Enhanced U-Net Model for Accurate Aerial Road Segmentation

Rayene Doghmane[1,*] and Karima Boukari[2,**]
[1] *Laboratory of Automatic and Signals of Annaba (LASA)*
[2] *Laboratory of Study and Research in Instrumentation and Communication of Annaba (LERICA)*
Faculty of Technology, Badji Mokhtar-Annaba University, Annaba, Algeria
[*] *Corresponding author: Rayene Doghmane (rayene.doghmane@univ-annaba.dz)*
[**] *Corresponding author: Karima Boukari (karima.boukari@univ-annaba.dz)*

Abstract In computer vision, Convolutional Neural Networks (CNNs) have become a foundation for image analysis. They excel in tasks such as object recognition, classification, and more, semantic segmentation. In order to achieve better accuracy, it is crucial to apply normalization techniques to the network for enhancing overall performance. This paper introduces an innovative approach that incorporates Batch Group Normalization (BGN) into the popular U-Net for binary semantic segmentation, with a particular focus on aerial road detection. Our research primarily focuses on evaluating the BGN-UNet's performance compared to traditional normalization techniques, such as Batch Normalization (BN) and Group Normalization (GN). With a batch size of 2, the U-Net model enhanced with Batch Group Normalization (BGN-UNet) achieves a remarkable Mean IoU of 98.4% in aerial road segmentation, demonstrating its superior accuracy in this task.

Keywords: image analysis, image recognition, normalization techniques, batch group normalization, semantic segmentation, BGN-UNet, aerial road detection.

1. Introduction

Road extraction proves to be a crucial task in the analysis of remote sensing imagery [4]. It plays a significant role in various aspects of society and the economy. Despite its importance, accurately extracting roads faces challenges due to the presence of non-road objects, and the complexity of the background. These factors contribute to the difficulty of achieving precise road extraction. Addressing these challenges frequently requires the utilization of pixel-wise semantic segmentation to extract road areas accurately (Fig. 1).

Semantic segmentation, a fundamental task in computer vision, involves the classification of individual pixels in an image into distinct object categories, thus aiding in a good comprehension of visual content [12]. In this field, the UNet architecture has proven to be reliable and effective across various applications [27, 28].

Our research focuses on the more recent innovation, Batch Group Normalization (BGN). When integrated into the UNet architecture, BGN exhibits the potential to enhance the accuracy and efficiency of convolutional neural networks for tasks like binary semantic segmentation, with a specific emphasis on road detection.

Figure 2 displays different normalization techniques: Batch Normalization (BN), Layer Normalization (LN), Group Normalization (GN), and Batch-Group Normalization

a **b**

Fig. 1. Binary Semantic Segmentation for Aerial Road Image. (**a**) Aerial Road Image. (**b**) Segmentation for the Aerial Road Image.

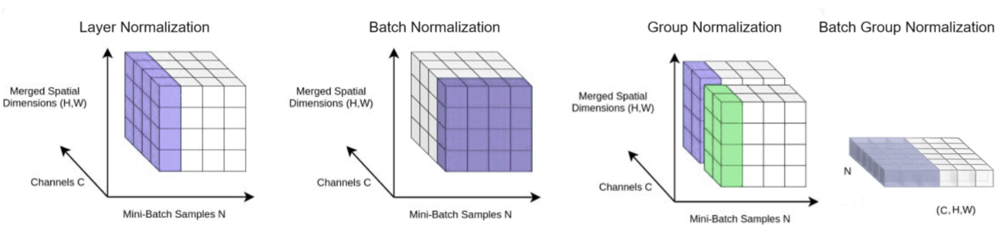


Fig. 2. Normalization Techniques.

(BGN). In each subfigure, you can see a feature map tensor, with axes representing the batch size (N), the number of channels (C), and the spatial dimensions (H, W). The pixels in purple are used to compute the statistics. BGN offers a unique perspective by combining the dimensions of channels, height, and width into a unified dimension and subsequently partitions this new dimension into distinct feature groups. This paper looks at how normalization methods in deep learning have changed over time. It also talks about how using BGN can make UNet better for tasks like semantic segmentation. The upcoming sections will offer comprehensive information on our research methods, the results of our experiments, and the discussions that follow. This will illuminate the power of combining U-Net and BGN in the constantly evolving fields of computer vision and deep learning.

The remainder of this paper is organized as follows. Section 2 reviews related works relevant to our study. Section 3 details the methodology, including the normalization techniques (Subsection 3.1), data preprocessing (Subsection 3.2), and data augmentation (Subsection 3.3) employed in the study. In Section 4, we describe the application of BGN-UNet to aerial road segmentation, with Subsection 4.1 in which the model is described, and Subsection 4.2 focusing on the contributions and novelty of the proposed model. Section 5 presents the results and discussion, along with an ablation study (Subsection 5.1), practical applications of BGN-UNet in road detection (Subsection 5.2) and the challenges and considerations for practical implementation of BGN-UNet (Subsection 5.3). Finally, Section 6 concludes the paper by summarizing the key findings and implications of this work.

2. Related works

A novel end-to-end generative adversarial network was introduced by Zhang et al. [38] to carry out the road extraction task in aerial images. The combination of DCGAN and CGAN was utilized in their model for extracting roads from aerial images, followed by the replacement of deconvolutional layers with FCN. Therefore, the performance of the model is significantly impacted by data from different sources. A deep learning model, called the Recurrent Convolutional Neural Network U-Net (RCNN-UNet), was proposed by Yang et al. [34] for road detection and centerline extraction. It is an end-to-end deep learning model that exploits the spatial context and rich low-level visual features through the design of the RCNN unit. However, this can be computationally intensive and may require significant resources, including computational power and memory.

In addition, a novel deep learning-based convolutional network called VNet model with 2D convolutional kernel to extract road networks from high-resolution remote sensing imagery was introduced by ABOLFAZL et al. [2] a new objective loss function based on cross-entropy and dice loss (CEDL) was used to combine local information and global information, diminish the influence of class imbalance, and improve road segmentation results. However, the use of 2D convolutional kernels and the fully convolutional architecture can lead to significant computational overhead, particularly when processing large datasets. This may result in longer training times and increased resource requirements.

Furthermore, an improved road detection algorithm [10] that integrates Deep Convolutional Neural Networks (CNNs) with a Random Forest classifier has been proposed to enhance the accuracy of analyzing Very High Resolution (VHR) remotely sensed images. However, the performance of the algorithm is highly dependent on the quality and quantity of training data, which can pose a limitation in areas with insufficient labeled datasets.

Recent advancements in road extraction from high-resolution remote sensing images have been marked by the RADANet model [9], which employs a deformable attention mechanism to enhance feature extraction in complex environments, demonstrating superior performance compared to traditional techniques. However, a major limitation is its difficulty in effectively utilizing the spatial relationships and structure of roads, making it challenging to improve extraction accuracy in complex road settings.

According to Shaofu et al. [19] the proposed method MS-AGAN offers an efficient, cost-effective, and reliable approach for dynamically updating road networks using high-resolution remote sensing images. However, despite its improved performance in road extraction, it may still encounter difficulties in complex scenes with high vegetation coverage and occlusions, leading to fragmentation and discontinuities in the extracted road networks.

Moreover, the authors of [18] propose an improved UNet++ network suitable for road extraction from high-resolution remote sensing images. By integrating the CBAM module and incorporating weight information in both channel and spatial dimensions of the feature map, this approach effectively suppresses the network's learning of non-road information, resulting in a more efficient and targeted model.

In summary, the proposed improvement of the UNet network for road extraction from remote sensing images offers notable strengths and weaknesses [30]. It enhances feature extraction through a CNN-Transformer architecture, improving segmentation accuracy with a double upsampling module and a combination of cross-entropy (CE) and Dice loss functions. This results in good training stability, robustness, and generalization across various datasets compared to established models like UNet, PSPNet, DeepLabV3, and TransUNet. However, the algorithm also exhibits high computational complexity and long training times, making it less suitable for mobile or embedded devices. Its resource-intensive nature may limit real-time applications, and the model's complexity could lead to overfitting on smaller datasets. Thus, while the approach shows promise in enhancing road extraction accuracy, its practical applicability is constrained by these limitations, indicating a need for future research into more efficient methods for image semantic segmentation.

