Computing and Informatics, Vol. 40, 2021, 1244–1262, doi: 10.31577/cai_2021_6_1244

INCSA-UNET: SPATIAL ATTENTION INCEPTION UNET FOR AERIAL IMAGES SEGMENTATION

İbrahim Delibaşoğlu

Software Engineering
Faculty of Computer and Information Sciences
Sakarya University
54050, Sakarya, Turkey
e-mail: ibrahimdelibasoglu@sakarya.edu.tr

Abstract. Building segmentation from aerial images is essential in applications such as facilitating urban planning and estimating the population. Fully convolutional networks (FCNs) and especially UNET have achieved promising results in segmentation problems, after deep learning methods have significantly advanced the performance of many computer vision problems. However, in Convolutional Neural Networks (CNNs) with the standard convolution operations, there are problems such as the overfitting and precise extraction of the boundaries of the objects with different sizes and shapes. In this study, we have used Inception blocks with UNET to enhance feature extraction by implementing two-level Inception approach covering the entire encoding stage. In the proposed architecture, structured form of dropout (DropBlock) is used to prevent overfitting, and spatial/channel attention modules are applied to enhance important features by focusing key areas. We evaluate the proposed INCSA-UNET architecture on publicly available Massachusetts dataset and apply two fold cross-validation experiments for better analyzes. The experimental results show that the proposed architecture does not significantly increase the number of parameters of UNET and has a significant improvement in terms of $F1$ and *Kappa* quantitative measures.

Keywords: Segmentation, deep learning, CNN, INCSA-UNET, attention

1 INTRODUCTION

The increase in spatial resolution of satellite imagery and camera-mounted unmanned aerial vehicles (UAV)/drones provide images with sufficient structural and texture information. These images with the high spatial resolution are widely used in mapping, estimating population and facilitating urban planning by automatically observing changes in urban areas [1, 2]. The automatic building detection/segmentation plays a significant role in urban applications. In the literature, many studies have been carried out on the detection of buildings in aerial images. In the first studies, it is seen that classical image processing-based methods including edge, shadow, shape and color based controls have been examined for building detection in the literature [3, 4, 5, 6]. To achieve promising results in classical methods, multi-stage processes must be applied. In a relatively more complex multi-stage algorithm, shadow is used as evidence for buildings, and classification is made at the pixel level with second-level graph optimization [7]. The basis of the algorithm is to find the initial shadow areas, and it is crucial to combine the regions found at this stage. The general problem of classical methods is that extracted features may be insufficient for different kinds of images.

Machine learning techniques such as fuzzy-genetic algorithm [8], support vector machine [9], maximum likelihood [10] have also been used to detect buildings from aerial images. Unfortunately, these pixel-level algorithms could not perform well due to a lack of ability to use object-level features. Any feature such as edges, shapes, and textures used in classical methods can be considered to improve classification accuracy. Occlusions and buildings with different structures and sizes are also other challenges [11]. Besides classical methods, recent studies show that more successful results have been obtained in the segmentation problem with Convolutional Neural Network (CNN) based methods. CNN is a kind of machine learning method, and it is extremely capable of learning how to extract high and low-level features by using labeled data. Thus, the feature extraction from the input image is included inside the model with convolutional filters during the training process. CNN-based segmentation methods combine feature extraction and classification of each pixel within an architecture. It mainly improves prediction performance by extracting deep features using large training sets. The automatic feature extraction with CNN has been very effective in remote sensing studies, e.g., change detection [12], hyperspectral image classification [13], and object detection [14].

The major contribution of this work is to propose a new UNET based architecture using DropBlock [15], spatial attention [16], channel attention and Inception blocks. We aim to improve the performance of classification by the proposed architecture. The underlying hypothesis behind the proposed architecture is that the model can effectively extract features in parallel Inception layers in addition to sequential layers in the encoding stage. It employs structured form of dropout (DropBlock), the original convolutional blocks of UNET and Inception blocks to prevent overfitting. Attention modules are also applied after the encoding step to

enhance important features. We compare the proposed architecture with different architectures, which have shown good performance on medical and aerial images segmentation.

The remainder of this paper is organized as follows: Section 2 presents the related work in the literature, Section 3 represents the details of proposed network architecture and implementation details, Section 4 describes the dataset and evaluation metrics, Section 5 is a discussion of performance and Section 6 presents conclusions.

