significantly worn conditions, respectively. This visual representation aids in comprehending how the VBmax parameter is quantified and its relevance in assessing tool wear. During the testing phase, operations were periodically halted to assess the tool's condition using a Mitutoyo TM-505 microscope, ideal for dimensional and angular measurements. This microscope, capable of inspecting screw shapes and gears with an additional reticle, facilitated the categorization of wear states as follows:

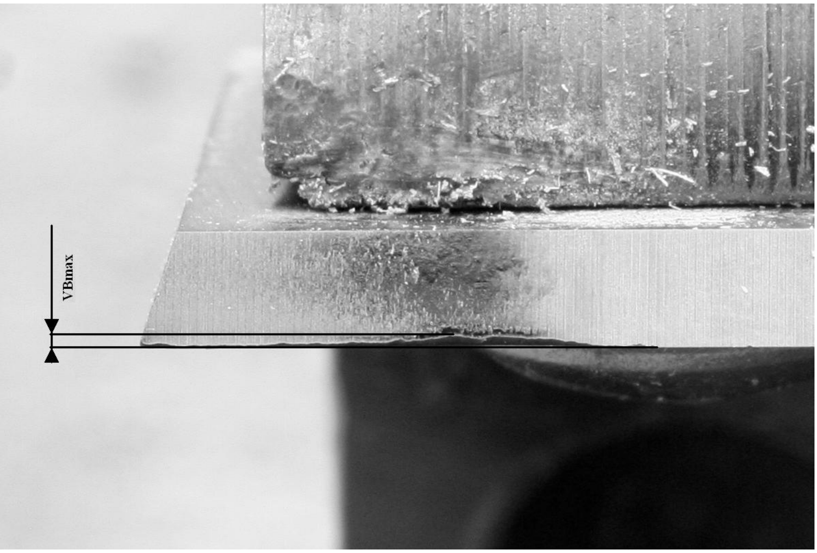


Fig. 1. Microscopic imagery showing the wear on a drill bit, highlighting the VBmax parameter pertinent to class evaluation.

- a ‘Green’ state is designated when VBmax is within the range of 0 to 0.15 mm, encompassing four levels of wear;
- a ‘Yellow’ state is defined for VBmax values between 0.151 and 0.299 mm, involving two wear levels;
- a ‘Red’ state is assigned for VBmax exceeding 0.299 mm, also with two levels of wear.

The experimental setup included a variety of sensors capable of monitoring 11 different parameters, such as:

- Force measurements in X and Y directions (Kistler 9601A sensor);
- Acoustic emission (Kistler 8152B sensor);
- Noise level (Brüel & Kjær 4189 sensor);
- Vibration intensity (Kistler 5127B sensor);
- Current and voltage ratings for the device, head, and servo (Finest HR 30 and Testec TT-Si9001 sensors).

All above items are depicted at Figure 2. This setup includes a detailed arrangement of sensors, acquisition cards, and the CNC machine tool used during the milling process. The configuration of these components is vital for capturing and analyzing data related to the milling process, such as vibrations, acoustic emissions, force measurements, and electrical signals. Understanding this setup is essential for appreciating how the data was collected, which forms the basis for the subsequent analysis using CNNs.

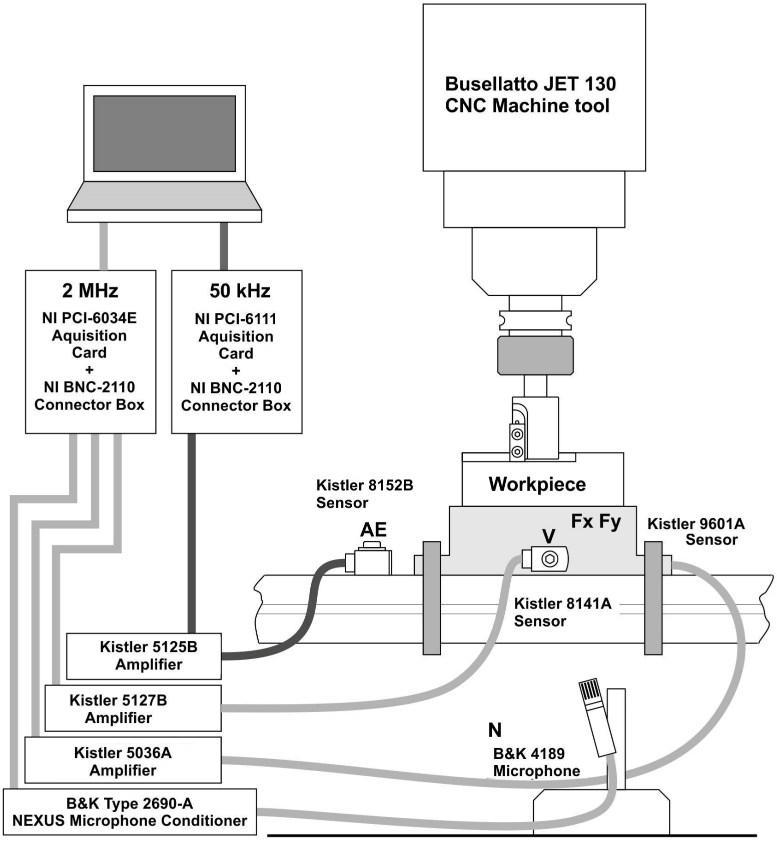


Fig. 2. Diagram of the experimental setup, illustrating the configuration of sensors, acquisition cards, and the CNC machine tool utilized in the experimental trials.

Data from these sensors were captured using National Instruments PCI-6111 and PCI-6034E measurement cards. The recording was conducted on a PC running National Instruments' Lab ViewTM software. The AE signal required a high-frequency card (2 MHz, 0.3 s window), while other signals were recorded with a 50 kHz card, using a 1.1 s window. Connections to the cards were made via BNC-2110 boxes.

To avoid training disruption by extraneous noise or sound variations, sensor positions relative to the workpiece and cutting zone were consistently maintained. The structure of the collected data is detailed in Table 1. It is organized into three columns and details various parameters measured across different categories of data. Each row represents a unique measurement type with its corresponding length and sampling rate in Hz. For

Tab. 1. Data Set Variable Configuration.

Variable Name	Measured Parameter	FS [Hz]
DataHigh	Acoustic Emissions	5 000 000
DataLow0	X-Axis Force	200 000
DataLow1	Y-Axis Force	200 000
DataLow2	Sound Level	200 000
DataLow3	Vibration Intensity	200 000
DataCurrent0	Device Current	50 000
DataCurrent1	Device Voltage	50 000
DataCurrent2	Head Unit Current	50 000
DataCurrent3	Head Unit Voltage	50 000
DataCurrent4	Servo Motor Current	50 000
DataCurrent5	Servo Motor Voltage	50 000

Tab. 2. Summary of Records and Data Classification.

Category	Number of Records	Number of Registered Signals per Record
Green Class	25	11
Yellow Class	25	11
Red Class	25	11
Total	75	

instance, the ‘High-Resolution Data’ category includes ‘Acoustic Emissions’ with a significant length of 27 999 960 and a very high sampling rate of 5 000 000 Hz. In contrast, ‘Standard Data’ encompasses measurements like ‘X-Axis Force’ and ‘Y-Axis Force’, each with 700 000 lengths and a sampling rate of 200 000 Hz. Lastly, ‘Electrical Data’ records parameters such as ‘Device Current’ and ‘Device Voltage’, each with a length of 30 000 and a sampling rate of 50 000 Hz.

The Table 2 condenses the dataset into three main classes: ‘Green’, ‘Yellow’, and ‘Red’. Each class has the same number of records, 25, and the same number of registered signals per record, 11. a total row is also included, summing the records to 75 and confirming the uniform number of signals per record across all classes. This table provides a high-level overview of the dataset’s classification structure.

The classification of tool wear during the milling process is significantly enhanced by the analysis of device current signals through spectrograms. In Fig. 3 a comparative study is presented of the spectral characteristics for three distinct tool condition classes – ‘Green’, ‘Yellow’, and ‘Red’ – as manifested in the device current signal. Spectrograms offer a two-dimensional representation where one axis represents time, the other frequency, and the color intensity indicates signal power at a given frequency and time.

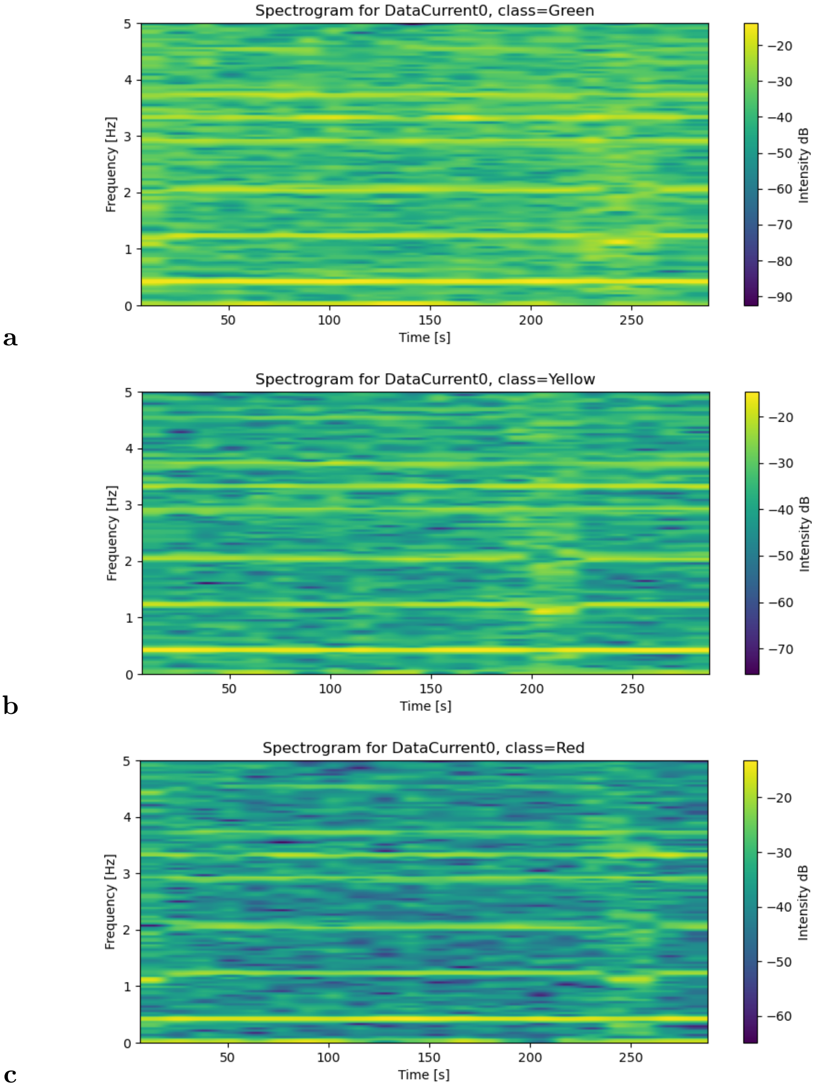


Fig. 3. Examples of spectrograms used in constructing a CNN model for classifying different conditions of a milling chipboard process. Each figure represents a unique class condition. (**a**) Spectrogram for class=‘Green’, indicating a specific condition or state. (**b**) Spectrogram for class=‘Yellow’, indicating a transition phase or intermediate state. (**c**) Spectrogram for class=‘Red’, indicating a critical condition or state.

The spectrogram corresponding to the 'Green' class, as shown in the first panel of Figure 3a, exhibits a pattern characterized by uniformity across the frequency bands. This uniformity correlates with a well-conditioned tool exhibiting consistent operational behavior without any signs of wear. The frequency bands remain relatively undisturbed, indicating stable current consumption and, implicitly, a lack of significant resistance or stress on the milling apparatus.

The 'Yellow' class spectrogram, depicted in the second panel of Figure 3b, begins to show variations in the signal's intensity across its frequency bands. These variations suggest the onset of tool wear, with certain frequencies becoming more pronounced, likely due to irregularities in the milling process as the tool encounters increasing resistance and degradation. This intermediate state of wear calls for closer monitoring to preempt any potential quality issues in the milling outcome.

The 'Red' class spectrogram, presented in the third panel of Figure 3c, shows a distinct pattern with significant disruptions in frequency stability. These disruptions reflect a critical level of tool wear, where the tool's inefficiency is evidenced by erratic fluctuations in current consumption. The 'Red' condition indicates an immediate need for tool replacement to prevent substandard milling results and potential damage to the milling equipment.

A comparative analysis of the three spectrograms underscores the efficacy of using device current signal analysis for tool wear monitoring. The progression from 'Green' to 'Red' class is marked by a discernible transition in the signal's spectral content, underscoring the potential of this approach in predictive maintenance and automated monitoring systems in industrial settings.

The spectral analysis of device current signals through spectrograms provides a robust mechanism for classifying tool condition in real-time. This method facilitates the early detection of tool wear, enabling timely interventions that can significantly enhance the efficiency and quality of the milling process.

3. Conversion of Time Series Signal into Spectrograms

This chapter outlines the methodology applied to transform the time series data collected from the milling process into spectrograms. These spectrograms serve as inputs for the subsequent classification using CNNs. The conversion process is crucial for extracting meaningful features from the time series data, which are then leveraged by the CNN for accurate classification.

In this study, each color-coded spectrogram corresponds to a distinct class condition, with 'Green' representing a normal state, 'Yellow' an intermediate state, and 'Red' a critical state. The spectrograms provide a visual and quantitative way to discern the differences between these conditions over time.

3.1. Data Preparation and Interpolation

Initially, the time series data undergo extensive preprocessing. Given the varying lengths of the time series, a uniform length is necessary to ensure consistent analysis. To achieve this, the data is interpolated to a common length, maintaining the integrity of the original series. The interpolation is conducted using Python's `interp1d` function [16] from the `scipy` library, which allows for linear interpolation of the data. The following code snippet (Algorithm 1) demonstrates the interpolation process:

Algorithm 1 Interpolate time series data

```
procedure InterpolateTS($data, TargetLength, SamplingRate$)
    $data_{Sampled} \leftarrow data[:: SamplingRate]$
    $X_{Old} \leftarrow$ linspace(0, 1, length($data_{Sampled}$))
    $X_{New} \leftarrow$ linspace(0, 1, $TargetLength$)
    $interpolator \leftarrow$ interp1d($X_{Old}, data_{Sampled}$, fillValue="extrapolate")
    return $interpolator(X_{New})$
end procedure
```

3.2. Spectrogram Generation

Post-interpolation, each time series is converted into a spectrogram. a spectrogram is a visual representation of the spectrum of frequencies in a signal as they vary with time. This conversion is crucial for visualizing the frequency content of the time series data, which is key for the CNN's feature extraction phase.

Spectrograms play a pivotal role in the analysis of time series data, especially in the context of monitoring the milling process of chipboards. They offer a visual representation of the frequency spectrum of signals as they change over time. a spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a particular waveform. It is a powerful tool for analyzing the frequency components of a signal that evolves over time. In our context, the spectrogram provides insights into the milling process by depicting the frequency content of vibrations, acoustic emissions, and other relevant signals.

The main parameters of the `specgram` [13] function used in this study are:

`seriesData` – the input signal data.

`Fs` (samplingRate) – the sampling frequency. It defines the number of data points sampled per second in the time series. In our case `Fs` depends on signal (Table 1).

`NFFT` – the number of data points used in each block for the Fast Fourier Transform (FFT). a higher number improves the frequency resolution of the spectrogram. In our case `NFFT`=256.

`noverlap` – the number of points of overlap between blocks. Increasing this value increases the continuity of the frequency spectrum over time but reduces temporal resolution. In our case `noverlap`=128.

The spectrograms are generated using the `specgram` function from Python. This function computes the spectrogram for each time series, showcasing the intensity of frequencies over time. Each spectrogram is then saved as an image file, which is later used as input for the CNN model. The code for generating the spectrograms is as follows (Algorithm 2):

Algorithm 2 Generate spectrograms for each time series

```
procedure GenerateSpectrograms($DataSet, minLength, SamplingRate$)
    for $k$ in range(length($DataSet$)) do
        for $i$ in range(length($DataSet[k]$)) do
            $DS \leftarrow DataSet[k][i]$
            Define $DataHigh0, DataLow0, \dots, DataCurrent5$ from $DS$
            $DataHigh0\_interp \leftarrow$ InterpolateTS($DataHigh0, \dots$)
            ...                                        ▷ Repeat for other data series
            for $SeriesName, SeriesData$ in $[('DataHigh0', DataHigh0\_interp), \dots]$ do
                Generate a spectrogram from $SeriesData$
                Save the spectrogram as an image file
            end for
        end for
    end for
end procedure
```

4. Model architecture with pretrained networks

4.1. CNN Pretrained networks

Convolutional Neural Networks are mainly known for their role in computer vision and image processing, which is a big part of deep learning. Traditionally associated with visual data analysis, the versatility of CNNs extends far beyond, as evidenced in this study. By repurposing these networks, initially designed for interpreting complex imagery, we have demonstrated their remarkable adaptability in processing time series data for industrial process monitoring. This innovative application leverages the inherent strengths of pretrained CNN models such as VGG16, VGG19, and RESNET34, traditionally employed in visual tasks, to analyze and interpret signals in the context of milling chipboard processes. This approach underscores a significant advancement in the application of AI, extending the capabilities of CNNs from their conventional domain of image-based analysis to the intricate realm of signal processing and time series analysis in an industrial setting.

CNNs, especially those with pretrained architectures, have revolutionized the field of deep learning, particularly in image processing and computer vision. Among the most prominent of these are VGG16 [17,19], VGG19 [17,20], and RESNET34 [2], each having unique characteristics and strengths.

VGG16 is renowned for its simplicity and depth. It consists of 16 layers, with a design focused on small convolution filters of size 3×3 and the use of max pooling to reduce spatial dimensions. The VGG16 network has proven its effectiveness in feature extraction due to its depth and uniform architecture.

VGG19, an extension of VGG16, includes 19 layers. The additional layers in VGG19 allow it to learn more complex features from images, making it slightly more powerful than VGG16 at the expense of increased computational cost. Its effectiveness in handling more intricate image patterns makes it a preferred choice for complex image recognition tasks.

RESNET34 marks a significant departure from traditional sequential architectures like VGG. The innovative use of residual connections, or 'skip connections', allows it to train deeper networks by addressing the vanishing gradient problem. This architecture comprises 34 layers and is exceptionally efficient in training due to these residual connections, which help preserve the gradient flow through the network. The RESNET34 architecture is known for its high performance in various tasks, including but not limited to image classification.

The application of these pretrained networks extends beyond traditional image processing. In various research and industry applications, these networks have been repurposed and fine-tuned for tasks such as signal processing, time-series analysis, etc. The pretrained aspect of these networks implies that they have been previously trained on large datasets, like ImageNet, allowing them to have a deep understanding of a wide range of features. This pre-training makes them incredibly versatile and efficient when adapted to new tasks.

In conclusion, the developed CNN model demonstrates significant potential for automating the monitoring of milling processes in the furniture manufacturing industry. By harnessing the power of advanced machine learning techniques, this model paves the way for more efficient, accurate, and automated tool condition monitoring.

The performance evaluation of the CNN model is a critical step in the development process. It allows us to assess the model's ability to generalize to new data, which is indicative of its practical utility in real-world applications. This section details the evaluation metrics and results obtained from testing the CNN model on the dataset for tool condition monitoring during the milling process.

4.2. Model architecture

The proposed model integrates the high-level feature extraction capabilities of the aforementioned pretrained networks, as illustrated in Figure 4. Each pretrained network serves as a separate feature extractor for a different input data stream. The architecture allows for the parallel processing of multiple data signals, each passing through a frozen base of a pretrained network. These signals correspond to the various parameters captured during the milling process, such as acoustic emissions, force measurements, and electrical signals.

Each data stream, denoted as DataHigh, DataLow0 through DataLow3, and DataCurrent0 through DataCurrent5, is processed through a distinct instance of the pretrained network. The pretrained network bases are kept frozen to preserve the knowledge they have acquired from vast amounts of visual data, ensuring the effectiveness of the feature extraction as seen in Figure 4.

Post feature extraction, the outputs from all pretrained networks are flattened and then concatenated into a single, common feature vector. This concatenated vector encapsulates comprehensive feature information important to the milling process, which is then passed through a series of fully connected (Dense) layers.

The first Dense layer consists of 512 neurons followed by batch normalization and a ReLU activation function. This is followed by a dropout layer with a dropout rate of 0.1 to mitigate the risk of overfitting. Subsequent layers reduce the dimensionality from 256 to 128 neurons, each followed by batch normalization, ReLU activation, and dropout layers.

The final layer in our architecture is a Dense layer with a softmax activation function. This layer outputs the probabilities corresponding to the different conditions of the tool – ‘Green’, ‘Yellow’, and ‘Red’ – indicating the level of wear and tear experienced by the milling apparatus.

The model’s versatility is underpinned by its ability to harness the strengths of each pretrained network. By adjusting the trainable parameters in the fully connected layers and maintaining the sophisticated feature extraction capabilities of the pretrained bases, our model is adept at providing nuanced classifications of tool conditions, thereby significantly enhancing the monitoring process.

The architecture of the model leverages the power of pretrained networks to analyze complex data streams effectively. The incorporation of these networks into our model reflects a significant step towards advanced monitoring and predictive maintenance in the domain of manufacturing.

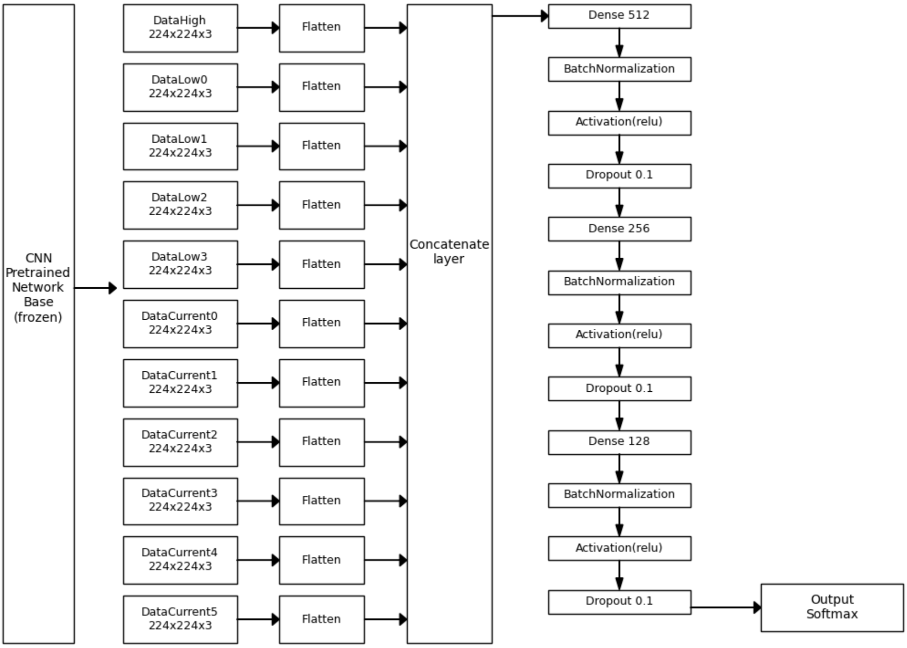


Fig. 4. Schematic representation of the CNN model utilizing pretrained networks for chipboard milling process monitoring.

