e-Prime - Advances in Electrical Engineering, Electronics and Energy 5 (2023) 100244

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

Advanced road extraction using CNN-based U-Net model and satellite imagery

Mohd Jawed Khan *, Pankaj Pratap Singh

Department of Computer Science & Engineering, Central Institute of Technology, Kokrajhar, Assam, India

ARTICLE INFO

Keywords:
Data augmentation
Road extraction
Deep learning
U-Net
BRISQUE
Semantic segmentation

ABSTRACT

Road extraction from high-resolution remote sensing imagery (HRRSI) is a challenging task due to low spectral variation and the presence of complex background elements, such as the shadows of buildings and trees. Many techniques have struggled to maintain proper edges and boundaries while retaining the geometric features necessary for accurate non-linear road extraction. In this paper, we address these issues by proposing a deep learning approach that utilizes a fine-tuned U-Net model with a modified feature space in the basic U-Net architecture for semantic segmentation. We also incorporate the BRISQUE preprocessing technique to improve the performance. We experimented with a subset of 200 high-quality images from the Massachusetts roads dataset, prioritized based on their rank. Due to computer memory constraints, the images were resized to 256 by 256 pixels. The proposed method produced the accuracy of 95.45%.

1. Introduction

Remotely sensed data became a useful resource as remote sensing technology developed. Because they are the lifeline for people in all eras, roads are now a significant concern. Although roads are the most likely mode of transportation, it takes a lot of attention to update and maintain the data in the geographic information system (GIS) [1–3]. Different levels of difficulty can arise when attempting to extract roads from remotely sensed imagery [4–6]. High-resolution images are very complicated, and it's hard to see things like tree shadows, vehicles, and buildings on the side of the road [7]. This is due to the phenomenon of similar objects showing similar spectral values as a road pixel value [8]. The topology of roads is complex in RS imagery [9]. Methodologies for data extraction can be broadly categorized into two groups: (i) traditional techniques and (ii) deep learning techniques [10]. Further traditional approaches are classified into three types. (a): Road segmentation at the feature level based on template matching, parallel and edge lines, Filter methods, and model methods [11–14]. (b): Multiresolution analysis is applied to extract roads at the object level as well as regional data [15] and (c): extraction of road knowledge based on combined multi-source data [16]. Deep Learning methods are divided into two categories: (a) CNN-based segmentation includes the DBN model [17]. CNN was used by SAITO to extract roads [18]. CNN is a model of cellular neural networks, utilizing the ANN approach [19], object-based deep learning methods [20], and (b): FCN includes CasNet [21], the use of CRFs and multi-scale networks [22], and the combination of residual learning with U-Net [23]. To establish the background for our approach, we review previous accomplishments in road extraction from remotely sensed images using deep learning algorithms. Although deep learning techniques are more efficient than other classical methods, they are inefficient when it comes to extracting road segments arising from complex conditions that affect road segments and are impeded by occlusion [24]. Used Gooey and Pleiades-1A satellites to recognize roads using a CNN-based method. The first stage was to identify each pixel and use a CNN to compute the chance that every pixel contained a road segment [25,26]. After retaining the edge information, minor gaps were connected, and a smooth map was generated using a line-integral convolution-based approach. Finally, by applying image processing techniques, we preprocessed the data. Over the years, deep learning techniques, particularly the U-Net architecture, have shown remarkable success in accurately delineating roads from these images. In recent studies, researchers have proposed various variants and improvements to the U-Net model, enhancing its performance in road extraction tasks [37,38]. Rather than using the standard rectified linear unit activation function, they enhanced their proposed technique by using exponential linear unit activation function (ReLU) metrics for a landscape method to eliminate erroneous road pixels and boost overall effectiveness, and by progressively rotating images in eight phases to supplement training

* Corresponding author.
E-mail address: mjkjawed@gmail.com (M.J. Khan).

https://doi.org/10.1016/j.prime.2023.100244
Received 6 June 2023; Received in revised form 10 July 2023; Accepted 10 August 2023
Available online 12 August 2023

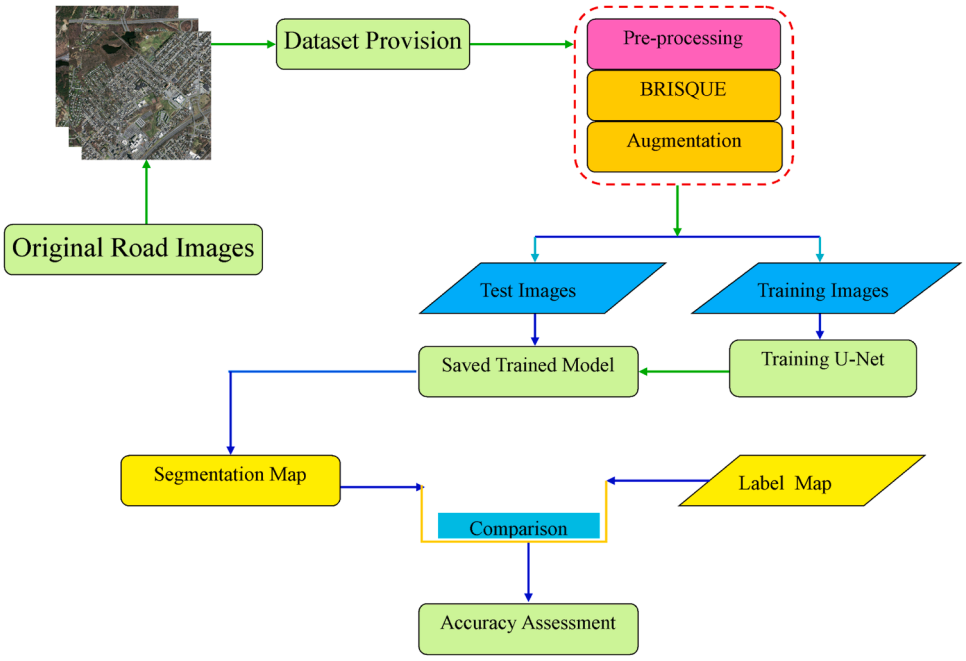

Fig. 1. Working methodology of the proposed approach.

data. The suggested technique outperforms older, state-of-the-art techniques for extracting roads from remotely sensed imagery. In another study, a nonlocal link network containing nonlocal blocks (NLBs) was utilized. As a consequence, deep learning algorithms are becoming more advanced, especially in the area of image processing. After preprocessing the dataset to address issues like the correlation of spatial and geometric information based on different road structures and properties, we used a U-Net architecture-based CNN model to extract roads out of remotely sensed data.