2 RELATED WORK

CNNs have made breakthroughs in many image analysis problem and fully convolutional networks (FCNs) were proposed to accomplish pixel-wise classification [17]. In FCN, convolutional layers replaced fully connected layers of CNN for classification and deconvolutional layers are used to upsample feature maps for same resolution as the input. Thus, FCNs created a precedent for pixel-based encoder–decoder architectures. UNET [18] modified the FCN by using advantage of both low-level and high-level features. UNET is a common and well-known backbone network, widely used in fields such as medical image segmentation and building segmentation. UNET consists of encoder(downsampling)-decoder(upsampling) layers and a "skip connection" between them. In a recent study, UNET is enhanced with DropBlock and spatial attention. This proposed lightweight network model, called SA-UNET [19], prevents overfitting, as shown in SD-UNET [20]. SA-UNET architecture is evaluated against UNET and SD-UNET, and it achieves state-of-the-art performance for two medical image segmentation datasets. Also, SCAU-Net [21] architecture investigates using spatial and channel attention modules to enhance the UNET architecture. It shows that using spatial attention is effective to enhance the UNET.

Due to the good performance of UNET, it is also used for building segmentation, and different UNET based architectures are proposed in the literature. Some papers modify the standart UNET because it is not deep enough to gain higher performance. A UNET based architecture combining RESNET [22] and UNET called RES-UNET is proposed in a study for building detection [23]. It applies a pre-processing step for input images and features such as differential vegetation index (NDVI) and the first component of the principal component are fed into the network. It uses pre-trained RESNET weights for feature extraction. Inception module [24] is a type of neural network architecture that leverages feature detection at different scales through filters with different kernel sizes. It allows us to make the networks wider. Another method [25] proposes using Inception modules for UNET architecture. Inception blocks and sequential convolutional filters are used in parallel layers of the network in the feature extraction(encoding) stage. It is named as "Inception UNET-V2" and evaluated against UNET, Inception UNET, and UNET++ [26]. It outperforms the other architectures for two

different aerial images dataset. Delibasoglu and Cetin [27] also reported that UNET expanded with Inception blocks has remarkably better performance compared to UNET and classical state-of-the-art methods for building segmentation. UNET++ architecture re-designs skip connections in UNET and adds new layers between encoder and decoder sub-networks. New dense convolution blocks bring the semantic level of encoder features closer to the features in the decoder, but it increases the trainable parameters and floating point operations (FLOPs). In UNET++, ImageNet weights can optionally be used as pre-trained weights for feature extraction.

Although the CNN-based segmentation methods have achieved promising results, there are still some problems such as precise extraction of the boundaries of the objects. Therefore, some post-processing methods such as fully connected CRFs or Markov random fields (MRFs) was used to improve segmentation result [28, 29]. In another study [30], it is reported that recurrent neural networks could also refine the segmentation results by employing a feedback connection. In this study, we aimed to enhance Inception based UNET with attention and DropBlock modules as a solution to the drawbacks by taking into account the performance of the SA-UNET and Inception based UNET proposed in [25]. In the proposed architecture, attention mechanism is used to weight lower-level features into the final output feature to build boundaries information. Inception blocks are used to enhance feature extracting in the encoding stage.