Researchers in deep learning are always looking for ways, like normalization methods, to improve the training efficiency, generalization, and overall performance of neural networks. Throughout the years, a series of normalization techniques have been explored by researchers to enhance the training and generalization of deep learning models. This chronological survey encompasses the inception of Batch Normalization in 2015 by [14], extending to the more recent introduction of Batch Group Normalization (BGN) by Zhou in 2020 [42]. The timeline of normalization methods began with Batch Normalization (BN), which revolutionized deep learning by stabilizing training dynamics and accelerating convergence. Subsequently, Group Normalization (GN), proposed by Wu and He in 2018 [33], addressed certain limitations of BN, particularly when dealing with small batch sizes. Furthermore, Layer Normalization [7], Weight Normalization [25], Instance Normalization [29], and Positional Normalization [17] were introduced, each catering to specific requirements and contributing to the maturation of deep learning models.

3. Methodology

3.1. Normalization techniques

Normalization plays a crucial role in the training of Convolutional Neural Networks (CNNs) to ensure stable and effective learning. A standard normalization layer involves four steps: (1) grouping the feature map into distinct feature groups; (2) computing mean and variance statistics for each feature group; (3) normalizing each feature group using the calculated statistics; and (4) adjusting the normalized feature map to preserve the representation ability of the Convolutional Neural Network (CNN).

Batch Norm is a normalization technique done between the layers of a Neural Network instead of in the raw data [14]. In a neural network, batch normalization (BN) is achieved through a normalization step that fixes the means and variances of each layer's inputs [35] as schematically shown in Figure 2. Normalization is applied separately to each group of data, called a mini-batch, during the training process. This is a general formulation of feature normalization expressed as:

$$x_i := \frac{1}{\sigma_i}(x_i - \mu_i)\ . \tag{1}$$

Here, x represents the feature computed by a layer, and i is an index. In the context of 2D images, $i = (i_N, i_C, i_H, i_W)$ is a 4D vector indexing the features in the order (N, C, H, W), where:

- N is the batch axis,
- C is the channel axis,
- H is the spatial height axis, and
- W is the spatial width axis.

In Batch Normalization, the transformation applied to the input feature to compute the normalized output is given by the following formula:

$$S_i = \{k \mid k_C = i_C\}\,, \tag{2}$$

where i_C (and k_C) denotes the sub-index of i (and k) along the C axis. This implies that pixels sharing the same channel index are normalized together. In other words, for each channel, Batch Normalization (BN) computes μ and σ along the (N, H, W) axes. In other words, the mean and variance are calculated along the batch dimension. Thus, the transformation helps to normalize the input and make the optimization process more stable during training.

According to [36], Batch Normalization does not work effectively for tasks requiring training with small batches, such as image segmentation, often due to memory limitations. Efforts have been made to explore alternative normalization techniques, including Group Normalization, where normalization is applied across partitions of features or channels, with different pre-defined groups [33]. In Group Normalization (GN), the normalization is performed across partitions of features or channels. The formula for Group Normalization is as follows:

$$S_i = \left\{ k \mid k_N = i_N, \left\lfloor \frac{k_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor \right\}, \tag{3}$$

where:
- G is the number of groups (pre-defined hyper-parameter, with $G = 32$ by default),
- C/G is the number of channels per group,
- $\lfloor \cdot \rfloor$ denotes the floor operation,
- $\left\lfloor \frac{k_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor$ means that the indexes i and k are in the same group of channels, assuming each group of channels is stored in a sequential order along the C axis.

In Batch Group Normalization (BGN) technique, the channel, height, and width dimensions are initially concatenated into a new dimension, resulting in a flattened representation denoted as $F_{N\times D}$, where $D = C \times H \times W$. The mean μ_g in (4) and variance σ_g^2 in (5) are then computed along both the batch and the new dimension.

$$\mu_g = \frac{1}{N \times S} \sum_{n=1}^{N} \sum_{d=(g-1).S+1}^{g.S} f_{n,d}\,, \tag{4}$$

$$\sigma_g^2 = \frac{1}{N \times S} \sum_{n=1}^{N} \sum_{d=(g-1).S+1}^{g.S} (f_{n,d} - \mu_g)^2\,, \tag{5}$$

where $g = 1, \ldots, G$ is a group index used in the group normalization technique and G is the number of groups that the new dimension is divided into, and is a hyper-parameter; $f_{n,d}$ is a member of $F(1)_{N\times D}$, representing a feature instance after merging the channel, height, and width dimensions into a new dimension; $D = C \times H \times W$, where C, H, and W are the channel, height, and width dimensions, respectively. Further, $S = M/G$ is the number of instances inside each divided feature group. The notation $g.S$ represents the range of feature instances included in group g for the calculation of the mean μ_g and variance σ_g^2. Specifically, the summation over d goes from $(g-1).S+1$ to $g.S$, indicating that each group g contains S feature instances along the new dimension D.

BGN dynamically adjusts the number of feature instances used for statistical calculation, employing the group technique from Group Normalization (GN). When dealing with a small batch size, a smaller value for G is chosen to combine the entire new dimension, preventing noisy statistics. Conversely, with a larger batch size, a larger G is selected to partition the new dimension into smaller segments, facilitating the calculation of more accurate and less confused statistics. That is why, In our training process, G=2 was used for a batch size of 2 to combine the entire new dimension, and G=32 was used for batch sizes 8 and 16.

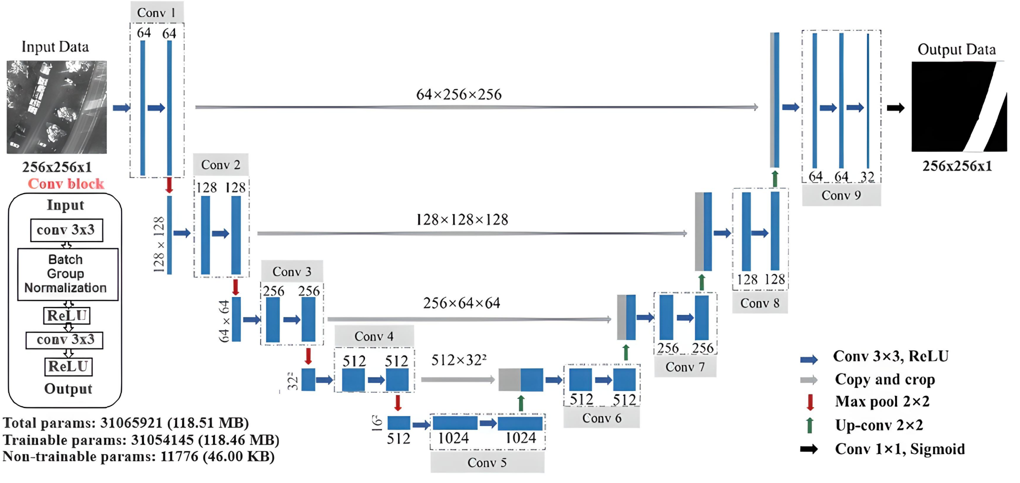


Fig. 3. BGN-UNet Architecture.

This multi-step approach allows BGN to capitalize on the generalization capabilities of GN, while also addressing the limitations of BN. Batch Group Normalization (BGN) has been implemented as a custom layer within the Keras deep learning framework.

Figure 3 provides a visual representation of the BGN-UNet model's architecture based on the original UNet architecture [23]. It illustrates how Batch Group Normalization (BGN) is incorporated into the UNet structure. Figure 4 compares the convolutional blocks in the standard UNet model with those in the BGN-UNet, emphasizing the differences in their structures and how BGN is employed.

BGN-UNet is defined for image segmentation tasks. Within this network, convolutional blocks, encoder blocks, and decoder blocks are included. The basic building block of the network is defined by the conv block function. It applies convolutional layers with BatchGroupNormalization and ReLU activation functions. This block is used to extract features from the input data. The encoder block function combines the convolutional block with max-pooling, allowing the network to progressively downsample the input image and capture high-level features. The decoder block function handles the upsampling and feature concatenation process. It uses transpose convolutional layers to increase the spatial resolution and combines the features from the encoder to refine the segmentation. The build unet function is responsible for constructing the entire UNet architecture, which consists of encoder and decoder blocks.