2. Methodology

The proposed approach to extract roads from aerial imagery as shown in Fig. 1.

2.1. Data preprocessing

Preprocessing of data puts the important role by utilizing various algorithms we perform to obtain it. Our Massachusetts road dataset contains 1171 images of Massachusetts. Because all images have a size of 1500 by 1500 pixels, training our modified in number of filters in U-Net on huge imagery takes a high amount of time therefore it is resource-intensive. using BRISQUE to preprocess the dataset and then patchifiying it to assist with model training Finally, we selected images based on the ranking of the images generated after using a blind/referenceless image spatial quality evaluator. It displays the rank of all the images in the dataset. The lower the ranking, the greater the image quality. We chose 200 of the finest images from the 1171 images in the Massachusetts roads dataset based on image rank. (b) Patched random data augmentation is used to get the huge dataset needed for our proposed research.

2.2. No-reference image quality assessment using BRISQUE

Image quality perceptions are very subjective. These types of measures are frequently used to evaluate the performance of algorithms in computer vision applications such as image compression, transmission, and processing [27]. Image quality assessment (IQA) is divided into two study areas: reference-based assessments and no-reference evaluations. Using BRISQUE, we primarily evaluated the quality of no-reference images. A reference image is not required for no-reference image quality evaluation; the algorithm just utilizes the imagery whose quality is being analyzed. A blind approach is often divided into two phases. This is the initial stage in calculating characteristics that illustrate the structure of an image, followed by a comparison of those features to human perceptions of its quality [28]. The BRISQUE model evaluates features solely from the pixels of an image (other models transform images into other spaces, such as wavelet or DCT). The spatial Normal Scene Statistics (NSS) model with locally normalized brightness coefficients provides the basis for the model. Natural Scene Statistical Analysis in the Spatial Domain The next section describes a spatial technique for evaluating the quality of NR images. Using local mean subtraction and divisive normalization, compute locally normalized intensity from a (possibly noisy) picture [29]. Log-contrast intensity should be treated as a local nonlinear procedure in order to lessen the neighbours as (local) mean displacement (LMD) from zero logarithmic contrast and level the local variation of the logarithmic contrast. We may construct [28] by using this approach to an intensity image (i, j) is represented in Eqs. (1) and (2):

$$\widehat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C} \tag{1}$$

where i ε 1, 2... M, j ε 1, 2... N represents indicators of spatial pattern. When the denominator approaches zero, M, N are the image height and width. (comparable to the scenario of an image patch corresponding to a plain sky), C = 1 prevents instabilities from occurring and If I(i, j) domain is [0, 255] then C=1 if the domain is [0, 1] then C=1/255.

$$\mu(i,j) = \sum_{k=-k}^{K} \sum_{l=-k}^{L} w_{k,l} I_{k,l}(i,j) \tag{2}$$

Then calculate the local deviation as $\sigma(i,j)$is represented in Eq. (3).

$$\sigma(i,j) = \sqrt{\sum_{k=-k}^{K} \sum_{l=-k}^{L} w_{k,l} I_{k,l}(i,j) - \mu(i,j)^2} \tag{3}$$

where w = $w_{k,l}$ | k = K,..., K, l = L, ….. L is a 2D (two-dimensional) circularly symmetric Gaussian weighting function was sampled to three standard deviations and thereafter resampled to unit volume. K = L = 3

Fig. 2.1. Original Image with its corresponding mask (a-b).

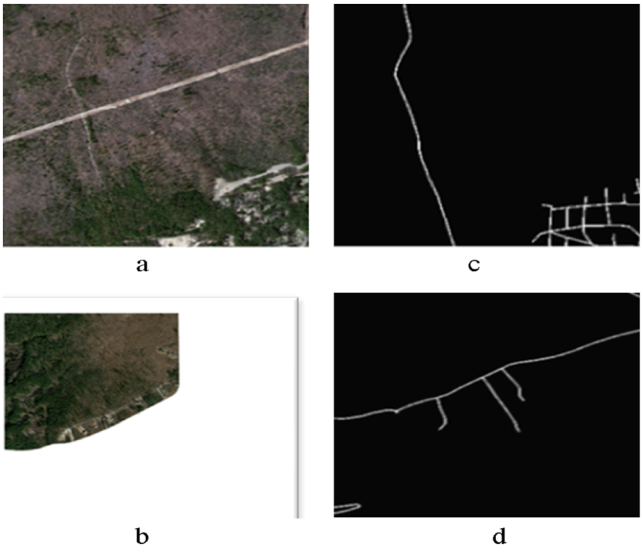

Fig. 2.2. Original Image (a-b) and its corresponding mask with its Brisque score (c-d) as: 72.81776068022214, 117.63437436370333.

in our implementation.

Fig. 2.2 Shows the selected samples of BRISQUE score among top 200 best quality images. (Lower the score better the image quality after applying BRISQUE technique).

We prefer to train any deep learning algorithm with tiny images since they generate better results. But what if we have very large images? One possibility is to break the larger imagery into smaller patches, which would allow us to train any algorithm. You are probably wondering what patches mean. An image patch, as the name implies, is a collection of pixels in imagery. Let's say I have a 20 by 20-pixel image. It can be broken into 1000, 2 × 2 pixel square patches. A user may desire to modify the appearance of an image by repositioning features of objects or adding or subtracting textures. After pachifiy a 200 selected imagery of original size in our road dataset, i.e., 1500 × 1500 to 512 × 512, From one imagery of size 1500 × 1500 we crop four patches of possible size 512 × 512, as shown in Figs. 3 and 4.

