On the Use of Convolutional Neural Networks with Patterned Stride for Medical Image Analysis

Luiz Zaniolo, Oge Marques
Florida Atlantic University, Boca Raton, FL (USA)

Abstract. The use of deep learning techniques for early and accurate medical image diagnosis has grown significantly in recent years, with some encouraging results across many medical specialties, pathologies, and image types. One of the most popular deep neural network architectures is the convolutional neural network (CNN), widely used for medical image classification and segmentation, among other tasks. One of the configuration parameters of a CNN is called *stride* and it regulates how sparsely the image is sampled during the convolutional process. This paper explores the idea of applying a patterned stride strategy: pixels closer to the center are processed with a smaller stride concentrating the amount of information sampled, and pixels away from the center are processed with larger strides consequently making those areas to be sampled more sparsely. We apply this method to different medical image classification tasks and demonstrate experimentally how the proposed patterned stride mechanism outperforms a baseline solution with the same computational cost (processing and memory). We also discuss the relevance and potential future extensions of the proposed method.

Key words: convolutional neural networks, patterned stride, medical image classification, deep learning.

1. Introduction

The use of deep learning architectures for medical image analysis has experienced significant growth in recent years [12], with impressive success stories and claims of superhuman performance across many tasks, image modalities, and diseases.

Convolutional Neural Networks (CNNs) are the most popular architecture for medical image classification tasks. CNN architectures have an end-to-end structure, which learn high-level representations from raw data [11] without the need for pre-selecting features relevant to that type of data. Ever since their initial success on ImageNet Large-Scale Visual Recognition Challenge (LSVRC) 2012 [10], CNNs have been extended to specialized image classification tasks.

The performance of a CNN can be improved by fine-tuning some of its parameters (e.g., number of layers, type of layers, weights, biases) and hyperparameters (e.g., learning rate, number of epochs, loss function, activation functions). One of the CNN parameters that can be specified by the designer of the network is called *stride*: it specifies how the filters in a convolutional layer convolve around the input volume. Larger stride values mean that the network will perform more sparse convolutions – and consequently

a smaller number of them – throughout the processing of the image. Smaller stride values mean more concentrated sampling and consequently larger number of convolutions performed. The majority of existing CNNs implement a fixed value of stride, usually between 1 and 3. The minimum stride value of 1 means that all pixels are processed, which sets an upper bound on the computational cost of the convolution operations.

This paper explores the idea of changing the stride value in CNNs depending on the position of the pixel within the image: a smaller stride value is used when processing the center of the image, while a larger one is used for pixels close to the edges, according to a predefined sampling pattern.

Guo et al. [6] proposed another method where image classification benefits from using a patterned stride. However, their algorithm addresses the image classification problem in a different way, by judging the image complexity based on extracted features, and using that result to decide the stride value, suggesting smaller stride values for complex images and larger strides for simple ones.

The proposed method is experimentally evaluated on three medical image analysis tasks – (i) skin lesion classification, (ii) brain tumor detection, and (iii) image modality classification – and compared against a baseline fixed stride approach that requires similar computational power for both training and inference phases.

2. Materials and Methods

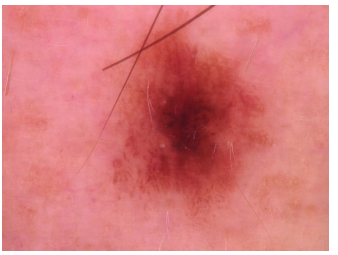
2.1. Datasets

In this study, we use three datasets: the HAM 10000 dataset used for skin lesion classification tasks; the Brain Tumor dataset, which contains images with healthy and unhealthy magnetic resonance imaging (MRI) brain slices; and the MedNIST dataset for image modality classification.

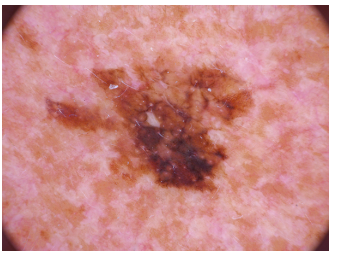
2.1.1. The HAM 10000 Dataset

The HAM10000 (*Human Against Machine with 10000 training images*) dataset [15] is a large collection of labeled multi-source dermatoscopic images in RGB color space, manually classified into one of seven different classes: *Melanocytic nevi*, *Melanoma*, *Benign keratosis-like lesions*, *Basal cell carcinoma*, *Actinic keratoses*, *Vascular lesions*, and *Dermatofibroma* (Figure 1).

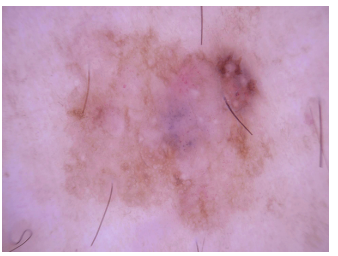
Training, validation, and test sets are available at the 2018 International Skin Imaging Collaboration (ISIC) challenge archive website [9] (see [2,15] for more information), which also includes a live challenge submission option, for continuous evaluation of automated classifiers using the dataset.



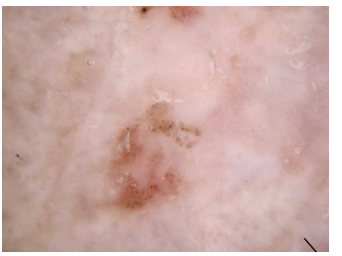
(a) Melanocytic nevi



(b) Melanoma

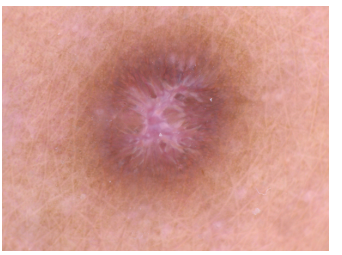


(c) Benign keratosis-like lesions

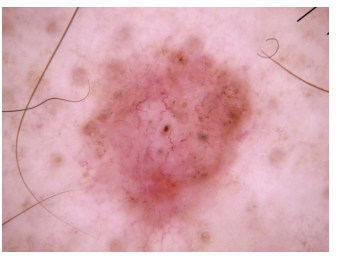


(d) Actinic keratoses

(e) Vascular lesions



(f) Dermatofibroma



(g) Basal cell carcinoma

Fig. 1: HAM 10000 dataset: examples of representative images for each class.