The choice of the final activation function is determined by the number of classes in the output. In the specific context of binary semantic segmentation, where the objective is to differentiate between two distinct classes, the choice of the activation function at the

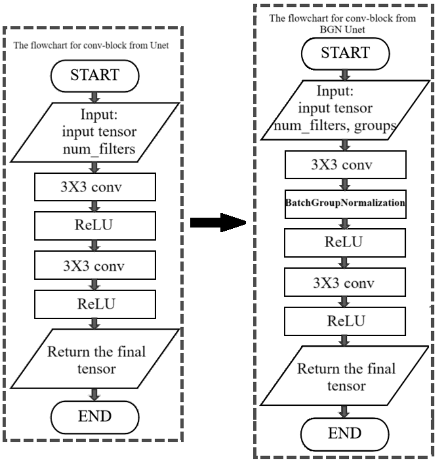


Fig. 4. Convolutional Blocks in UNet and BGN-UNet.

output layer is the sigmoid function which is preferable as output activation function [5]. This function assigns a continuous value to each pixel within the segmented image [31], ranging from 0 to 1. Pixels closer to a value of 1 are indicative of belonging to the road class, while those closer to 0 are representative of the non-road class. In essence, the BGN-UNet is designed for image segmentation tasks, where it takes an input image and produces a segmented image as the output. It effectively combines convolutional layers, normalization techniques, and upsampling to capture detailed features and produce accurate segmentations.

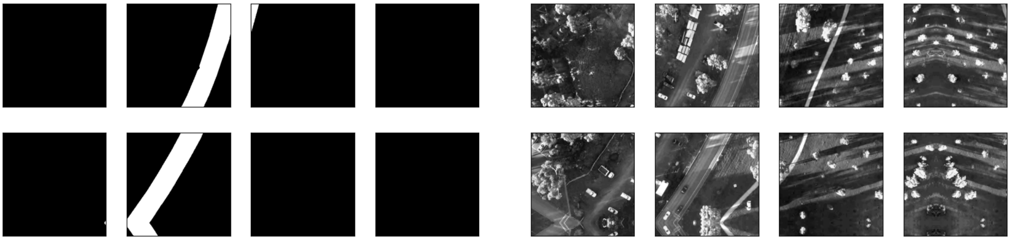
3.2. Data preprocessing

To make it easier to understand and analyze the data, it is imperative to systematically structure and format the dataset. The way data is prepared can vary significantly based on what we want to achieve with the data and the methods we plan to use for analysis [1]. In our research, a road dataset comprised of two distinct sequences is employed. The first sequence consists of 224 images, each possessing dimensions of 848×480 pixels, while the second sequence encompasses 109 images, each characterized by dimensions of 1280×720 pixels, all of these come with corresponding ground truth.

The dataset used for this study can be downloaded from internet and was previously utilized by [39]. However, we would like to note that the original source for downloading the dataset [40] appears to be unavailable at this time. In the meantime, the dataset can be accessed via the link in [41].

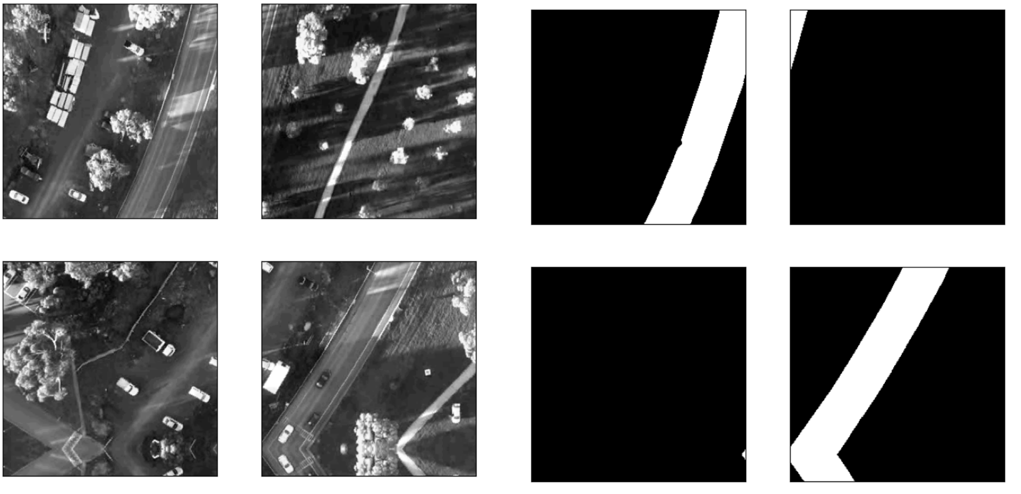
Data preprocessing is an indispensable phase in the pipeline of image analysis, particularly in the context of road segmentation. The following steps encapsulate the procedures employed to transform the raw data into a structured and manageable dataset, each step is guided by a specific rationale:

1. **Color space conversion to grayscale**
   In the first step of our process, the original RGB images are transformed into grayscale versions. This is done to simplify the data and make it more efficient for our model. Grayscale images reduce complexity and memory usage since they remove color information while preserving the critical road segmentation details [15]. This approach improves computational efficiency and helps us optimize resource usage.
2. **Padding for dimension alignment**
   Our goal here is to apply reflection padding to resize images in a way that their dimensions become divisible by 256. Reflection padding helps maintain the continuity and information within the image. The rationale behind using padding lies in its importance for achieving uniformity in image dimensions allowing the images to be divided into smaller images (patches) each measuring 256×256 pixels. After applying padding, the images in the first sequence have a size of 1024×512 pixels, and in the second sequence, the images are resized at 1280×768 pixels.
3. **Patch creation for training**
   Our objective is to create image patches as a solution for segmenting larger images. This process involves dividing the larger images into smaller without overlapping. By partitioning the images into patches, a diverse training dataset is generated, consisting of 1792 patches from the first image sequence and 970 patches from the second sequence. The expression (6) below illustrates how to calculate the number of created patches as it is shown in Figure 5. The choice of a patch size of 256×256 has been made for the sake of computational efficiency, striking a balance between capturing adequate spatial information and maintaining a manageable computational load. It is observed in the literature and established practices for similar tasks that a patch size of 256×256 is commonly chosen, reflecting a widely adopted approach in the field [16, 21].
   $$n = \left(\frac{\text{SIZE\_X}}{\text{patch\_size}}\right)\left(\frac{\text{SIZE\_Y}}{\text{patch\_size}}\right) \tag{6}$$
4. **Elimination of non-informative patches**
   Pruning non-informative patches is very important for data quality and model efficiency. By retaining only patches with pertinent road-related content as shown in Figure 6, it is ensured that the training dataset is composed exclusively of relevant information, enhancing the model's accuracy and mitigating the inclusion of noise or irrelevant details. The selection of patches was automated based on the presence of information in the corresponding masks, avoiding manual elimination [8].



a b

Fig. 5. Creating patches. (**a**) Patches from padded graysclae image; (**b**) patches from padded ground truth.



a b

Fig. 6. Elimination of non-informative patches. (**a**) Informative images patches; (**b**) informative ground truth patches.

5. **Fusion of image sequences**

   The fusion of image sequences yields a unified dataset of 906 images. This fusion enhances the dataset's richness, incorporating diverse scenarios from both sequences, and increases the model's capacity to generalize across different road environments and improving the robustness of the road segmentation model.

6. **Data normalization**

   Data normalization is the process of transforming raw data values to another form with properties [3]. In this case, our data must be more suitable for neural network algorithms which require data that are on a 0-1 scale.

Each of these steps contributes to the overall data preprocessing strategy, ensuring that the dataset is meticulously prepared and optimized for the road segmentation task. Given that our dataset is relatively small, with images of 1024×512 and 1280×768 pixels, we have the opportunity to train our UNet model from scratch, which is particularly advantageous for adapting to the specific characteristics of our dataset.

3.3. Data augmentation

Data augmentation is a technique used to increase the size of a dataset by applying various transformations to the original images. This technique is particularly useful in deep learning tasks, where a large amount of data is required to train the model effectively [26]. In this study, data augmentation was employed to augment the size of the dataset, which consisted of 906 grayscale images with dimensions of 256×256, for aerial road segmentation using the UNet model and batch group normalization.