2.3. Random data augmentation

We can't capture an image of how every object appears in the actual world. As a result, when developing computer vision models, we must teach them how to recognize broad representations of objects. This is where data augmentation may help.The network model is then prepared to learn and recognize the desired invariance of feature rendition as a result of data augmentation. For our application, the segmentation procedure shouldn't be affected by changes to the imagery used in the learned feature rendering of roads. Since color-oriented and scale parameters are essential for road segments and the scanned Siegfried maps already have a large amount of noise, they really should not be changed throughout data augmentation. As a result, we use rotation and flipping as the other imagery feature alteration strategies. The majority of earlier images or image patch rotation or flipping approaches On the other hand, Siegfried maps have a variety of distinctive features. Examples of things that shouldn't be rotated or flipped include numbers and triangulation locations. Labels can only be rotated slightly and cannot be flipped since big-degree rotation (such as larger than 90°) and flipping are not edge-preserving actions [30]. As a result, as shown in Fig. 5, we

Fig. 3. Sample of four Patch images (512 * 512) of each 200 best quality images.

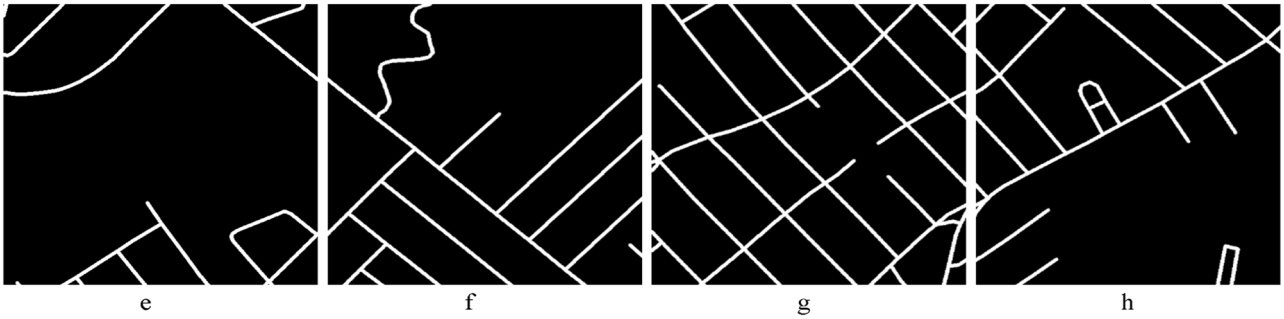

Fig. 4. Sample of corresponding mask of four patch images (512*512) of each 200 best quality images.

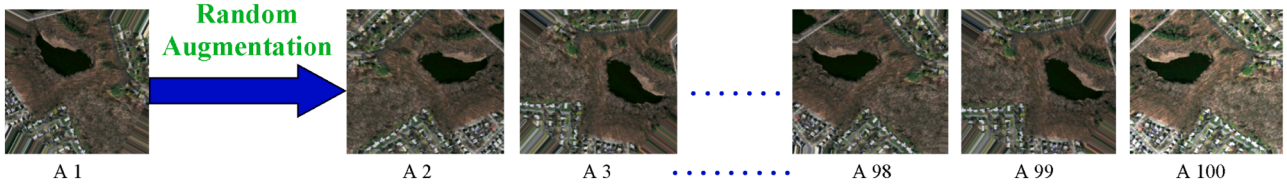

Fig. 5. From each patch like A, we generated 100 images from A- A100 (100 images) after applying random augmentation.

Table 1
Brief description of fine –tuned U-Net with modified filters.

Trained by	Resizing images to 256 × 256 × 3 for the model
Hidden Layers	4
Optimizer	Adam
Loss function	Binary Cross-entropy

generate 100 images from each patch by randomly rotating and flipping the patched images.

2.4. Proposed U-Net architecture

The U-Net architecture is a popular deep learning model used for image segmentation tasks [36], including road extraction. It consists of an encoder pathway that captures context and a decoder pathway that enables precise localization. Here, we describe the U-Net architecture mathematically for road extraction with an input size of 256 × 256 × 3 and other hyperparameters shown in Table 1. As size increases, so does the training period, and to aid backward propagation during training and establish a link between the low-level features and the corresponding high-level information, numerous feature channels are introduced [31–33]. As shown in Fig. 6.

1. Encoder Pathway

The encoder pathway performs a series of convolutional and pooling operations to extract high-level features from the input image.

(a) **Input:** The input image has dimensions 256 × 256 × 3, representing width, height, and the RGB color channels.
(b) **Convolutional Layers:** The convolutional layers consist of multiple filters applied to the input image. Let's denote the number of filters in each layer as **F_enc** . The output of the **i** -th convolutional layer can be represented as

Conv_i, **where** **i** **ranges from 1 to** **N_enc**.