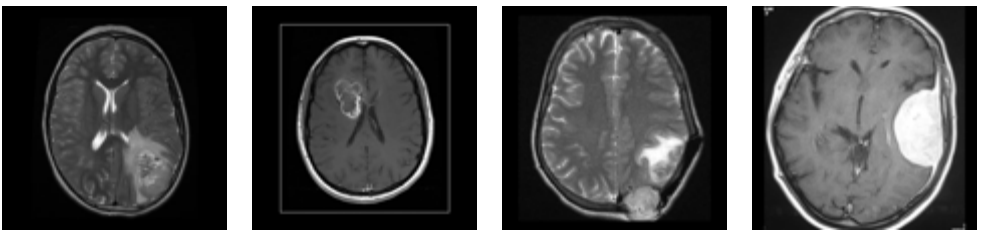


Fig. 2: Examples of tumorous brains.

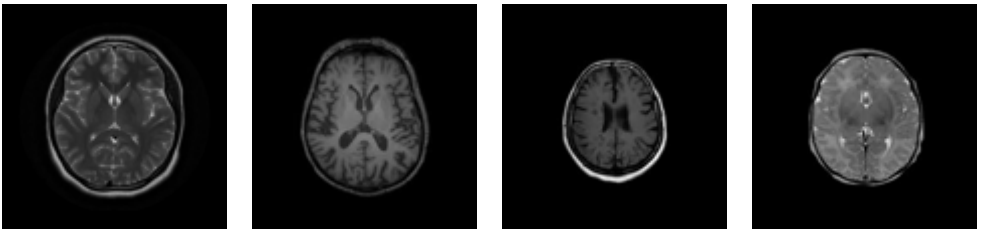


Fig. 3: Examples of healthy brains.

2.1.2. The Brain Tumor Dataset

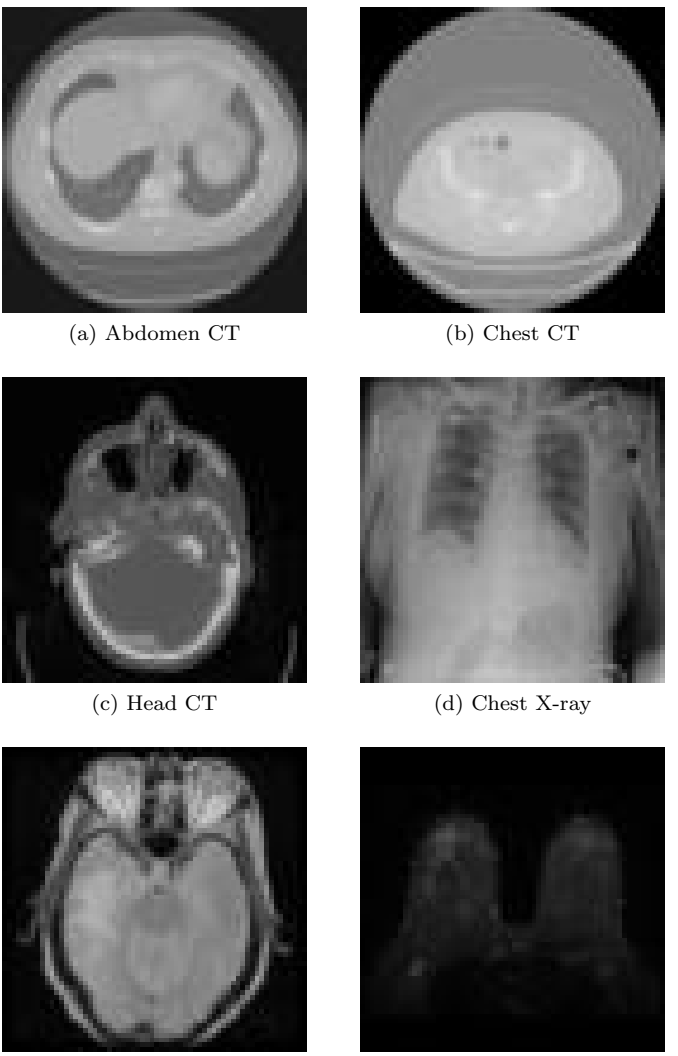
The Brain Tumor dataset [1] contains a total of 253 grayscale Brain MRI slices: 155 images that exhibit tumor and 98 slices that do not (Figures 2 and 3).

This dataset has been published on Kaggle [5] (arguably the world's most famous machine learning and data science community) and used in several experiments using different network architectures to test performance and accuracy in classification tasks.

2.1.3. The MedNIST Dataset

The MedNIST dataset [4] is a collection of grayscale medical images categorized into six different classes: *Abdomen computed tomography (CT)*, *Chest CT*, *Head CT*, *Chest X-ray*, *Brain magnetic resonance (MR)*, and *Breast MR* (Figure 4).

This dataset was created with the purpose of teaching basic deep learning concepts. In a follow up work [3], the dataset was used to demonstrate how to optimize a simple image classifier, thereby guiding researchers in the process of building an environment to execute a complete deep learning application, understanding the deep learning workflow, and focusing more specifically on parameters adjustments and their influence on the overall performance.



(a) Abdomen CT (b) Chest CT

(c) Head CT (d) Chest X-ray

(e) Brain MR (f) Breast MR

Fig. 4: MedNIST dataset: examples of representative images for each class.