5. Numerical Experiments

5.1. Cross-Validation Methodology

In this study, we applied cross-validation as a robust approach to evaluate the performance of our machine learning models. Specifically, we used the stratified K-fold cross-validation technique, ensuring that each fold is a good representative of the whole. Stratified K-fold divides the dataset into K equally sized folds (in our case $K = 10$), with each fold containing approximately the same percentage of samples of each target class as the complete set.

For each fold in the cross-validation, every model was trained on the training subset and evaluated on the validation subset. The evaluation metrics included accuracy, confusion matrix, and a detailed classification report highlighting precision, recall, and F1-score for each class.

We also conducted a detailed error analysis by examining the confusion matrices.

Particular attention was given to extreme classification errors between the most dissimilar classes.

The following pseudocode provides an overview of our cross-validation implementation (Algorithm 3):

Algorithm 3 Cross-Validation for Model Evaluation

```
$data, labels \leftarrow$ loaddata()
$kf \leftarrow$ StratifiedKFold($n_{splits} = 10$)
for $fold, (train_{index}, val_{index})$ in $kf.split(data, labels)$ do
    $model \leftarrow$ createmodel()
    $train_{data}, val_{data} \leftarrow data[train_{index}], data[val_{index}]$
    $train_{labels}, val_{labels} \leftarrow labels[train_{index}], labels[val_{index}]$
    $model.fit(train_{data}, train_{labels})$
    $YPred \leftarrow model.predict(val_{data})$
    Calculate and Analyze $YPred$ against $val_{labels}$
    Generate classification reports and confusion matrices for current fold
end for
Generate Final classification report and confusion matrix for all 10-folds.
```

The results of the cross-validation process, including accuracy scores, confusion matrices, and classification reports, are compiled and analyzed to assess the overall performance and robustness of every the models.

6. Hardware Configuration

Below Table 3 details the hardware setup utilized in our computational experiments. The hardware configuration played a crucial role in the successful implementation and execution of our CNN models. The table below summarizes the key hardware components used in our setup.

Our system was equipped with dual Intel(R) Xeon(R) Gold 5420+ processors. These high-performance processors are well-suited for demanding computational tasks, such as training deep neural networks. They ensure efficient parallel processing and fast computation speeds, which are essential for handling large datasets and complex algorithms.

The system's memory architecture included 2240 KiB of L1 cache, 56 MiB of L2 cache, and 52 MiB of L3 cache. Additionally, it boasted a substantial RAM capacity of 256 GiB, comprising 16 DIMM slots, each fitted with a 16 GiB module operating at 4800 MHz. This extensive memory setup was critical for managing the high data throughput and storage requirements of our CNN models.

For operating system and software, the setup included 2×128 GB NVMe disks, known

Tab. 3. Summary of hardware configuration.

Type	**Hardware Item**
Processor #1	Intel(R) Xeon(R) Gold 5420+
Processor #2	Intel(R) Xeon(R) Gold 5420+
Memory Cache L1	2240KiB L1 cache
Memory Cache L2	56MiB L2 cache
Memory Cache L3	52MiB L3 cache
Memory RAM	256GiB System Memory (16*16GiB DIMM 4800 MHz)
OS Disk	2x128 GB NVMe disk
Storage	26TB RAID
Network	4xNetXtreme BCM5720 Gigabit Ethernet PCIe
GPU #1	NVIDIA A40, 48 GB GDDR6 with ECC
GPU #2	NVIDIA A40, 48 GB GDDR6 with ECC
GPU #3	Integrated Matrox G200eW3 Graphics Controller

for their high-speed data transfer rates. The main data storage was a robust 26 TB RAID system, offering both large storage capacity and data redundancy, which is crucial for maintaining data integrity in large-scale computational experiments.

The network configuration comprised 4 NetXtreme BCM5720 Gigabit Ethernet PCIe adapters. This setup provided high-speed network connectivity, ensuring efficient data transfer within our computational network and facilitating remote access to the computational resources.

The computational rig was equipped with two NVIDIA A40 GPUs, each offering 48 GB of GDDR6 memory with ECC. These GPUs were instrumental in accelerating the training and inference processes of our CNN models. The integrated Matrox G200eW3 Graphics Controller served as an auxiliary GPU, primarily handling display outputs and less intensive graphical tasks.

This hardware configuration provided a robust and efficient platform for conducting our advanced computational experiments, particularly in training and evaluating deep neural network models.

7. Results and Discussion

This section presents the results obtained from the application of different pretrained CNN models, namely VGG16, VGG19, and RESNET34, in the monitoring of the milling chipboard process. The performance of each model is evaluated based on the confusion matrices and various performance metrics as summarized in Table 1.

7.1. Performance Evaluation

The evaluation of the pretrained CNN models involved analyzing their ability to accurately classify tool conditions into ‘Green’, ‘Yellow’, and ‘Red’ states. Figures 5a, b and c depict the confusion matrices for the VGG16, VGG19, and RESNET34 models, respectively (Table 4).

- The VGG16 and VGG19 models both achieved an accuracy of 80%, whereas the RESNET34 model showed a slightly lower accuracy of 70.67%.
- Critical errors, which are severe misclassifications, were slightly higher for the VGG19 model (3 errors) compared to VGG16 and RESNET34 (2 errors each).
- The training times varied among the models, with VGG19 taking the longest at approximately 1 hour and 10 min, followed by VGG16 at 55 min, and RESNET34 being the fastest at 32 minutes.

7.2. Class-Specific Analysis

A detailed analysis of each class’s performance reveals significant insights into the models’ classification capabilities:

- For the VGG16 model (Table 5), the ‘Green’ class showed the highest precision and specificity, indicating its effectiveness in correctly identifying well-conditioned tools

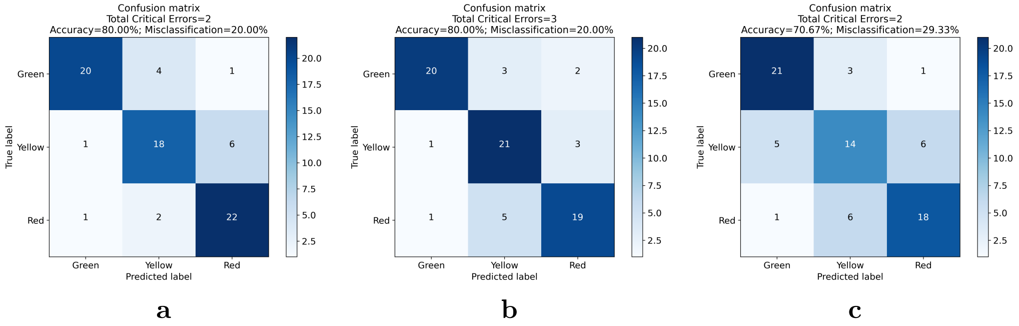


Fig. 5. Confusion matrices for the CNN model with three pretrained networks: (**a**) VGG16, (**b**) VGG19, (**c**) RESNET34.

Tab. 4. Comparison of Performance Metrics for Different Pretrained Neural Networks.

Pretrained network	**Accuracy [%]**	**Critical Errors**	**Training Time**
VGG16	80.00	2	55 min
VGG19	80.00	3	1 h 10 min
RESNET34	70.67	2	32 min

Tab. 5. Comparative analysis of VGG16 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	90.91	80.00	85.11	96.00
‘Yellow’	75.00	72.00	73.47	88.00
‘Red’	75.86	88.00	81.48	86.00

and minimizing false positives. The ‘Yellow’ and ‘Red’ classes had lower precision and sensitivity, pointing towards challenges in distinguishing between slightly worn and critically worn tool conditions.

- The VGG19 model (Table 6), while similar in precision and sensitivity for the ‘Green’ class as VGG16, showed better sensitivity for the ‘Yellow’ class and a balanced precision-sensitivity trade-off for the ‘Red’ class.
- The RESNET34 model (Table 7) exhibited lower overall precision and sensitivity across all classes, with the ‘Yellow’ class being the most challenging in terms of precision and sensitivity.

Tab. 6. Comparative analysis of VGG19 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	90.91	80.00	85.11	96.00
‘Yellow’	72.41	84.00	77.78	84.00
‘Red’	79.17	76.00	77.55	90.00

Tab. 7. Comparative analysis of RESNET34 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	77.78	84.00	80.77	88.00
‘Yellow’	60.87	56.00	58.33	82.00
‘Red’	72.00	72.00	72.00	86.00

7.3. Discussion

The results indicate that while all three models are capable of classifying tool conditions effectively, there are distinct differences in their performance. VGG16 and VGG19 models are more accurate but require longer training times, which might be a trade-off in real-time applications. The RESNET34 model, despite its lower accuracy, offers a faster training process, which could be beneficial in scenarios where rapid model deployment is necessary.

The higher precision in the ‘Green’ class across all models suggests that they are well-suited for identifying tools in good condition. However, the lower sensitivity in ‘Yellow’ and ‘Red’ classes, especially in the RESNET34 model, highlights the need for further model optimization to improve the detection of worn and critically worn tools.

In conclusion, the choice of a pretrained CNN model for tool condition monitoring in milling processes should be based on the specific requirements of accuracy, training time, and the ability to distinguish between different wear conditions. Future work may explore combining these models or employing ensemble methods to enhance overall performance and reliability.

8. Conclusion

This study presented a comprehensive approach to monitoring and analyzing the milling chipboard process in furniture manufacturing using Convolutional Neural Networks with pretrained networks like VGG16, VGG19, and RESNET34. Our research demonstrated the effectiveness of these models in accurately classifying tool conditions into ‘Green’, ‘Yellow’, and ‘Red’ states based on their wear levels, utilizing spectrogram representations of time-series data collected during the milling process.

The experimental results showed that VGG16 and VGG19 achieved higher accuracy compared to RESNET34, although with a longer training time. This highlights a trade-off between accuracy and computational efficiency that must be considered in practical applications. While all models proved capable in identifying well-conditioned tools, challenges remain in differentiating between slightly worn and critically worn conditions, especially for the ‘Yellow’ and ‘Red’ classes.

Furthermore, the application of machine learning in this domain has opened new approaches for predictive maintenance and quality control in manufacturing. By enabling early detection of tool wear, the implemented models can significantly contribute to optimizing the milling process, reducing downtime, and ensuring product quality.

This research also underscores the potential of transfer learning, where pretrained models, originally developed for different tasks, can be successfully adapted and applied to specific industrial processes. It paves the way for further exploration of deep learning techniques in manufacturing, where similar strategies can be employed to enhance various aspects of production and monitoring systems.

In conclusion, the findings of this study contribute valuable insights into the application of advanced neural networks in industrial settings. Future work may focus on further refining these models, exploring ensemble methods for improved accuracy, and extending this approach to other manufacturing processes to fully harness the potential of AI in industrial automation and monitoring.

References

[1] M. Gao, D. Qi, H. Mu, and J. Chen. A transfer residual neural network based on ResNet-34 for detection of wood knot defects. *Forests*, 12(2), 2021. doi:10.3390/f12020212.

[2] P. Iakubovskii (qubvel). Classification models Zoo – Keras (and TensorFlow Keras). https://github.com/qubvel/classification_models, 2023. [Accessed: 2023-12-01].

[3] P. Iskra and R. E. Hernández. Toward a process monitoring and control of a cnc wood router: Development of an adaptive control system for routing white birch. *Wood and Fiber Science*, 42(4):523–535, 2010. https://wfs.swst.org/index.php/wfs/article/view/567.

[4] A. Jegorowa, J. Górski, J. Kurek, and M. Kruk. Initial study on the use of support vector machine (SVM) in tool condition monitoring in chipboard drilling. *European Journal of Wood and Wood Products*, 77(5):957–959, 2019. doi:10.1007/s00107-019-01428-5.

[5] A. Jegorowa, J. Górski, J. Kurek, and M. Kruk. Use of nearest neighbors (K-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas: Ciencia y Tecnologia*, 22(2):189–96, 2020. doi:10.4067/S0718-221X2020005000205.

[6] A. Jegorowa, J. Kurek, I. Antoniuk, W. Dołowa, M. Bukowski, et al. Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard. *Wood Science and Technology*, 55(1):271–293, 2021. doi:10.1007/s00226-020-01245-7.

[7] R. J. Kuo. Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Engineering Applications of Artificial Intelligence*, 13(3):249–261, 2000. doi:10.1016/S0952-1976(00)00008-7.

[8] J. Kurek, I. Antoniuk, J. Górski, A. Jegorowa, B. Świderski, et al. Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network. *Machine Graphics & Vision*, 28(1/4):13–23, 2019. doi:10.22630/MGV.2019.28.1.2.

[9] J. Kurek, I. Antoniuk, J. Górski, A. Jegorowa, B. Świderski, et al. Data augmentation techniques for transfer learning improvement in drill wear classification using convolutional neural network. *Machine Graphics and Vision*, 28(1/4):3–12, 2019. doi:10.22630/MGV.2019.28.1.1.

[10] J. Kurek, I. Antoniuk, B. Świderski, A. Jegorowa, and M. Bukowski. Application of siamese networks to the recognition of the drill wear state based on images of drilled holes. *Sensors (Switzerland)*, 20(23):1–16, 2020. doi:10.3390/s20236978.

[11] J. Kurek, B. Świderski, A. Jegorowa, M. Kruk, and S. Osowski. Deep learning in assessment of drill condition on the basis of images of drilled holes. In: *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, vol. 10225, pp. 375–381. SPIE, 2017. doi:10.1117/12.2266254.

[12] S. Mascarenhas and M. Agarwal. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification. In: *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1, pp. 96–99, 2021. doi:10.1109/CENTCON52345.2021.9687944.

[13] matplotlib.pyplot.specgram – matplotlib 3.5.1 documentation. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.specgram.html, 2023. [Accessed: 15-11-2023].

[14] S. Osowski, J. Kurek, M. Kruk, J. Górski, P. Hoser, et al. Developing automatic recognition system of drill wear in standard laminated chipboard drilling process. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, pp. 633–640, 2016. doi:10.1515/bpasts-2016-0071.

[15] S. S. Panda, A. K. Singh, D. Chakraborty, and S. K. Pal. Drill wear monitoring using back propagation neural network. *Journal of Materials Processing Technology*, 172(2):283–290, 2006. doi:10.1016/j.jmatprotec.2005.10.021.

[16] SciPy.interpolate.interp1d — scipy v1.8.0 reference guide. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html, 2023. [Accessed: 15-11-2023].

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2015. ArXiv.1409.1556. doi:10.48550/arXiv.1409.1556.

[18] K. Szwajka and T. Trzepieciński. Effect of tool material on tool wear and delamination during machining of particleboard. *Journal of Wood Science*, 62(4):305–315, 2016. doi:10.1007/s10086-016-1555-6.

[19] tf.keras.applications.vgg16.vgg16. https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg16/VGG16. [Accessed: 2023-12-01].

[20] tf.keras.applications.vgg19.vgg19. https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg19/VGG19. [Accessed: 2023-12-01].

[21] W. Wei, Y. Li, T. Xue, S. Tao, C. Mei, et al. The research progress of machining mechanisms in milling wood-based materials. *BioResources*, 13(1):2139–2149, 2018. doi:10.15376/biores.13.1.Wei.

Xception-based Architecture with Cross-sampled Training for Image Quality Assessment on KonIQ-10k

Tomasz M. Lehmann*, Przemysław Rokita
Warsaw University of Technology, Warsaw, Poland
**Corresponding author: Tomasz M. Lehmann (tomasz.lehmann@dokt.pw.edu.pl)*

Abstract. Image quality assessment is a crucial task in various fields such as digital photography, online content creation, and automated quality control, as it ensures an optimal visual experience and aids in maintaining consistent standards. In this paper, we propose an efficient method for training image quality assessment models on the KonIQ-10k dataset. Our novel approach utilizes a dual-Xception architecture that analyzes both the image content and additional image parameters, outperforming traditional single convolutional models. We introduce cross-sampling methods with random draw sampling of instances from majority classes, effectively enhancing prediction quality in the Mean Opinion Score (MOS) ranges that are underrepresented in the database. This methodology allows us to achieve near state-of-the-art results with limited computing costs and resources. Most importantly, our predictions across the entire spectrum of MOS values maintain consistent quality. Because of using a novel and highly effective method for image sampling, we achieved these results with much lower computational cost, making our approach the most effective way of MOS estimation on the KonIQ-10k database.

Key words: image quality assessment, computer vision, Xception

1. Introduction

Image quality refers to the fidelity of imaging systems in capturing and processing signals to form an image, and the weighted sum of visually important features as perceived by the human eye [1]. This dual perspective is crucial in applications like diagnostics, environmental monitoring, visual media, security, and manufacturing, impacting decision-making and operational efficiency. Both the technical fidelity and subjective appeal highlight the need for robust image quality assessment.

Objective image quality assessment is divided into no-reference, reduced-reference, and full-reference methods, based on the original image's availability. Full-reference metrics compare a test image with the original; reduced-reference uses limited original information, and no-reference assesses the image independently. These methods provide automated metrics for estimating image quality [2].

Blind Image Quality Assessment (BIQA) stands out as the most complex yet most applicable among the three types of image quality assessments because it does not need a reference image. In many cases, such references are not accessible. Deep learning advancements have shown significant potential in enhancing BIQA alongside other areas like image recognition and object identification. The development of BIQA methods would benefit substantially from a vast and varied database that includes naturally occurring image distortions. Nevertheless, the training of deep learning models for

Machine GRAPHICS & VISION 32(2):109–127, 2023. DOI: 10.22630/MGV.2023.32.2.6 .

BIQA is currently constrained by the limited scope and synthetic nature of existing databases [3, 4, 5]. Moreover, large-scale quality assessments in a controlled environment are not feasible, given the extensive time and participant involvement required. In the study referenced as [6], the researchers introduced a groundbreaking dataset named KonIQ-10k, consisting of 10,073 images each with an associated quality score. Additionally, they developed a CNN-based model, KonCept512, which surpassed competing models [7,8,9] in performance on both the KonIQ-10k and the LIVE-itW [10] databases.

In this paper, we focused on replicating results with the streamlined Xception architecture [11], which has fewer parameters (22.8 million when the architecture proposed by the original KonIQ-10k dataset authors contains around 56 million variable parameters). Architectures with fewer parameters typically learn faster and are more effective on small datasets because they are less prone to overfitting and require fewer computational resources for training. This is a significant advantage in the context of novel approaches to data distribution during training steps. We explored the effectiveness of a hybrid deep learning model that utilizes dual CNN extractors. Moreover, we have confirmed that incorporating undersampling (due to methods of random sampling from major classes and data duplication in minor classes, referred to as cross-sampling) [13] to reduce training times does not detrimentally impact the overall results. All these additional improvements make our model easier and much faster to train, as well as quicker and lighter for inference.

1.1. Related Works

Image Quality Assessment (IQA) is crucial in many fields, playing a key role in ensuring the precision and efficiency of numerous advanced decision-making and operational systems. IQA is generally divided into subjective and objective categories. Subjective IQA depends on human evaluations, leading to the Mean Opinion Score (MOS) system, which reflects the average perceived image quality. However, its time-consuming and costly nature limits its practical use.

Objective IQA, especially no-reference or Blind Image Quality Assessment (BIQA), has greatly advanced with deep learning. While traditional BIQA methods focused on manually selected features, recent trends lean towards automatic representation learning from raw images to predict quality scores. Deep learning in BIQA, such as the application of deep belief networks by Ghadiyaram et al. and the VGG16 network in DeepBIQ and BLINDER models [14, 15], showcases the effectiveness of deep neural networks in this area. These models estimate image quality by analyzing various image sections and averaging their MOS, considering both individual and overall image quality scores.

Pixel-by-Pixel IQA (pIQA) [16] marks a significant progress in this sector, introducing an innovative way to compute the MOS for each pixel and sum it up for an overall image score. This method surpasses older IQA techniques and aligns closely with human vision, representing a major step forward in objective IQA.

Deep learning-based Full-Reference IQA (FR-IQA) [17] methods have also been developed, focusing on the similarity between an original and altered image. However, their effectiveness varies with the image's complexity, encouraging further exploration of deep learning for more effective feature extraction.

Transfer learning has been utilized to address the challenges of small training datasets in BIQA. Examples include RankIQA [18], which employs a Siamese Network for image quality ranking, and MEON [19], which uses a multi-task approach with shared initial layers for distortion detection and quality prediction. MS-UNIQUE [17], an FR-IQA method, leverages multiple linear decoders trained on large datasets to gauge visual quality by comparing feature vectors of original and distorted images. Talebi et al. proposed a framework based on object-classification architectures [20], while Zhang et al. used a Siamese network for MOS-based image pair ranking [8].

The KonIQ-10k [6] dataset is a notable recent contribution, specifically created for BIQA prediction. The KonCept512 model, based on the Inceptionv2 architecture, has achieved top-tier results on both the KonIQ-10k and LIVE-itW datasets. However, this solution has room for optimization in MOS range prediction and computational efficiency. Inceptionv2, being a large structure, demands significant computing resources, which may pose challenges for training on commonly available and free personal computers and notebooks.

2. Experimental setup

2.1. Dataset

The KonIQ-10k dataset, the largest of its kind for Image Quality Assessment (IQA), includes 10 073 images, each evaluated for quality. Notable for its ecological validity, the dataset prioritizes authenticity in distortion types, content variety, and quality measures. Developed through extensive crowdsourcing, it incorporates over 1.2 million quality evaluations from 1 459 participants, offering a robust foundation for advancing IQA models.

In Figure 1 several images from the specified database are displayed.