$$\text{Conv\_1} = \text{Conv2D}(\text{Input}, \text{F\_enc}) \tag{4}$$

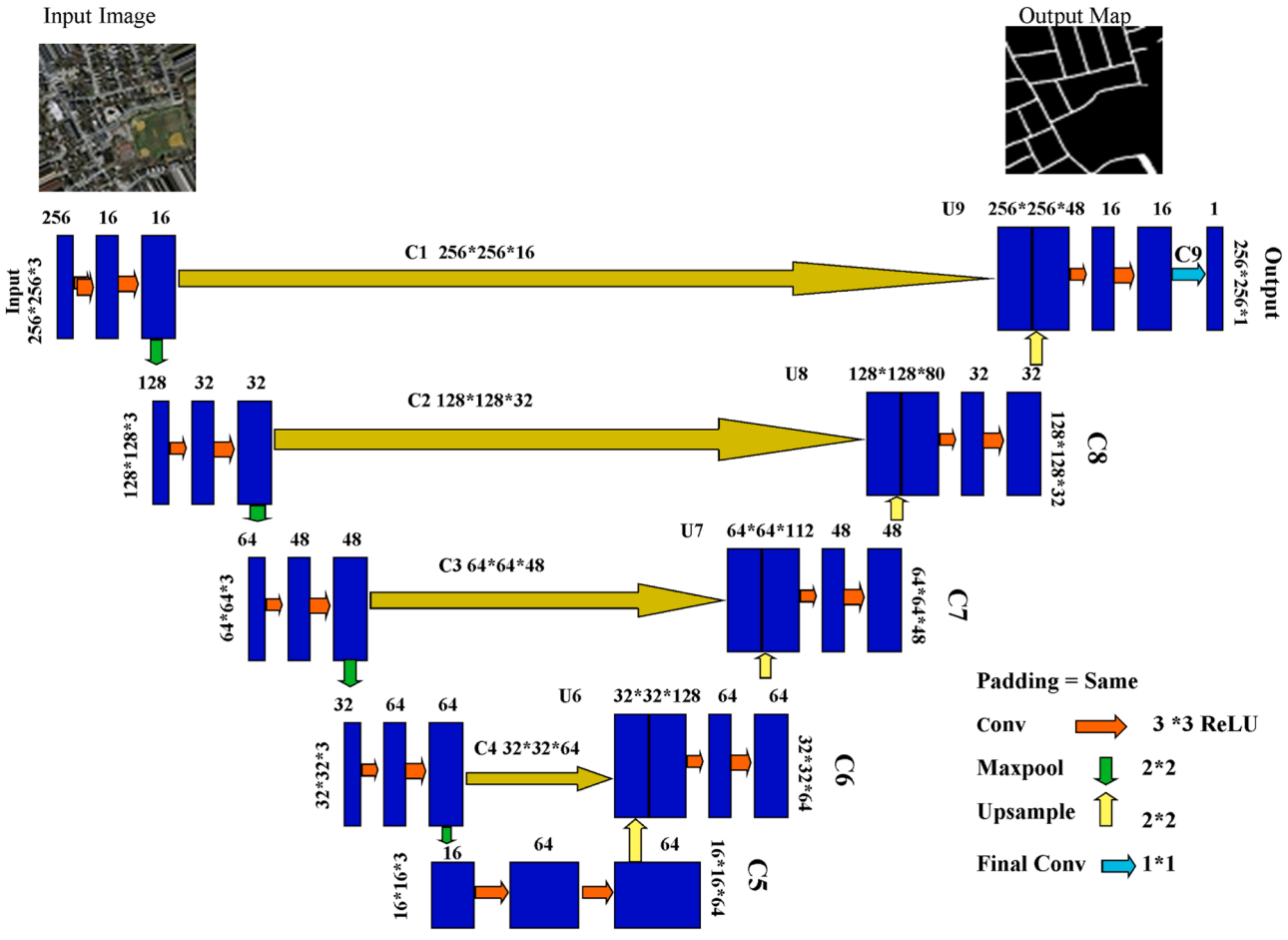

Fig. 6. Architectural detail of fine-tuned U-Net with modified feature space from original U-Net.

Table 2
Performance evaluations of our method and competing deep learning-with other road extraction techniques on Massachusetts road dataset.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	IoU (%)
CNN [35]	94.42	74.69	71.87	73.25	56.69
U- Net [36]	95.20	76.91	74.00	74.66	59.57
G.L Dense U-Net [4]	95.46	81.83	70.48	75.71	60.92
Fine Tuned U-Net (Proposed Approach)	95.48	91.77	69.91	79.36	60.97

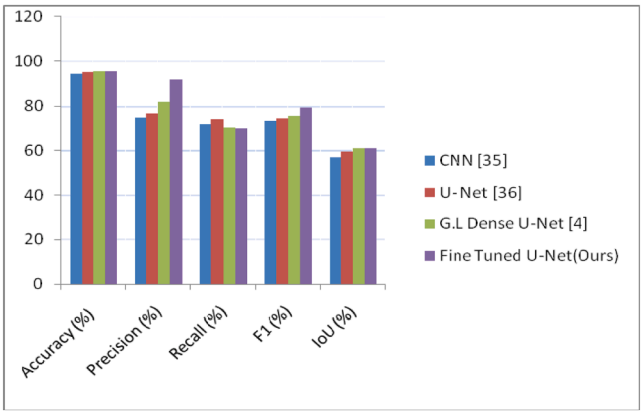

Fig. 7. Comparing various approaches: a visual representation.

$$\mathbf{Conv\_2} = \mathbf{Conv2D}(\mathbf{Conv\_1},\ \mathbf{F\_enc}) \tag{5}$$

$$\mathbf{Conv\_N\_enc} = \mathbf{Conv2D}\big(\mathbf{Conv\_}(\mathbf{N\_enc}-\mathbf{1}),\ \mathbf{F\_enc}\big) \tag{6}$$

(c) Pooling Layers: After each convolutional layer, a pooling operation is performed to downsample the feature maps and captures larger contextual information. We can represent the pooling operation as 'MaxPooling2D'.

$$\mathbf{Pool\_1} = \mathbf{MaxPooling2D}(\mathbf{Conv\_1}) \tag{8}$$

$$\mathbf{Pool\_2} = \mathbf{MaxPooling2D}(\mathbf{Conv\_2}) \tag{9}$$

$$\mathbf{Pool\_N\_enc} = \mathbf{MaxPooling2D}(\mathbf{Conv\_N\_enc}) \tag{10}$$

2. Decoder Pathway

The decoder pathway aims to recover the spatial resolution of the extracted features and generate a segmentation mask.