2.2. CNN Architectures

CNNs have demonstrated excellent performance in image classification tasks. Classical examples of successful networks include AlexNet [10] and VggNet [14]. Modern CNN architectures are deeper and use more complex connections among layers, such as ResNet [7] and DenseNet [8].

When an input image is processed at the convolutional layer, the image goes through a series of convolution operations taking as input different pixels of the image. These operations are performed using masks (usually 3×3 or 5×5) in a similar way to spatial filters in classical image processing, except for two significant differences: (i) the masks' coefficients (weights) are learned by training the CNN (rather than fixed by the specific image processing technique); and (ii) the number of pixels (known as *stride*) by which a mask is shifted after performing the convolution in a certain portion of the image can be chosen by the CNN designer.

The choice of stride value in CNNs impacts the number of computations (additions and multiplications) required to process each image: smaller strides require more computation than larger strides. Typical stride values are 1, 2, and 3. A stride of 1 means that all pixels form the input image will be processed, which sets an upper bound on the computational cost of the convolution operations.

In this paper we extend the ideas first presented in [16] to explore and test the idea of varying the stride between 1 and 3 depending on the relative position of the pixels to be used in the convolution operation within the image: pixels that are closer to the center of the image will be processed using a smaller stride whereas pixels closer to the edges of the images are processed with a larger stride.

Figure 5 illustrates the process using a generic input image whose size is 15×15[1], where the blue pixels are pre-selected for the convolution operation. The resulting effect is the assignment of corresponding stride values as follows:

- pixels 1 to 5: stride 3,
- pixels 5 to 7: stride 2,
- pixels 7 to 9: stride 1,
- pixels 9 to 11: stride 2,
- pixels 11 to 15: stride 3.

Medical images sizes are usually larger, therefore the original strategy has to be expanded to accommodate larger images. It is easy to imagine how the process would scale up. The sampling process has to be done in a way where there is a given percentage of the pixels that would be sampled with stride 1, and others with strides 2 and 3. This has to be arranged in a way as to sample half of the pixels in each dimension. For

[1] In our experiments the same basic idea was adapted and implemented with different image sizes, depending on the dataset used.

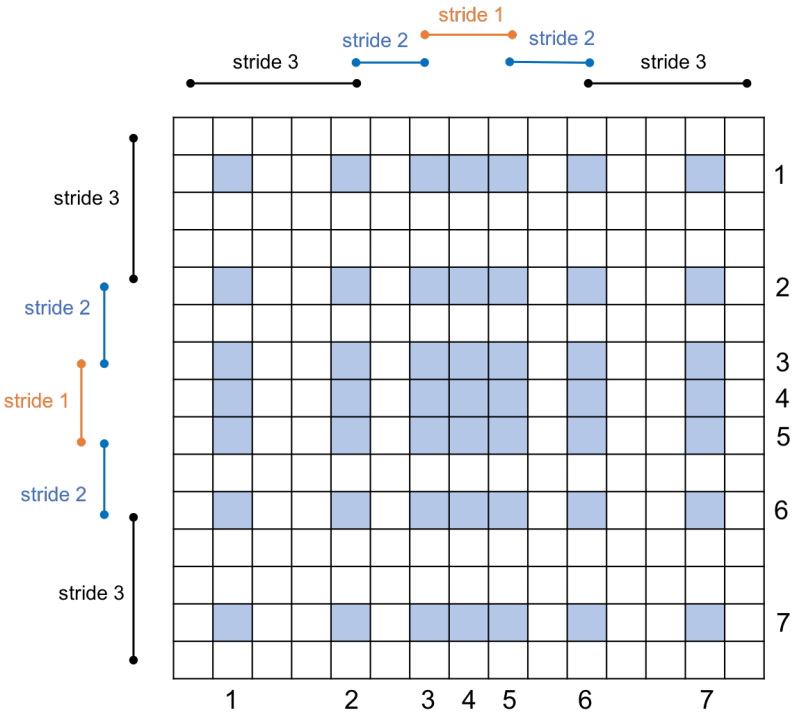


Fig. 5: Simple example of the proposed patterned stride mechanism for a generic image of 15×15 pixels.

example, in the MedNIST dataset all images are 64×64 pixels. The stride arrangement used for both dimensions in this dataset was as follows:

- pixels 1 to 8: stride 3,
- pixels 8 to 30: stride 2,
- pixels 30 to 35: stride 1,
- pixels 35 to 57: stride 2,
- pixels 57 to 64: stride 3.

In [16], the authors formalized their assumptions about the advantages of the patterned stride approach with two complementary testable hypotheses:

- H1: The use of patterned stride in images whose main contents are in the **central** portion of the image will lead to **improved** performance (when compared to the baseline case of comparable computational complexity, i.e., fixed stride = 2).

- H2: The use of patterned stride in images whose main contents are **not** in the **central** portion of the image will lead to **decreased** performance (when compared to the baseline case of comparable computational complexity, i.e., fixed stride = 2).

In this paper we focus on new examples of successful use of patterned stride, introduce and explain three new hyperparameters (P_1, P_2, P_3), and present image score calculations to provide a rough estimate of how effective the patterned stride mechanism can be in different datasets.

2.3. Hyperparameters P_1, P_2, P_3

One of the assumptions that are needed to compare this new method against fixed stride 2 is to create a model which has similar computational and memory requirements. In order for this to happen, the first layer has to reduce the sampling points by half. The sampling points can be spread in different ways throughout the image. The proposed method postulates a concentrated sampling in the center and sparse at the edges, varying the stride from 1 to 3 in different parts of the images as seen in figure 5. However, the number of pixels in each area (stride 1, 2 or 3) can vary while still keeping the same total number of sampling points. Figure 6 shows an example where the stride 2 area is enlarged compared to strides 1 and 3. We propose the introduction of three new hyperparameters, that will define the percentage of sampling points in each stride region, P_1, P_2 and P_3 which are the percentage of total sampling points in each corresponding stride region (1, 2 and 3) respectively.