For quality indicators, the dataset incorporates measures well-correlated with human perception. These include brightness, colorfulness, Root Mean Square (RMS) contrast, and sharpness, as revealed by preliminary subjective studies. Other factors like image bitrate, resolution, and JPEG compression quality were also assessed. In our research, we narrowed our focus to brightness, contrast, sharpness, and bitrate because these showed a stronger correlation with the Mean Opinion Score (MOS) what is depicted in dataset's correlation matrix (Figure 2). In our study, we employed four key indicators – brightness, contrast, sharpness, and bitrate – as inputs to the Xception architecture [11].

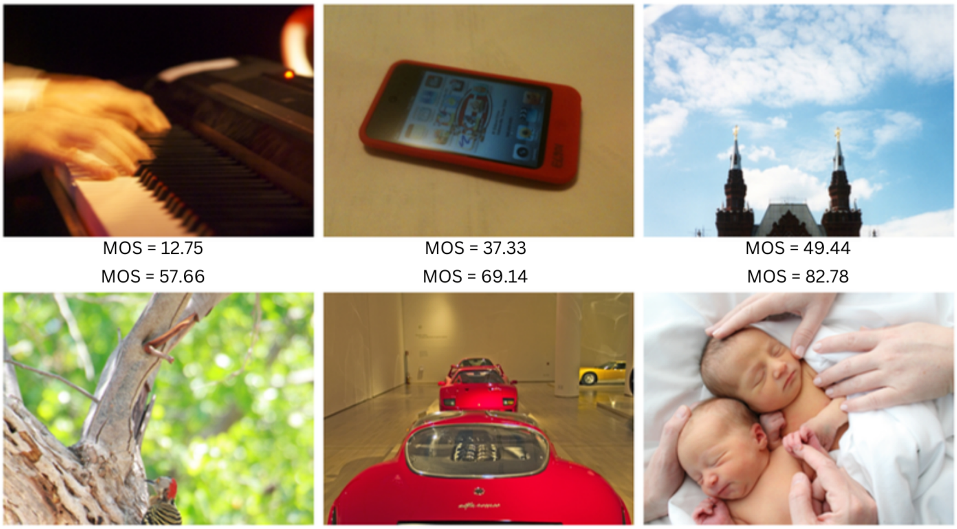


Fig. 1. The image showcases six examples from the KonIQ-10k database, each accompanied by an MOS value derived from the dataset labels positioned either above or below them. Images demonstrating relatively lower quality (MOS < 50) exhibit characteristics such as being cropped, blurred, or noisy. Conversely, as the MOS increases, the images become clearer and more precisely cropped.

This informed the extraction of a feature vector that was integral to our hybrid neural network, allowing for improved image quality predictions based on the KonIQ-10k dataset's findings.

As indicated in the original KonIQ-10k paper, we also divided our dataset into three subsets (training, validation, testing), adhering to the same distribution ratio: 7 058 elements for training, 1 000 for validation, and 2 015 for testing.

2.2. Proposed methods

2.2.1. Model architecture

In our study, we chose the Xception architecture over InceptionResnetv2 [21] due to its efficient use of parameters without compromising on performance. The significantly lower number of trainable parameters was also a crucial factor in our decision, given our intention to use the cross-sampling method, which limits the number of data points in the training set. This choice is supported by comparative evaluations in which Xception surpassed other residual-connected CNNs, including ResNet50, ResNet152 [22], and

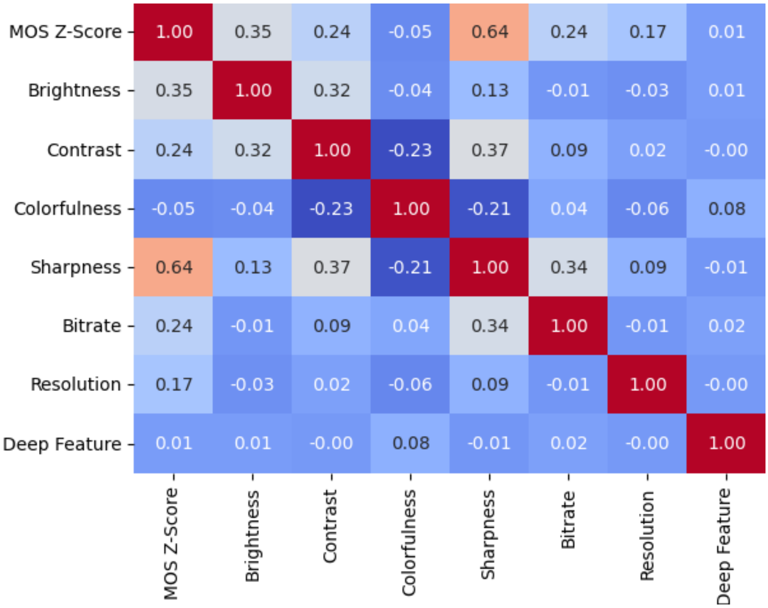


Fig. 2. The pairwise Pearson correlation coefficients visualization among various image quality indicators from the KonIQ-10k dataset. It encapsulates the relationships between Mean Opinion Score (MOS) Z-Score, Brightness, Contrast, Colorfulness, Sharpness, Bitrate, and Resolution, along with extracted Deep Features. The depicted correlations offer a concise overview of the interdependencies between perceived image attributes.

SENet154 [23], in terms of processing efficiency while still maintaining competitive accuracy. This efficacy positions Xception as the preferred model for our image analysis tasks. The most significant differences between the Xception and InceptionResNetV2 architectures lie in their structural design and efficiency.

Xception revolutionizes the traditional Inception architecture by adopting depthwise separable convolutions, which streamline the model by reducing the number of parameters without sacrificing efficiency. Unlike InceptionResNetV2, which enhances the Inception model with residual connections for increased depth and complexity, potentially improving accuracy at the expense of greater computational demands, Xception optimizes for both computational efficiency and performance. This is achieved by decoupling the mapping of cross-channel and spatial correlations in the feature maps, a strategy that allows Xception to surpass the performance of its Inception counterparts in benchmark tasks while requiring fewer computational resources.

Xception is an advanced deep learning model that utilizes depthwise separable convolutions as a fundamental building block, optimizing computational efficiency and model performance. It diverges from InceptionResNetV2 by employing depthwise separable convolutions, which decouple the mapping of cross-channel correlations and spatial correlations in feature maps, instead of the Inception modules with mixed convolutions. This architectural choice facilitates a more efficient use of model parameters and enables the Xception network to outperform its Inception counterparts on benchmark tasks with fewer computational resources. In our research, we observed that a single training step for the Xception model is around twice as fast.

The Xception framework, embodying "Extreme Inception", is composed of 36 convolutional layers structured into 14 modules, all based on depthwise separable convolutions (Fig. 3). This design principle posits that the correlations within the feature maps of convolutional neural networks can be effectively separated, leading to a model that is both powerful and efficient. The architecture, characterized by its simplicity akin to the VGG16 model but diverging from the more intricate Inception designs, is detailed in its foundational publication [11]. Our findings indicate that Xception's training process is notably faster, with a single step taking roughly half the time compared to more complex models.

To assess the impact of extracted image parameters (brightness, contrast, sharpness, and bitrate) on the quality of results, we utilized a pioneering dual-Xception architecture. One Xception model, pre-trained on the ImageNet dataset [24], was used to extract a feature tensor the size of the original architecture's last linear layer (1024 neurons). A second Xception was adapted to accept four floating-point values (the aforementioned parameters) as input and was not pre-trained. The tensor returned by this part of the proposed structure also had a dimension of 1024. Outputs from both models were then merged using a matrix concatenation operation. The combined feature vector (of size 2048) was processed through a Leaky ReLU activation function and a final linear layer with a single output size.

The Dual-Xception architecture is presented in the diagram in Figure 4.

While this approach increased computational complexity, the goal was to evaluate the model's sensitivity to the parameters introduced into the second model. We examined the models' performance across six scenarios: image-only Xception (with and without cross-sampling), parameters-only Xception (with and without cross-sampling), and the combined architecture (with and without cross-sampling).

In the initial stage, we also compared the proposed dual-Xception architecture with more traditional methods. Our goal was to develop a single, cohesive structure capable of learning from both image data and additional parameter information. While it was possible to use two or three separate models for this purpose, maintaining the entire training pipeline in a unified form was crucial, where we trained just one neural network function using only one optimizer and loss function.

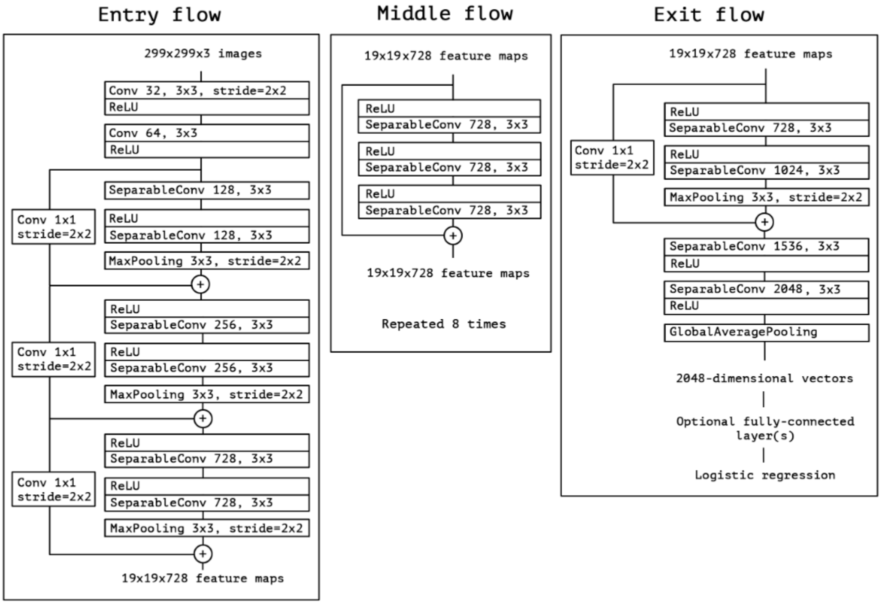


Fig. 3. The complex structure of the Xception architecture is derived directly from the paper [11] (replicated here from [12] according to arXiv License Information). It is important to note that all convolutions are followed by batch normalization, and all SeparableConvolution structures use a depth multiplier.

The dual-Xception model's sophisticated approach offers distinct advantages by separately processing scalar attributes with an Xception model, rather than merely concatenating them with image features or using a basic linear layer. This method acknowledges the complexity of scalar attributes like brightness and contrast, allowing for a nuanced abstraction and integration with image-derived features. It enhances the model's ability to grasp the intricate, non-linear relationships between these attributes and image quality, potentially improving accuracy.

However, the dual-Xception framework's complexity and computational demands are notable drawbacks, increasing training time and data requirements to avoid overfitting. The complexity may also complicate model adjustments and necessitate meticulous regularization.

We decided to evaluate the proposed novel architecture by comparing the mentioned architecture with simpler networks illustrated in the Figure 5.

On the left, we observe that a simple linear neural network layer is employed as a replacement for the parameter-focused Xception part, thereby reducing computational

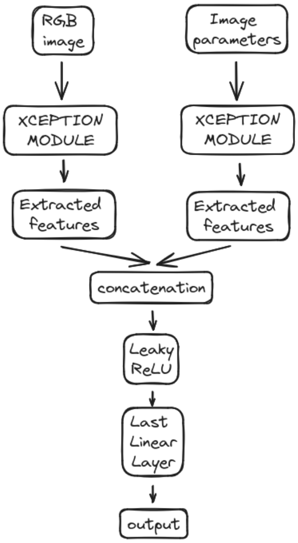


Fig. 4. The Dual-Xception architecture consists of two primary components. The first is the standard Xception model trained on RGB images. The second is a modified version of Xception, designed to process a 4-dimensional input reflecting image parameters: brightness, contrast, sharpness, and bitrate. The duality lies in how these separate models are combined: features extracted from both networks are concatenated and then passed through additional network layers (including an activation function and a linear layer) to predict the final image quality, measured by the Mean Opinion Score (MOS).

complexity. The resulting tensor was then concatenated with features extracted from the RGB image. On the right, we note that scalars are treated as tensors themselves without any preprocessing or feature engineering methods.

2.2.2. Loss function

In this paper, we propose the use of Mean Square Error (MSE) as a metric for assessing the average squared difference between the estimated Mean Opinion Score (MOS) and the labeled ground truth. MSE is a widely adopted approach in a multitude of computer vision-based predictive models. The equation for the loss function is presented below:

$$\mathrm{MSE}(x,y) = \frac{1}{n}\sum_{n}(x_n - y_n)^2\,, \tag{1}$$

where:
n – number of images in the batch,
x_n – ground truth MOS,

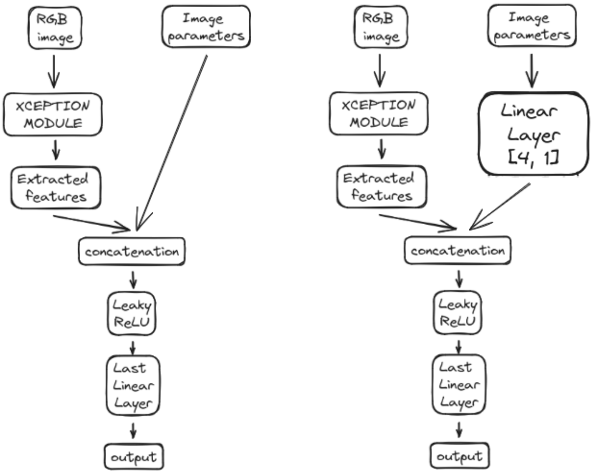


Fig. 5. On the left, we see an architecture where a linear layer replaces the parameter-oriented Xception module. The input size matches the number of included parameters, while the output size is set to 1. On the right, we treat the scalar values of parameters as features in themselves, without proposing any transformations except tensorification. In both cases, a crucial step is the concatenation operation, where features extracted from the RGB image are merged with information extracted from parameters in two distinct manners.

y_n – predicted MOS.

To assess the impact of cross-sampling on MSE within narrower MOS intervals, we tracked this metric across the following ranges: 0-20, 20-40, 40-60, 60-80, and 80-100. The data was categorized according to the labeled MOS values.

2.2.3. Cross-sampling

To optimize computing time and enhance results in MOS ranges with fewer training examples, we employed an cross-sampling strategy.

Under-sampling is a technique employed to address imbalances in datasets by decreasing the size of the more dominant class to match that of the minority class. This method is part of a suite of tools that data scientists use to extract more accurate insights from datasets that initially exhibit a skew in class distribution. In our solution, we have also incorporated randomness into the drawing of samples at every step. Therefore, we prefer to refer to this procedure as cross-sampling, and we adhere to this terminology throughout this paper.

Given the substantial imbalance in our dataset across the ranges 0-20, 20-40, 40-60, 60-80, and 80-100, we applied a straightforward algorithm to ensure the model paid greater attention to underrepresented classes. If the number of images in a range was more than double but less than quadruple the quantity in the smallest range, the dataset was randomly reduced during each training epoch to a maximum of twice the size of the smallest class. If the quantity was between four to six times larger, the limit was set to three times the size of the smallest class, and so forth, capping at four times the size for any larger quantities. This approach effectively created a more balanced training environment, promoting better learning from minority classes.

2.2.4. Training procedures

The model training was conducted on an NVidia RTX 3070 GPU with 8GB memory. Each training session involved 60 epochs, with a batch size of 4, at a resolution of 512x384. The duration of training varied between 5 to 20 hours, depending on the architecture and whether the dataset was processed with cross-sampling. The number of training epochs was suggested by the authors of KonIQ-10k, and in the subsequent chapters, we demonstrate that it was not a very accurate approximation. After each epoch, the model was evaluated on a validation set to monitor for signs of overfitting. Ultimately, the best-performing model – characterized by the lowest loss – was selected for testing. The ADAM algorithm [25], known for its adaptive learning rate methods, served as the optimizer. The initial learning rate was set at 0.0001, halving every 20 epochs.

For our metrics, we utilized PLCC and SROCC, both of which are widely employed in the evaluation of image quality.

PLCC stands for Pearson Linear Correlation Coefficient, which measures the linear correlation between two variables, providing a value between −1 and 1. A PLCC of 1 indicates perfect positive correlation, while −1 indicates a perfect negative correlation. It is often used in image quality assessment to compare the similarity between the quality ratings of images by an algorithm and subjective ratings by humans.

SROCC denotes Spearman's Rank Order Correlation Coefficient, a non-parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. In the context of image quality assessment, it ranks the images based on quality and compares the algorithm's rankings with those from human assessments.

$$\mathrm{PLCC}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}\,, \tag{2}$$

where:
n – number of observations,
X_i – i^{th} observation of variable X,
Y_i – i^{th} observation of variable Y,
$\overline{X}$ – mean of all observations of variable X,
$\overline{Y}$ – mean of all observations of variable Y.

$$\mathrm{SROCC}(X,Y) = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}\,, \tag{3}$$

where:
n – number of observations,
d_i – difference between the ranks of the i^{th} observations of variables X and Y.

3. Experimental results

The authors of the referenced paper conducted experiments to find the optimal input resolution for their model by training on the original resolution (1024 × 768) and two lower resolutions (512 × 384 and 224 × 224). They found that the models trained on the smallest resolution (224 × 224) performed worse than the others, suggesting a significant loss of quality-related information during the down-sampling process. Interestingly, the models trained on the medium resolution (512 × 384) outperformed those trained at the original resolution.

For our study, we exclusively utilized the Xception architecture, pre-trained on the ImageNet dataset. The results demonstrated that both PLCC and SROCC metrics were slightly better for the 512 × 384 resolution compared to the 1024 × 768, and substantially better than the 224 × 224 resolution.

These findings led us to exclusively utilize the 512 × 384 resolution in subsequent operations.

3.1. Comparative Analysis of Dual-Xception Architectures

In this analysis, we evaluate the effects of modifications to the Xception architecture on performance outcomes. Initially, the original Xception model was trained using the KonIQ-10k image dataset.

Subsequently, we compared three models derived from architectures proposed in section 2.2.1.

The first and simplest architecture incorporates tensored parameters as additional features, combining this information with features extracted from the RGB image through concatenation.

The second approach introduces a simple linear layer to process parameters, functioning as a parameter-oriented branch of the architecture.

The third solution employs a modified Xception model designed to accept a 4-dimensional input of image parameters: brightness, contrast, sharpness, and bitrate, in lieu of standard RGB images. In this design, the parameters are fed into the initial layer of the modified Xception network, effectively substituting the first convolutional layer that typically processes RGB values. While a Multi-Layer Perceptron (MLP) could handle these

Tab. 1. Results comparing three methods of integrating features from images with image parameters are presented. The first method, labeled 'Plain Parameters', involved no preprocessing but directly combined tensored parameter values. The second method, 'Linear Layer', utilized a linear neural network to extract features from image parameters. The third method utilized our novel dual-Xception architecture.

	Plain parameters	**Linear Layer**	**dual-Xception**
PLCC	0.912	0.916	0.920
SROCC	0.896	0.898	0.903
MSE	6.32	6.22	6.06
MSE (0-20)	10.18	13.31	12.01
MSE (20-40)	8.82	8.41	8.48
MSE (40-60)	6.73	6.60	6.66
MSE (60-80)	5.10	4.87	4.56
MSE (80-100)	6.61	7.24	6.10
Parameters	20.8M	20.8M	41.6M

parameters, the modified Xception model still utilizes depthwise separable convolutions, which are central to Xception's design. This approach might leverage the convolutions for effective feature extraction and transformation from non-image data, representing a novel strategy that could treat the spatial hierarchies and patterns within the parameters similarly to image features. Future research could explore different architectures to refine this extractor and further reduce computational demands.

We evaluated the models using three main metrics: PLCC, SROCC, and MSE, as detailed in the previous chapter. Additionally, we assessed the MSE within the ground truth range of the MOS parameters, dividing these ranges into five categories (MOS from 0 to 20, MOS between 20 and 40, etc.). A lower MSE value indicates more accurate predictions. For PLCC and SROCC, the ideal value is 1, with the value decreasing as prediction quality deteriorates (down to a minimum value of 0).

In Table 1 we present results comparing three methods of combining features extracted from images with image parameters. In the first column, titled "Plain Parameters", we show the metrics for the method where we used no preprocessing but combined tensored parameter values directly. In the middle column, titled "Linear Layer", we present results achieved by using a linear neural network as a feature extractor from image parameters. In the third column, we present results for our pioneering dual-Xception architecture. We observe that the differences in results are relatively small. However, in most cases, the dual-Xception-based network achieved better results.

The convolutional-based extractor might be better for scalars because it can capture non-linear relationships and subtle patterns within the data, which a simple linear model might overlook. Given the high performance of current models, even slight improvements are considered significant achievements in the field. This underscores the potential of

Tab. 2. Training results using standard Xception on KonIQ-10k images (Images), Xception trained on tabular data comprising image parameters (Parameters), and the outcomes using the Dual-Xception a approach (Images+Parameters).

	Images	**Parameters**	**Images+Parameters**
PLCC	0.91	0.74	0.92
SROCC	0.90	0.74	0.90
MSE	6.32	10.34	6.06
MSE (0-20)	11.17	26.42	12.01
MSE (20-40)	8.18	14.44	8.48
MSE (40-60)	6.64	9.28	6.66
MSE (60-80)	5.38	8.63	4.56
MSE (80-100)	5.91	15.08	6.10

convolutional approaches, like the dual-Xception architecture, in enhancing predictive accuracy, even in scenarios where traditional models already perform exceptionally well.

In Table 2 we analyzed three constructed models: the standard Xception architecture (termed “Images”), Xception trained solely on the tabular image attributes parameters such as brightness, contrast, sharpness, and bitrate (referred to as “Parameters”), and a combined architecture. This combined model integrates feature extractors from the first two models and produces a corrected result through a linear neural layer, employing the concatenation of tensors from the initial models.

The table shows that the Dual-Xception model excelled, particularly in assessing image quality within the 60 to 80 MOS range. This superior performance is likely due to the specific parameters within this range being highly indicative of image quality, as demonstrated by the improved results in the same quality range when using only the “Parameters” model. Notably, we did not employ the cross-sampling technique in this study, which might lead to skewed results from an imbalanced dataset.

In the graphs presented in Fig. 6 we observe the variation in the SROCC and PLCC metrics throughout the training epochs. It is evident that the Dual-Xception model initially registered lower values compared to the conventional approach. However, around the 8th to 10th epoch, the performance of the standard approach plateaued, whereas the more complex Dual-Xception architecture continued to improve as it progressed further in training.

3.2. Comparative Analysis of Models with Cross-Sampling

The images in the KonIQ-10k dataset are significantly unbalanced, impacting the final results and making model training less effective. This is demonstrated in Table 3, where the number of samples in the smallest and largest groups differs by nearly 30 times.