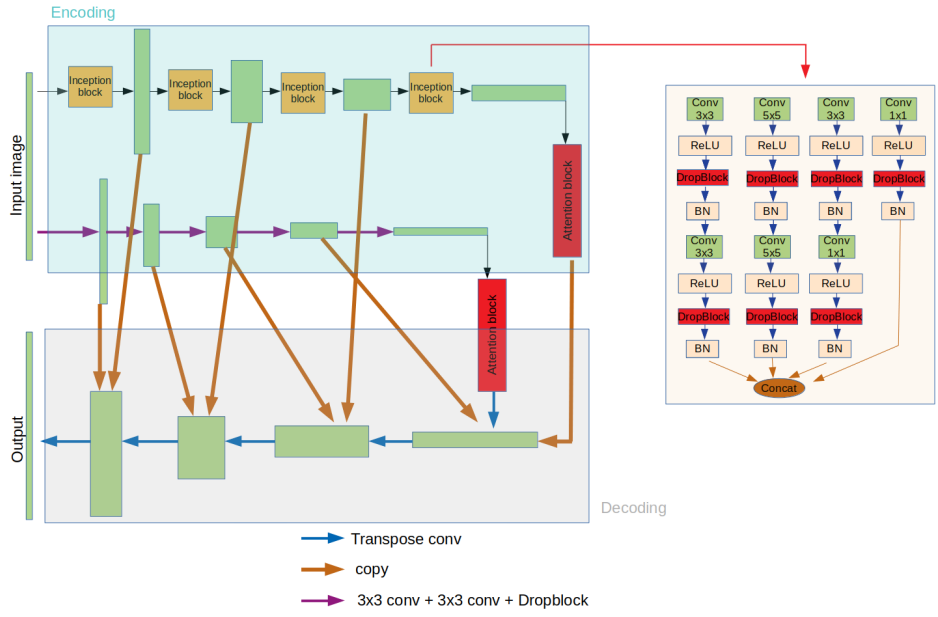

Figure 1. Diagram of proposed INCSA-UNET architecture

3 METHODOLOGY

3.1 Network Architecture

Figure 1 shows the proposed architecture (INCSA-UNET) with Inception blocks (INC), spatial attention (SA) and DropBlock. Proposed architecture consists of three main parts: two parallel layers (classical sequential and Inception) in the encoding stage and decoding part on the bottom which is composed of conventional convolutional layers. Inception block consists of filters with multiple sizes operating on the same level. Besides, another set of parallel blocks is added parallel with Inception blocks. Thus, it could be considered that a complex "Inception" block is implemented that covers the entire encoding process in the proposed architecture. The skip connections make the learning easier, and it is seen that the proposed architecture has skip connections to transmit extracted features with both Inception and classical sequential layers. The architecture applies DropBlock after each convolutional layer in Inception and other layers. In proposed architecture, max pooling is applied after DropBlock layer in encoding stage while DropBlock is at the last step in the decoding stage. DropBlock is better than dropout to regularize CNNs by preventing overfitting problem, as shown in different studies [15, 20, 19, 31]. Encoding layer for classical UNET, Inception UNET-V2 and proposed INCSA-UNET is shown in Figure 2.

Proposed encoding stage consists of 5 classical sequential and 4 Inception layers parallel to each other. Let us assume, x_l is the input feature of a layer, f^{nxn} denotes filter with $n \times n$ kernel size, α denotes rectified linear unit ($ReLU$) activation function, f^{DB} denotes DropBlock and f^{BN} denotes Batch Normalization. The output of classical sequential layer in encoding stage is:

$$x_{l+1} = MaxPooling\left(\alpha\left(\alpha\left(x_l f^{3\times3}\right) f^{3\times3}\right) f^{DB}\right). \tag{1}$$

The output of Inception layer in encoding stage ($x_{l+1}^{Inception}$) is obtained by concatenating paralel filters output (Equations (2), (3), (4) and (5)) as given in Equation (6):

$$x_a = \alpha\left(\alpha\left(x_l f^{3\times3}\right) f^{DB} f^{BN} f^{3\times3}\right) f^{DB} f^{BN}, \tag{2}$$

$$x_b = \alpha\left(\alpha\left(x_l f^{5\times5}\right) f^{DB} f^{BN} f^{5\times5}\right) f^{DB} f^{BN}, \tag{3}$$

$$x_c = \alpha\left(\alpha\left(x_l f^{3\times3}\right) f^{DB} f^{BN} f^{1\times1}\right) f^{DB} f^{BN}, \tag{4}$$

$$x_d = \alpha\left(x_l f^{1\times1}\right) f^{DB} f^{BN}, \tag{5}$$

$$x_{l+1}^{Inception} = [x_a, x_b, x_c, x_d]. \tag{6}$$

Details of proposed Inception block are also shown in Figure 3. Filters with $3 \times 3 - 3 \times 3$, $5 \times 5 - 5 \times 5$, $3 \times 3 - 1 \times 1$ and 1×1 kernel size are used in Inception block, as shown in Equations (2), (3), (4) and (5). x_a, x_b, x_c and x_d represent