While it's true that a small dataset might limit how well the model generalizes to larger datasets, image augmentation techniques can help improve performance by generating additional training samples through techniques like rotation and flipping. This effectively increases the diversity of the training data without needing more labelled samples. In fact, augmentation can help prevent overfitting and improve model robustness, leading to better performance even when applied to larger datasets [37]. Although a small dataset is a limitation, these techniques can mitigate its impact and enhance generalizability.

Even though the study uses a small dataset, the method, along with data augmentation, can also work well with larger datasets. We plan to test its effectiveness on larger datasets in future research.

Also, according to [24] the size of the dataset required may depend on various factors such as the complexity of the task and the number of parameters. This statement implies that for simpler tasks or those with fewer parameters, smaller datasets may suffice for effective training.

The data augmentation techniques presented in [6], including horizontal and vertical flipping to help the model learn to recognize road patterns in different orientations, as well as rotation, were implemented. These techniques were applied to the original images to generate new training examples. The augmented dataset was then used to train the UNet model with batch group normalization. In our approach to aerial road segmentation, data augmentation played a crucial role, enabling an increase in the size of our dataset and an improvement in the accuracy of our model. The utilization of a highly accurate and efficient model for aerial road segmentation was achieved through the combination of UNet and batch group normalization with data augmentation.

4. Aerial road segmentation using BGN-UNet

4.1. The model

The UNet model is constructed with a series of convolutional and decoder blocks, each meticulously designed to capture and refine features for binary segmentation tasks. The convolutional block ((`conv_block`) is a crucial building block, comprising two consecutive 3×3 convolutional layers. Notably, Batch Group Normalization is incorporated into the (`conv_block`), enhancing the stability and efficiency of the network during training. Following the convolutional layers, Batch Group Normalization is applied, followed by Rectified Linear Unit (ReLU) activation, synergistically contributing to feature extraction. On the other hand, the decoder block (`decoder_block`) leverages a transposed convolutional layer with a 2×2 kernel for effective upsampling. The upsampled features undergo concatenation with corresponding features from the encoder block and are subsequently processed through the (`conv_block`) to extract informative features. Throughout the UNet architecture, the encoder systematically downsamples the input image through convolution and max-pooling operations. Conversely, the decoder adeptly upsamples the features to generate a segmentation map. In the context of this binary segmentation task, the final layer of the model employs a 1×1 convolution with a sigmoid activation function, facilitating the precise prediction of pixel-wise binary masks. This thoughtful architectural choice, integrating 3×3 convolutions, Batch Group Normalization, and the appropriate activation function, underscores the model's efficacy in capturing spatial information and thereby enhancing its performance in accurately delineating objects of interest in the input images.

To evaluate the performance of the proposed model, experiments were conducted using an aerial road dataset described in the research paper titled Efficient Road Detection and Tracking for Unmanned Aerial Vehicles [39]. Specifically, 1792 aerial images from this dataset were utilized. As previously explained in the data preprocessing section 3.2, before training the model, the data is prepared. To enhance manageability, the large images are partitioned into smaller 256 by 256 pixel images, commonly referred to as patches, and are represented in grayscale. This approach simplifies data processing and model training. The dataset used in this study comprises a total of 906 informative patches extracted from road aerial images. To facilitate the training and evaluation of our model, the dataset is randomly partitioned into three distinct sets. These sets are designated for various purposes: one is allocated for training the model, another is reserved for validation during the training process, and the final set serves as the test set for the evaluation of model performance. The division is conducted using the `train_test_split` function, which separates the image and mask datasets into 724 images for training (80%), 91 images for validation (10%), and 91 images for testing (10%), in order to ensure the consistency of results across experiments.

Figure 7 highlights the flowchart of methodologies used in this paper. The model

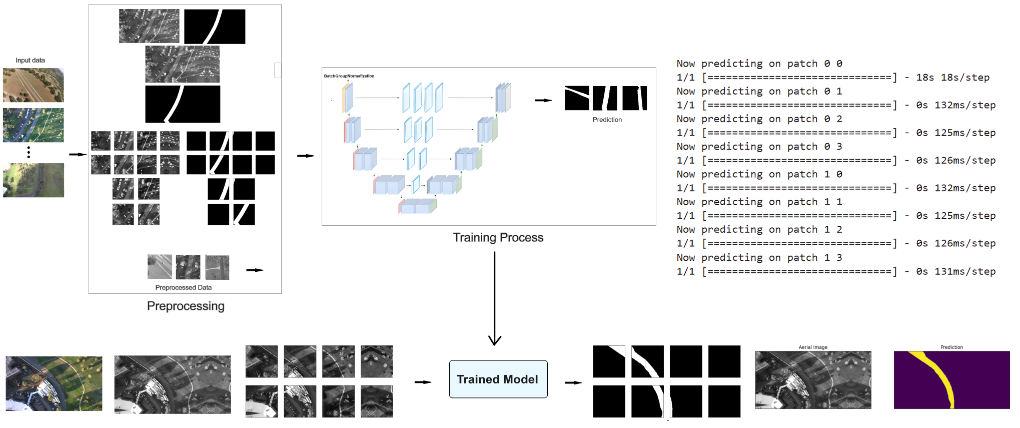


Fig. 7. Flowchart of the proposed methodology.

hyperparameters, as outlined in the Table 1, encompass settings for the training process and data information. In general, our model produces favorable results with minimal distortion and negligible false detections, as shown in Figure 8. In Figure 9, four randomly selected patches from the test set are showcased, segmented using the BNG-UNet model. These patches present varying levels of difficulty while consistently demonstrating amazing segmentation results.

Tab. 1. Model hyperparameters.

Parameter	**Value**
Learning Rate	1×10^{-3}
Loss Function	Binary Cross-entropy
Epochs	25
Batch Size	16
Group	32
Optimizer	Adam Optimizer
Validation Split	0.20, Random State = 42
Total Params	31 065 921 (118.51 MB)
Trainable Params	31 054 145 (118.46 MB)
Non-Trainable Params	11 776 (46.00 KB)
Image Data Shape	(906, 256, 256, 1)
Mask Data Shape	(906, 256, 256, 1)
Max Pixel Value in Image	255
Labels in the Mask	[0, 255]
Patch Size	$256\times256\times1$

Fig. 8. Semantic Segmentation of the Aerial Road Image with BGN-UNet

4.2. Contributions and novelty

This paper presents several significant contributions and novel aspects in the domain of aerial road segmentation using an Enhanced U-Net model, specifically incorporating Batch Group Normalization technique (BGN). The key contributions are outlined below:

A novel normalization technique, Batch Group Normalization (BGN), has been incorporated into the U-Net architecture. This method addresses the performance limitations of Batch Normalization (BN) at very small or extremely large batch sizes by leveraging the grouping strategy employed in Group Normalization (GN). It combines the channel, height, and width dimensions into a unified representation, partitions this dimension into feature groups, and computes the statistics across both the feature groups and the entire mini-batch to enhance performance. By effectively stabilizing training dynamics, BGN enhances model convergence and accuracy.

Building upon this, the proposed BGN-UNet architecture modifies the traditional U-Net framework by implementing BGN layers, which allows for better feature extraction and representation in aerial imagery. This adaptation is particularly beneficial for binary semantic segmentation tasks, where precise delineation of road areas is critical.

In line with these theoretical improvements, our experimental results demonstrate that the BGN-UNet achieves a Mean Intersection over Union (IoU) of 98.4% (See section 5), significantly outperforming traditional normalization techniques such as Batch Normalization (BN) and Group Normalization (GN). This remarkable accuracy underlines the effectiveness of our proposed model in real-world applications.

Furthermore, a thorough comparative analysis of the BGN-UNet model is provided against several state-of-the-art models for aerial road segmentation. This evaluation highlights the superior performance of the approach and discusses its robustness in handling complex urban environments with various road types and conditions.