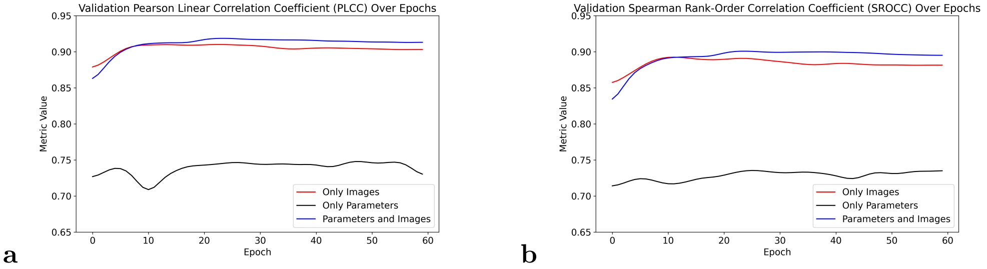


Fig. 6. Validating with two measures in function of epochs for the three types of models: (**a**) PLCC; (**b**) SROCC. Function color represents applied architecture (blue - Dual-Xception, red - Image Xception, black - Parameters Xception

To enhance the predictability of our models for lower-quality ranges, we employed and compared several cross-sampling methods. Additionally, we experimented with weighted MSE loss to focus the algorithm more on better predictions in the minority classes. In table below, we can observe the exact quantity of images in every range of MOS in the KonIQ-10k dataset. We note that in the class where MOS is between 0 and 20, there are fewer than 200 image instances, while the sum of images with MOS between 60 and 80 is 30 times larger.

In the preliminary phase of our research on under and over-sampling, we explored three distinct methodologies. The initial method involved constraining the number of elements in each class to correspond with the highest count found in the smallest class for the training dataset (method 1). Here, *class* denotes a grouping of instances within identical Mean Opinion Score (MOS) ranges. For every training iteration, instances were randomly selected from the categorized dataset, ensuring varied images for classes with a higher image count. In the second method, we limited the number of elements to the minimum count unless the image count exceeded 1 000. In such cases, we increased the image count for the specified range by double (method 2), leading to a twofold increase in data for the 40-60 and 60-80 ranges compared to other classes. The third approach

Tab. 3. Number of images samples in each of the five MOS ranges.

MOS range	**Samples**
(0, 20>	183
(20, 40>	1189
(40, 60>	3081
(60, 80>	5408
(8, 100>	212

Tab. 4. Comparison of the outcomes from four data sampling techniques.

	Without C-S	**Method 1**	**Method 2**	**Method 3**
PLCC	0.91±0.01	0.89±0.01	0.91±0.01	0.91±0.02
SROCC	0.88±0.01	0.87±0.01	0.89±0.01	0.89±0.02
MSE	6.40±0.09	7.30±0.12	6.30±0.15	7.35±0.20
MSE (0-20)	8.64±0.11	10.01±0.13	9.46±0.15	7.99±0.31
MSE (20-40)	8.61±0.11	8.57±0.16	8.41±0.14	8.14±0.20
MSE (40-60)	6.83±0.08	8.12±0.20	6.78±0.18	7.77±0.22
MSE (60-80)	5.42±0.05	6.37±0.24	5.24±0.18	6.89±0.25
MSE (80-100)	5.54±0.08	5.31±0.06	5.64±0.08	5.12±0.10
Ranges standard deviation	1.57±0.15	1.85±0.21	1.80±0.18	0.72±0.27

(method 3) employed a similar technique, but the number of elements in the largest classes was quadrupled relative to the smallest class. For three other ranges, we tripled the number of image instances, necessitating the repeated use of identical images. To mitigate overtraining risks, we applied image augmentation techniques, including rotations, flips, minor contrast modifications, and noise addition, while avoiding substantial alterations to maintain the integrity of the MOS ground truth.

During the validation phase, we modified our criteria for selecting the optimal model. Previously, the model with the lowest total Mean Squared Error (MSE) on the validation set was chosen. However, given our emphasis on minimizing the standard deviation across all five MOS ranges in this phase, we adopted a different approach. The optimal model was now determined based on the minimal sum of MSEs from each of the five MOS ranges.

It is important to highlight that the randomness in data allocation and manipulation, especially for the largest class, led to a relatively high standard deviation, thereby compromising the model's predictability. This study entailed ten training iterations per method, with notable disparities in results for edge cases.

The results for each model are presented in the table below. To ensure a more reliable comparison, we employed the same model selection criteria for validation across every method, including the primary architecture discussed in the previous chapter. This is why the results do not completely align with those presented in Tab. 1.

As depicted in the table above, due to the substantial reduction of data, cross-sampling can significantly decrease computing time without compromising performance quality. The data in the KonIQ-10k database are not evenly balanced, presenting a valuable opportunity to optimize our models even with relatively modest computing resources.

What is important to mention is that due to limitations of the dataset in every single training step, the time of the training procedure was reduced four times for method

Tab. 5. The table presents the computing time for a single training epoch for four selected models. From the left, we have the KonCept512 model provided by the KonIQ-10k authors, our dual-Xception model without cross-sampling implementation, and three cross-sampling methods explained above in this chapter.

	KonCept512	**Without C-S**	**Method 1**	**Method 2**	**Method 3**
Time	273 s	357 s	54 s	85 s	110 s
Parameters	56.0 M	40.8 M	40.8 M	40.8 M	40.8 M

number 3 and up to 7 times for the most limited method 1. This makes it a significantly more effective method, capable of enhancing computations by several times. It is a direct result of the dataset sampling procedure where we operate only on a small subset of the original dataset.

In Table 5 we illustrate the changes in computational time per epoch following the implementation of a given cross-sampling technique and compare this to the computing time of the current state of the art. The presented time refers to the duration of a single training epoch iteration. Similar to the authors of the state-of-the-art method, we trained our models over 60 epochs. The given time represents the average value from the entire training process. The training was conducted on an NVIDIA RTX 3070 GPU with 8GB of memory and a batch size of 4.

We can observe that the computational time for the original KonCept512 model is shorter than that of our dual-Xception architecture without the implementation of the cross-sampling method. This discrepancy may be due to different implementation approaches. The authors do not provide a PyTorch implementation of their network, and for this analysis, we opted to use the HuggingFace implementation, which might be slightly more optimized. However, we can still see that by employing our training procedure, which depends on the dataset's quantitative operations, we were able to outperform this state-of-the-art model by more than fivefold.

3.3. Comparison with other algorithms

We compared our results from the normal training procedure with our outcomes where we used a drastically limited dataset, following cross-sampling methods, against other solutions that were or still are seen as state-of-the-art in blind image assessment. Results are presented in Tab. 6.

We observed that the KonCept512 model remains the most effective. However, none of the authors of the papers included in our comparison provided details on accuracy changes across the entire spectrum of the MOS parameter. We have developed a very simple, easy-to-train, and extremely fast solution that guarantees prediction quality for all ranges of MOS values.

Tab. 6. Comparison of performance scores of several well-known and influential methods on the KonIQ-10k dataset. Results for each model, except our own one, are derived from [6].

Method	**SROCC**	**PLCC**
BIQI [26]	0.56	0.62
BLIINDS-II [27]	0.59	0.60
BRISQUE [28]	0.71	0.71
CNN [4]	0.57	0.59
DeepBIQ (VGG16) [5]	0.87	0.89
DeepBIQ (InceptionResNetV2) [5]	0.91	0.91
KonCept512 [6]	0.92	0.94
Ours without C-S	0.90	0.92
Ours with C-S	0.89	0.91

4. Conclusions

We introduced an efficient yet easy-to-train model that achieved near state-of-the-art performance with a dataset significantly smaller in size. Our method ensures excellent predictions across the entire MOS parameter range on the KonIQ-10k dataset. By using the cross-sampling method, we optimized single epoch processing time by up to 5 times without a significant decrease in result quality. We can assert that our solution is easier and faster for training and inference. The achieved results are also examined across the entire spectrum of MOS image values, making our model unique.

For future work, we could delve into more sophisticated research regarding optimal under/over-sampling techniques. Our proposed methodology, while experimental, suggests that different dataset partitioning might yield even better results. Another aspect worth investigating is the application of weighted training loss that varies according to the ground truth parameter value. However, this approach may not significantly reduce training time, especially if we continue to utilize sampling methods.

We also propose to intensify research on how to optimize feature extractors that operate on numerical data. We need to investigate alternatives to the Xception architecture that may offer comparable results.

References

[1] N. Burningham, Z. Pizlo, and J. P. Allebach. Image Quality Metrics. In Hornak, Joseph P. (ed.). *Encyclopedia of imaging science and technology*, Wiley, New York, 2002. doi:10.1002/0471443395.img038.

[2] I. H. AL-Qinani. A Review Paper on Image Quality Assessment Techniques. *International Journal of Modern Trends in Engineering & Research*, 6(8):1–7, 2019. doi:10.21884/IJMTER.2019.6023.SVDQQ.

[3] S. Bosse, D. Maniry, T. Wiegand, and W. Samek. A deep neural network for image quality assessment. *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3773–3777. doi:10.1109/ICIP.2016.7533065.

[4] L. Kang, P. Ye, Y. Li, and D. Doermann, Convolutional neural networks for no-reference image quality assessment. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1733–1740, 2014. doi:10.1109/CVPR.2014.224.

[5] S. Bianco, L. Celona, P. Napoletano, and R. Schettini. On the use of deep learning for blind image quality assessment. *Signal, Image and Video Processing*, 12(2)355–362, 2018. doi:10.1007/s11760-017-1166-8.

[6] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe. KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–40561, 2020. doi:10.1109/tip.2020.2967829.

[7] V. R. Dendi, C. Dev, N. Kothari, and S. S. Channappayya. Generating image distortion maps using convolutional autoencoders with application to no reference image quality assessment. *IEEE Signal Processing Letters*, 26(1):89–93, 2018. doi::10.1109/LSP.2018.2879518.

[8] W. Zhang, K. Ma, G. Zhai and and X. Yang. Learning to blindly assess image quality in the laboratory and wild. *Proc. 2020 IEEE International Conference on Image Processing (ICIP)*, pp. 111–115, 2020. doi:10.1109/ICIP40778.2020.9191278.

[9] D. Varga, T. Szirányi, and D. Saupe. DeepRN: A content preserving deep architecture for blind image quality assessment. *Proc. 2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2018. doi:10.1109/ICME.2018.8486528.

[10] D. Ghadiyaram and A. C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387, 2016. doi:10.1109/TIP.2015.2500021.

[11] F. Chollet, Xception: Deep learning with depthwise separable convolutions. *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017. doi:10.1109/CVPR.2017.195.

[12] F. Chollet, Xception: Deep learning with depthwise separable convolutions. *arXiv*, preprint arXiv:1610.02357, 2017. doi:10.48550/arXiv.1610.02357.

[13] R. Mohammed, J. Rawashdeh and M. Abdullah. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. *Proc. 2020 11th International Conference on Information and Communication Systems (ICICS)*, pp. 243–248, 2020. doi:10.1109/ICICS49469.2020.239556.

[14] D. Ghadiyaram and A. C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1)372–387, 2016. doi:10.1109/TIP.2015.2500021.

[15] F. Gao, J. Yu, S. Zhu, Q. Huang, and Q. Tian. Blind image quality prediction by exploiting multi-level deep representations. *Pattern Recognition*, 81:432–442, 2018. doi:10.1016/j.patcog.2018.04.016.

[16] W.-H. Kim, et al. Pixel-by-pixel Mean Opinion Score (pMOS) for no-reference image quality assessment. *ArXiv*, preprint arXiv:2206.06541, 2022. doi:10.48550/arXiv.2206.06541.

[17] M. Prabhushankar, D. Temel, and G. AlRegib. MS-UNIQUE: Multi-model and sharpness-weighted unsupervised image quality estimation. *Electronic Imaging*, 29(12):30–35:art00006, 2017. doi:10.2352/ISSN.2470-1173.2017.12.IQSP-223.

[18] X. Liu, J. van de Weijer, and A. D. Bagdanov. RankIQA: Learning from rankings for no-reference image quality assessment. *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1040–1049, 2017. doi:10.1109/ICCV.2017.118.

[19] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo. End-to-end blind image quality assessment using deep neural networks. *IEEE Transactions on Image Processing*, 27(3):1202–1213, 2018. doi:10.1109/TIP.2017.2774045.

[20] H. Talebi and P. Milanfar, NIMA: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018. doi:10.1109/TIP.2018.2831899.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. *Proc. 31st AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, pp. 4278–4284, 2017. doi:10.1609/aaai.v31i1.11231.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi:10.1109/CVPR.2016.90.

[23] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018. doi:10.1109/CVPR.2018.00745.

[24] J. Deng, W. Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi:10.1109/cvprw.2009.5206848.

[25] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. *Proc. 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 7–9, 2015. doi:https://arxiv.org/abs/1412.6980.

[26] A. Moorthy and A. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010. doi:10.1109/LSP.2010.2043888.

[27] M. A. Saad, A. C. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Transactions on Image Processing*, 21(8):3339–3352, 2012. doi:10.1109/TIP.2012.2191563.

[28] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695-4708, 2012. doi:10.1109/TIP.2012.2214050.

Semantic Segmentation of Diseases in Mushrooms using Enhanced Random Forest

Rakesh Kumar Yacharam[1,*], Dr. V. Chandra Sekhar[2]

[1]*ECE Department, G. Narayanamma Institute of Technology and Science (For Women), Hyderabad, India*

[2]*ECE Department, Matrusri Engineering College, Hyderabad, India*

[*]*Corresponding author: Rakesh Kumar Yacharam (rakeshyacharam@gmail.com)*

Abstract. Mushrooms are a rich source of antioxidants and nutritional values. Edible mushrooms, however, are susceptible to various diseases such as dry bubble, wet bubble, cobweb, bacterial blotches, and mites. Farmers face significant production losses due to these diseases affecting mushrooms. The manual detection of these diseases relies on expertise, knowledge of diseases, and human effort. Therefore, there is a need for computer-aided methods, which serve as optimal substitutes for detecting and segmenting diseases. In this paper, we propose a semantic segmentation approach based on the Random Forest machine learning technique for the detection and segmentation of mushroom diseases. Our focus lies in extracting a combination of different features, including Gabor, Bouda, Kayyali, Gaussian, Canny edge, Roberts, Sobel, Scharr, Prewitt, Median, and Variance. We employ constant mean-variance thresholding and the Pearson correlation coefficient to extract significant features, aiming to enhance computational speed and reduce complexity in training the Random Forest classifier. Our results indicate that semantic segmentation based on Random Forest outperforms other methods such as Support Vector Machine (SVM), Naïve Bayes, K-means, and Region of Interest in terms of accuracy. Additionally, it exhibits superior precision, recall, and F1 score compared to SVM. It is worth noting that deep learning-based semantic segmentation methods were not considered due to the limited availability of diseased mushroom images.

Key words: mushroom diseases, semantic segmentation, computer aided, Machine Learning, significant feature extraction, Random Forest classifier.

1. Introduction

To enhance mushroom yields, farmers frequently invest in controlled environment cultivation rooms. Despite these earnest endeavours, the persistence of diseases in substrates and mushrooms remains a challenge [1,2]. Even with precautionary measures in place, a lack of knowledge, crop management skills, and occasional human errors can contribute to the onset of diseases affecting both mushrooms and their substrate bags. Among the most prevalent mushroom diseases are cobweb, wet bubble, bacterial blotches, dry bubble, and mites. These diseases not only affect individual mushrooms but also have the potential to spread from one mushroom to another. Over time, they can proliferate throughout the cultivation bag, extending to other bags in the room, resulting in significant losses for the farmer. Timely detection and proactive measures are paramount for mitigating the impact of these diseases. The manual identification of mushroom diseases poses a formidable challenge, necessitating a profound expertise in disease recognition

Machine GRAPHICS & VISION 32(2):129–146, 2023. DOI: 10.22630/MGV.2023.32.2.7 .

and the capacity to implement effective corrective actions. Farmers frequently find themselves compelled to seek guidance from scientific experts, and any oversight in this process can result in significant losses. Moreover, the early-stage detection of diseases with the naked eye proves to be a daunting task, underscoring the imperative for computer-aided detection methods. Traditional methods for identifying diseases in mushrooms often depend on non-computer-aided techniques that utilize chemical and biological approaches in scientific laboratories. Nevertheless, these methodologies are often time-consuming. In response to this challenge, we introduce an innovative computer-aided methodology that harnesses the power of machine learning, particularly an Enhanced Random Forest technique. Our proposed method integrates an optimized selection of features to reduce complexity and elevate overall effectiveness in implementation.

The subsequent sections of this article are structured as follows: Section 2 presents a comprehensive literature review that encompasses mushroom diseases, various disease detection methods, and the application of machine learning techniques in disease segmentation. In Section 3, the design methodology is detailed, incorporating established segmentation methods such as K-Means clustering, Region of Interest extraction, Colour Threshold, Support Vector Machine (SVM), and the proposed Enhanced Random Forest method, with a specific focus on parameter optimization techniques. Section 4 provides a presentation of results and a discussion on mushroom disease segmentation, covering both existing standard methods and the newly proposed approach, followed by a comparative analysis. Finally, the conclusions of the work are summarized in Section 5.

2. Related work and motivation

The literature review outlines various methods and approaches employed in the detection and segmentation of mushroom diseases, emphasizing both manual and computer-aided techniques and also outlines some of the different disease detection methods. The proposed research aims to address the existing gap in automated mushroom disease segmentation by introducing a semantic segmentation method based on Enhanced Random Forest. This method is compared against established state-of-the-art techniques such as SVM, Naïve Bayes, K-means, ROI, and Colour Threshold methods. The review begins by discussing manual segmentation methods involving chemical and biological processes, such as Biological Material-RNA [3] analysis and Isolation of dsRNA [4] coupled with electron microscopy. These methods are noted for their time-consuming nature. Computer-aided methods, as outlined in [5] and [6], leverage techniques like Naïve Bayes, Sequential Minimal Optimization, and Ripple Down Rules (RIDOR) for classifying 16 images of diseased mushrooms. Human intervention is required in the conversion of these images to a suitable file format for classification. Additionally, an automated pixel-to-pixel image processing-based software, proposed by [7] and [8], is designed to inspect white button mushroom crops and detect signs of illness and pests.

For diagnosing mushroom diseases, [9] and [10] have developed a rule-based expert system. This system necessitates text-based responses from farmers as inputs to detect diseases. While these methods utilize computer-aided techniques, it is important to emphasize that human intervention is still required to detect diseased mushrooms in a given image. The various image segmentation approaches [11] in general are threshold, edge and region based applied for disease segmentation in different fields like agriculture or medical diagnosis. In [12] the region growing method is used to segment disease spots on leaf and eliminate background by interactively selecting growing seeds in the ACCF map. Then morphological operation is used on the region growing method results.

In the identification of diseases in Arecanut, a two-step procedure is employed, incorporating K-Means clustering and the Otsu approach, as outlined in [13]. The colour-based K-Means clustering method is applied during pre-processing to effectively isolate Arecanut from the background. Subsequently, the Otsu thresholding method is employed to transform RGB images into monochrome images, facilitating the detection of diseases. The diseased area of the Arecanut is then accurately determined using the connected components method.

In addition to conventional state-of-the-art methods, numerous machine learning approaches have been proposed for disease classification in plants and fruits. For instance, in the segmentation and classification of plant leaf diseased areas, image thresholding, K-Means clustering, and Neural networks are utilized [14]. In the case of apple fruits, the identification of infected areas involves employing a Global threshold for segmentation. Further classification of the infected areas on apple fruits is accomplished using a Machine Learning Technique, specifically the multi-class Support Vector Machine (SVM) [15].

The random forest, as introduced in [17], operates on an ensemble of trees, each dependent on the values of a random vector. This vector is sampled independently, yet uniformly, across all trees within the forest. This strategic approach embodies a diversified learning mechanism, cultivating robustness against noise and enhancing adaptability in predictive modelling. T. K. Ho, in [18], proposed tree-based classifiers to arbitrarily increase capacity, thereby improving accuracy for both training and testing data. The approach involves developing multiple trees in randomly selected subspaces, particularly effective for handwritten digits.

Achieving accurate and effective semantic segmentation poses a challenge due to the necessity of classifying each pixel, a computationally demanding task. In addressing this challenge, [19] presents a random forest-based semantic segmentation algorithm that achieves precise and effective pixel-wise classification of body poses. The Random Forest (RF) approach [20] for pixel-level segmentation in images contributes in three significant ways. First, it demonstrates the applicability of Nearest Neighbour Matching and Texton Class Histograms to the Random Forest structure. Second, it underscores the importance of discriminative learning and geographic context for Random Forest, emphasizing how the architecture can enhance classifier performance. Lastly, segmentation performance is elevated by utilizing Random Forest to integrate multiple features, including colour, textons, HOG features, and filter banks. A Flexible Random Forest model [21] has been developed to address a diverse and extensive range of video and image tasks, presenting a discussion that combines both theoretical insights and practical implementations.

In addition to employing machine learning techniques for segmenting diseases across various fields, deep learning methods are also utilized in disease segmentation. Several convolutional neural network-based techniques presented in [22] aim to enhance the accuracy of semantic segmentation. Deep learning approaches excel over other methods, partially due to their ability to learn intricate representations, coupled with hierarchical structures and non-linear activations. Notable deep learning-based semantic segmentation models such as DeepLab, CCNet, SegNet, ICNet, and RefineNet [23,24,25,26,27,28] have been developed for segmenting images of various types, including high-resolution and real-time image segmentation.

From the existing literature, it follows evidently that there is limited research on mushroom disease segmentation, and the current studies in this domain require human intervention rather than an automated approach. In response to this gap, a semantic segmentation method is proposed based on Enhanced Random Forest (ERF) for mushroom disease segmentation. This method is compared against standard state-of-the-art techniques such as SVM, Naïve Bayes, K-means, ROI, and colour threshold methods. Due to the scarcity of diseased mushroom images from diverse sources, deep learning-based semantic segmentation methods like DeepLab, SigNet, ICNet, etc., were not considered in the comparative analysis.

3. Methodology and methods

3.1. Existing standard methods of segmentation

Image processing is a multidisciplinary field encompassing the manipulation, analysis, and interpretation of visual information extracted from considered images. Within this domain, diverse approaches are employed to extract insightful information and enhance image quality for various applications. Noteworthy methods in digital image processing include K-Means clustering, Region of Interest (ROI) extraction, the colour threshold method, and the application of Naïve Bayes, as well as SVM classification techniques, which are subject to comparative analysis. These techniques find application across various domains such as computer vision, remote sensing, medical imaging, and the detection of diseases in agricultural crops.