Since we still need to comply with the requirement of having half of the sampling points in the first convectional layer, we have to guarantee that the number of sampling points on stride regions 1 and 3 are the same. Another intuitive way to explain this is that any increase in a more concentrated sampling area (stride 1) has to be compensated by an increase of a sparse area (stride 3).

Therefore the choice of P_1, P_2, and P_3 is bound by the following conditions:

$$P_1 = P_3 \ ,$$

and

$$P_1 + P_2 + P_3 = 1 \ .$$

The optimal values used for P_1, P_2, and P_3 for each experiment will be described in Section 3.

2.4. Image Score

To guide the selection of hyperparameters P_1, P_2, and P_3, we created a new method to calculate the input images' score. The objective of the score is to calculate how the meaningful part of the image frame is concentrated in the center. Image scores

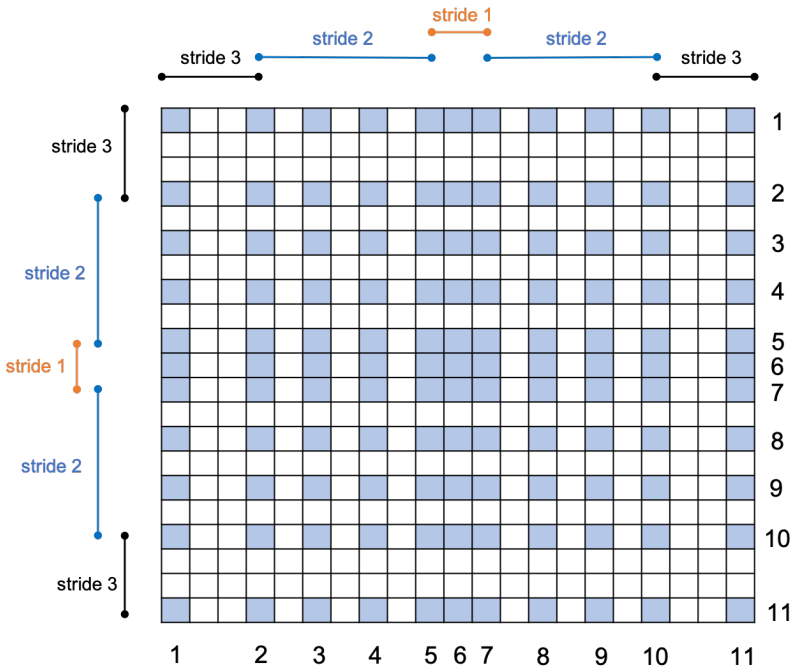


Fig. 6: Simple example of variation of stride areas for a generic image of 21×21 pixels.

range between 0 and 1, where 0 is an image where all the information is completely concentrated in the center and 1 means the opposite. Figure 7 shows an example of image score calculation using a quadrilateral bounding polygon, where the meaningful part of the image is shown in yellow.

The score is calculated as:

$$S = \frac{1}{2}\left(\frac{x_c + y_c}{2} + \sum_{k=1}^{s} \frac{x_k + y_k}{2n}\right) ,$$

where n is the number of sides of the bounding polygon and x_c, y_c, x_k, and y_k are the partial scores of each individual point calculated as follows:

$$x_c = \frac{|p_{cx} - i_{cx}|}{d_x} ,$$

$$y_c = \frac{|p_{cy} - i_{cy}|}{d_y} ,$$

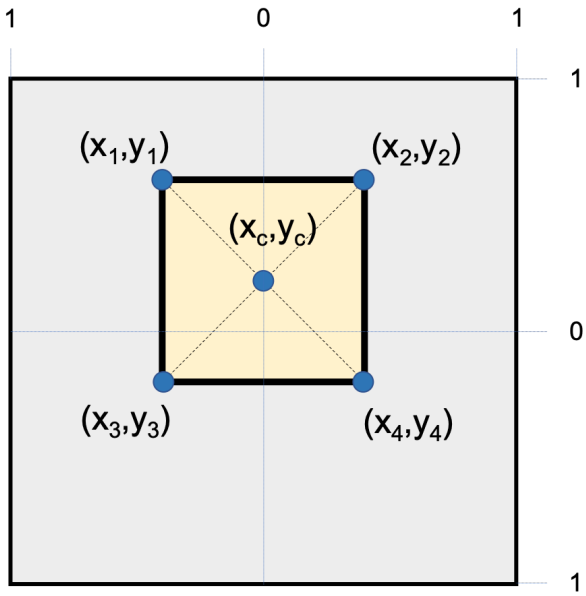


Fig. 7: Image score bounding box.

$$x_k = \frac{|p_{kx} - i_{cx}|}{d_x} \ ,$$

$$y_k = \frac{|p_{ky} - i_{cy}|}{d_y} \ ,$$

where

p_{cx} – x coordinate of bounding polygon center,
p_{cy} – y coordinate of bounding polygon center,
p_{kx} – x coordinate of k^{th} point of the bounding polygon,
p_{ky} – y coordinate of k^{th} point of the bounding polygon,
i_{cx} – x coordinate of frame center,
i_{cy} – y coordinate of frame center,
d_X – image width in pixels,
d_Y – image height in pixels.

This method was used to calculate the scores for the datasets used in the experiments and will be reported together with the results of each experiment in Section 3.

3. Experiments and Results

This section describes in detail the experiments performed to test the performance of the patterned stride mechanism in a simple CNN (Figure 8), whose layers are described below:

1. Convolutional 2D layer: this is the layer in which we modify the stride parameter,
2. Batch Normalization layer,
3. ReLu layer,
4. Max Pooling layer,
5. Convolutional 2D layer,
6. Batch Normalization layer,
7. ReLu layer,
8. Max Pooling layer,
9. Convolutional 2D layer,
10. Batch Normalization layer,
11. ReLu layer,
12. Fully connected layer,
13. Softmax layer,
14. Classification layer[2].