In summary, the distinctive contributions of this paper lie in the innovative integration of BGN into the U-Net architecture, achieving superior segmentation accuracy, conducting comprehensive evaluations against existing models, and addressing practical

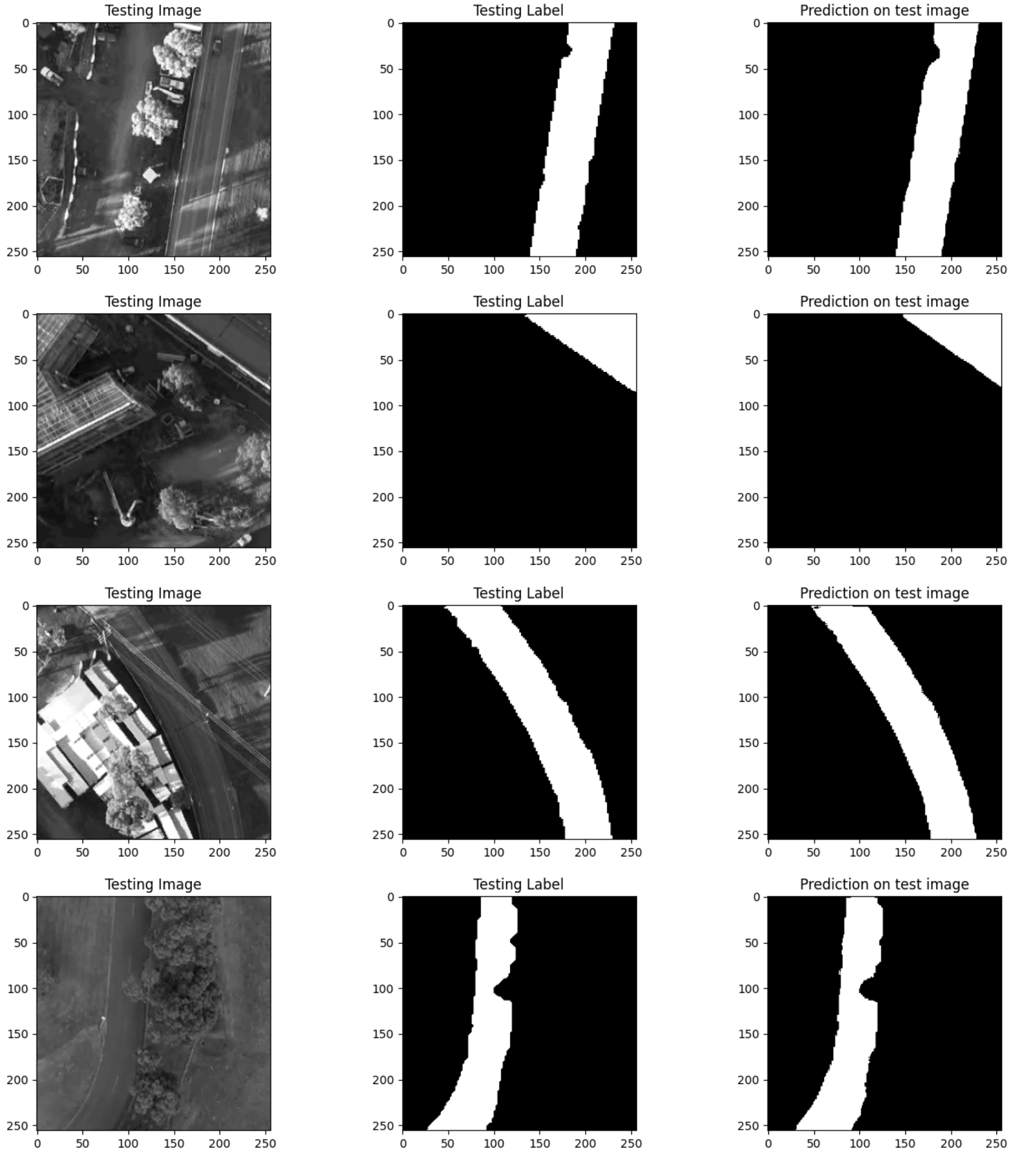


Fig. 9. Road segmentation using BGN-UNet

challenges in aerial road detection. These elements collectively underscore the novelty and significance of our research in advancing methodologies for semantic segmentation in aerial imagery.

5. Results and discussion

Among the various normalization methods, two specific techniques, Batch Normalization (BN) and Group Normalization (GN), have been chosen as the baseline methods to be used in the U-Net architecture. These methods will serve as reference points for comparing the performance of the novel Batch-Group Normalization (BGN) technique when combined with the UNet architecture, which is being introduced and evaluated. By comparing BGN-UNet to these established normalization techniques, we can assess its effectiveness and potential advantages. This evaluation allows us to understand the performance of the novel Batch-Group Normalization (BGN) technique when integrated with the U-Net architecture.

In our UNet model, BGN has been utilized to optimize the network's performance. Testing was conducted in image segmentation, a challenging task, with a specific focus on its performance with aerial imagery. Initial results suggest that BGN might offer advantages over traditional BN and GN methods.

In our comprehensive experimental design, an assessment of three distinct models BN-UNet, GN-UNet, and BGN-UNet was conducted. The impact of varying batch sizes (2, 8, and 16) across a consistent 25 epochs training period was systematically explored. Despite the constraints of a small dataset and only one GPU, our model still achieves good results. The selection of the Adam optimizer, a widely recognized choice in deep learning, greatly facilitated our training process by ensuring efficient convergence and adaptive learning rates [22]. To fit our approach to binary segmentation, focusing on the detection of road and non-road classes, the binary cross-entropy loss function was employed. Cross-entropy serves as a loss function in neural networks in machine learning, offering a metric to gauge the likeness between predicted and actual values [13]. While our resources may appear limited, our results consistently demonstrated that BGN-UNet outperformed both BN-UNet and GN-UNet, particularly in terms of Mean Intersection over Union (IoU).

In our experiments with three different normalization methods BN-UNet, GN-UNet, and BGN-UNet using varying batch sizes of 2, 8, and 16, interesting results were observed in terms of Mean IoU, which is a measure of segmentation accuracy. For the smallest batch size 2, the BGN-UNet achieved a Mean IoU of 0.9727, while BN-UNet and GN-UNet had Mean IoU scores of 0.9673 and 0.9687, respectively. As the batch size was increased to 8, it was observed that BGN-UNet outperformed both GN-UNet and BN-UNet, achieving a higher accuracy with a Mean IoU value of 0.9729. In comparison, GN-UNet had a Mean IoU of 0.9724, while BN-UNet lagged behind with a score of

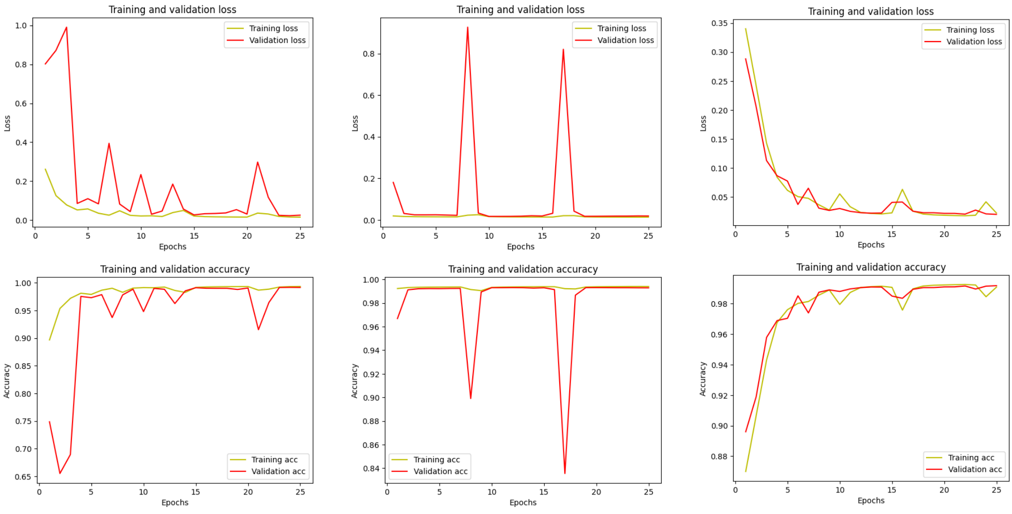
Tab. 2. Mean IoU for Different Batch Sizes and Models

Model	**Batch Size 2**	**Batch Size 8**	**Batch Size 16**
BN-UNet	0.96733713	0.96600366	0.97286165
GN-UNet	0.9687331	0.97235453	0.9730376
BGN-UNet	**0.9726844**	**0.9729512**	**0.9740099**
BGN-UNet+data augmentation	**0.98135**	**0.9821**	**0.9840**

0.9660. When using a batch size of 16, BGN-UNet recorded the highest Mean IoU of 0.9740, outperforming both BN-UNet 0.9729 and GN-UNet 0.9730. In summary, BGN-UNet consistently achieved the best segmentation accuracy across all batch sizes, outperforming both BN-UNet and GN-UNet. These results highlight the effectiveness of Batch-Group Normalization (BGN) in improving the U-Net model's performance in semantic segmentation tasks. The Table 2 shows the Mean IoU values for various batch sizes in the BN-UNet, GN-UNet, and BGN-UNet models.