(a) **Convolutional Transpose Layers (Upsampling):** Convolutional transpose layers are used to upsample the feature maps. The number of filters in each transpose layer can be denoted as 'F_dec'. The output of the 'i'-th transpose layer can be represented as

ConvT_i', where 'i' ranges from 1 to 'N_dec'.

$$\mathbf{ConvT\_1} = \mathbf{Conv2DTranspose}(\mathbf{Pool\_N\_enc},\ \mathbf{F\_dec}) \tag{11}$$

$$\mathbf{ConvT\_2} = \mathbf{Conv2DTranspose}(\mathbf{ConvT\_1},\ \mathbf{F\_dec}) \tag{12}$$

$$\mathbf{ConvT\_N\_dec} = \mathbf{Conv2DTranspose}\big(\mathbf{ConvT\_}(\mathbf{N\_dec}-\mathbf{1}),\mathbf{F\_dec}\big) \tag{13}$$

(b) Concatenation: At each decoder layer, the corresponding feature maps from the encoder pathway are concatenated to provide both high-level and low-level information.

$$\mathbf{Concat\_1} = \mathbf{Concatenate}([\mathbf{Conv\_N\_enc}, \mathbf{ConvT\_1}]) \tag{14}$$

$$\mathbf{Concat\_2} = \mathbf{Concatenate}\big(\big[\mathbf{Conv\_}(\mathbf{N\_enc}-\mathbf{1}), \mathbf{ConvT\_2}\big]\big) \tag{15}$$

$$\mathbf{Concat\_N\_dec} = \mathbf{Concatenate}([\mathbf{Conv\_1}, \mathbf{ConvT\_N\_dec}]) \tag{16}$$

(c) Convolutional Layers: After concatenation, additional convolutional layers are applied to refine the features.

$$\mathbf{ConvT\_1} = \mathbf{Conv2D}(\mathbf{Concat\_1},\ \mathbf{F\_dec}) \tag{17}$$

$$\mathbf{ConvT\_2} = \mathbf{Conv2D}(\mathbf{ConvT\_1},\ \mathbf{F\_dec}) \tag{18}$$

$$\mathbf{ConvT\_N\_dec} = \mathbf{Conv2D}\big(\mathbf{ConvT\_}(\mathbf{N\_dec}-\mathbf{1}),\ \mathbf{F\_dec}\big) \tag{19}$$

3. Output Layer

(a) The output layer consists of a single convolutional layer with a ReLU activation function, which outputs a probability map indicating the presence or absence of a road at each pixel.

$$\mathbf{Output} = \mathbf{Conv2D}(\mathbf{ConvT\_N\_dec},\ \mathbf{1},\ \mathbf{activation}=\text{'}\mathbf{ReLU}\text{'}) \tag{20}$$

This mathematical description outlines the general structure of a U-Net architecture for road extraction tasks with an input size of 256 × 256 × 3.The proposed method performs good by producing its output as the specific hyperparameters such as the number of filters is different, as used in original U-Net architecture ('*F_enc*' *and* '*F_dec*'), the number of encoder and decoder layers is four ('*N_enc*' *and* '*N_dec*'), and the activation functions are ReLU is manipulated as shown in Fig. 6 based on my specific requirements to address the issue of geometric and spatial issues in extracting roads from remote sensing imagery and is represented in Eqs. (4)–(20).

2.5. Cross entropy loss function

Widely used loss function is cross entropy for image segmentation. Several preceding methods, such as weighted cross-entropy, that reweight the data so that background areas are given less relevance than foreground areas are used to address this issue. When approaching the goal of separating the road from the background as a binary classification task, each pixel in the particular imagery may either correspond to the road or the background. Typically, the cross-entropy loss function of binary classification is used [34] and is represented in Eq. (21).

$$L = \frac{-1}{n}\sum_{K=1}^{n}(t_i \log y_i + (1-t_i)\log(1-y_i)) \tag{21}$$

where t i denotes the ground truth of the ith pixel. $t_i = 0$ denotes that the *i*th pixel belongs to the background, whereas ti = 1 denotes that it belongs to the road. Y_i (0,1) is a ReLu function's is used to calculate the value for the *i*th pixel. As we per y_i pixel reached towards 1, and the in the data *i*th pixel is mostly related to road information. By iteratively modifying the network weights, the loss function *L* is lowered by our training algorithm.