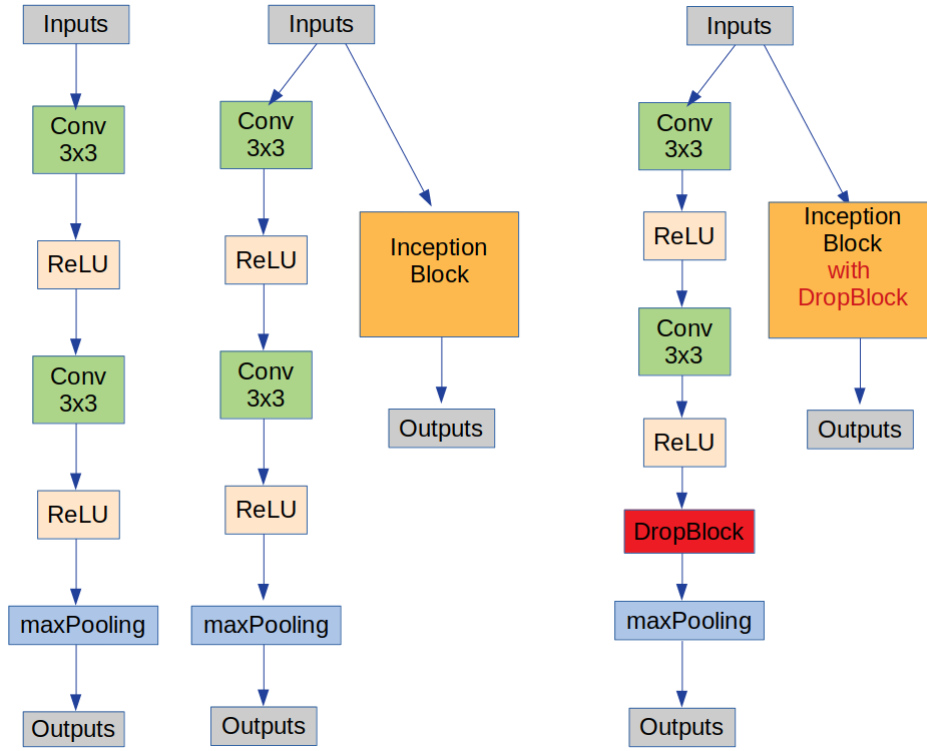

Figure 2. Encoding blocks: UNET (left), Inception UNET-V2 [25] (middle), Proposed with DropBlocks (right)

different sequential filters used in proposed Inception block. Each convolution is followed by *ReLU* activation function, DropBlock and Batch Normalization (BN), as illustrated in Figure 3. In the last step of encoding, $x_{l+4}^{Inception}$ and x_{l+5} features are concatenated after applying attention module. Unlike the convolutional block of Inception UNET-V2, using DropBlock prevents overfitting, as shown in training accuracies for training set-I and II in Figure 6. Figure 6 also shows the validation accuracies and it is seen that validation accuracy is better for training set-I containing more training images while convergence is slightly lower with DropBlock modules. The training graphs obtained with both training sets show that using DropBlock reduces the accuracy for training set while increasing the accuracy for validation. Considering that the training set is better predicted in case of overfitting, DropBlock seems to prevent this situation and clearly increases the validation accuracy.