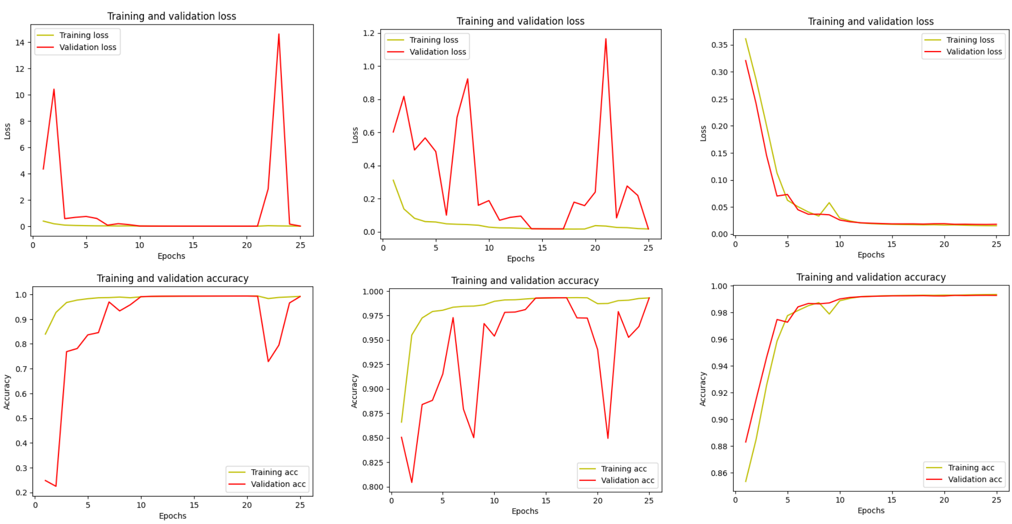
When predicting patches from a large image (1024×512) using the BGN-UNet model, it has been observed that the first patch takes approximately 18 seconds to process (compared to 12 seconds for the BN-UNet model and 15 seconds for the GN-UNet model.), while the remaining 7 patches are predicted almost instantaneously. This initial delay is due to model initialization overhead, which includes tasks such as loading weights, setting up the neural network in memory, and performing preprocessing steps. Once these tasks are completed, the model is fully operational, allowing for the rapid prediction of subsequent patches. BN-UNet is generally faster because it relies on batch normalization, which uses batch statistics, resulting in faster initial processing. On the other hand, GN-UNet uses group normalization, which normalizes within groups without depending on batch size, leading to a middle-ground processing time compared to BN-UNet and BGN-UNet. In summary, the 18 seconds for the first patch mainly come from initialization and setup overhead, while the near-zero time for the following patches reflects efficient reuse of the initialized model, significantly speeding up the process for the remaining patches – see Fig. 7.

Our results show that BGN-UNet is very adaptable and efficient, making it a valuable tool for tasks like binary segmentation, especially in situations where data is limited, and computational resources are constrained. To summarize, BGN-UNet appears to perform well with small datasets in binary segmentation tasks, as our findings indicate. The visual representations of training and validation Metrics as shown in Figs. 10, 11, and 12 provide a comprehensive overview of the performance evaluation of the three models under varying batch conditions. Favorable results with minimal distortion and negligible false detections can be observed in the BGN-UNet graphs.



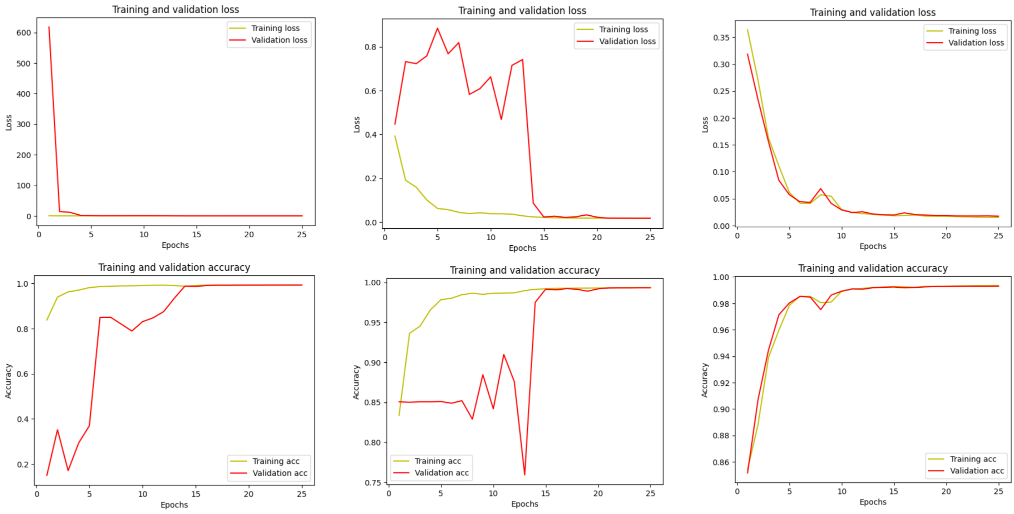
a **b** **c**

Fig. 10. Training and validation metrics for batch size 2: (**a**) BN-UNet; (**b**) GN-UNet; (**c**) BGN-UNet.



a **b** **c**

Fig. 11. Training and validation metrics for batch size 8: (**a**) BN-UNet; (**b**) GN-UNet; (**c**) BGN-UNet.



a **b** **c**

Fig. 12. Training and validation metrics for batch size 16: (**a**) BN-UNet; (**b**) GN-UNet; (**c**) BGN-UNet.

5.1. Ablation study

The choice of batch size significantly influences the training dynamics and convergence of deep learning models. To understand the effect of batch size on model performance, we conducted experiments using three different batch sizes: 2, 8, and 16. The performance metric used for evaluation was the Mean IoU (Intersection over Union). The results for the BGN-UNET model are as follows: for Batch Size 2, the Mean IoU was 0.98135; for Batch Size 8, the Mean IoU increased slightly to 0.9821; and for Batch Size 16, the highest Mean IoU of 0.9840 was achieved.

As the batch size increased, we observed a slight improvement in Mean IoU, with batch size 16 yielding the highest performance. No signs of overfitting were detected, and the model remained stable across all batch sizes, indicating that varying the batch size had minimal impact on overall performance. However, the slight increase in performance with larger batch sizes suggests that batch size 16 may have been more effective in stabilizing gradients during training, leading to better results.

In contrast, when using traditional normalization techniques, we observed different performance metrics. The results when using Batch Normalization are as follows: for Batch Size 2, the Mean IoU was 0.9673; for Batch Size 8, the Mean IoU slightly decreased to 0.9660; and for Batch Size 16, the Mean IoU improved to 0.9729. When using Group Normalization, the results were: for Batch Size 2, the Mean IoU was 0.9687; for Batch

Size 8, the Mean IoU increased to 0.9724; and for Batch Size 16, the highest Mean IoU of 0.9730 was achieved.

These results indicate that while larger batch sizes can enhance performance in some cases, the integration of Batch Group Normalization (BGN) allows for effective training even with small batch sizes, mitigating some of the common issues associated with traditional normalization techniques that struggle under similar conditions.

Overall, these findings illustrate how batch size affects various aspects of model training, including convergence behavior and generalization capability. The slight improvements observed with larger batch sizes suggest a potential benefit in stabilizing gradients during training. However, BGN's effectiveness with smaller batches highlights its advantage in scenarios where data availability is limited. In future work, we aim to explore larger batch sizes of 32 and 64 to further assess their impact on model performance and training dynamics.