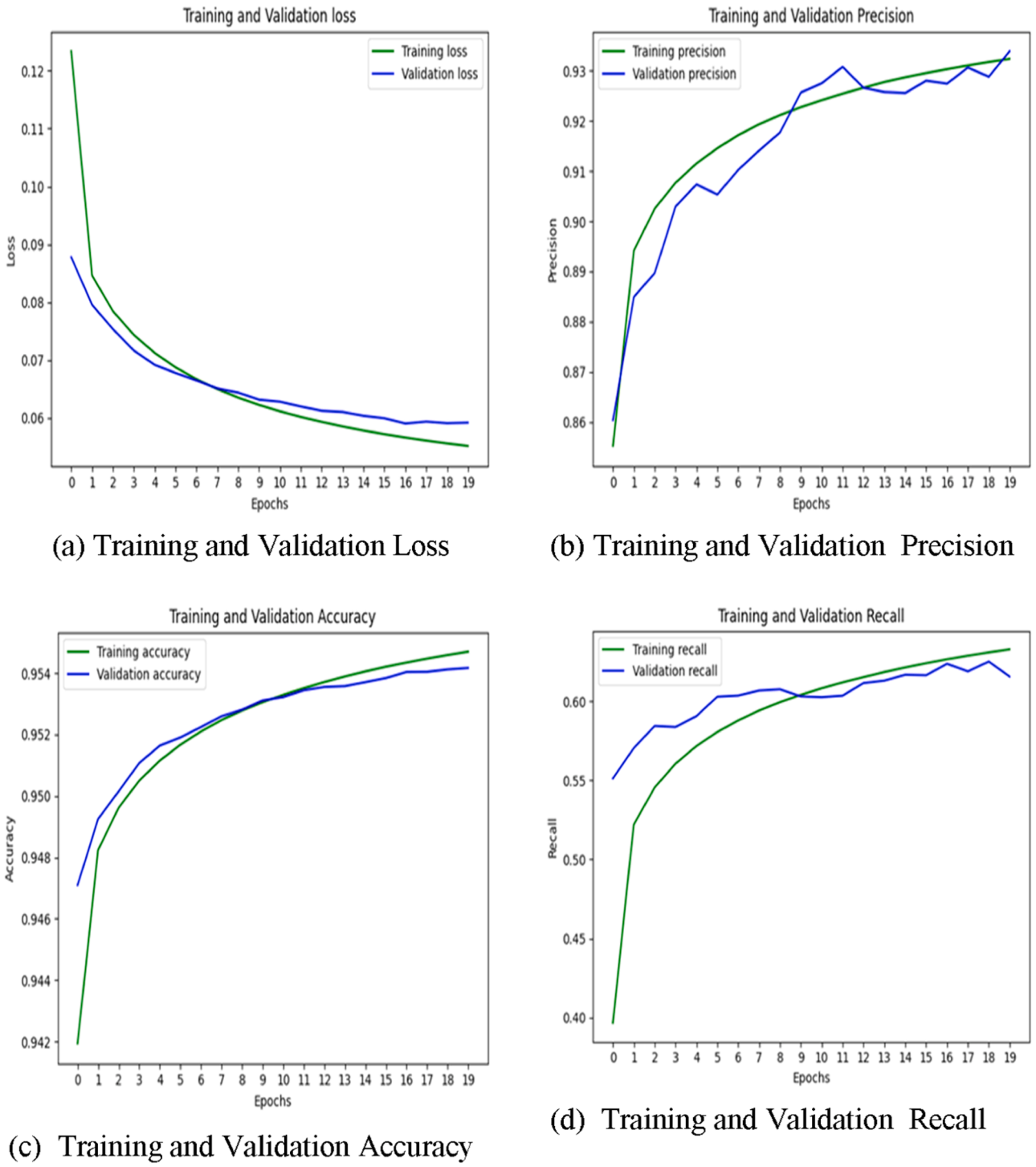

Fig. 8. Obtained performance curves as training and validation loss, training and validation precision, training and validation accuracy, and training and validation recall.

3. Results

3.1. Dataset used and its implementation details

The Massachusetts road dataset was employed [35]. It serves as a standard and is the strongest existing dataset for roads. There are 1171 pieces of aerial imagery in this dataset, with original spatial dimensions of 1500 × 1500. To evaluate and validate later in the U-Net model, we used the largest existing road dataset. After applying BRISQUE (a lower BRISQUE score shows good-quality images), 200 images of size 1500 × 1500 with full information and high quality were chosen to overcome computational limitations. The original image was patchified into four images of 512 by 512 size. From 200 photos, 800 images were generated as the new dataset, and 126 patches were removed as redundant masks, and the respective images where no road is found, i.e., the patchified mask has only black values (every value is 0 in the numpy array), with no road presence, were deleted. The remaining 674 images used for random data augmentation and created 100 images from each patch, so 674 × 100, i.e., 67,400 total images in the final dataset for implementation of the proposed work. The dataset was divided into

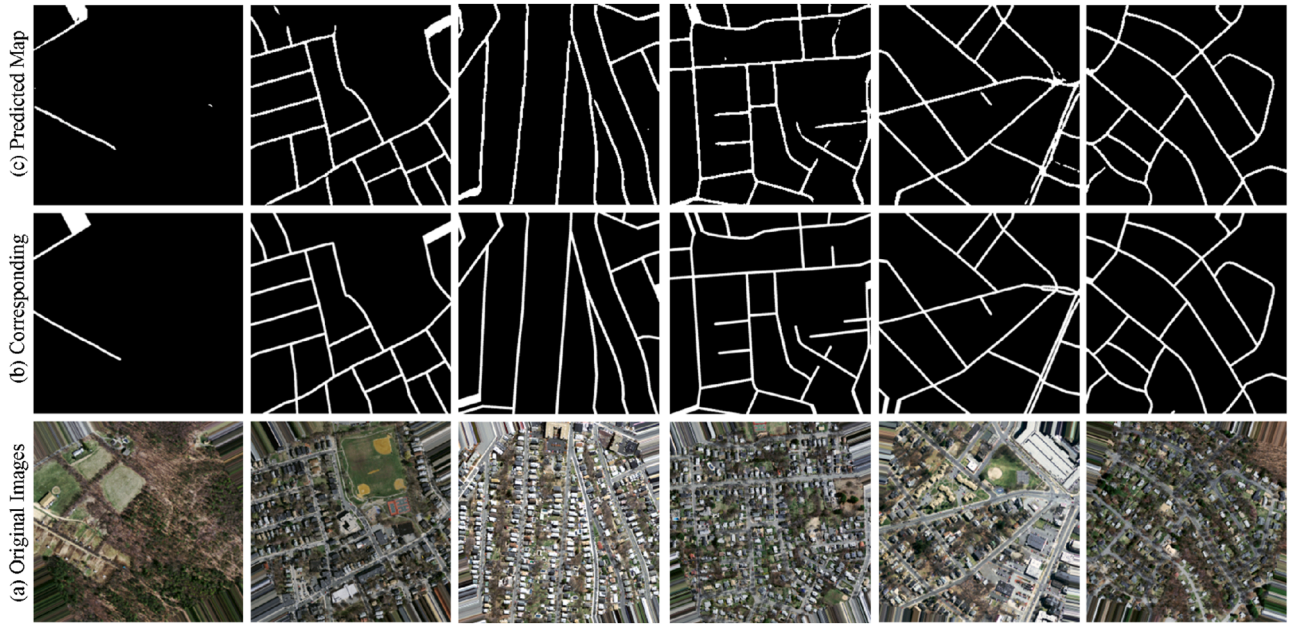

Fig. 9. Overview of selected images as results (a) Original images (b) Corresponding masks (c) Predicted maps as Output.

90%, i.e., 60,660 images, for training and 10% for testing, i.e., 7740 images. In order to identify roads from either the test data or the presented U-Net model, a batch size of 1 was used for the model's testing across 20 epochs. The extracted labels and the actual data were compared to determine their quality. The NVidia Quadra P5000 GPU processor, which has a computing capacity of 6.2 and 16 GB of memory, was used to carry out the full procedure for the proposed approach for extracting roads from remotely sensed imagery. Tensor flow was utilized at the backend.