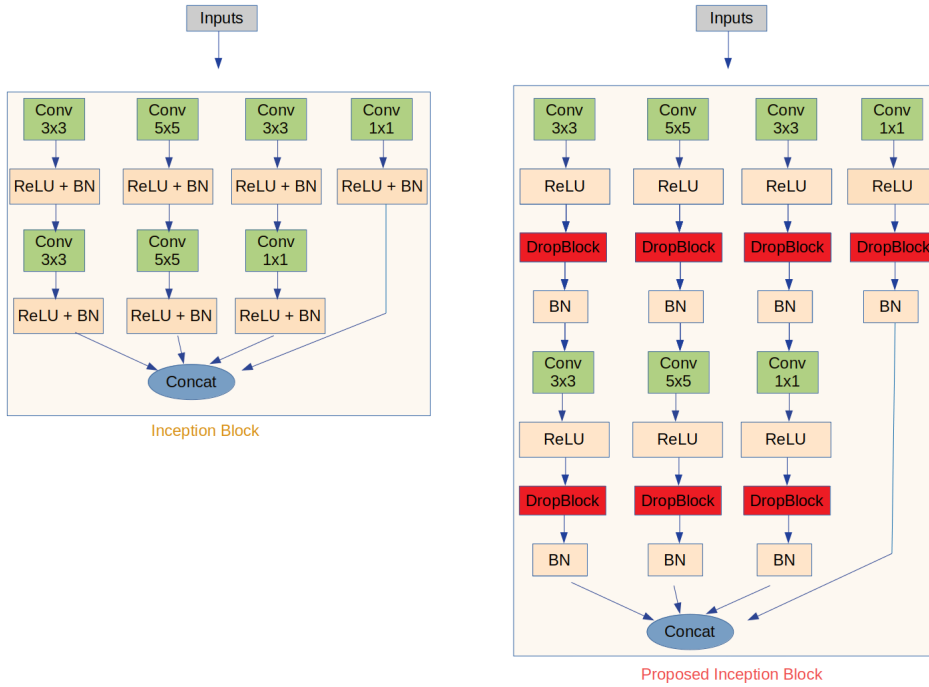

Figure 3. Inception block [25] (left), Proposed Inception block with DropBlocks (right)

3.1.1 Attention Modules

Attention modules are used to enhance important features/some spatial locations by focusing on key features/areas. In this study, we improve our two-level Inception approach using DropBlock with two types of attention mechanisms. Firstly, spatial attention block is used to build a spatial attention map by establishing the spatial relationship in the proposed architecture. It applies average and max pooling operations on input feature ($F^{H,W,C}$) along the channel axis in the first step to generate one dimensional features ($F_{av}^{H,W,1}$ and $F_{max}^{H,W,1}$). Then, extracted features are concatenated and spatial attention map (F_{map}) is generated with a convolutional layer with a kernel size of 7 ($f^{7\times7}$) followed by the Sigmoid activation function (θ), as shown in Equation (7). Figure 4 shows the diagram of spatial attention module. In the last step, F_s output feature is obtained by multiplying attention map learned focusing spatial location (F_{map}) and input feature (F) for adaptive feature refinement, as calculated in Equation (8).

$$F_{map} = \theta\left([MaxPooling(F), AveragePooling(F)]f^{7\times7}\right), \tag{7}$$

$$F_s = F_l \varnothing F_{map}. \tag{8}$$

In addition to spatial attention, channel attention module similar to used in [32] is also implemented. The diagram of proposed channel attention module is shown in Figure 5. It uses two consecutive layers features, $F_{l-1}^{H,W,C}$ and $F_{l}^{H,W,C}$. *Gating module* including $f^{1\times1}$, BN, and $ReLU$ is applied for $F_{l-1}^{H,W,C}$. *Gating module* output (F_g) is upsampled and concatenated with $F_{l}^{H,W,C}$ filtered with $f^{2\times2}$, as shown in Equation (9) on which Ω respresents the upsampling operation. Then, α, $f^{1\times1}$, θ, and Ω operations are applied to acquire feature weights (w), as shown in Figure 5 and Equation (10). In the last step, F_s output feature is obtained by multiplying w and input feature (F_l) to generate a global distribution of channel features, as calculated in Equation (11).

$$F_{con} = [\Omega(F_g), F_l f^{2\times2}], \tag{9}$$

$$w = \Omega(\theta(\alpha(F_{con}) * f^{1x})), \tag{10}$$

$$F_s = (F_l \varnothing w) f^{1\times1} f^{BN}. \tag{11}$$

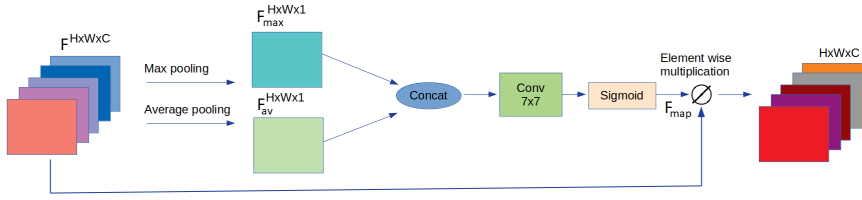

Figure 4. Spatial attention module [19]