5.2. Practical applications of BGN-UNet in road detection

The BGN-UNet model offers significant potential in road detection tasks across multiple domains due to its advanced segmentation capabilities and adaptability to high-resolution imagery. This subsection explores several practical applications where BGN-UNet could contribute to improving accuracy and efficiency, including urban road mapping, autonomous vehicle navigation, infrastructure monitoring, disaster response, and traffic analysis. Each example highlights specific contexts in which BGN-UNet's performance may address current challenges and enhance practical outcomes.

- **Urban Road Mapping**

  BGN-UNet can be employed to accurately map urban road networks from high-resolution satellite or aerial imagery. Example: Similar to the C-UNet model, which improved road extraction accuracy in remote sensing images, BGN-UNet could enhance urban planning and traffic management by providing precise road layouts.
- **Autonomous Vehicle Navigation**

  In autonomous driving systems, BGN-UNet can be utilized for real-time lane and road boundary detection. Example: A project using a UNet model for lane detection demonstrated high accuracy on diverse driving scenarios, showcasing how deep learning models can effectively identify drivable areas under various conditions.
- **Infrastructure Monitoring**

  BGN-UNet can assist in monitoring the condition of roads by detecting cracks and other surface anomalies. Example: Research has shown that U-Net architectures can be adapted for crack detection in tunnels and roads, emphasizing the model's capability to automate infrastructure inspections and enhance maintenance strategies.
- **Disaster Response and Recovery**

After natural disasters, BGN-UNet can help assess road damage by analyzing satellite imagery to identify blocked or damaged routes. Example: Similar methodologies have been applied in post-disaster scenarios where rapid assessment of road conditions is crucial for effective emergency response.

- **Traffic Analysis and Management**
  The model can be used to analyze traffic patterns by segmenting roads from video feeds or images captured by drones. Example: The integration of deep learning models has shown promise in extracting road features from very-high-resolution images, which could be adapted for real-time traffic analysis.

5.3. Challenges and considerations for practical implementation of BGN-UNet

While our paper primarily focuses on the advantages of the BGN-UNet model compared to traditional normalization techniques like Batch Normalization (BN) and Group Normalization (GN), we acknowledge the importance of discussing the potential challenges associated with implementing BGN-UNet in practical applications. Key considerations include the following.

- **Execution runtime**
  Implementing BGN can significantly extend the execution runtime during model training compared to simpler normalization techniques. However, the improved results it yields justify the added computational cost.
- **Memory Requirements**
  BGN-UNet can demand higher memory usage due to the need to maintain statistics for multiple groups within a batch. As seen in the literature [20], larger batch sizes need more memory for activations and gradients.
- **Sensitivity to Hyperparameters**
  The effectiveness of BGN-UNet may depend on the careful tuning of hyperparameters, such as the number of groups and batch size. our results demonstrate that, despite variations in batch size, BGN-UNet consistently outperforms the other models.
- **Generalization Across Domains**
  While BGN-UNet has shown promise in specific tasks like aerial road segmentation, its generalizability to other domains remains an open question. While U-Net performs well in biomedical applications [32], the integration of BGN would further enhance its performance. Future work will explore how BGN-UNet performs across various tasks. This investigation will help clarify the effectiveness of BGN-UNet in diverse applications and identify potential limitations in its adaptability.
- **Complexity of Implementation**
  Incorporating BGN into existing architectures may require more complex modifications compared to standard normalization techniques. This can be a challenge for practitioners with limited experience in deep learning.

In conclusion, while BGN-UNet offers notable advantages for training stability and performance, it is essential to consider these potential challenges when implementing it in practical scenarios. Future work could explore these aspects further, providing insights into optimizing BGN-UNet for various applications and understanding its limitations.

6. Conclusion

Our study presents an innovative approach by incorporating Batch Group Normalization (BGN) technique into the well-known UNet architecture for binary semantic segmentation, with a particular focus on road detection. We evaluated the performance of BGN-UNet in comparison to BN-UNet and GN-UNet, and our experimental results underscored the superior performance of the BGN-UNet model. The careful preprocessing of the dataset played a significant role in the success of this segmentation task. Integrating BGN as a custom layer in the Keras deep learning framework allowed us to make good use of its benefits and incorporate it into the UNet architecture. The research concluded that BGN-UNet is a valuable network for aerial road segmentation, even in situations with limited data and constrained computational resources. The overall outcome of our study was to enhance UNet model, offering an innovative approach to semantic segmentation with consistently superior results. Our proposed model experienced relatively faster convergence compared to baseline networks such as BN-UNet and GN-UNet, easily achieving 0.984 Mean IoU benchmark within only 25 epochs of training. We believe that further enhancements can be made to our model, not only in terms of training on a single GPU and limiting the training to 25 epochs with a maximum batch size of 16, but also in exploring further optimizations in hyperparameter tuning, dataset augmentation, and potentially leveraging distributed computing resources. These steps may enhance our model's performance even further. While existing research has demonstrated the effectiveness of various Convolutional Neural Networks (CNNs) for aerial image analysis, there remains a gap in the application of advanced normalization techniques to improve segmentation accuracy specifically for road detection. This study aims to address these gaps by introducing the enhanced U-Net model with Batch Group Normalization (BGN). By advancing the U-Net model with Batch Group Normalization, we not only aim to bridge existing gaps in segmentation accuracy but also to inspire innovation and improvement in the broader field of computer vision, underscoring the importance of continuous advancements in all domains.

BGN-UNet has performed well in aerial road segmentation, but its ability to handle other tasks like medical image segmentation and autonomous driving is still uncertain. While U-Net was originally designed for biomedical image segmentation, the differences in data types and challenges in these tasks may influence the performance of BGN-UNet. However, it is expected that BGN-UNet could enhance the capabilities of U-Net in these areas. Future research will test BGN-UNet in these tasks to better understand its effectiveness and potential limitations across a wider range of applications. Also, we will keep improving our model and see how well it works with different types of datasets.

Acknowledgement

We acknowledge the use of Google Colab [11] for providing access to free GPU instances, which were instrumental in conducting the research and performing the necessary computations related to the semantic segmentation of aerial images using the UNet model and batch group normalization.

References

[1] Z. S. Abdallah, L. Du, and G. I. Webb. Data preparation. In: D. Phung, G. I. Webb, and C. Sammut (Eds.), *Encyclopedia of Machine Learning and Data Science*, pp. 1–10. Springer US, New York, NY. 2023. doi:10.1007/978-1-4899-7502-7_62-2. Living reference work entry [Accessed: 2023].

[2] A. Abdollahi, B. Pradhan, and A. Alamri. VNet: An end-to-end fully convolutional neural network for road extraction from high-resolution remote sensing data. *Ieee Access* 8:179424–179436. 2020. doi:10.1109/ACCESS.2020.3026658.

[3] A. Abdollahi, B. Pradhan, and N. Shukla. Road extraction from high-resolution orthophoto images using convolutional neural network. *Journal of the Indian Society of Remote Sensing* 49:569–583. 2021. doi:10.1007/s12524-020-01228-y.

[4] A. Abdollahi, B. Pradhan, N. Shukla, S. Chakraborty, and A. Alamri. Deep learning approaches applied to remote sensing datasets for road extraction: A state-of-the-art review. *Remote Sensing* 12(9):1444. 2020. doi:10.3390/rs12091444.

[5] M. I. Ahmed, M. Foysal, M. D. Chaity, and A. B. M. A. Hossain. DeepRoadNet: A deep residual based segmentation network for road map detection from remote aerial image. *IET Image Processing* 18:265–279. 2023. doi:10.1049/ipr2.12948.

[6] S. Arkhangelskiy. Data augmentation on GPU in Tensorflow. In: *Becoming Human. Exploring Artificial Intelligence & What it Means to be Human*. Medium. 2017. https://becominghuman.ai/data-augmentation-on-gpu-in-tensorflow-13d14ecf2b19.

[7] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. In: *Proc. Neural Information Processing Systems – NIPS 2016 Deep Learning Symposium*. 2016. https://openreview.net/forum?id=BJLa_ZC9.

[8] C.-I. Cira, M.-Á. Manso-Callejo, R. Alcarria, B. Bordel Sánchez, and J. González Matesanz. State-level mapping of the road transport network from aerial orthophotography: An end-to-end road extraction solution based on deep learning models trained for recognition, semantic segmentation and post-processing with conditional generative learning. *Remote Sensing* 15(8):2099. 2023. doi:10.3390/rs15082099.