3.2. Metrics for performance evaluation

The proposed technique used to extract roads from the Massachusetts road remote sensing dataset was assessed for accuracy using four metrics: accuracy, precision, recall, and intersection over union (IOU) parameters and is represented in Eqs. (22)–(26). These metrics can be calculated using the number of false positive (FP), false negative (FN), true negative (TN), and true positive (TP) pixels as follows:

3.2.1. Accuracy

Accuracy is calculated based on pixels' class as the total number of two correct predictions (TP + TN) divided by the total number of a dataset (P + N).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

3.2.2. Precision

The precision is also based on belongingness of the pixel as the proportion between the numbers of true positive and the total number of picture elements belongs to that class.

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

3.2.3. Recall

The Recall is described as proportion between the numbers of true positive and the total numbers of picture elements actually belonging to that class.

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

3.2.4. F1 Score

$$F1\ Score = \frac{2 \times precision \times recall}{precision + recall} \tag{25}$$

3.2.5. Intersection over Union (IoU)

IoU is the proportion of the overlapping area of (masks) of ground truth and predicted area to the total area in the given dataset. However, in the task of road extraction, it is also known as *Jaccard Index*. IoU is illustrative in the following way.

$$IoU = \frac{TP}{TP + FP + FN} \tag{26}$$

4. Discussion

Fig. 9 clearly demonstrates the results obtained using proposed U-Net model. Our research shows that the suggested approach can more precisely extract and categorize roads from the RS image. The U-Net approach predicts few FN pixels, because it is susceptible to shadows and tree occlusion, FP pixels reduce its accuracy. The U-Net model cannot efficiently because parking lots, shadows, and buildings frequently have similar spectral and textural characteristics, these other components can be used to separate roads from them. As a result, several little regions are classified incorrectly. Additionally, some of the extracted road segments are not continuous, and there is a lack of connectivity between the roads at intersections. The four metrics specified in Section 3 were also used to numerically evaluate the proposed U-Net deep learning network' correctness, and the obtained results are illustrated in Table 2 and also represented in Fig. 7. The U-net model achieves scores of 95.48% for accuracy and 60.97% for IoU accuracy metrics, supporting the conclusions from the visually displayed in Figs. 8 and 9. The challenging common spatial and spectral features contained in the dataset, as well as the complex backgrounds and occlusions, have an impact on the improvements of 0.02% and 0.04%, respectively.

5. Conclusion

The U-Net architecture and its variants have revolutionized road extraction from remote sensing images. These deep learning-based approaches have shown remarkable performance in capturing complex

road patterns and overcoming challenges posed by diverse issues. By leveraging the power of convolutional neural networks and advanced architectural modifications. Specific hyperparameters, such as the number of filters, are different, as used in the original U-Net architecture, the number of encoder and decoder (hidden) layers is four, and the activation function is manipulated as ReLU. The dataset used for experimentation was processed using the BRISQUE approach, it picked 200 high-quality images from a total of 1171. The proposed model achieves scores of 95.48% for accuracy and 60.97% for IoU accuracy metrics, which have an impact on improvements of 0.02% and 0.04%, respectively. Our model outperforms earlier models in terms of performance. The U-Net approach predicts few FN pixels, because it is susceptible to shadows and tree occlusion, FP pixels reduce its accuracy. As a result, several little regions are classified incorrectly. In the future, shadow and occlusion handling can be done by modifying the U-net architecture.

Declaration of Competing Interest

The author declare that they have no known competing interests or relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] P.P. Singh, R.D. Garg, A two-stage framework for road extraction from high-resolution satellite images by using prominent features of impervious surfaces, Int. J. Remote Sens. 35 (24) (2014) 8074–8107.

[2] N.Y. Abderrahim, S. Abderrahim, A. Rida, Road segmentation using u-net architecture, in: 2020 IEEE International conference of Moroccan Geomatics (Morgeo), IEEE, 2020, pp. 1–4.

[3] X. Liu, X. Wang, G. Wright, J.C. Cheng, X. Li, R. Liu, A state-of-the-art review on the integration of Building Information Modeling (BIM) and Geographic Information System (GIS), ISPRS Int. J. Geo-Inf. 6 (2) (2017) 53.

[4] Y. Xu, Z. Xie, Y. Feng, Z. Chen, Road extraction from high-resolution remote sensing imagery using deep learning, Remote Sens. 10 (9) (2018) 1461.

[5] R. Alshehhi, P.R. Marpu, W.L. Woon, M.Dalla Mura, Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks, ISPRS J. Photogramm. Remote Sens. 130 (2017) 139–149.

[6] W. Xia, Y.Z. Zhang, J. Liu, .L Luo, K. Yang, Road extraction from high resolution image with deep convolution network—a case study of GF-2 image, Multidiscip. Digit. Publish. Inst. Proc. 2 (7) (2018) 325.

[7] M.O. Sghaier, I. Hammami, S. Foucher, R. Lepage, Stroke width transform for linear structure detection: application to river and road extraction from high-resolution satellite images, in: Image Analysis and Recognition: 14th International Conference, ICIAR 2017, Montreal, QC, Canada, July 5–7, Springer International Publishing, 2017, pp. 605–613. Proceedings 14.

[8] L. Gao, W. Shi, Z. Miao, Z. Lv, Method based on edge constraint and fast marching for road centerline extraction from very high-resolution remote sensing images, Remote Sens. 10 (6) (2018) 900.

[9] A. Abdollahi, B. Pradhan, N. Shukla, S. Chakraborty, A. Alamri, Deep learning approaches applied to remote sensing datasets for road extraction: a state-of-the-art review, Remote Sens. 12 (9) (2020) 1444.

[10] Z. Hong, D. Ming, K. Zhou, Y. Guo, T Lu, Road extraction from a high spatial resolution remote sensing image based on richer convolutional features, IEEE Access 6 (2018) 46988–47000.

[11] Y. Cao, Z. Wang, L. Yang, Advances in method on road extraction from high resolution remote sensing images, Remote Sens. Technol. Appl. 32 (1) (2017) 20–26.