3.1. Experimental setups

All experiments were performed using MATLAB. Since MATLAB doesn't provide an option to select variable strides for convolutional layers, some modifications in the MATLAB source code were needed in other to perform the desired operation.

The train/test split was 95/5, i.e., 95% of the images were used for training/validation and 5% for tests. Categorical cross-entropy loss and stochastic gradient descent with momentum (SGDM) optimizer were used.

Network parameter values were experimentally selected to achieve the best performance for each task. For the HAM10000 dataset, the network was trained using 20 epochs and a decaying learning rate (LR), starting with 0.0003 and reducing it by half every 2 epochs. For the Brain Tumor dataset, we used 40 epochs and a LR of 0.0003. For the MedNIST dataset, we used 8 epochs and a fixed LR of 0.0003.

3.2. Results

Test runs were performed in four different network configurations, where the stride parameter was set to 1, 2, 3 or *patterned*, depending on the run. Final results for the

[2]The number of nodes in this layer will vary depending on the experiment according to the number of classes.

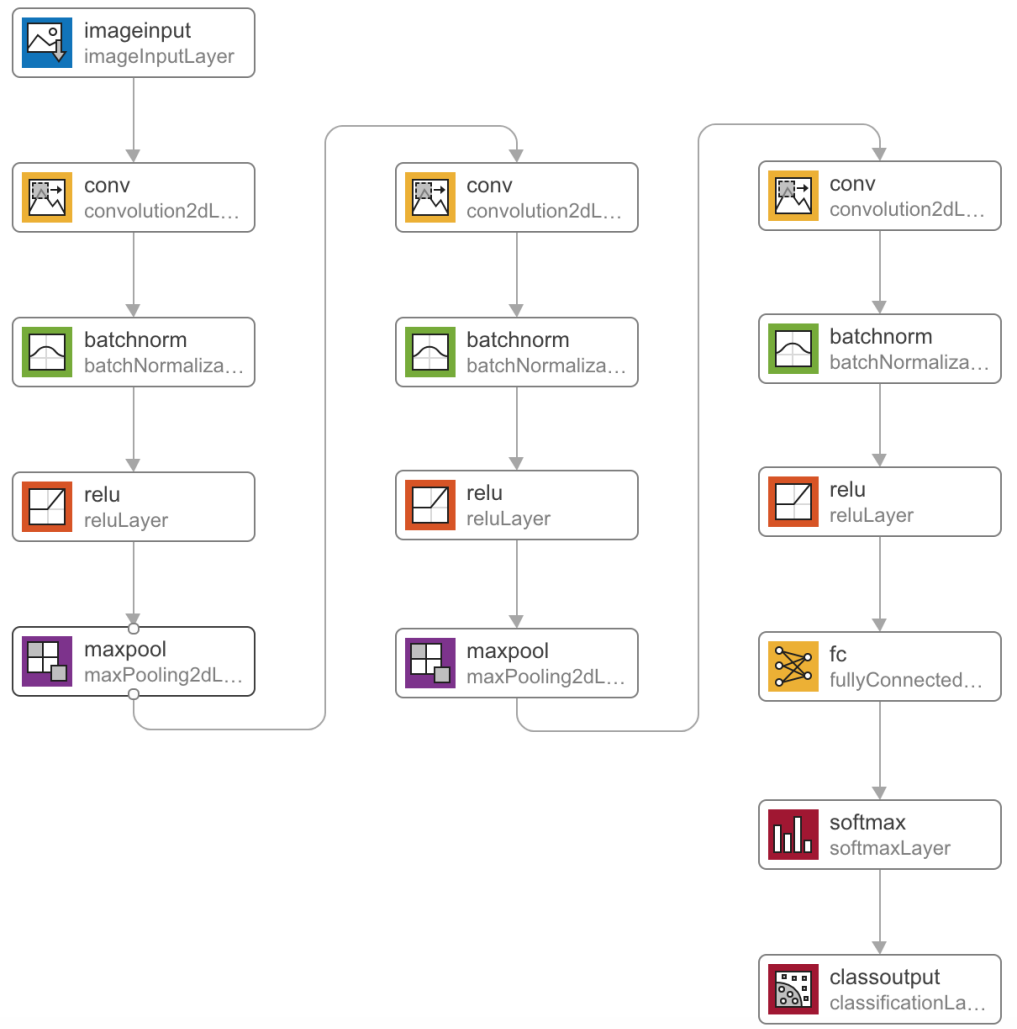


Fig. 8: CNN architecture used for the classification experiments.

Tab. 1: Skin lesion classification using the HAM10000 dataset and patterned stride: classification accuracy results and time measurements.

Stride	Accuracy	Training Time	Inference Time
1	77.0 %	258 min	31.9 ms
2	76.4 %	105 min	10.9 ms
3	75.5 %	70 min	5.8 ms
Patterned	78.1 %	105 min	10.9 ms

Tab. 2: Brain tumor detection using the Brain Tumor dataset and patterned stride: classification accuracy results and time measurements.

Stride	Accuracy	Training Time	Inference Time
1	89.6 %	12.5 min	50.6 ms
2	84.0 %	4.2 min	15.3 ms
3	81.6 %	2.4 min	7.4 ms
Patterned	87.2 %	4.2 min	15.3 ms

HAM10000 dataset, Brain tumor dataset, and MedNIST dataset are shown in Tables 1, 2, and 3, respectively. In each table the respective classifier's accuracy and the elapsed time for training and inference for each case are reported. The tables confirm the computational cost for patterned stride is comparable to the cost for stride 2.