[9] L. Dai, G. Zhang, and R. Zhang. RADANet: Road augmented deformable attention network for road extraction from complex high-resolution remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 61:1–13. 2023. doi:10.1109/TGRS.2023.3237561.

[10] A. Fakhri and R. Shah-Hosseini. Improved road detection algorithm based on fusion of deep convolutional neural networks and random forest classifier on VHR remotely-sensed images. *Journal of the Indian Society of Remote Sensing* 50(8):1409–1421. 2022. doi:10.1007/s12524-022-01532-9.

[11] Google Research. Welcome to Colab. 2023. https://research.google.com/colaboratory/. Online Service [Accessed: 2023].

[12] S. Hao, Y. Zhou, and Y. Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing* 406:302–321. 2020. doi:10.1016/j.neucom.2019.11.118.

[13] X. Hu and H. Yang. DRU-net: a novel U-net for biomedical image segmentation. *IET Image Processing* 14(1):192–200. 2020. doi:10.1049/iet-ipr.2019.0025.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proc. 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR. 2015. https://proceedings.mlr.press/v37/ioffe15.html.

[15] V. M. Ionescu. CPU and GPU gray scale image conversion on mobile platforms. In: *Proc. 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6. IEEE. 2017. doi:10.1109/ECAI.2017.8166501.

[16] R. Jaturapitpornchai, M. Matsuoka, N. Kanemoto, S. Kuzuoka, R. Ito, et al. Newly built construction detection in SAR images using deep learning. *Remote Sensing* 11(12):1444. 2019. doi:10.3390/rs11121444.

[17] B. Li, F. Wu, K. Q. Weinberger, and S. Belongie. Positional normalization. In: *Proc. 32th Int. Conf. Neural Information Processing Systems (NeurIPS)*, pp. 1622–1634. Curran Associates. 2019. https://proceedings.neurips.cc/paper/2019/hash/6d0f846348a856321729a2f36734d1a7-Abstract.html.

[18] K. Li, M. Tan, D. Xiao, T. Yu, Y. Li, et al. Research on road extraction from high-resolution remote sensing images based on improved UNet++. *IEEE Access* 12:50300–50309. 2024. doi:10.1109/ACCESS.2024.3385540.

[19] S. Lin, X. Yao, X. Liu, S. Wang, H.-M. Chen, et al. MS-AGAN: Road extraction via multi-scale information fusion and asymmetric generative adversarial networks from high-resolution remote sensing images under complex backgrounds. *Remote Sensing* 15(13):3367. 2023. doi:10.3390/rs15133367.

[20] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, et al. Mixed precision training. In: *Proc. 6th International Conference on Learning Representations (ICLR)*. 2018. https://openreview.net/forum?id=r1gs9JgRZ.

[21] N. Narisetti, M. Henke, K. Neumann, F. Stolzenburg, T. Altmann, et al. Deep learning based greenhouse image segmentation and shoot phenotyping (deepshoot). *Frontiers in Plant Science* 13:906410. 2022. doi:10.3389/fpls.2022.906410.

[22] R. O. Ogundokun, R. Maskeliunas, S. Misra, and R. Damaševičius. Improved CNN based on batch normalization and Adam optimizer. In: *Proc. Computational Science and Its Applications – ICCSA 2022 Workshops*, vol. 13381 of *Lecture Notes in Computer Science*, pp. 593–604. Springer. 2022. doi:10.1007/978-3-031-10548-7_43.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – Proc. MICCAI 2015: 18th International Conference*, vol. 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer. 2015. doi:10.1007/978-3-319-24574-4_28.

[24] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, et al. Ten deep learning techniques to address small data problems with remote sensing. *International Journal of Applied Earth Observation and Geoinformation* 125:103569. 2023. doi:10.1016/j.jag.2023.103569.

[25] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: *Advances in Neural Information Processing Systems 29 (NIPS 2016)*. Curran Associates. 2016. https://proceedings.neurips.cc/paper/2016/hash/ed265bc903a5a097f61d3ec064d96d2e-Abstract.html.

[26] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data* 6:60. 2019. doi:10.1186/s40537-019-0197-0.

[27] F. Sultonov, J.-H. Park, S. Yun, D.-W. Lim, and J.-M. Kang. Mixer U-Net: An improved automatic road extraction from UAV imagery. *Applied Sciences* 12(4):1953. 2022. doi:10.3390/app12041953.

[28] T. Tiwari and M. Saraswat. A new modified-unet deep learning model for semantic segmentation. *Multimedia Tools and Applications* 82(3):3605–3625. 2023. doi:10.1007/s11042-022-13230-2.

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv, arXiv:1607.08022. 2016. doi:10.48550/arXiv.1607.08022.

[30] R. Wang, M. Cai, Z. Xia, and Z. Zhou. Remote sensing image road segmentation method integrating CNN-Transformer and UNet. *IEEE Access* 11:144446–144455. 2023. doi:10.1109/ACCESS.2023.3344797.

[31] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina. Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density. *International Journal of Information System and Technology* 1(1):43–54. 2017. doi:10.30645/ijistech.v1i1.6.

[32] G. Wieczorek, I. Antoniuk, M. Kruk, J. Kurek, A. Orłowski, et al. BCT Boost segmentation with U-net in Tensorflow. *Machine Graphics and Vision* 28(1/4):25–34. 2019. doi:10.22630/MGV.2019.28.1.3.

[33] Y. Wu and K. He. Group normalization. In: *Computer Vision – Proc. ECCV 2018*, vol. 11217 of *Lecture Notes in Computer Science*, pp. 3–19. 2018. doi:10.1007/978-3-030-01261-8_1.

[34] X. Yang, X. Li, Y. Ye, R. Y. Lau, X. Zhang, et al. Road detection and centerline extraction via deep recurrent convolutional neural network U-Net. *IEEE Transactions on Geoscience and Remote Sensing* 57(9):7209–7220. 2019. doi:10.1109/TGRS.2019.2912301.

[35] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi. A deep normalization and convolutional neural network for image smoke detection. *Ieee Access* 5:18429–18438. 2017. doi:10.1109/ACCESS.2017.2747399.

[36] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: *Proc. International Conference on Learning Representations (ICLR)*. 2017. `https://openreview.net/forum?id=Sks9_ajex`.

[37] W. Zeng. Image data augmentation techniques based on deep learning: A survey. *Mathematical Biosciences and Engineering* 21(6):6190–6224. 2024. doi:10.3934/mbe.2024272.

[38] X. Zhang, X. Han, C. Li, X. Tang, H. Zhou, et al. Aerial image road extraction based on an improved generative adversarial network. *Remote Sensing* 11(8):930. 2019. doi:10.3390/rs11080930.

[39] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi. Efficient road detection and tracking for unmanned aerial vehicle. *IEEE transactions on intelligent transportation systems* 16(1):297–309. 2014. doi:10.1109/TITS.2014.2331353.

[40] H. Zhou and L. Wei. UAV dataset. `https://sites.google.com/site/hailingzhouwei/`. Unaccessible at present.

[41] H. Zhou and L. Wei. UAV dataset. `https://drive.google.com/file/d/1DiQBsm5wmN0fFNnZs6i7oG_SHEo6HLej/`. Copy of [40].

[42] X.-Y. Zhou, J. Sun, N. Ye, X. Lan, Q. Luo, et al. Batch group normalization. arXiv, arXiv:2012.02782. 2020. doi:10.48550/arXiv.2012.02782.

Rayene Doghmane received her Master's degree in Electronics, specializing in Networks and Multimedia, from Badji Mokhtar University, Annaba, Algeria, in 2015. She is currently a Ph.D. student at the LASA laboratory, University of Annaba. Her research interests include image processing, computer vision, machine learning and network systems.

Karima Boukari is a senior lecturer at the University of Badji Mokhtar Annaba in Algeria. She received her engineering degree in automation from the University of Badji Mokhtar Annaba in 1996 and her Ph.D. in 2009. She is currently a lecturer at the Faculty of Technology of the same university. She is also the director of the Laboratory of Study and Research in Instrumentation and Communication of Annaba (LERICA). Her main research interests are: image processing, biometrics, diagnosis and fault detection.