[12] R. Gaetano, J. Zerubia, G. Scarpa, G. Poggi, Morphological road segmentation in urban areas from high resolution satellite images, in: 2011 17th International Conference on Digital Signal Processing (DSP), IEEE, 2011, pp. 1–8.

[13] D. Chaudhuri, N.K. Kushwaha, A. Samal, Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 5 (5) (2012) 1538–1544.

[14] S. Leninisha, K. Van, Water flow based geometric active deformable model for road network, ISPRS J. Photogramm. Remote Sens. 102 (2015) 140–147.

[15] W. Yi, Y. Chen, H. Tang, L. Deng, Experimental research on urban road extraction from high-resolution RS images using probabilistic topic models, in: 2010 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2010, pp. 445–448.

[16] H. Chen, L. Yin, L. Ma, Research on road information extraction from high resolution imagery based on global precedence, in: 2014 Third International Workshop on Earth Observation and Remote Sensing Applications (EORSA), IEEE, 2014, pp. 151–155.

[17] V. Mnih, G.E. Hinton, Learning to detect roads in high-resolution aerial images, in: European Conference on Computer Vision, Berlin, Heidelberg, Springer, 2010, pp. 210–223.

[18] S. Saito, T. Yamashita, Y. Aoki, Multiple object extraction from aerial imagery with convolutional neural networks, Electron. Imaging 2016 (10) (2016) 1–9.

[19] J.S. Wijesingha, R.W. Kumara, P. Kajanthan, R.M. Koswatte, K.R. Bandara, Automatic road feature extraction from high resolution satellite images using LVQ neural networks, Asian J. Geoinform. 13 (1) (2013) 30–36.

[20] H.Wu J.Zhao, L. Chen, Road surface state recognition based on SVM optimization and *image segmentation processing*, J. Adv. Transport. (2017).

[21] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, C. Pan, Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network, IEEE Trans. Geosci. Remote Sens. 55 (6) (2017) 3322–3337.

[22] C.Liu G.Fu, R. Zhou, T. Sun, Q. Zhang, Classification for high resolution remote sensing imagery using a fully convolutional network, Remote Sens. 9 (5) (2017) 498.

[23] P. Li, Y. Zang, C. Wang, J. Li, M. Cheng, L. Luo, Y. Yu, 'Road network extraction via deep learning and line integral convolution, in: Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Beijing, China, 2016, pp. 1599–1602.

[24] Z. Zhong, J. Li, W. Cui, H. Jiang, Fully convolutional networks for building and road extraction: preliminary results, in: Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Beijing, China, 2016, pp. 1591–1594.

[25] P. Panboonyuen, K. Vateekul, Jitkajornwanich, S. Lawawirojwong, An enhanced deep convolutional encoder-decoder network for road segmentation on aerial imagery, in: Proc. Int. Conf. Comput. Inf. Technol, Cham, Switzerland, Springer, 2017, pp. 191–201.

[26] Y. Wang, J. Seo, T. Jeon, 'NL-LinkNet: Toward lighter but more accurate road extraction with nonlocal operations, in: IEEE Geosci. Remote Sens. Lett, 2021 early access, Jan. 26.

[27] H. Maître, From Photon to Pixel: The Digital Camera Handbook, John Wiley & Sons, 2017.

[28] A. Mittal, A.K. Moorthy, A.C. Bovik, No-reference image quality assessment in the spatial domain, IEEE Trans. Image Process. 21 (12) (2012) 4695–4708.

[29] D.L. Ruderman, The statistics of natural images, Netw. Comput. Neural Syst. 5 (4) (1994) 517–548.

[30] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.

[31] J. Y, Z.X. Pan, B. Lei, C.B. Ding, Automatic color correction for multisource remote sensing images with Wasserstein CNN, Remote Sens. 9 (5) (2017) 483.

[32] Y. Tan, Y. Yu, S. Xiong, J. Tian, Semi-automatic building extraction from very high resolution remote sensing imagery via energy minimization model, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2016, pp. 657–660.

[33] W.C. Kang, Y.M Xiang, F. Wang, H. J.You, EU-Net: An efficient fully convolutional network for building extraction from optical remote sensing images, Remote Sens. 11 (23) (2019) 2813.

[34] Y. Lin, D. Xu, N. Wang, Z. Shi, Q. Chen, Road extraction from very-high-resolution remote sensing images via a nested SE-Deeplab model, Remote Sens. 12 (18) (2020) 2985.

[35] V. Mnih, Machine Learning for Aerial Image Labeling, University of Toronto (Canada, 2013.

[36] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: InInternational Conference on Medical Image Computing and Computer-Assisted Intervention, Cham, Springer, 2015, pp. 234–241.

[37] S. Liang, J.Li Z.Hua, Hybrid transformer-CNN networks using superpixel segmentation for remote sensing building change detection, Int. J. Remote Sens. 44 (8) (2023) 2754–2780.

[38] Z. Yang, D. Zhou, Y. Yang, J. Zhang, Z. Chen, A novel road extraction method for remote sensing images via combining high-level semantic feature and context, IEEE Geosci. Remote Sens. Lett. 19 (2022) 1–5.

Mohd Jawed Khan received B. Tech. degree in Computer Science & Engineering from Uttar Pradesh Technical University, Lucknow India in 2008 and M.Tech degree in Computer Engineering from Maharishi Dayanand University, Rohtak Haryana, India in 2012. He is pursuing Ph.D. from Central Institute of Technology kokrajhar Assam India. His research areas are Image processing in the area of remote sensing and its applications.

Pankaj Pratap Singh received B. Tech. degree in Computer Science & Engineering and M.Tech.(IT spec. in Intelligent system) IIIT Allahabad. He has completed his Ph.D. in Geomatics Department Indian Institute of Technology (IIT) Roorkee, India. He is currently working as Assistant Professor in the Department of Computer Science & Engineering in Central Institute of Technology Assam, India. His research interests Satellite Image Processing, Artificial Intelligence, Data Mining. He has published more than 50 research papers in reputed International Journals and Conferences and also authored 7 book chapters.