We performed tests with different values of hyperparameters P_1, P_2, and P_3 and reported results for optimal parameter selection. The optimal values for each dataset are shown in Table 4.

Tab. 3: Image modality classification using the MedNIST dataset and patterned stride: classification accuracy results and time measurements.

Stride	Accuracy	Training Time	Inference Time
1	99.8 %	16.5 min	18.6 ms
2	98.3 %	5.0 min	6.1 ms
3	95.9 %	2.4 min	3.5 ms
Patterned	98.9 %	5.0 min	6.1 ms

Tab. 4: Optimal values for $P1$, P_2, and P_3 of each dataset.

Dataset	P_1	P_2	P_3
HAM10000	0.24	0.52	0.24
Brain Tumor	0.46	0.08	0.46
MedNIST	0.16	0.68	0.16

4. Discussion

Experimental results on three different classification tasks, in three different datasets, have confirmed the hypothesis that the proposed patterned stride mechanism outperforms the fixed stride options (with stride equal to 2 or 3) in all test cases.

For the HAM10000 dataset, the results were even better, with an accuracy even higher than the stride 1 configuration (at a fraction of the computational cost). This is consistent with the fact that the most informative portion of the skin lesion images in the HAM10000 dataset is usually centered (and the surrounding area contains very little information), which is the optimal case for the proposed patterned stride scheme.

On a related note, figures 9 and 10 show the confusion matrices for the case of patterned stride and stride 2 in the MedNIST classification task. In Figure 9, which displays the results for patterned stride, one can see that two of the images from the brain MR category were misclassified, one as head CT and the other as breast MR. In Figure 10, which displays the confusion matrix for stride 2, one can observe that there were two instances of misclassification between the CTHead and MRBrain classes, which did not happen for the patterned stride case (Figure 9). This result is particularly interesting because for these two classes, since the general brain shape is the same, the differences had to be found in internal image elements. This confirms that the patterned stride has a better performance when the central area of the image contains the most informative pixels within the image.

Even though the patterned stride mechanism delivered a better classification accuracy in the MedNIST classification task, there were some cases where it classified an image incorrectly. One example is shown in figure 11, where an image from the brain magnetic resonance class was classified as breast magnetic resonance. In this case, the input image was very dark and the key to classify it correctly lies on the observation of the external shape where the breast magnetic resonance has some distinguishing factors. Since the patterned stride network concentrates its effort in central places, it wasn't able to get the necessary nuances of the image for a correct classification.

Another interesting observation is the relationship between image scores and hyperparameters P_1, P_2, P_3. Table 5 shows the comparison of image scores for each dataset with the optimal values used for the parameters and indicates that larger scores need

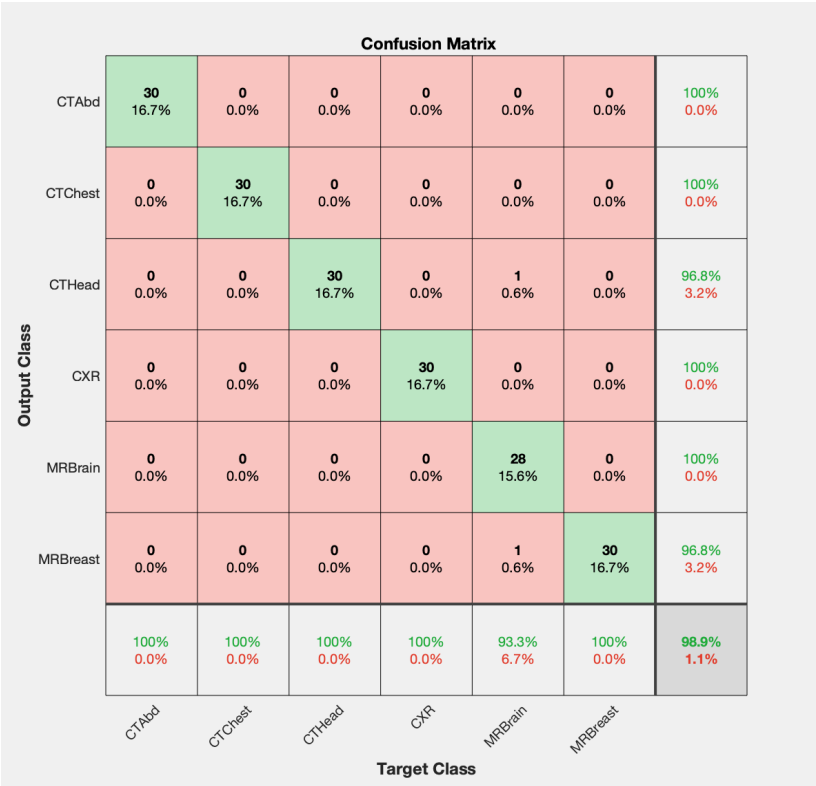


Fig. 9: Confusion matrix for the patterned stride case using the MedNIST dataset.

Tab. 5: Comparison between image scores and hyperparameters P_1, P_2, and P_3 for each dataset.

Dataset	Image Score	P_1	P_2	P_3
Brain Tumor	0.29	0.46	0.08	0.46
HAM10000	0.35	0.24	0.52	0.24
MedNIST	0.44	0.16	0.68	0.16

to use smaller values for P_1 and P_3 and larger values for P_2. These results confirm the usefulness of the image scores as a preliminary estimate of the effectiveness of the patterned stride mechanism across different datasets.