3.2. Random Forest-based semantic segmentation

In the proposed approach for segmenting mushroom diseases, we employ Random Forest-based semantic segmentation. Throughout this segmentation process, the initial preprocessing of image data is conducted to enhance both its quality and relevance. Following this, feature estimation procedures are implemented to capture pertinent image attributes. Subsequently, significant features are extracted to facilitate the discrimination of key regions. To attain the final segmentation results, a pixel-wise Random Forest classifier is applied.

3.2.1. Data set

Firstly, the collection process involves gathering images of both diseased and healthy mushrooms for training purposes. Ground truth images are subsequently generated with the input of experts, who provide valuable insights into the distinctive characteristics of mushroom diseases. Subsequently, the images undergo a pre-processing stage before feature extraction. Each image considered for training is represented by the notation I_{pxq}, where 'p' and 'q' denote the row and column of the matrix 'I.' In this matrix, every element, designated as I_i, corresponds to a pixel with an intensity value.

3.2.2. Image pre-processing

The original images are in the RGB colour model, but they have been converted to a single grayscale representation using the green channel for ease of processing. This conversion is advantageous due to the contrast property, which is particularly beneficial over the red (R) and blue (B) channels. The green channel response exhibits a lower contrast, while the blue channel demonstrates a less dynamic range.

3.2.3. Feature estimation

The preprocessing of training images involves the consideration of both diseased and non-diseased classes for feature estimation, aiming to accurately segment the diseased portions. Various features are computed from the training images, including Canny and Sobel edge detectors, as well as Roberts, Scharr, Prewitt, Bouda, and Kayyali edge detectors, which are derived from the Sobel operator. Additionally, Gaussian features with σ values of 3 and 7, Median with a size of 3, and Gabor filters with a kernel size of 9×9 and variations in orientation (ϕ), scaling (σ), and wavelength (γ) are employed. These features, each assigned corresponding weights, play a pivotal role in node splitting and classification decision within the random forest framework. The computation of features involves the mathematical operation of convolution between the input image and the filter, as expressed as follows

$$Y(i,j) = \sum_{k=1}^{m}\sum_{l=1}^{n} I(i+k-1, j+l-1)K(k,l)\,, \tag{1}$$

where $i = \{1, 2, \ldots,\ M-m+1\}$ and $j = \{1, 2, \ldots,\ N-n+1\}$, $M \times N$ is the size of input image, m, n is the size of the filter (kernel). The kernel or filter coefficient of Roberts, 3×3 Sobel, 3×3 Prewitt, 3×3 Kayyali [29], Bouda [30] are as follows

$$K_{\text{Roberts}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \ K_{\text{Sobel}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \ K_{\text{Prewitt}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix},$$

$$K_{\text{Kayyali}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, K_{\text{Bouda}} = \begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 2\sqrt{2} & 0 & -2\sqrt{2} \\ \sqrt{2} & 0 & -\sqrt{2} \end{bmatrix}, K_{\text{Scharr}} = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}.$$

The Gaussian filter kernel of size $(2k+1) \times (2k+1)$ convolved with input image is given by

$$K_{ij} = \frac{1}{2\pi\sigma^2} e^{\frac{-(i-(k+1)^2+j-(k+1)^2)}{(2\sigma^2)}} \quad 1 \leq i, j \leq 2k+1\,. \tag{2}$$

Canny edge filter works with derivatives in direction of the edge textures on Gaussian filtered output [31]. Gabor filter function [31] is represented by the product of Gaussian function and exponential function given as

$$g_{\theta,\gamma,\lambda,\phi,\sigma}(x,y) = exp\left[\frac{-(x^2+\gamma^2 y^2)}{(2\sigma^2)}\right] exp\left[\frac{i2\pi x}{\lambda\sigma+\phi}\right], \tag{3}$$

where θ gives rotation of Gabor envelope, λ Regulates the width of the Gabor function strips, γ regulates the Gabor filter height, σ regulates the Gabor filter overall size, ϕ is phase offset of sinusoid and is equal to zero in the considered case.

3.2.4. Significant features extraction

The features assessed in Section 3.2.3 lack significance. Extracting significant features proves beneficial in alleviating the computational burden during both training and testing phases, leading to time savings. To achieve this, we advocate for feature extraction through constant mean-variance thresholding. This approach helps eliminate features with low variance and zero-valued variables. Additionally, employing the Pearson correlation coefficient aids in identifying highly correlated features, which may result in feature duplication. By avoiding such redundancies, we aim to enhance computational efficiency.

Constant mean-variance thresholding

A non-constant feature extraction method employs mean-variance thresholding. This technique aids in discerning features that are either constant, approximately constant, or zero-valued. Such features lack significance in determining the pixel's class within a given image.

Let V represent a variable (one of the features among the total set of features). It becomes a significant feature if it satisfies the following condition

$$\begin{array}{lll} \text{if} \quad v_i \leq (m-2\sigma^2) & v_i & \text{is not significant} \\ \text{otherwise} & v_i & \text{is significant}\,, \end{array} \tag{4}$$

where v_i is feature considered at that instant, m is mean and σ^2 is variance of the feature set, respectively.

Pearson correlation coefficient

The Pearson correlation coefficient is a widely utilized statistical measure that gauges the strength of the relationship between two variables. This measure is particularly prevalent in the context of linear regression analysis. When employed for feature reduction purposes, the Pearson correlation coefficient (5), denoted as p_r, is calculated for two features, f_1 and f_2.

$$p_r = \frac{N(\sum f_1 * f_2) - (\sum f_1)(\sum f_2)}{\sqrt{[N \sum f_1^2 - (\sum f_1)^2][N \sum f_2^2 - (\sum f_2)^2]}}\,, \tag{5}$$

where f_1 is first feature set and f_2 is adjacent feature set under consideration. N is number of elements in the feature set. Higher the p_r value between f_1 and f_2, higher the correlation, and it indicates duplication of these features, which is not effective for training.

3.2.5. Random Forest pixel-wise classifier

It constitutes an ensemble of decision trees, trained through the bagging method, typically with the maximum number of samples set to the size of the training set. This algorithm yields greater tree diversity, introducing a higher bias for lower variance and consequently leading to a superior model. In a Random Forest, during the splitting process at each node, a random subset of features is considered to grow the tree, searching for the best feature among this random subset. Moreover, it is possible to enhance the randomness of trees by also employing random thresholds for each feature, instead of searching for the best possible threshold as done in a traditional decision tree.

Let the dataset be denoted as $D = \{f_1, f_2, \ldots, f_n\}$, where each point represents a feature. We randomly select a subset of features (pixels) from this dataset, with the pixels, in turn, corresponding to the class labels originally present in the dataset. A feature is represented by f_i [32] and is given by

$$f_i(p, P, V) = \sum_{i=1}^{n} w_i P_{(p+u_i/V_{p,h})}\,, \tag{6}$$

where p is a pixel under consideration, P is the input image, w is weight, V is depth map, h is channel index of P, u is offset parameter vector.

Bootstrap data sets $D_i | i = 1, 2, \ldots, B$ are generated from the data set D, by randomly selecting the significant features discussed in Section 3.2.4, where repeating is allowed. Each randomly chosen Bootstrap set D_i helps in constructing Decision Tree T_i. Root node of the Tree can be any one of the features in D_i. At each decision tree nodes splitting

S_j is done and and the best split is chosen as that which has the highest information gain:

$$S_j = S_j^L \cup S_j^R \,, \tag{7}$$

where S_j^L is left split, S_j^R is right split. Information gain in terms of entropy for the node split is given by (8), where entropy at the node is given by (9):

$$I_j = H(s_j) - \sum_{i \in (L,R)} \frac{S_j^i}{s_j} H(S_j^i) \,, \tag{8}$$

$$H(s) = \sum_{i}^{C} -P_i(s) \log_2 P_i(s) \,, \tag{9}$$

where S_j is target population before the split, $H(s)$ is entropy of s, $H(S_j^i)$ is entropy of S_j^i, S_j^i are data points falling into right or left subtree based on $i \in (L, R)$. P_i is the probability of a class i in the data s. The conditional probability from each tree T_i for a data point p being a class c considered at each node is given by

$$PT_i(c/p, P, V) < Q \quad \text{or} \quad PT_i(c/p, P, V) > Q \,, \tag{10}$$

where Q is the threshold. Then, the majority voting out of the total decision trees is considered to decide the p being class c.

Algorithmic steps

Mushroom images, both diseased and non-diseased, are initially divided into training and test sets. Subsequently, mask images are generated for the training dataset. The algorithmic flow can be outlined in the following steps:

Step 1: Consider mushroom images

Step 2: Image Preprocessing – Convert the image from RGB to grayscale and resize it to a standard size of 128 × 128.

Step 3: Feature Extraction – Extract features such as Gabor features, original image pixels, Canny edge, Roberts, Sobel, Scharr, Prewitt, Bouda, Kayyali, Gaussian ($\sigma = 3$ and $\sigma = 7$), Median ($\sigma = 3$), and variance with a size of 3 from the mushroom images.

Step 4: Extract significant features using a constant mean-variance threshold and Pearson correlation for training.

Step 5: Image Pre-processing for Labelled Mask Images – Convert the images to grayscale and resize them to a standard size of 128 × 128.

Step 6: Split the Data into Train and Test sets – The data is split with a train size of 70% and a test size of 30%. The random state parameter is set to 20.

Step 7: Build a Random Forest (RF) Classifier model. The RF model is trained using significant features, and 20 estimators are employed, representing twenty decision trees with a depth of 10.

Step 8: Test the model by predicting on the test data, and calculate the accuracy.

4. Results and discussions on mushroom disease segmentation

The proposed method has been employed on a dataset comprising mushroom diseases to effectively segment diseases from input mushroom images. Non-diseased images, along with diseases such as Dry Bubble, Wet Bubble, Cobweb, Bacterial Blotch, and mites on mushroom images, were selected for experimentation using the Random Forest semantic segmentation algorithm. The proposed approach, founded on a random forest classifier combined with a robust feature extraction process, outperforms the SVM classifier, Naïve Bayes, and other methods, including the K-Means clustering method, Region of Interest, and Colour Threshold method.

Approximately 250 mushroom images, encompassing both diseased and non-diseased instances, were gathered from diverse organizations and a popular website (i.e., from [33,34,35]). These images were subsequently divided into training (70%) and test (30%) datasets. Ground truth images were generated for the training set to characterize the diseases present in mushrooms and distinguish the background parts of the images.

Features from the Gabor filter are extracted from the mushroom image using a kernel size of 9x9. The parameters for extraction include theta values ranging from 0 to 45 degrees, γ values of 0.05 and 0.5, σ with values of 1 and 3 and λ values of 0, 45, 90, and 135. In addition to Gabor features, other features extracted include original image pixels, and edges by Canny, Roberts, Sobel, Scharr, Prewitt, Bouda, and Kayyali (extracted from the Sobel operator), Gaussian with $\sigma = 3$ and 7, median and variance in the windows of size 3×3.

The total number of features amounts to 43, with 32 originating from the Gabor filter and the remaining 11 from other filters. Since some of these features are not significant, a process involving Pearson correlation and a constant mean-variance threshold is applied to extract meaningful features. This process eliminates constant and highly correlated features, thereby enhancing the performance of the Random Forest semantic segmentation results. Additionally, it contributes to the reduction of computation time and complexity.

The Random Forest semantic segmentation model is constructed using 20 decision trees with a depth of 10. The results of the Random Forest simulation on the mushroom diseased image dataset are compared with the Naïve Bayes method, basic standard methods such as Region of Interest, Colour Threshold method, unsupervised K-means clustering algorithm, and other supervised machine learning techniques. Support Vector Machine with a Radial Basis Function kernel is employed in the comparison process.

4.1. Subjective analysis

The experiment produced a series of images, showcasing both the original input images and the resulting images, as illustrated in Figure 1. The first row denotes the disease names, the second row displays the original images of diseased mushrooms, and the

Tab. 1. Summary on number of diseases segmented correctly by various segmentation methods.

Segmentation method	Cobweb Diseases (total 65)	Dry bubble Diseases (total 70)	Wet bubble Diseases (total 41)	Mites Diseases (total 24)	Bacterial blotch Diseases (total 50)
Enhanced Random Forest	64	69	40	22	49
SVM	59	69	39	21	43
Naïve Bayes	58	68	38	22	45
K-means	61	64	36	22	47
Region of Interest	58	68	35	22	47
Colour Threshold	59	65	36	22	43

third row exhibits the resultant images generated by the proposed Enhanced Random Forest. Following suit, the fourth row presents Naïve Bayes' resultant images, the fifth row displays SVM's resultant images, the sixth row exhibits K-means' resultant images, and the seventh row illustrates the ROI resultant images. Additionally, the seventh row showcases resultant images corresponding to the Colour Threshold method [16], all aligned with the respective diseased images are shown in the 8th row.

Upon subjective analysis, it is discerned that the Enhanced Random Forest excels in accurately segmenting disease areas for cobweb, dry bubble, wet bubble, and bacterial blotch diseases, outperforming Naïve Bayes, SVM, K-means, ROI, and Colour Threshold techniques. However, it exhibits suboptimal performance in some mite images, occasionally extracting background elements alongside the diseased portions.

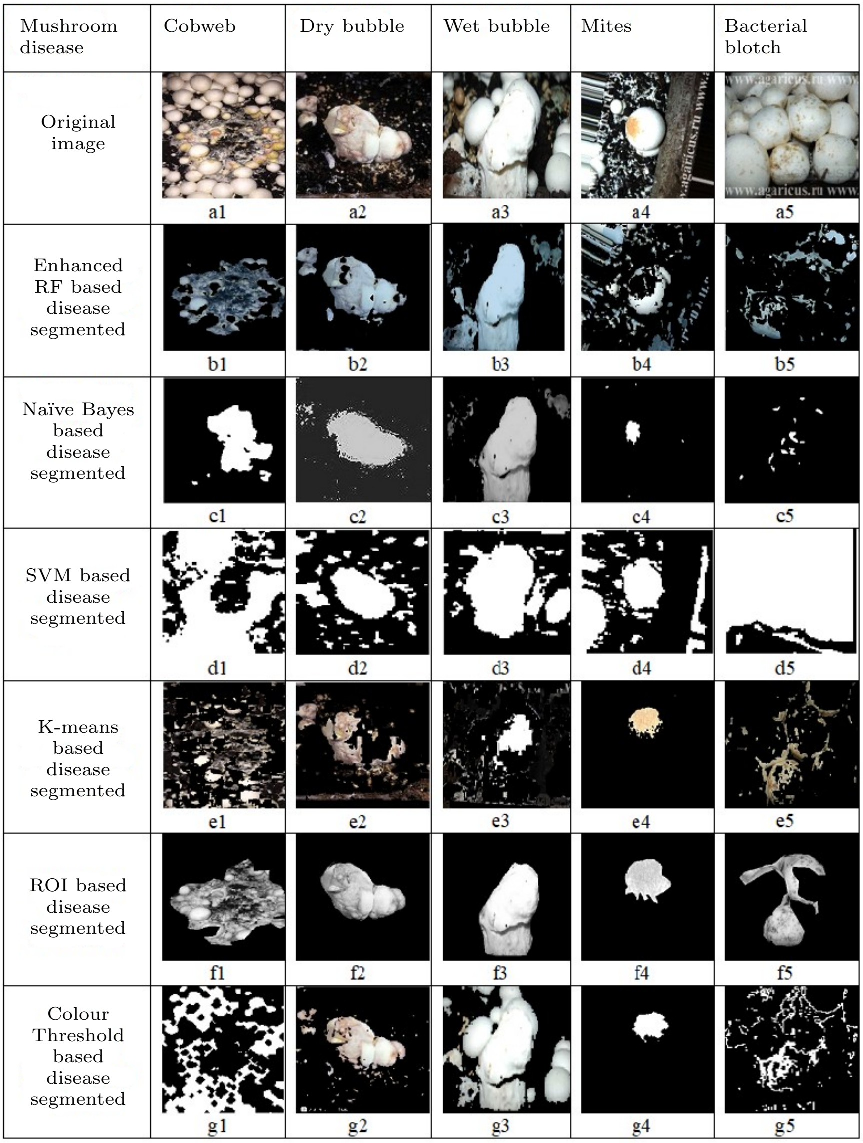
4.2. Objective analysis

To evaluate the performance of the proposed Enhanced Random Forest, a comparative analysis was conducted with several other methods, including Naïve Bayes, Support Vector Machine, K-means, ROI, and colour threshold. The dataset used for this assessment comprised 250 images depicting various mushroom diseases. Table 1 presents the statistics for the number of correctly segmented mushroom disease images out of the total 250, categorized as cobweb (65), dry bubbles (70), wet bubbles (41), mites (24), and bacterial blotches (50). The graphs illustrating the segmentation results obtained by different methods across various mushroom disease categories are shown in Figure 2.

Table 2 shows accuracies derived from statistics from Table 1, calculated as

$$\mathrm{ACC} = N_{iC}/N_i\,, \tag{11}$$

where ACC – accuracy, N_{iC} – number of images correctly identified, and N_i – total number of images. Notably, the Enhanced Random Forest outperforms Naïve Bayes,



Mushroom disease	Cobweb	Dry bubble	Wet bubble	Mites	Bacterial blotch
Original image	a1	a2	a3	a4	a5
Enhanced RF based disease segmented	b1	b2	b3	b4	b5
Naïve Bayes based disease segmented	c1	c2	c3	c4	c5
SVM based disease segmented	d1	d2	d3	d4	d5
K-means based disease segmented	e1	e2	e3	e4	e5
ROI based disease segmented	f1	f2	f3	f4	f5
Colour Threshold based disease segmented	g1	g2	g3	g4	g5

Fig. 1. Comparative results of Enhanced RF (ERF), Naïve Bayes, SVM, K-Means, ROI, Colour Threshold methods for different diseases. 1st row: a1-a5 – sample original input images; 2nd row: b1-b5 – ERF methods results; 3rd row: c1-c5 – RF methods results; 4th row: d1-d5 – SVM method results; 5th row: e1-e5 – K-Means method results; 6th ro: f1-f5 – ROI method results; and 7th row: g1-g5 – Colour Threshold method results for the corresponding input diseased images a1-a5.

Tab. 2. Accuracy of ERF, RF and SVM classifier based semantic segmentation and other standard segmentation methods.

Segmentation method	Accuracy
Enhanced Random Forest	0.98
SVM	0.93
Naïve Bayes	0.92
K-means	0.92
Region of Interest	0.90
Colour Threshold	0.92

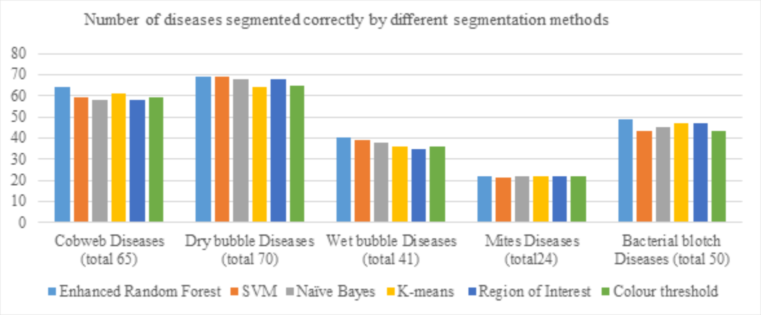


Fig. 2. Bar graph of number of diseases segmented correctly by different segmentation methods.

achieving the highest accuracy of 98%. This suggests that the Enhanced Random Forest, utilizing features selected through constant mean-variance thresholding and Pearson correlation coefficient, is more effective in mushroom disease segmentation tasks compared to SVM, widely-used supervised machine learning technique which achieved an accuracy of 93%. The second-highest accuracy of SVM indicates its correct classification of 93% pixels or regions in mushroom disease images. Naïve Bayes also a supervised machine learning technique used for disease segmentation, demonstrated an accuracy of 92.4%, performing well but slightly less accurately than SVM and the Enhanced Random Forest in segmenting mushroom diseases. K-means is an unsupervised machine learning clustering algorithm, achieved a respectable accuracy of 92% and almost equals Naïve Bayes, showcasing its effectiveness in a classification context, albeit slightly behind other machine learning techniques. Region of Interest (ROI), a conventional segmentation technique for identifying specific areas in an image, achieved an accuracy of 90%, indicating its lesser effectiveness compared to other methods in mushroom disease segmentation tasks. Colour Thresholding, a basic method relying on colour information for image object segmentation, attained an accuracy of 92%, aligning with K-means, Naïve Bayes and slightly below other machine learning methods.

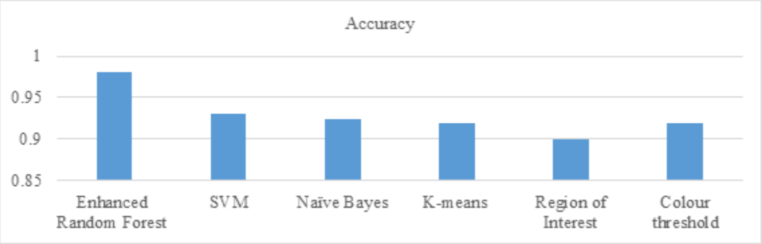


Fig. 3. Bar graph of accuracies attained by the tested methods.

The accuracy results of the proposed enhanced Random Forest and other standard segmentation methods are shown as a bar graph in Figure 3.

The metrics of precision, recall, and F1 score are taken into account, offering a comprehensive perspective on the effectiveness of the proposed semantic segmentation methods in identifying regions affected by mushroom diseases in a given image. Specifically, the metric of specificity evaluates the model's capability to accurately predict non-disease regions. It's worth noting that, in the context of semantic segmentation for disease detection, True Negatives (TN) are of lesser relevance, leading to the infrequent use of specificity as a metric in this particular domain.

$$\mathrm{PRE} = TP/(TP+FP)\,, \tag{12}$$

$$\mathrm{REC} = TP/(TP+FN)\,, \tag{13}$$

$$\mathrm{F1} = (2*\mathrm{PRE}*\mathrm{REC})/(\mathrm{PRE}+\mathrm{REC})\,, \tag{14}$$

where PRE – precision, REC – recall, F1 – F1 score.