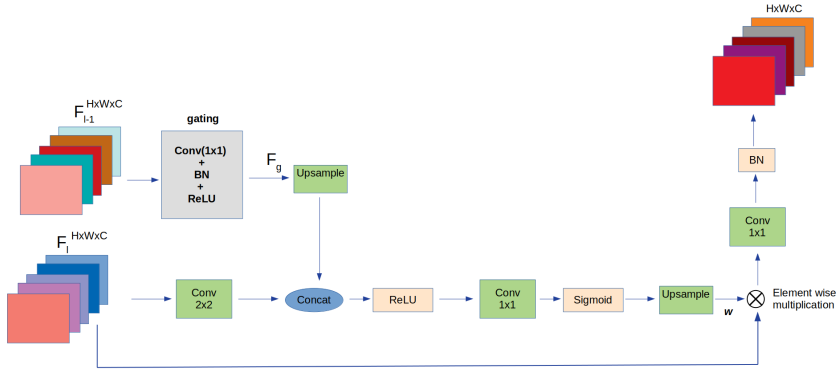

Figure 5. Channel attention module

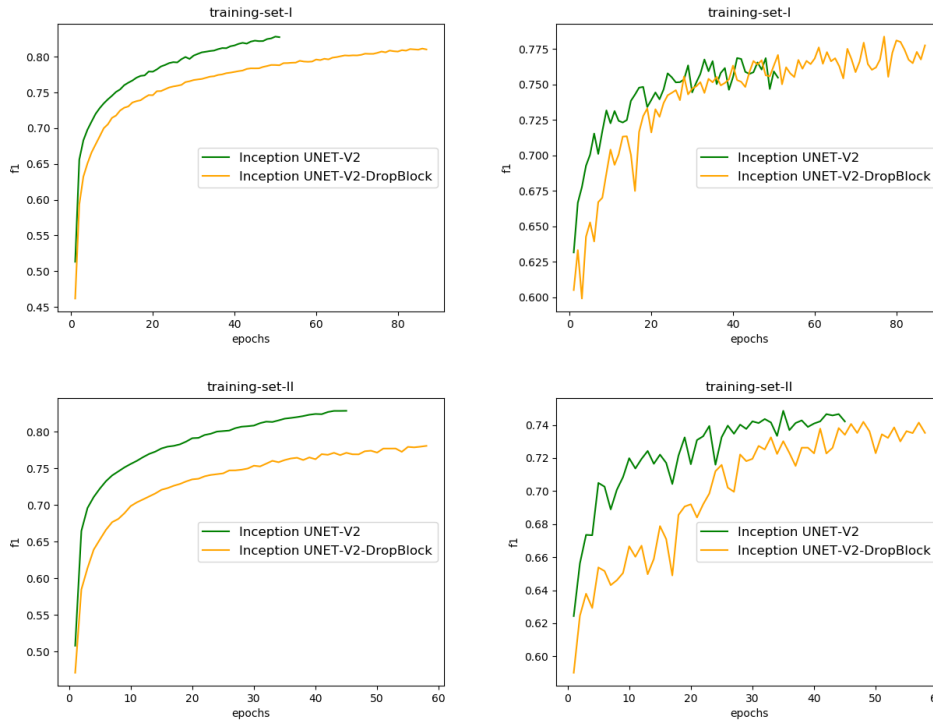

Figure 6. Effect of DropBlock module in training (left) and validation (right) accuracy

3.2 Implementation Details

The proposed architecture is implemented using TensorFlow library. Adam algorithm is used for training and initial learning rate is set to 0.00001. NVIDIA RTX 2070 is used to accelerate training. Early stopping criteria with the patient value of 10 is used for validation during training process. It is important to monitor whether the network is overfitting.

Dice coefficient is used as a loss function to measure the error between estimated segmentation mask and ground truth to optimize the network parameters. Figure 8 and Figure 9 show training and validation accuracies for the training set-I and the training set-II, respectively. INCSA-UNET* represents the architecture using channel attention, while INCSA-UNET represents the architecture using spatial attention in the figures. Accordingly, the biggest number of training epochs is obtained with SA-UNET for training set-I, while INCSA-UNET* has the biggest number of training epochs for training set-II. UNET also has a higher training epochs for both training sets and it is seen that Inception UNET has the least number of training epochs of all. Proposed INCSA-UNET has the best validation accuracy for both validation sets while UNET++ has the lowest performance. It is seen that using

DropBlock and attention modules in INCSA-UNET increases the convergence time during training compared to Inception UNET-V2, but training and validation accuracies show that INCSA-UNET prevents overfitting and has better performance, as detailed in Section 5.