The proposed method has its limitations, more notably: (i) it does not provide better

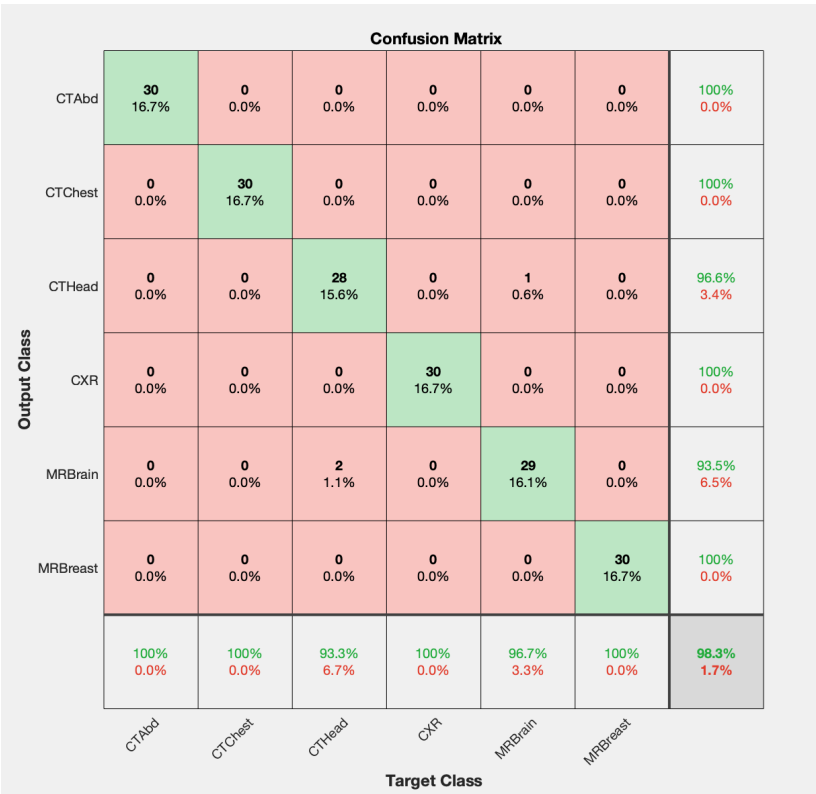


Fig. 10: Confusion matrix for the stride 2 case using the MedNIST dataset.

results than the fixed stride 2 baseline in cases where the informational content of an image is spread throughout the image (as opposed to concentrated on its center)[3]; and (ii) it does not work for cases where most of the useful information is concentrated in a small portion of the image, but away from its center.

To address (i), a simple two-class classifier could be used to determine if a dataset (and associated classification problem) is suitable for the patterned stride mechanism or not. Such classifier could use an easy-to-compute measure of image complexity or homogeneity (e.g., entropy) as its main feature. In its simplest form, if the entropy is higher than a certain threshold, the dataset is not a good candidate for the patterned stride approach. More sophisticated features and classifiers, of course, could be used.

A potential solution to (ii) could be the use of alternative sampling patterns (see

[3]See counterexamples in [16] that support hypothesis H2.

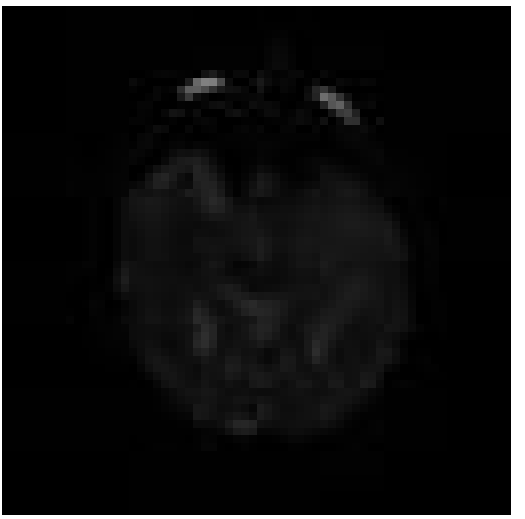


Fig. 11: Brain magnetic resonance classified as breast magnetic resonance by the patterned stride network.

Figure 12 for examples of five different predefined patterns[4] for *centered* as well as *top-right*, *top-left*, *bottom-right*, and *bottom-left* cases) and a selector algorithm that could be used to determine – for each individual image in the dataset – which of the (five) patterns is most appropriate. Such algorithm could use any type of region-of-interest (ROI) detection scheme, e.g., off-the-shelf face detectors for images involving a face, or ROI computation using saliency maps [13] for cases where the object of interest is also the most salient in the scene.

5. Conclusion

We have extended a method for implementing a patterned stride mechanism in CNNs and successfully demonstrated experimentally that the use of patterned stride leads to higher accuracy than a fixed stride baseline case of same computational complexity in three different medical image classification tasks and datasets.

The proposed approach could be extended to other datasets, more complex CNN architectures, and different tasks using CNNs, such as semantic segmentation and object detection.

[4]The examples show the sampling distribution for a 15×15 image, but can be extrapolated to any image size.