Table 3 presents the confusion matrix, while Table 4 displays the performance metrics of pixel-based classifiers for ERF, SVM, and NB. Additionally, Figure 4 illustrates the corresponding bar graph. The objective analysis of ERF, SVM, and NB pixel-level classifiers employs performance metrics such as Precision, Recall, and F1 score, as outlined in equations (12), (13) and (14), respectively. These metrics are computed based on the False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN) values derived from the confusion matrix. TP represents pixels correctly identified by the semantic segmentation model as part of the actual mushroom diseased area in a given image, while FP signifies pixels identified by the model as part of the disease region but not belonging to the actual diseased area. FN corresponds to pixels within the mushroom disease region that are incorrectly classified as not belonging to the disease by the semantic segmentation model, and TN represents pixels correctly identified as not belonging to the mushroom disease region when they actually do not belong. It is noteworthy that semantic segmentation models typically emphasize the identification of positive mushroom diseased regions more than the negative (healthy) regions, rendering True Positive less relevant in the context of semantic segmentation.

Tab. 3. Confusion matrix of Enhanced Random Forest – ERF, Naïve Bayes – NB, and Support Vector Machine – SVM.

Classes	Correctly Segmented labels			Incorrectly Segmented labels		
class	ERF	SVM	NB	ERF	SVM	NB
1 (diseased)	5246	5172	5031	98	178	312
2 (undiseased)	1039	1042	1061	171	162	150

Tab. 4. Classifier performance metrics at pixels level: ERF – Enhanced Random Forest, NB – Naïve Bayes, SVM – Support Vector Machine methods.

Class	Precision			Recall			F1-Score		
class	ERF	SVM	NB	ERF	SVM	NB	ERF	SVM	NB
1 (diseased)	0.97	0.95	0.95	0.99	0.97	0.98	0.98	0.97	0.97
2 (undiseased)	0.91	0.85	0.88	0.87	0.87	0.79	0.89	0.85	0.83

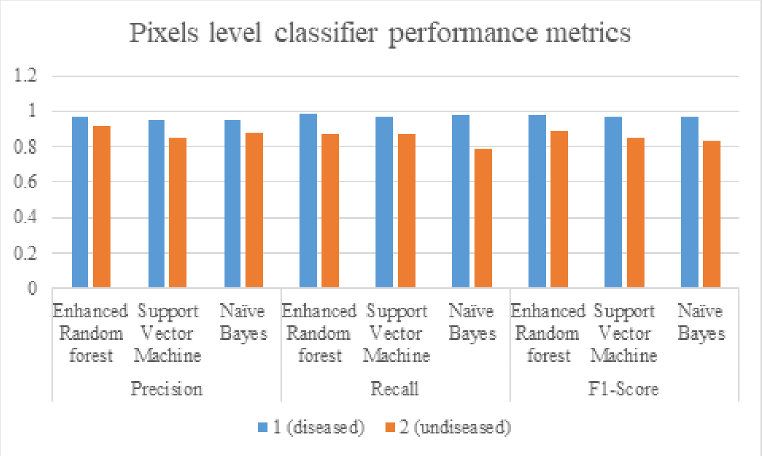


Fig. 4. Bar graph of pixels level classifier performance metrics of ERF – Enhanced Random Forest, NB – Naïve Bayes, SVM – Support Vector Machine.

5. Conclusion

In mushroom cultivation, the segmentation of mushroom diseases stands out as a crucial task, playing a pivotal role in estimating disease severity, suggesting necessary preventive actions, and mitigating potential losses for farmers. This work presents a novel approach to automatically perform semantic segmentation specifically for identifying various mushroom diseases, employing the Enhanced Random Forest method. Despite the existence of advanced deep learning methods for semantic segmentation in image object recognition, the decision to utilize machine learning methods in this context stems from the limited availability of data. This proposed method is further strengthened by the integration of substantial feature extraction, achieved through constant mean-variance thresholding and the Pearson correlation coefficient. The effectiveness of the proposed method is underscored by its impressive accuracy rate of 98 percent, outperforming Naïve Bayes, SVM, k-means, ROI, and colour threshold methods. A comprehensive comparative analysis, based on confusion matrices and performance metrics such as precision, recall, and F1-score values, further highlights the superiority of our approach over Naïve Bayes and SVM. These metrics serve as reliable indicators of the efficiency of our semantic segmentation approach in accurately identifying and delineating mushroom diseases in images. This study substantiates the efficacy of the Enhanced Random Forest approach as a valuable tool for managing and preventing mushroom diseases, emphasizing the importance of employing machine learning techniques in situations characterized by limited data availability.

Acknowledgement

We thank Dr. Anil Kumar Rao, Scientist ICAR-Directorate of Mushroom Research Solan (HP), India, and organization, for providing some of the data-set (diseased mushroom images) for our experimentation. Rakesh Kumar Y thanks the Management and the Principal of G. Narayanamma Institute of Technology and Science (For Women), Hyderabad, India, for allowing him to do his part time research work and providing computer with internet and supporting software to do the research.

References

[1] S. Sharma, S. Kumar, and V. P. Sharma. *Diseases and Competitor Moulds of Mushrooms and their Management.* Technical Bulletin, National Research Centre for Mushroom (ICAR), India, 2007. `https://dmrsolan.icar.gov.in/Disease___Competitor_Moulds__Dr._S.R._Sharma_.pdf`.

[2] J. T. Fletcher and R. H. Gaze. *Mushroom pest and disease control. A colour handbook.* CRC Press, London, United Kingdom, 2007. doi:10.1201/b15139

[3] E. Daniel, G. Julian, G. Helen, and B. Kerry. Viral agents causing brown cap mushroom disease of *Agaricus bisporus*. *Applied and Environmental Microbiology Journal*, 81(20):7125–7134, 2015. doi:10.1128/AEM.01093-15.

[4] I. O. Elibuyuk and H. Bostan. Detection of a virus disease on white button mushroom (*Agaricus bisporus*) in Ankara, Turkey. *International Journal of Agriculture and Biology*, 12(4):597–600, 2010. `http://www.fspublishers.org/published_papers/86156_..pdf`.

[5] J. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* Microsoft Research Technical Report No. MSR-TR-98-14, Apr 1998. `https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/`.

[6] D. R. Chowdhury and S. Ojha. An empirical study on mushroom disease diagnosis: A data mining approach. *International Research Journal of Engineering and Technology*, 4(1):529–534, 2017. `https://www.irjet.net/archives/V4/i1/IRJET-V4I190.pdf`.

[7] P. Goyal, E. Din, and D. Kapoor. A software for diagnosis and management of diseases and pests in white button mushrooms. *International Journal of Advanced Research in Computer and Communication Engineering*, 08(2):3136–3139, 2013. `https://www.ijarcce.com/upload/2013/august/37-H-pratibha goyal -A software for diagnosis and.pdf`.

[8] A. Jensen, P. Boll, T. I., and B. Pathak. Pl@nteInfo® – a web based system for personalized decision support in crop management. *Computers and Electronics in Agriculture*, 25:271–293, 2000. doi:10.1016/S0168-1699(99)00074-5.

[9] M. Y Minirah, M. Rozlini, and M. Y. Siti An expert system development: Its application on diagnosing oyster mushroom diseases. In: *13th International Conference on Control, Automation and systems*, pp. 20–23. Gwangju, Korea (South), 2013. doi:10.1109/ICCAS.2013.6703917.

[10] M. Y Minirah, M. Rozlini, and M. Y. Siti Design and rules development of expert system for diagnosing oyster mushroom diseases. In: *Proc. of the Computer and Information Science (ICCIS)*, pp. 286–289, 2012. doi:10.1109/ICCISci.2012.6297255.

[11] T. Zuva, O. O. Olugbara, S. O. Ojo, and S. M. Ngwira. Image segmentation, available techniques, developments and open issues. *Canadian Journal on Image Processing and Computer Vision*, 2(3):20–29, 2011. `https://www.researchgate.net/publication/264854010_Image_Segmentation_Available_Techniques_Developments_and_Open_Issues`.

[12] N. Jothiaruna, K. J. A. Sundar, and B. Karthikeyan. segmentation method for disease spot images incorporating chrominance in comprehensive color feature and region growing. *Computers and Electronics in Agriculture*, 165:104934, 2019. doi:10.1016/j.compag.2019.104934.

[13] S. Siddesha and S. K. Niranjan. Detection of affected regions of disease arecanut using k-means and Otsu method. *International Journal of Scientific and Technology Research*, 9(2):3404–3408, 2020. `http://www.ijstr.org/paper-references.php?ref=IJSTR-0120-30236`.

[14] T. N. Tete and S. Kamlu. Plant disease detection using different algorithms. In: *Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering-ACSIS*, vol. 10, pp. 103–106, 2017. `https://annals-csis.org/Volume_10/drp/24.html`.

[15] A. S. M. Shafi, B. Rahman, and M. M. Rahman. Fruit disease recognition and automatic classification using msvm with multiple features. *International Journal of Computer Applications*, 181(10):104934, 2018. doi:10.5120/ijca2018916773.

[16] Y. R. Kumar and V. Chandra Sekhar and A. K. Rao, An automatic multi-threshold image processing technique mushroom disease segmentation, *International Journal of Current engineering research*, 7(6):110-115, 2020 `http://troindia.in/journal/ijcesr/vol7iss6/110-115.pdf`.

[17] T. K. Ho. Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, p. 278–282, Aug 1995. doi:10.1109/ICDAR.1995.598994.

[18] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

[19] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587503.

[20] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In: *Proceedings of the British Machine Vision Conference*, p. 54.1–54.10, 2008. https://bmva-archive.org.uk/bmvc/2008/papers/207.html.

[21] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, London, 2013. doi:10.1007/978-1-4471-4929-3.

[22] B. Kang, Y. Lee, and T. Q. Nguyen. Depth-adaptive deep neural network for semantic segmentation. *IEEE Transactions on Multimedia*, 20(9):2478–2490, 2018. doi:10.1109/TMM.2018.2798282.

[23] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. doi:10.1109/TPAMI.2017.2699184.

[24] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. ICNet for real-time semantic segmentation on high-resolution images. In: *Computer Vision – Proc. ECCV 2018*, Lecture Notes in Computer Science, vol. 11207, p. 418–434. Springer International Publishing, 2018. doi:10.1007/978-3-030-01219-9_25.

[25] Z. Huang, X. Wang, Y. Wei, L. Huang, et al. CCNet: Criss-Cross attention for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(06):6896-6908, 2023. doi:10.1109/TPAMI.2020.3007032.

[26] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017. doi:10.1109/TPAMI.2016.2572683.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. doi:10.1109/TPAMI.2016.2644615.

[28] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, Jul 2017, pp. 5168-5177. doi:10.1109/CVPR.2017.549.

[29] E. Kawalec-Latała. Edge detection on images of pseudo impedance section supported by context and adaptive transformation model images. *Studia Geotechnica et Mechanica* 36(1):29–36, Mar 2014. doi:10.2478/sgem-2014-0004.

[30] B. Bouda, L. Masmoudi, and D. Aboutajdine. Cvvefm: Cubical voxels and virtual electric field model for edge detection in color images. *Signal Processing*, 88(4):905–915, 2008. doi:10.1016/j.sigpro.2007.10.006.

[31] P. Kuppusamy, M. M. Basha, and C.-L. Hung. Retinal blood vessel segmentation using random forest with gabor and canny edge features. In: *International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, Villupuram, India, Mar 2022, pp. 1-4. doi:10.1109/ICSTSN53084.2022.9761339.

[32] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha and S. Kundu. Improved random forest for classification. *IEEE Transactions on Image Processing*, 27(8):4012–4024, 2018. doi:10.1109/TIP.2018.2834830.

[33] Forest Mushroom Research Center (FMRC), Seoul, Korea (South). https://www.fmrc.or.kr.

[34] Imagenet. https://www.image-net.org.

[35] Mushroom World. http://www.mushroom.world/.

Rakesh Kumar Yacharam obtained his B.Tech from Osmania University, Hyderabad in 2006 and ME Degree from Osmania University in 2009. Presently pursing Ph.D From Osmania University, Hyderabad and working as Assistant Professor, ECE Department, G. Naryanamma Institute of technology and science for women, Hyderabad, India. His research areas of interest are Digital Image Processing and Cognitive Radio. He published around 22 papers in journals and conferences. He received 3 Awards, completed one funded project and published one patent.

Dr. V. Chandra Sekhar obtained his B.Tech from JNTU College of Engineering, Ananthapuram, in 1982. He acquired the ME Degree from college of engineering, Guindy, Chennai, in 1984. He recived Ph.D. from JNTU, Hyderabad, in 2012. His research areas of interest are coding techniques, image processing and wireless communication. Presently he is working as professor in ECE Department of Matrusri Engineering College, Hyderabad, India.

Restoration of Remote Satellite Sensing Images using Machine and Deep Learning: A Survey

Meriem Abdellaoui*, Souad Benabdelkader, Ouarda Assas
Electronics Department (LEA Laboratory), Faculty of Technology,
University of Batna 2 (Mostafa Benboulaid), Batna, Algeria
**Corresponding author: Meriem Abdellaoui (abdellaouimeriem23@gmail.com)*

Abstract. Remote sensing satellite images are affected by different types of degradation, which poses an obstacle for remote sensing researchers to ensure a continuous and trouble-free observation of our space. This degradation can reduce the quality of information and its effect on the reliability of remote sensing research. To overcome this phenomenon, the methods of detecting and eliminating this degradation are used, which are the subject of our study. The original aim of this paper is that it proposes a state of art of recent decade (2012-2022) on advances in remote sensing image restoration using machine and deep learning, identified by this survey, including the databases used, the different categories of degradation, as well as the corresponding methods. Machine learning and deep learning based strategies for remote sensing satellite image restoration are recommended to achieve satisfactory improvements.

Key words: image restoration, remote sensing images, Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Convolutional Neural Network (CNN).

1. Introduction

Remote sensing images are used to obtain a variety of data, including spying on enemy territories for the military purposes (the main purpose of building satellites), climate prediction and control, which has been one of the main civilian activities of remote sensing (Meteosat satellites), disaster prevention, measuring the ozone layer, detecting and controlling forest fires or oil slicks, mapping, etc. In other words, the general aim of remote sensing is to record and recognize the globe. Although satellites have made progress, researchers often encounter problems when taking satellite images, in terms of sensor malfunctions or the presence of clouds that prevent optimal exploitation of the data. In order to solve this problem, restoration treatments are frequently required to restore the polluted parts of these images and to exploit them afterwards. This document aims to present the studies carried out during the last decade (2012-2022) and their techniques for reconstructing satellite images based on automatic and deep learning, the different kinds of noise and the databases used. The papers consulted for our study were taken from references, publications and conferences relevant to our topic. This paper is structured as follows: Section 2 presents the databases used, Section 3 describe the main sources of degradation, Section 4 describe various techniques to detect and reduce them. Section 5 presents applications of these techniques. Section 6 concludes the paper.

Machine GRAPHICS & VISION 32(2):147–167, 2023. DOI: 10.22630/MGV.2023.32.2.8 .

2. Databases

As of April 30, 2018, the United States was leading the space launches; indeed, it has successfully launched 859 satellites in orbit. China launched 250, Russia 146, while in the rest of the world launched 631 satellites. In 2021, interestingly, the Chinese space industry has surpassed that of the United States, in terms of satellite launches, , while in Africa there was no rocket fired [1]. In turn, China has planned to organise more than 40 space expeditions in the coming months [2], where the Chinese super administrator China Aerospace Science and Technology Corporation (CASC), said in early 2022.

The remote sensing images were collected in several databases. Where, in this section, we will present the different satellites used in the work done according to the continents. In the United States, the Geostationary Operational Environmental Satellite (GOES) series [3] is the main group of meteorological satellites moving in geostationary orbit to provide frequent images of the Earth's surface and cloud cover to the National Weather Service, such as Meteosat [4] and [5]). The GOES satellites were planned by the National Aeronautics and Space Administration (NASA) [6].

In America, Land Satellite (Landsat) is widely used, and it is the main space program of Earth observation for non-military purposes. It was created by the American space agency NASA. Landsat was initially known under the acronym Earth Resources Technology Satellite (ERTS-1) [7] and its program is a technical and scientific success. This type of satellite is employed in [8]-[20]. In California, Vandenberg AFB launched QuickBird which is a commercial high resolution Earth Observation Satellite. QuickBird is used in [21]. PatternNet dataset has been suggested since 2017; it is composed of a large number of high-resolution and large-scale remote sensing images collected for remote sensing image databases. These images are collected from Google Maps (GMA) and Google Earth Imagery (GEI) for cities in the United States (US). PatternNet is utilized in [22]. National Oceanic and Atmospheric Administration (NOAA) is the American agency responsible for the investigation of the ocean and the atmosphere. On board of NOAA there is the primary sensor named Advanced Very High Resolution Radiometer (AVHRR), valuable for monitoring weather and vegetation on the surface of the globe, sea surface temperature, storms, etc. NOAA-AVHRR is applied in [23]. Space Plug-and-play Architecture Research Cubesat (SPARC) is a military research nano satellite shared by the United States and Sweden. SPARC is applied in [20].

In Europe, the system (satellite for Earth observation, SPOT) is a family of French civilian remote sensing satellites for earth observation. It was designed and launched by the French National Centre for Space Studies (CNES) [24], in a joint effort with Belgium and Sweden. It is used to acquire remote sensing information for commercial purposes. SPOT images have some applications in areas that require continuous imagery, such as guard service and agriculture. In addition, the Sentinel satellites are a family of European Earth observation satellites, focused on the environment and security [25]. Meteosat is a European meteorological satellite placed in geostationary orbit [26]. These satellites are developed under the supervision of the European Space Agency (ESA) on behalf of the European meteorological satellite operator (EUMETSAT). The European satellites Meteosat, Spot and Sentinel are used in [4,5], [27,28], and in [29,31], respectively.

In Asia, the Indian Remote Sensing (IRS) satellite series includes all Earth observation satellites launched and operated by the Indian Space Research Organization (ISRO) [32]. The Indian space agency is responsible for the planning and operation of satellites and launchers. In China, China National Space Administration (CNSA) is the space agency responsible for the Chinese Space Program. Gaofen is a family of Chinese earth observation satellites for civilian use. Its objective is to provide near real-time data for natural disaster prevention and treatment, climate change monitoring, mapping, resource and environmental monitoring, and agricultural support. Ziyuan (meaning Resources) is a Chinese Earth observation satellite. It is a high-resolution imaging satellite operated by the Ministry of Land and Resources (MLR) of the People's Republic of China. The satellite is utilized to provide imagery to monitor resources, land use and ecology, and for use in urban planning and disaster management. GaoFen is applied in [18,33,34,35,36,37] and ZiYun-3 is used in [18,33,37] and [38]. In addition, Yaogan (meaning remote sensing satellite) is a complete Chinese platform of earth observation and remote sensing satellites for military use. Officially, the Chinese authorities consider them as observation satellites designed for crop evaluation, disaster prevention, urban planning and scientific experimentation. But it is generally accepted that, given their orbit, payload and launch rate, they are in fact military satellites [22]. UC Merced is the Google image dataset of the University of California, Merced (UC Merced) or (UCM). Its images were extracted in a manual way from large images from the database of The United States Geological Survey (USGS), which is a U.S. government agency responsible for monitoring the seismic phenomenon. UC Merced is applied in [22]. WHU-RS19 is a set of remote sensing satellite images exported from Google Earth, and it was released by Wuhan University responsible for providing high-resolution satellite images [22]. As for the aerial image dataset (AID) [39], it is a new large-scale dataset, obtained by collecting sample images from Google Earth. RSSCN7 is a satellite image database collected from the private research company Remote Sensing Systems (RSS) that processes microwave data from a variety of NASA satellites. RSSCN7 is utilized in [39].

3. Sources of image degradation

The main sources of degradation can be divided into several categories: physical degradation, linked to the imperatives of physics, notably the radiative nature of sunlight and air turbulence. Mechanical degradation linked to the camera (the impacts of photographic grain), electronic degradation linked to errors in information transmission (transmission in the camera to the radio device), and optical degradation linked to the properties of the imaging system (the lenses), and so on. For each type of degradation, the image processing operations that can be applied to reduce its effects depend on the source of the degradation [40]. This is why image restoration processing is often essential to correct the distortions introduced and thus improve the quality of these images, so that they can be used later.

The presence of degradation is the major problem of images obtained from satellites. It can take various forms such as: clouds [4, 5, 10, 12, 13, 14, 17, 18, 20, 21, 23, 28, 31, 33, 35, 41], the cloud and its shadow [8, 19, 34, 37], haze [9, 11, 39, 42], thin cloud [15], thick cloud [16,23,38], thick cloud and cloud shadow [29], cloud and snow [30], noise [36,43,44], shadow [45], noise and blur [46] and jitter [22].

4. Techniques used

With the rapid development of remote sensing image acquisition technology, there are often degraded regions in these images due to poor atmospheric conditions or internal malfunction of satellite sensors that cause the loss of collected information and also make target detection, object recognition and other post-processing tasks very difficult, generating erroneous results. Detection and elimination of degradation can therefore improve the efficiency of remote sensing image interpretation. Image restoration involves restoring missing data from the original image from the degraded image. The considerable number of application areas of image restoration techniques demonstrates the importance of this operation in the field of image processing, from cosmic and astronomical images to medical images [47] and police investigations. In this section, the bibliography surveyed the different detection and removal techniques employed by researchers to restore remote sensing images.

4.1. Classification of restoration techniques

Relevant approaches to remote sensing image restoration can be divided into two broad categories: approaches based on classical algorithms and approaches based on Artificial Intelligence [48]. Some examples of approaches based on classical algorithms are:

- Clear-Sky Background Differencing (CSBD) algorithm based on image characteristics [50, 51].
- Automatic Cloud Cover Assessment (ACCA) based on the relationship between objects of cloud and cloud shadow [52].
- Background Subtraction Adaptive Threshold (BSAT) method [53].
- Spectral indices method-cloud index (CI) and clod shadow index (CSI) [54].
- Himawari-8 Cloud and Haze Mask (HCHM) algorithm [55].
- Fmask algorithm (Cloud Displacement Index) CDI [56].

Regarding the approaches based on Artificial Intelligence here are some types:

- Multi-Scale Residual Convolutional Neural Network (MRCNN) [11].
- Simple Linear Iterative Cluster (SLIC), Deep Convolutional Neural Networks (CNNs) [21].
- Multiple Convolutional Deep Neural Networks (ConvNets), Conditional Random Field (CRF) [45].
- Image Despeckling Convolutional Neural Network (ID-CNN) [57].

4.2. Techniques based on classical algorithms

Classical algorithms are considered as the methods that have specific known steps to follow for a specific input image. The output of cloud removal and detection depends on the input image and the algorithms employed (input + program = output); moreover, in classical techniques there is no learning. In contrast, machine learning is a field of Artificial Intelligence that allows systems to learn automatically based primarily on the input image and existing data (input + output = program).

4.3. Artificial intelligence based techniques

Artificial Intelligence (AI) is a term used in 1956 by John McCarthy. It is the science and engineering of making intelligent machines, it is a thought that suggests that hardware can learn and think on its own, without being coded with commands [49]. AI has offered promising solutions to the problem of image processing, especially the restoration of remote sensing images, allowing greater flexibility which makes it more robust than traditional techniques. Machine learning (ML) is a field of study in AI, its basic idea is the study of computer algorithms that can improve automatically through experience and the use of data. AI has two phases: the first is learning or training where the ML must first be trained by processing a large number of input patterns and their associated reference output patterns, once trained, the ML is able to recognize similarities when presented with a new input pattern, resulting in a predicted output pattern presented by the second phase. Deep learning (DL) is the sub-domain of ML derived from AI.

Machine and deep learning is about creating huge neural network models capable of making accurate choices based on data, DL is suitable for situations where the data is complex. DL algorithms have been progressing day by day for a very long time in the improvement of image processing algorithms and have developed in many fields. Notably, space research, intelligent robots, security and surveillance, autonomous vehicles, voice, facial and fingerprint recognition, social networks where Facebook uses it to break down the message in online discussions, financial forecasting, automated commerce, identification of defective parts and localization of malware or false statements. In the health field, DL algorithms analyze information extracted from wearable watches, artificial pacemakers and various monitoring sensors placed in the human body. DL elements have made it possible to detect many diseases including epilepsy, hypoglycemia and atrial fibrillation. As for the gaming industry, the Xbox uses DL. to detect body movements and respond by exciting game fans. In addition, in language processing, DL can understand speech, convert it into written form and translate one language into another. Likewise, all the intelligent computer systems that are equipped with DL, have contributed to the enormous success, which we are currently witnessing [51].

5. Application of machine and deep learning for remote sensing restoration

Restoration of remote sensing images using machine and deep learning is the objective of our paper. However, considerable research is available in the literature to provide noise-free images or at least images with reduced degradation impacts, in particular, due to the arrival of new satellite images. This leads us to classify these algorithms in three categories as follows: Some of them deal with suppression, others with detection, while the last ones deal with both at the same time. The bibliography survey has reviewed different techniques.

5.1. Detection techniques

Noise frequently exists in remote sensing images, diminishing the quality of the image and leading to erroneous or inaccurate interpretations and thus causing many obstacles to remote sensing image applications. To remedy this, it is essential to first detect this noise and then remove it. Recently, several new studies have appeared for this type of technique, we present them below:

- Multilayer Perceptron (MLP) [4].
- Fuzzy Logic, Neural Network [5].
- Fully Convolutional Neural Networks Fully Convolutional Network (CloudFCN) [10].
- Multiscale Features-Convolutional Neural Network (MF- CNN) [12].
- Spectral Rationing + Fuzzy C-Means Clustering (FCM) [17].Cloud Detection Neural Network (CDnet), Deep Convolutional Neural Network (DCNN) [18].
- Deep Convolutional Neural Networks, SegNet, Remote Sensing Network (RS-Net) [20].
- Adaptive Simple Linear Iterative Clustering (A-SCLI), Multiple Convolutional Neural Networks (MCNNs) [33].
- Machine Learning and Multi-Feature, Multilevel Feature Fused Segmentation Network (MFFSNet) [34].
- Linear Stripe Noise Detection (LSND)[34].Convolutional Neural Network -3D Multiscale (3D-CNN) [37].

The paper [4] adopted the Multilayer Perceptron (MLP) approach which is a multilayer perceptron neural network to detect clouds in the Meteosat second generation Spin Enhanced Visible and Infrared Imager (MSG SEVIRI) images with the CLoud Mask

(CLM) provided by EUMETSAT. The MLP model is a feed forward artificial neural network classifier. The connections between the perceptrons in an MLP are direct and each perceptron is connected to all the perceptrons in the next layer, except for the output layer which gives the result. This approach is useful in cases where there is not enough auxiliary data. Furthermore, it is believed that the multilayer perceptron can be improved by increasing the size and diversity of the training and test sets, and by systematically testing other types of artificial neural networks and training algorithms. This proposed model was able to detect not only thick and bright clouds but also thin or less bright clouds. In addition, the execution time is about 20 s, which gives a significant impact on reducing the computational load when large data sets need to be processed.

Automatic detection of daytime land and marine clouds from Meteosat second generation rotationally enhanced visible and infrared imager (MSG SEVIRI) images based on fuzzy logic and neural networks was the proposed topic of the authors of [5]. They used the threshold mechanism and auxiliary data such as numerical weather prediction (NWP) for the development of the model. The analysis of the results obtained by the neural network compared to fuzzy logic also demonstrates its high accuracy and the usefulness of using artificial intelligence techniques in remote sensing imagery applications. This approach was not only able to detect thick clouds but also thin and less bright clouds.

Correct detection of cloudy pixels in Landsat 8 remote sensing images that relies on deep learning using fully Convolutional neural networks named FCN and CloudFCN are developed by [10] and [13] respectively. The deep learning process aims at extracting local and global semantic features at the pixel level of cloudy areas in an image. In addition, a gradient-based total identification is designed to perceive and exclude snow/ice areas in ground truths from the training set. The proposed techniques provide distinct and diverse detailed performance tests, which confirm that fully Convolutional network architectures are indeed a powerful and effective tool for cloud detection in remote sensing images, and can outperform previous techniques. Although these designs have become a standard deep learning approach for image segmentation, a direct deficiency of this work is the coverage of cloud shadows, fog and haze.

The Multiscale Features Convolutional Neural Network (MF-CNN) method described in the paper [12] is based entirely on a neural network and aims to solve the problem of reliably detecting thin clouds at the pixel level, while providing excessive accuracy for detecting thick clouds and non-cloudy pixels in remote sensing images. The design consists of first stacking the visible near- infrared, shortwave, cirrus, and thermal infrared bands of Landsat 8 imagery to obtain the combined spectral information. To learn the global multiscale features of the stacked images, the MF-CNN model is then used. The high-level semantic information acquired in the feature learning procedure is integrated with the low-level spatial information to classify the imagery into thick, thin, or cloud-free regions. The proposed method leads to the identification of complex cloud types and shapes. Experimental comparison of the results of the MF-CNN model with those of traditional machine learning, deep learning, and the classical Fmask and F_Score method of thick and thin clouds are needed to further evaluate the performance of the proposed model.

The authors of [17] have automatically detected clouds in Landsat ETM+ images without any manual intervention. The proposed approach is to conduct a color transformation on the input image. Then, by using the spectral image rationing technique a report image will be produced. Finally, it gathers the report image using Fuzzy C-Means clustering (FCM) to detect the clouds in an automatic way. The spectral rationing technique uses the value of the ratio between croma and luma to build the report image to detect clouds in satellite images. This method is effective in detecting thick clouds and thin clouds in average time.

The topic addressed by [18] is to detect clouds through a neural network of (CDnet) with an encoder-decoder structure, a feature pyramid module (FPM) and a boundary refinement block (BR) used for cloud mask extraction via ZY-3, GF-1 WFV and Landsat-8 satellite vignettes. The objective of this paper is threefold: First, the FPM module extracts multi-scale contextual data without lack of resolution and coverage. Second, the BR block refines object boundaries by exploiting high-level semantic capabilities and mid-level visual properties for category recognition of image areas. Finally, the encoder-decoder network structure recovers the segmentation results step by step with a size equivalent to the input image. Experimental results show its efficiency and robustness using only three bands of the multi-spectral images, but its drawback is the localization of boundaries for thin clouds.

The authors of [19] proposed a deep Convolutional Neural Network (CNN) to surface clouds and their shadows in Landsat 7 and Landsat 8 images. The authors performed a detailed CNN-based semantic segmentation named SegNet for extracting multi-level spatial and spectral features computed on the full input image to identify pixels as clouds, thin clouds, cloud shadows or bright areas. According to the extensive qualitative and quantitative analysis compared to FCMask, the adapted SegNet technique achieves promising performance in terms of overall accuracy for cloud and cloud shadow detection.

A formula for cloud detection in satellite imagery using deep learning and a remote sensing network (RS-Net) based on the U-net structure employing a fusion of spatial and spectral models has been planned by the authors of [20]. The model is trained using Landsat 8 Biome and SPARCS datasets. The high performance of this approach, which uses only the RGB and RGBI (Red/Green/Blue/Infrared) bands, outperformed the Fmask algorithm.

The author of [28] evaluated the performance of the proposed algorithm to automatically detect clouds from panchromatic SPOT5-HRS multi-temporal satellite images. This algorithm is designed by a regional growth procedure. The sheaths that correspond to the cloud are picked by a pixel-to-pixel comparison between existing images based on a strong change in reflection between two images. Although this method works on images with a single panchromatic channel and no longer requires a thermal band, the drawback is the false positive detection of many clouds which requires improved post-processing.

Multi-level cloud detection is a challenge for high-resolution remote sensing images based on a deep learning framework, this was the topic proposed by the authors of [21, 23]and [30]. First, the image is segmented into good quality super-pixels using the Simple Linear Iterative Clustering (SLIC) method. Then, a pair of image patches is extracted from each super-pixel and fed into a two-branch deep Convolutional Neural Network (CNN) designed to extract the multi-scale features of each super-pixel which effectively predicts the class of that super-pixel. Finally, the final cloud detection result is obtained using the predictions of all super-pixels. Through qualitative and quantitative analysis, and by evaluating the approach used with those previously performed, it was found that the performance of the approach used not only detects clouds at multiple levels, but also distinguishes between thin and thick clouds in [21] and [23] and between clouds and snow in [30].

The article [33] adopted the same techniques as the previous articles [21, 23, 30]. Because it performs multilevel cloud detection by applying the Adaptive Simple Linear Iterative Clustering (A-SCLI) algorithm to segment the satellite image into superpixels. Except that the CNN used by the authors of [21, 23] and [30] is replaced by a Multiple Convolutional Neural Network (MCNNs) which has the same task. The proposed method performed on GF-1, GF-2 and ZY-3 databases to distinguish between thin clouds, thick clouds and cloud shadows. Cloud and cloud shadow detection using multi-level feature fusion segmentation network (MFFSNet) for automatic training is performed by the authors in [34]. First, they used a fully convolutional network for training the cloud features and their shadows. Then, the extraction of the contextual relationship between the cloud and its shadow is performed by a new pyramid. Finally, to combine the semantic and spatial information of different levels to achieve better multi-scale object management and produce detailed segmentation boundaries, a special multi-level feature fusion structure is designed. The experimental aspect shows that MFFSNet outperforms the latest methods and achieves a high level of accuracy.

The authors of [35] investigated the machine learning strategy and fusion of several features, based on a comparative analysis of spectral, textural, and other typical variations between clouds and backgrounds in the images for cloud detection. By processing the Gao Fen-1 and Gao Fen-2 image database selected in southern China, object-oriented post-processing was applied using rectangles and a length-to-width ratio shape index, which further minimizes the classification errors of highly reflective images, thus increasing the accuracy. The proposed algorithm can be applied to totally different varieties, sizes and densities of clouds, and to any image source. Despite the reliability shown by this approach, the training samples must necessarily be selected manually. This analysis is intended to meet the requirements of the Chinese disaster reduction project, which focused on drought and flooding in southern China.

The authors of [36] have developed a new approach for band noise detection in GaoFen-2 high resolution remote sensing images using a deep learning technique called Linear Stripe Noise Detection (LSND). First, through linear transformations a large scale dataset is generated by simulating a wide variety of remote sensing images with band noise. Then, the target recognition of the band noise was performed using Deep Convolutional Neural Networks. On the experimental basis the LSND algorithm indicated its validity in terms of accuracy and time.

The basic concept of the paper [37] focuses on a multi-scale (3D-CNN) network of high- resolution multi-spectral imagery for the detection of clouds and their shadows in GF-1 WFV and ZY-3 data sets. The extraction of contextual data of clouds and their shadows at various levels was performed by a multi-scale learning module. In addition, a joint spectral-spatial information of the 3D convolution layer developed to discover the joint spatial-spectral correlations in the input data. The proposed network significantly improved the accuracy of shadow and cloud detection and could even distinguish between high-albedo objects (snow and ice) and low-albedo objects (water and mountain shadow).

In the paper [39] the researchers combined wavelet transform and deep learning technology to remove deep haze in remote sensing images where the haze was not evenly distributed in the image. First, the input image information is extracted from the first-order low-frequency Subband of its 2D stationary wavelet transform. Then, the network learns the more abundant image features and improves the overall ability to detect non-uniform haze in remote sensing images. Qualitatively and quantitatively, the proposed approach has superior advantages over traditional methods for removing non-uniform haze in remote sensing images.

The techniques for detecting noise in satellite images are summarized in Table 1, where the techniques used, the databases, the form of noise as well as the reference of each article and its year of publication are described.

5.2. Elimination techniques

Remote sensing images are frequently degraded, which minimizes the efficiency and accuracy of image interpretation. The removal of degradation from satellite images is an essential task after its detection. For this reason, many research efforts have been directed towards the removal of degradation from satellite images such as:

- Spatial Procedures for the Automated Removal Cloud and cloud Shadow (SPARCS) [8].
- Multi-Scale Residual Convolutional Neural Network (MRCNN) [11].
- Convolutional Neural Network (CNN) [15]. Progressively Spatio-Temporal Patch Group Deep Learning [29].
- Sentinel-1/2 Cloud Removal Time Series (SEN12MS-CR-TS) [31].

Tab. 1. Summary of detection techniques

Techniques used	**Databases**	**Forms of noise**	**Ref.**	**Year**
• Multilayer Perceptron Neural Networks (MLP)	Meteosat Second Generation Spinning Enhanced Visible and Infrared Imager (MSG-SEVIRI)	Cloud	[4]	2015
• Fuzzy Logic • Neural Network	Meteosat Second Generation Spinning Enhanced Visible and Infrared Imager (MSG SEVIRI)	Cloud	[5]	2018
• Fully Convolutional Neural Networks (FCN)	Landsat 8	Cloud	[10]	2018
• Multiscale Features Convolutional Neural Network (MF-CNN)	Landsat 8	Cloud	[12]	2018
• Fully Convolutional Network (CloudFCN)	Carbonite-2 Landsat 8	Cloud	[13]	2018
• Spectral Rationing • Fuzzy C-Means Clustering (FCM)	Landsat ETM+	Cloud	[17]	2013
• Cloud Detection Neural Network (CDnet) • Deep Convolutional Neural Network (DCNN)	ZY-3 GF-1 WFV Landsat 8	Cloud	[18]	2019
• Deep Convolutional Neural Network • SegNet	Landsat 7 Landsat 8	Cloud and cloud shadow	[18]	2019
• Remote Sensing Network (RS-Net) • Deep Learning	Landsat 8 Biome, SPARCS	Cloud	[20]	2019
• Simple Linear Iterative Clustering (SLIC) • Deep Convolutional Neural Networks (CNNs)	Quickbird	Cloud	[21]	2016
• Simple Linear Iterative Clustering (SLIC) • Convolutional Neural Networks (CNN)	NOAA/AVHRR	Cloud	[23]	2017
• Automatic method	SPOT5-HRS	Cloud	[28]	2012
• Simple Linear Iterative Clustering (SLIC) • Convolutional Neural Networks (CNN)	Sentinel-2A	Cloud and snow	[30]	2018
• Adaptive Simple Linear Iterative Clustering (A-SCLI) • Multiple Convolutional Neural Networks (MCNNs)	GF-1 GF-2 ZY-3	Cloud	[33]	2018
• Multilevel Feature Fused Segmentation Network (MFFSNet)	GF-1	Cloud and cloud shadow	[34]	2018
• Machine Learning • Multi-Features	GF-1 GF-2	Cloud	[35]	2016
• Linear Stripe Noise Detection (LSND)	GF-2	Noise	[36]	2022
• Convolutional Neural Network • 3D Multiscale (3D-CNN)	GF-1 WFV ZY-3	Cloud and cloud shadow	[37]	2020
• Wavelet Transform • Deep Learning	AID RSSCN7 BH	Haze	[39]	2021

- Wavelet Transform, Deep Learning [39].
- Reliable Cloudy Image Synthesis Model [41].
- Hyper Spectral Image denoising by Network (HSI-DeNet), Convolutional Neural Network (CNN) [43].

Spatial Procedures for Automated Removal of Cloud and Shadow (SPARCS) was the methodology proposed by the paper [8] using Landsat TM and ETM+ single date satellite images. This approach firstly uses a neural network to determine the membership of each pixel of an image scene to the classification of clouds, cloud shadow, water, snow/ice and clear sky. Then, it applies a series of spatial procedures to determine pixels with questionable membership using data, e.g., membership values of adjacent pixels and an estimate of the location of cloud shadows from solar geometry. For this approach to be applicable, it must meet the following rules: SPARCS uses only single-date images, and does not depend on auxiliary data sets. Furthermore, this strategy is fully computerized and does not require the determination of new boundaries for different scenes. The usefulness of the SPARCS method is demonstrated by comparing it to a state-of-the-art method used, FMask.

In the articles [9] and [11] the authors were able to efficiently remove haze in each band of Landsat 8 OLI (Operational Land Imager) multi-spectral images using a combination of a Convolutional Neural Network (CNN) with Residual and Multi-Scale Residual Convolutional Network (MRCNN) architecture respectively. These two techniques are similar. The basic idea of these algorithms is as follows. Starting with, haze removal based on Convolutional Neural Network (CNN), where different CNN individuals are connected to each other to learn the correspondence between the hazy image and the clear image, and a fusion unit is used to adaptively integrate the outputs of these individuals to generate the restored image. Then, through multi-scale convolutional the multi-scale features of the haze are extracted and the residual architecture to minimize the learning difficulty is adopted. Finally, the haze as a function of wavelength to generate a haze very close to the real conditions is simulated, thus training the designed network. The experimental results showed the validity of the proposed algorithm compared to the existing algorithms, to remove the haze in each band of multi-spectral images under different scenes with remarkable accuracy.

The paper [15] focuses on thin cloud removal in multi-spectral remote sensing images using Convolutional Neural Network (CNN) and a traditional imaging model. U-Net is used to estimate the reference thin cloud thickness map, while, the Slope-Net is used to estimate the thickness coefficient of each band. Thus, the cloud thickness maps of different bands are obtained. Finally, using the traditional thin cloud imaging model, the thin cloud thickness maps are subtracted from the cloud image. In order to evaluate the reliability and credibility of this experiment, a qualitative and quantitative analysis was performed on synthetic and real Landsat 8 OLI satellite images. The results showed that the suggested method can keep a better color quality by removing thin clouds in multi-spectral images with various land cover types.

The fundamental concept of the paper [29] is the combination of global and local spatiotemporal information from remote sensing images with the nonlinear learning capability of the Deep Neural Network for the removal of thick clouds and their shadows in multi-temporal images from the Sentinel-2 MSI and Landsat 8 OLI satellites. The significant advantages of this method over previous methods are: thick cloud coverage over large-scale areas, all temporal images have clouds or shadows and the deficient use of a single temporal image. A global-local DCNN network provided to optimize the formation model across cloudy and non-cloudy regions, taking into account global consistency and local particularity. The proposed system applied a global-local loss function in the supervised learning technique to optimize the training model across cloud-covered and non-cloud regions. In addition, weighted aggregation and progressive generation are used to reconstruct the holistic results. Experimental analysis proved the accuracy of removing thick clouds and their shadows from single and multi-temporal images of small/large scale scenes.

The authors of [31] designed an algorithm known as SEN12MS-CRTS for the reconstruction of Sentinel-1 and Sentinel-2 optical satellite images and the removal of multimodal and multi- temporal clouds. The validity and efficiency of SEN12MS-CRTS has been proven by considering two methods: first, a 3D multi-modal and multi-temporal Convolutional Neural Network that predicts a cloud-free image from a time series covered with clouds. Second, a network for sequence-to-sequence cloud removal which is the first case where a model preserving temporal information has been predicted in the context of cloud removal. Both strategies take advantage of the presence of coregistered and matched SAR (Synthetic Aperture Radar) measurements contained in the data set. Interestingly, the benefits of using multi-modal and multi-temporal data to reconstruct noisy data have highlighted the contribution of the dataset to the remote sensing community. The reliable model for cloudy image synthesis is the Convolutional Neural Network (CNN) based cloud removal approach in satellite images was proposed in reference [41]. First, the extraction of cloud masks from real cloud images by the layer separation method and the dark channel selection method. Second, the refinement of cloud masks by reflecting the color of the background surface as a function of cloud thickness. Finally, using the synthesized cloud images the hierarchical cloud suppression network is trained with a multi-scale scheme. An experimental evaluation indicated the validity of the proposed technique compared to state-of-the-art methods for accurate cloud removal in satellite images.

To improve the quality of multi-spectral remote sensing images contaminated by haze, the authors of [42] developed an efficient and reliable haze removal algorithm based on a learning framework. A linear regression model with relevant haze features was established, and the gradient descent methodology applied to the training model. From a hazy image, a correct transmission map was obtained by learning the coefficients of the linear model. Similarly, this algorithm estimated the atmospheric light in order to limit the influence of highlighting surfaces on the acquisition of atmospheric light. This method has shown its reliability to obtain a better image quality in the context of removing fine haze while preserving colors compared to the traditional strategies. The authors of [43] performed noise removal in hyper-spectral images (HSI), including random noise, structural stripe noise and dead pixels/traces, based on the deep Convolutional Neural Network (CNN) through the (HSI-DeNet) approach. The objective of this algorithm has overcome the problems faced by researchers of the same concern, which are the following. First, the proposed HSI-DeNet technique can be taken as a tensor method using filter learning in each layer without destroying the spectral and spatial structures. Secondly, the HSI-DeNet can take into account both different forms of noise in the HSI. Furthermore, this approach can be adapted for single and multiple images by slightly changing the filter channels of the first and last layer. Finally, the excessive speed of this method for testing, made it more practical for real applications. The quantitative and qualitative evaluation of HSI-DeNet on different types of simulated and real HSI images proved its high performance and extreme restoration runtime compared to the compared methods.

The methods for noise removal in satellite images are summarized in Table 2, where the techniques used, the datasets, the form of noise and the reference of each paper and its year of publication are indicated.