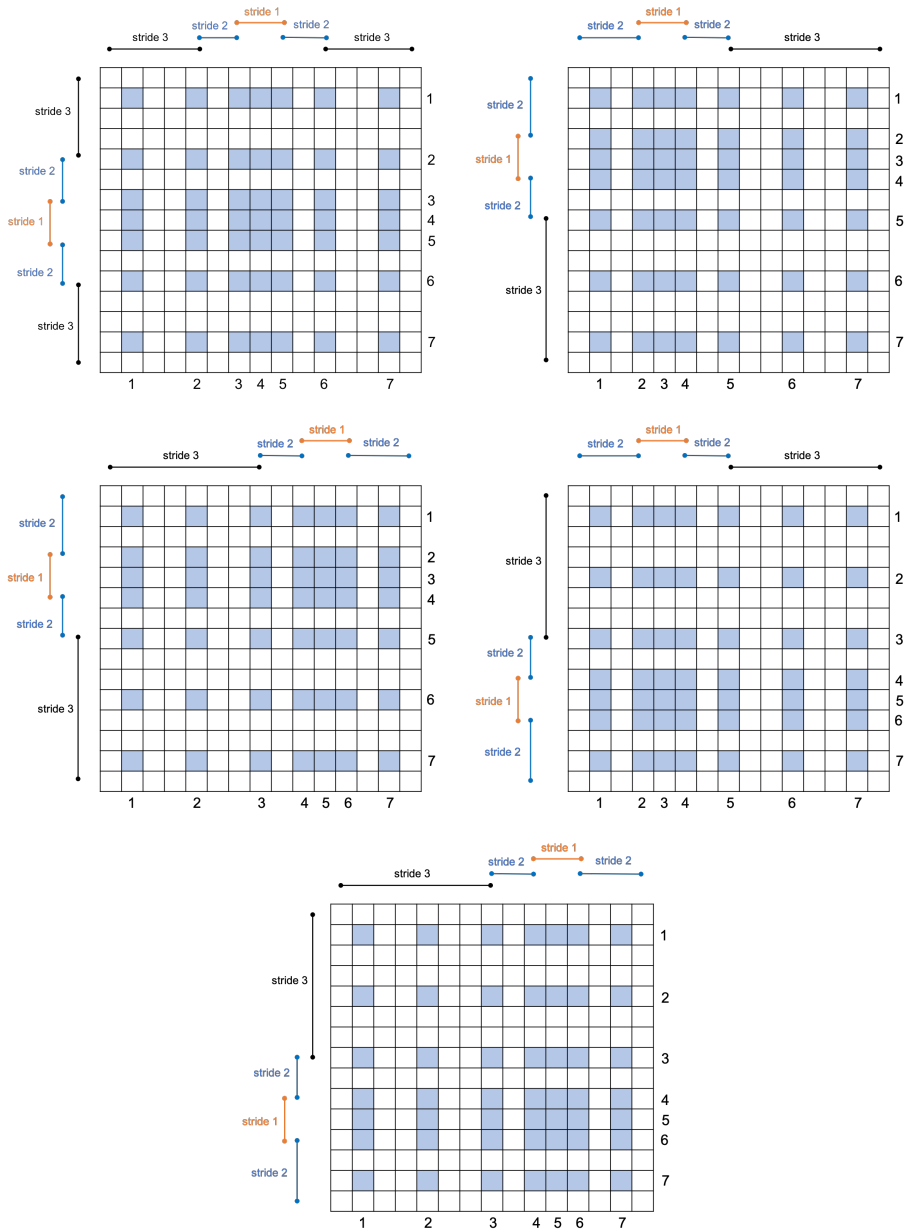


Fig. 12: Adjustable patterned stride sampling patterns.

References

[1] N. Chakrabarty. Brain MRI images for brain tumor detection, 2019. `https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection`. Dataset [accessed Feb 2021].

[2] N. Codella, V. Rotemberg, P. Tschandl, et al. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv preprint*, 2019. arXiv:1902.03368.

[3] B. J. Erickson. Magician's Corner: 2. Optimizing a simple image classifier. *Radiology: Artificial Intelligence*, 1(5):e190113, 2019. doi:10.1148/ryai.2019190113.

[4] B. J. Erickson. Magician's Corner: How to start learning about deep learning. *Radiology: Artificial Intelligence*, 1(4):e190072, 2019. doi:10.1148/ryai.2019190072.

[5] A. Goldbloom, B. Hamner, J. Moser, et al. Kaggle: Your Machine Learning and Data Science Community. `https://www.kaggle.com`. [accessed Feb 2021].

[6] C. Guo, Y.-l. Liu, and X. Jiao. Study on the influence of variable stride scale change on image recognition in CNN. *Multimedia Tools and Applications*, 78(21):30027–30037, 2018. doi:10.1007/s11042-018-6861-0.

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

[8] G. Huang, Z. Liu, L. Van Der Maaten, et al. Densely connected convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2017*, pages 4700–4708, Honolulu, HI, USA, 21-26 Jul 2017. IEEE. doi:10.1109/CVPR.2017.243.

[9] H. Kittler, N. C. F. Codella, M. E. Celebi, et al. ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection. `https://challenge2018.isic-archive.com`.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in neural information processing systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[11] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi:10.1038/nature14539.

[12] G. Litjens, T. Kooi, B. E. Bejnordi, et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017. doi:10.1016/j.media.2017.07.005.

[13] O. Marques, L. M. Mayron, G. B. Borba, and H. R. Gamba. Using visual attention to extract regions of interest in the context of image retrieval. In R. Menezes, editor, *Proc. 44st Ann. Southeast Regional Conf.*, pages 638–643, Melbourne, FL, USA, 10-12 Mar 2006. ACM. doi:10.1145/1185448.1185588.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, 2014. arXiv:1409.1556.

[15] P. Tschandl, C. Rosendahl, and H. Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5:180161, 2018. doi:10.1038/sdata.2018.161.

[16] L. Zaniolo and O. Marques. On the use of variable stride in convolutional neural networks. *Multimedia Tools and Applications*, 79:13581–13598, 2020. doi:10.1007/s11042-019-08385-4.

Luiz Zaniolo received his Master's degree in Computer Engineering from Florida Atlantic University, Boca Raton, FL, USA and is pursuing a Ph.D. degree in Computer Science at the same institution.

Oge Marques received his Ph.D. degree in 2001 from Florida Atlantic University (FAU), Boca Raton, FL, USA. He is Professor of Computer Science and Engineering at FAU since 2001. He is the author of 11 books and more than 120 scholarly publications in the area of Visual Artificial Intelligence. Dr. Marques is a Sigma Xi Distinguished Speaker, a Fellow of the Leshner Leadership Institute of the American Association for the Advancement of Science (AAAS), Tau Beta Pi Eminent Engineer, and a Senior Member of both the IEEE (Institute of Electrical and Electronics Engineers) and the ACM (Association for Computing Machinery).