5.3. Detection and suppression techniques

As degradation detection and removal are exceptionally interrelated and complementary, there is a need for an integrated framework that handles both tasks simultaneously. Another rich family of techniques to solve the remote sensing image detection and removal problem using machine and deep learning is described below:

- Cloud Detection Network (CDN), Cloud Removal Network (CRN) [14].
- Spatial-Temporal-Spectral based on a Deep Convolutional Neural Network (STS-CNN) [16].
- Image Restoration Based on Generative Adversarial Networks (RestoreGAN) [22].
- Deep PnP Low-Rank Tensor Approximation (DPLRTA) [44]. Multiple Convolutional Deep Neural Networks (ConvNets), Conditional Random Field (CRF) [45].
- Image restoration via deep learning (RestoreNet-Plus) [46].

The technique of detecting and removing clouds and cloud shadows simultaneously in Landsat-8 remote sensing bitemporal images through cascaded convolutional neural network (CNN) has been the proposed topic by [14]. Its design is organized as follows: the cloud images and the corresponding temporal images are processed by two fully convolutional networks (FCN) in cascade that structure the fundamental body of the system. The first FCN with multi-scale aggregation and channel attention mechanism, Tab. 2. Summary of removal techniques

Techniques used	**Databases**	**Forms of noise**	**Ref.**	**Year**
• Spatial Procedures for Automated Removal of Cloud and Shadow (SPARCS)	Landsat TM Landsat ETM+	Cloud and cloud shadow	[8]	2014
• Convolutional Neural Network (CNN) • The residual structure	Landsat 8 OLI	Haze	[9]	2018
• Multi-Scale Residual Convolutional Neural Network (MRCNN)	Landsat 8 OLI	Haze	[11]	2018
• Convolutional neural network (CNN) combined with an imaging model	Landsat 8 OLI	Thin cloud	[15]	2021
• Progressively Spatio-Temporal Patch Group Deep Learning	Sentinel-2 MSI Landsat 8 OLI	Thick cloud and cloud shadow	[29]	2020
• Sentinel-1/2 Cloud Removal Time Series (SEN12MS-CR-TS)	Sentinel-1 Sentinel-2 Landsat 8 OLI	Cloud	[31]	2022
• Convolutional Neural Network (CNN) • Reliable Cloudy Image Synthesis Model	Satellite images	Cloud	[41]	2019
• Learning Framework	Remote Sensing Multispectral Images	Haze	[42]	2019
• Hyperspectral Image denoising byNetwork (HSI-DeNet) • Convolutional Neural Network (CNN)	Hyperspectral Images	Noise	[43]	2018

aims to detect clouds and shadows using the Cloud Detection Network (CDN), while the second FCN with the detected cloud and shadow masks, the cloud image and a temporal image, is used for cloud removal and reconstruction of missing data provided by the Cloud Removal Network (CRN). The restoration was accomplished by a methodology of self-training designed to learn the correspondence between pairs of clean pixels of bitemporal images, thus avoiding the need for manual labels. The experimental aspect showed that the suggested algorithm was able to simultaneously detect and remove clouds and shadows from remote sensing images, thus outperforming traditional methods in all indicators, with a significant margin.

A pioneering work is done in the paper [16]. In this paper, the author adopts the Spatial- Temporal-Spectral (STS-CNN) method based on Deep Convolutional Neural Network, which reconstructed the missing data in a Landsat ETM + (Enhanced Thematic Mapper Plus) remote sensing image through a unified spatial-temporal-spectral (STS) framework based on a deep convolutional neural network (CNN). The basic idea of this paper is to use the unified framework to solve the following three problems: first, recovering deadlines in the Aqua Moderate Resolution Imaging Spectroradiometer (MODIS) band 6, second, correcting the scan lines (SLC) of Landsat ETM+, and third, removing thick clouds. Although existing methods can only handle a single task of reconstructing missing information, the proposed strategy was found to be effective in recovering deadlines in Aqua MODIS band 6, solving the SLC problem and removing thick clouds. The STS-CNN approach had some shortcomings such as, spectral distortion and blurring appeared during the removal of thick clouds by using temporal information.

The paper [22] proposed RestoreGAN architecture for jitter detection and restoration of remote sensing image, based on a Generative Adversarial Network (GAN) to learn and correct in an automatic way the features of the contaminated scene from a single remote sensing image. While two Convolutional Neural Networks (CNN) are designed, the first one serves to separate the inputs and the second one to adjust the distortions. After the validation and verification proofs of the RestoreGAN method on PatternNet, UC Merced, WHU-RS19 and Yaogan-26 databases respectively, the proposed system demonstrated its performance in terms of Deformation Metric (DM) compared to UnrollingCNN, GenCNN and ContGAN methods.

The topic of [38] presents a deep learning based method for the detection and removal of thick clouds from optical images of the ZY-3 satellite. For the first task convolutional neural network (CNN) architecture is used, while for the second one which is the recovery of image information under the clouds, content, texture and spectrum generation (CTS) networks based on classical CNN are used. It should be noted that the framework of the proposed CNN structure can use multi-source data (content, texture and spectrum) as a unified input. Although, the experimental results on simulated and real images have shown the effectiveness is robustness of the approach to remove particular types of thick, thin and shadow clouds. But it stands helpless in front of the changing land cover.

For hyper-spectral image recovery (HSI), the authors of [44] treated the Plug-and-Play (PnP) framework, due to its scalability and flexibility, as a bridge between traditional HSI restoration techniques and deep noise removal networks. The proposed approach, Deep PnP Low-Rank Tensor Approximation (DPLRTA), is a three-step process: Tensor Modeling, Low-Rank Tensor Decomposition, and Noise Removal by Implicit Convolutional Neural Network (ICNN) by regularization. PnP is a bridge that connects these three steps. Simulation and real experiments on Pavia City Centre and HYDICE Urban data respectively proved that DPLRTA can effectively preserve the detail, fundamental shape and texture data of HSI.

The authors of [45] automatically detected and removed shadows in real-world scenes from a single image using a fusion of Convolutional Deep Neural Networks (ConvNets) and a Conditional Random Field (CRF). The approach aims to automatically learn the most relevant features in a supervised manner for shadow detection based on multiple networks (ConvNets). Properties were also learned at the super-pixel level and on the dominant boundaries of the image. Posterior predictions based on the learned features introduced in a field model (CRF) to obtain shadow masks. With the help of the detected shadow masks, a Bayesian formulation that constitutes the concept of this shadow elimination process was used to appropriately extract the shadow matte with the recovered image, and then eliminate it. The proposed framework proved its performance

Tab. 3. Summary of detection and suppression techniques

Techniques used	**Databases**	**Forms of noise**	**Ref.**	**Year**
• Cloud Detection Network (CDN) • Cloud Removal Network (CRN)	Landsat 8	Cloud and cloud shadow	[14]	2020
• Spatial-Temporal-Spectral based on a Deep Convolutional Neural Network (STS-CNN)	Landsat ETM+	Thick Cloud	[16]	2018
• Image Restoration Based on Generative Adversarial Networks (RestoreGAN)	PatternNet UC Merced WHU-RS19 Yaogan-26	Jitter	[22]	2021
• Detection: Convolutional Neural Network (CNN) • Removal: Content-Texture-Spectral (CTS-CNN)	ZY-3	Thick Cloud	[38]	2019
• Deep PnP Low-Rank Tensor Approximation (DPLRTA) • Convolutional Neural Network (CNN)	Pavia City Centre (data simulation) HYDICE Urban (data real)	Noise	[44]	2020
• Multiple Convolutional Deep Neural Networks (ConvNets) • Conditional Random Field (CRF)	UCF CMU UIUC	Shadow	[45]	2015
• Image restoration via deep learning (RestoreNet-Plus)	Optical Synthetic Aperture Imaging (OSAI)	Noise and Blur	[46]	2021

on various databases (UCF shadow, CMU shadow and UIUC shadow) unlike previous research.

The paper [46] suggested an improved RestoreNet-Plus network for image restoration of a synthetic aperture optical imaging system based on deep learning. To establish a hidden nonlinear correspondence between the output and input without analytical expression, a neural network is used. While learning, the neural network is able to fit an input model to an output model that approximates the inverse problem process. Analysis of the experimental results indicated that RestoreNet-Plus is a better alternative compared to other methods in terms of noise suppression and restoration of synthetic aperture optical imaging.

The strategies for detecting and suppressing or removing noise in satellite images are summarized in Table 3, which shows the approaches used, the databases, the form of noise as well as the reference of each paper and its year of publication.

6. Conclusion

The bibliographic survey carried out between 2012 and 2022 on the techniques of restauration of satellite imagery data by machine and deep learning is analyzed. The satellites used in each work, the recognition of different forms of degradation, including clouds, haze and shadows are examined. The type of technique suitable for their treatment;

detection, elimination and algorithms that process both are studied. The study of the literature shows that the most used databases are, in descending order, the American satellites (Landsat), the Asian satellites (GF-1/2 and ZY-3), then the European satellites (Spot, Sentinel, Meteosat). In terms of the most processed type of degradation, clouds come first, followed by clouds and their shadows, and haze. The most widely used techniques are primarily the simple iterative linear cluster (SLIC) with convolutional neural networks (CNN) and fully convolutional neural networks (FCN). In addition, we note that the methods that deal with detection are more than those of suppression. It should be noted that, despite the great success and wide dissemination of American databases, there has recently been competition between Asia and America in terms of launching remote sensing satellites. Although AI-based strategies are pioneering in all areas compared to traditional algorithms, complementary efforts are needed to achieve promising results and performance in terms of reliability and calculation time. The accuracy of degradation detection and suppression can be increased by integrating special zones and time conditions according to various weather models. In addition, to overcome the constraints and disadvantages of current algorithms, it is crucial to combine atmospheric parameters with the artificial neural networks.

References

[1] Number of satellites in space world 2022. Statista. `https://www.statista.com/statistics/264472/number-of-satellites-in-orbit-by-operating-country/` (accessed Mar 11, 2022).

[2] In 2022, space development will remain at the heart of China's priorities. Opinion, Jan 05, 2022. `https://www.Opinion.com/international/in2022` (accessed Mar 11, 2022).

[3] NASA. GOES Satellite Network. `https://science.nasa.gov/mission/goes/` (accessed Apr 09, 2022).

[4] A. Taravat, S. Proud, S. Peronaci, F. Del Frate, and N. Oppelt. Multilayer perceptron neural networks model for meteosat second generation SEVIRI daytime cloud masking. *Remote Sensing*, 7(2):1529–1539, 2015. doi:10.3390/rs70201529.

[5] M. Reguiegue and F. Chouireb. Automatic day time cloud detection over land and sea from MSG SEVIRI images using three features and two artificial intelligence approaches. *Signal, Image and Video Processing*, 12(1):189–196, 2018. doi:10.1007/s11760-017-1145-0.

[6] NASA. `http://www.nasa.gov` (accessed Apr 09, 2022).

[7] R. S. Williams Jr. and W. D. Carter. ERTS-1, a new window on our planet. *U.S. Geological Survey*, Professional Paper 929, USGS 1976. doi:10.3133/pp929.

[8] M. J. Hughes and D. J. Hayes. Automated detection of cloud and cloud shadow in single-date Landsat imagery using neural networks and spatial post-processing. *Remote Sensing*, 6(6):4907–4926, 2014. doi:10.3390/rs6064907.

[9] M. Qin, F. Xie, W. Li, Z. Shi, and H. Zhang. (2018). Dehazing for multispectral remote sensing images based on a convolutional neural network with the residual architecture. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(5):1645–1655, 2018. doi:10.1109/JSTARS.2018.2812726.

[10] S. Mohajerani, T. A. Krammer, and P. Saeedi. Cloud detection algorithm for remote sensing images using fully convolutional neural networks. *arXiv*, preprint arXiv:1810.05782, 2018. doi:10.48550/arXiv.1810.05782.

[11] H. Jiang and N. Lu. Multi-scale residual convolutional neural network for haze removal of remote sensing images. *Remote Sensing*, 10(6):945, 2018. doi:10.3390/rs10060945.

[12] Z. Shao, Y. Pan, C. Diao, and J. Cai. Cloud detection in remote sensing images based on multiscale features-convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):4062–4076, 2019. doi:10.1109/TGRS.2018.2889677.

[13] A. Francis, P. Sidiropoulos, and J.-P. Muller. CloudFCN: Accurate and robust cloud detection for satellite imagery with deep learning. *Remote Sensing*, 11(19):2312, 2019. doi:10.3390/rs11192312.

[14] S. Ji, P. Dai, M. Lu, and Y. Zhang. Simultaneous cloud detection and removal from bitemporal remote sensing images using cascade convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1):732–748, 2020. doi:10.1109/TGRS.2020.2994349.

[15] Y. Zi, F. Xie, N. Zhang, Z. Jiang, W. Zhu, and H. Zhang. Thin cloud removal for multispectral remote sensing images using convolutional neural networks combined with an imaging model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:3811–3823, 2021. doi:10.1109/JSTARS.2021.3068166.

[16] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei. Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network. IEEE Transactions on Geoscience and Remote Sensing, 56(8):4274–4288, 2018. doi:10.1109/TGRS.2018.2810208.

[17] S. R. Surya and P. Simon. Automatic cloud detection using spectral rationing and fuzzy clustering. *Proc. 2013 2nd International Conference on Advanced Computing, Networking and Security*, Mangalore, India, 2013, pp. 90–95. doi:10.1109/ADCONS.2013.44.

[18] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, and K. Li. CDnet: CNN-based cloud detection for remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing, 57(8):6195–6211, 2019. doi:10.1109/TGRS.2019.2904868.

[19] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. Remote sensing of environment, 225):307–316, 2019. doi:10.1016/j.rse.2019.03.007.

[20] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. Remote sensing of environment, 229:247–259, 2019. doi:https://doi.org/10.1016/j.rse.2019.03.039.

[21] M. Shi, F. Xie, Y. Zi, and J. Yin. Cloud detection of remote sensing images by deep learning. *Proc. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Beijing, China, 2016, pp. 701–704. doi:10.1109/IGARSS.2016.7729176.

[22] Z. Wang, Z. Zhang, L. Dong, and G. Xu. Jitter detection and image restoration based on generative adversarial networks in satellite images. *Sensors*, 21(14):4693, 2021. doi:10.3390/s21144693.

[23] F. Xie, M. Shi, Z. Shi, J. Yin, and D. Zhao. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3631–3640, 2017. doi:10.1109/JSTARS.2017.2686488.

[24] CNES – The National Center for Space Studies. `https://cnes.fr/en/` (accessed Apr 09, 2022).

[25] COPERNICUS Programme. Sentinel Online. `https://sentinels.copernicus.eu/web/sentinel/` (accessed Apr 09, 2022).

[26] EUMETSAT – European Organisation for the Exploitation of Meteorological Satellites. Meteosat series. `https://www.eumetsat.int/our-satellites/meteosat-series` (accessed Apr 09, 2022).

[27] M. Le Goff, J.-Y. Tourneret, H. Wendt, M. Ortner, and M. Spigai. Deep learning for cloud detection. *Proc. 8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, Madrid, 2017, pp. 1–6, 2017. doi:10.1049/cp.2017.0139.

[28] N. Champion. Automatic cloud detection from multi-temporal satellite images: Towards the use of pléiades time series. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 24(B3:559–564, 2012. doi:10.5194/isprsarchives-XXXIX-B3-559-2012.

[29] Q. Zhang, Q. Yuan, J. Li, Z. Li, H. Shen, and L. Zhang. Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:148–160, 2020. doi:10.1016/j.isprsjprs.2020.02.008.

[30] L. Wang, Y. Chen, L. Tang, R. Fan, and Y. Yao. Object-based convolutional neural networks for cloud and snow detection in high-resolution multispectral imagers. Water, 10(11):1666, 2018. doi:10.3390/w10111666.

[31] P. Ebel, Y. Xu, M. Schmitt, and X. Zhu. SEN12MS-CR-TS: A remote sensing data set for multimodal multi-temporal cloud removal. *IEEE Transactions on Geoscience and Remote Sensing*, 60:5222414, 2022. doi:10.1109/TGRS.2022.3146246.

[32] K. Kasturirangan and V. Jayaraman. Indian Remote Sensing Satellite, IRS-1A. A forerunner for operational era. `https://www.isro.gov.in/Indian_Remote_Sensing_Satellite_1A.html` (accessed Apr 09, 2022).

[33] Y. Chen, R. Fan, M. Bilal, X. Yang, J. Wang, and W. Li. Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks. *ISPRS International Journal of Geo-Information*, 7(5):181, 2018. doi:10.3390/ijgi7050181.

[34] Z. Yan et al. Cloud and cloud shadow detection using multilevel feature fused segmentation network. *IEEE Geoscience and Remote Sensing Letters*, 15(10):1600–1604, 2018. doi:10.1109/LGRS.2018.2846802.

[35] T. Bai, D. Li, K. Sun, Y. Chen, and W. Li. Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion. *Remote Sensing*, 8(9):715, 2016. doi:10.3390/rs8090715.

[36] B. Li et al. Stripe noise detection of high-resolution remote sensing images using deep learning method. *Remote Sensing*, 14(4):873, 2022. doi:10.3390/rs14040873.

[37] Y. Chen et al. Cloud and cloud shadow detection based on multiscale 3D-CNN for high resolution multispectral imagery. *IEEE Access*, 8:16505–16516, 2020. doi:10.1109/ACCESS.2020.2967590.

[38] Y. Chen, L. Tang, X. Yang, R. Fan, M. Bilal, and Q. Li. Thick clouds removal from multitemporal ZY-3 satellite images using deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:143–153, 2019. doi:10.1109/JSTARS.2019.2954130.

[39] B. Jiang et al. Deep Dehazing Network for Remote Sensing Image with Non-Uniform Haze. *Remote Sensing*, 13(21):4443, 2021. doi:10.3390/rs13214443.

[40] T. Julliand, V. Nozick, and H. Talbot. Image noise and digital image forensics. *Proc. International Workshop on Digital-Forensics and Watermarking*, Tokyo, Japan, Oct 7-10, 2015. Lecture Notes in Computer Science, vol. 9569, pp. 3–17. doi:10.1007/978-3-319-31960-5_1.

[41] K.-Y. Lee and J.-Y. Sim. (2019). Cloud removal of satellite images using convolutional neural network with reliable cloudy image synthesis model. *Proc. 2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019, pp. 3581-3585. doi:10.1109/ICIP.2019.8803666.

[42] S. Shao, Y. Guo, Z. Zhang, and H. Yuan. Single remote sensing multispectral image dehazing based on a learning framework. *Mathematical Problems in Engineering*, 2019:4131378, 2019. doi:10.1155/2019/4131378.

[43] Y. Chang, L. Yan, H. Fang, S. Zhong, and W. Liao. HSI-DeNet: Hyperspectral image restoration via convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):667–682, 2019. doi:10.1109/TGRS.2018.2859203.

[44] H. Zeng, X. Xie, H. Cui, Y. Zhao, and J. Ning. Hyperspectral image restoration via CNN denoiser prior regularized low-rank tensor recovery. *Computer Vision and Image Understanding*, 197-198:103004, 2020. doi:10.1016/j.cviu.2020.103004.

[45] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Automatic shadow detection and removal from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):431–446, 2015. doi:10.1109/TPAMI.2015.2462355.

[46] J. Tang et al.. RestoreNet-Plus: Image restoration via deep learning in optical synthetic aperture imaging system. Optics and Lasers in Engineering, 146:106707, 2021. doi:10.1016/j.optlaseng.2021.106707.

[47] A. Tafsast, M. L. Hadjili, H. Hafdaoui, A. Bouakaz and N. Benoudjit. Automatic Gaussian mixture model (GMM) for segmenting 18F-FDG-PET images based on Akaike information criteria. *Proc. 2015 4th International Conference on Electrical Engineering (ICEE)*, Boumerdes, Algeria, 2015, pp. 1–4, 2015. doi:10.1109/INTEE.2015.7416845.

[48] S. Mahajan and B. Fataniya. Cloud detection methodologies: Variants and development—A review. *Complex & Intelligent Systems*, 6:251–261, 2020. doi:10.1007/s40747-019-00128-0.

[49] IBM. What is AI? `https://www.ibm.com/topics/artificial-intelligence` (accessed Apr 09, 2022).

[50] I. El Naqa, and M. J. Murphy. What Are Machine and Deep Learning?. In: El Naqa, I., Murphy, M.J. (eds.) *Machine and Deep Learning in Oncology, Medical Physics and Radiology*. Springer, Cham, 2022, pp. 3–15. doi:10.1007/978-3-030-83047-2_1.

[51] J. Yang, Q. Min, W. Lu, Y. Ma, W. Yao, T. Lu, J. Du, and G. Liu. A total sky cloud detection method using real clear sky background. *Atmospheric Measurement Techniques*, 9(2):587–597, 2016. doi:10.5194/amt-9-587-2016.

[52] B. Zhong, W. Chen, S. Wu, L. Hu, X. Luo and Q. Liu. A cloud detection method based on relationship between objects of cloud and cloud-shadow for chinese moderate to high resolution satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(11):4898-4908, 2017. doi:10.1109/JSTARS.2017.2734912.

[53] J. Yang, W. Lu, Y. Ma, and W. Yao. An automated cirrus cloud detection method for a ground-based cloud image. *Journal of Atmospheric and Oceanic Technology*, 29:527–537, 2012. doi:10.1175/JTECH-D-11-00002.1.

[54] H. Zhai, H. Zhang, L. Zhang, and P. Li. Cloud/shadow detection based on spectral indices for multi/hyperspectral optical remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:235–253, 2018. doi:10.1016/j.isprsjprs.2018.07.006.

[55] H. Shang, H. Letu, Z. Peng, and Z. Wang. Development of a daytime cloud and aerosol loadings detection algorithm for himawari-8 satellite measurements over desert. *Proc. ISPRS Workshop on Remote Sensing and Synergic Analysis on Atmospheric Environment*, 7–8 Nov 2018, Guangzhou, China. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-3/W5, pp. 61–66, 2018. doi:10.5194/isprs-archives-XLII-3-W5-61-2018.

[56] D. Frantz, E. Ha, A. Uhl, J. Stofels, and J. Hill. Improvement of the Fmask algorithm for Sentinel-2 images: separating clouds from bright surfaces based on parallax efects. *Remote Sensing of Environment*, 215:471–481, 2018. doi:10.1016/j.rse.2018.04.046.

[57] P. Wang, H. Zhang and V. M. Patel. SAR image despeckling using a Convolutional Neural Network. *IEEE Signal Processing Letters*, 24(12):1763–1767, 2017. doi:10.1109/LSP.2017.2758203.