4 EXPERIMENTS

4.1 Dataset

Proposed INCSA-UNET architecture is evaluated on Massachusetts building dataset [33]. The dataset contains images with a spatial resolution of 1 meter and 1 500 × 1 500 resolution, covering the urban and suburban areas of Boston. Sample images from dataset are shown in Figure 7. The dataset is split into a training set of 137 images, a test set of 10 images, and a validation set of 4 images. In this study, 27 training images with many missing regions are excluded, and the remaining 110 images are used for the training.

Figure 7. Sample images and corresponding labelled buildings from dataset

We extract patches from images to train the network with lower resolution images. A total of 6 076 sub-images are obtained by subtracting 224 × 224 resolution patches from all images in the dataset with the sliding window approach. We create two training sets by using 6 076 sub-images to apply two-fold cross-validation experiments. 5 390 sub-images are extracted from 110 training set images for training set-I. 196 and 490 sub-images are extracted for validation and test, respectively. In the training set-II, 3 675 sub-images are used for training, 2 205 sub-images are used for test, and 196 sub-images are used for validation. In short, about 88 percent of the total number of sub-images is used for training in training set-I, while about 60 percent is used in training set-II.

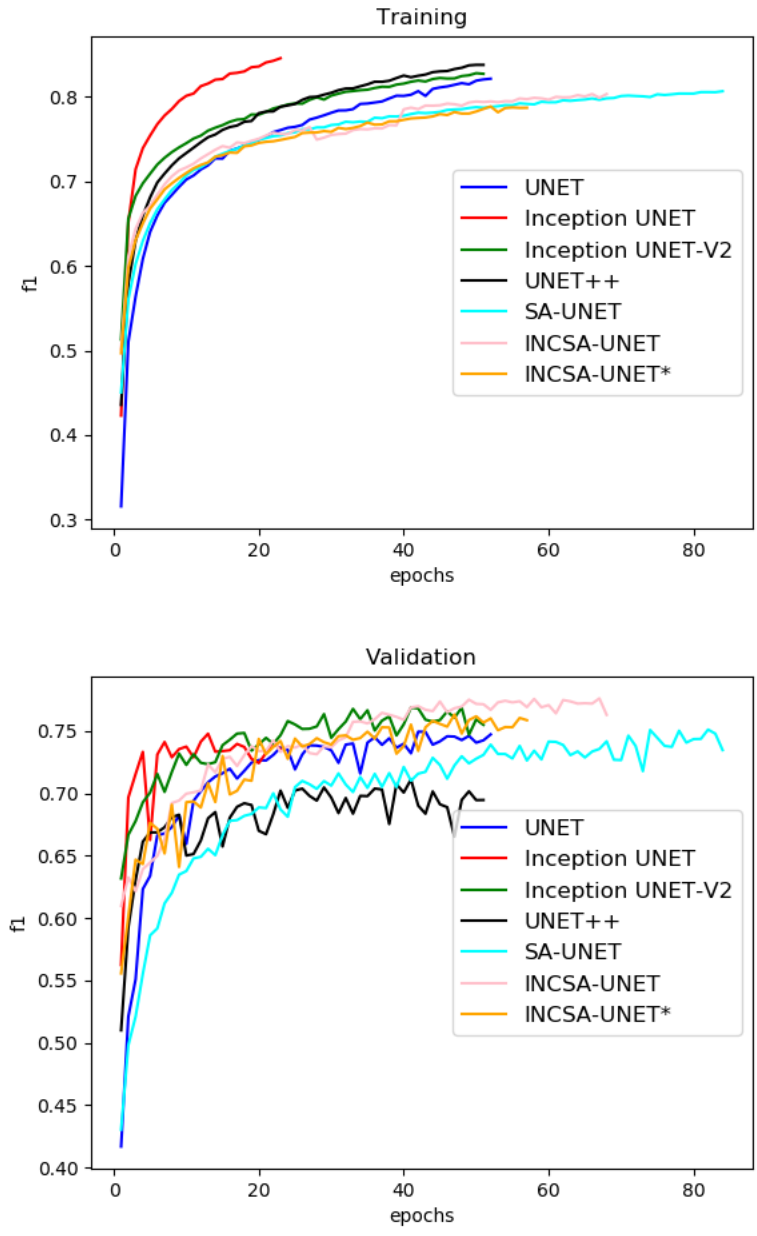

Figure 8. Training (top) and validation (bottom) accuracies of training set-I

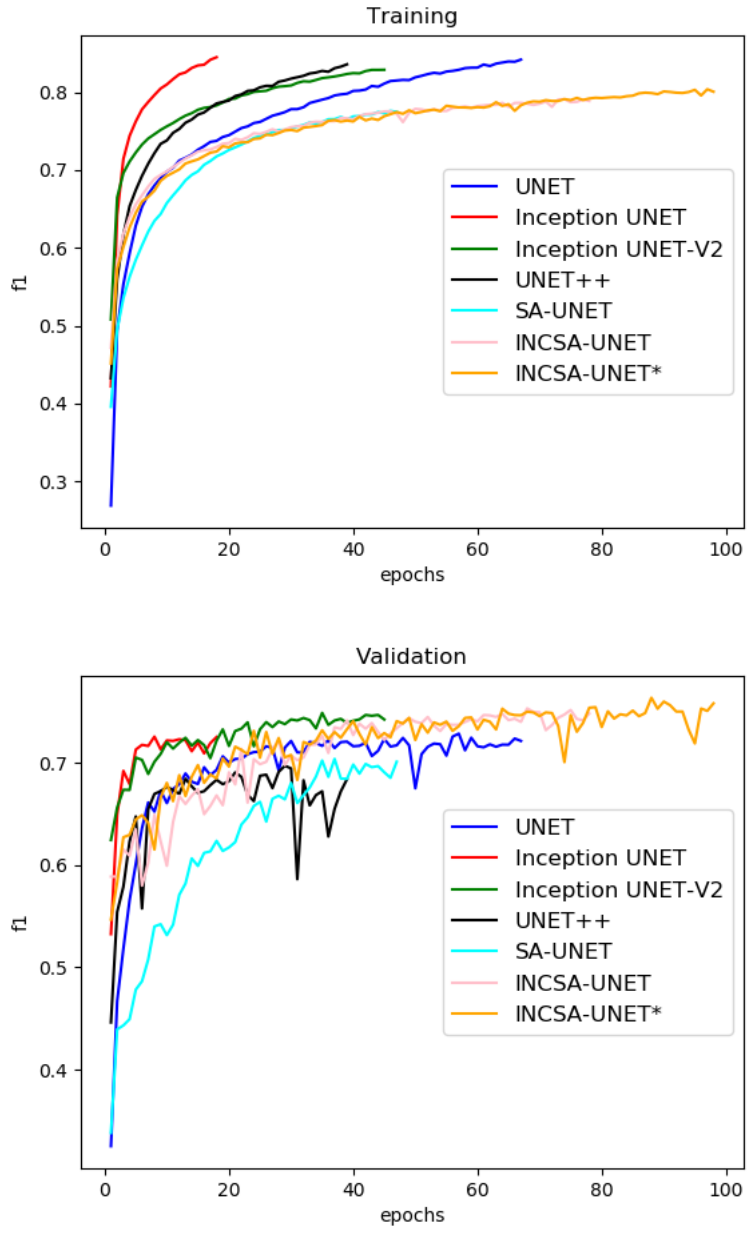

Figure 9. Training (top) and validation (bottom) accuracies of training set-II

4.2 Evaluation Metrics

In order to evaluate proposed model, we compare the segmentation results with the corresponding ground truth. Specifity (Sp), Precision (P), Recall (R), F-score ($F1$) and *Kappa* metrics are used for quantitative performance comparison. Sp measures how many of the background pixels are correctly predicted. P represents the ratio of the number of pixels correctly classified as buildings to the total number of pixels estimated as buildings. R is the ratio of the number of pixels correctly classified as buildings to the actual number of building pixels. The $F1$ is the harmonic mean of the P and R values. In Equations (12), (13) and (14): TP (True Positive) refers to the number of correctly classified building pixels, TN (True Negative) refers to the number of the background pixels classified by the model as background, FN (False Negative) refers to the number of building pixels classified by the model as background, and FP (False Positive) refers to the number of background pixels classified by the model as building. Cohen's *Kappa* coefficient is also used to measure the reliability of comparison [34]. *Kappa* score can handle imbalanced data well according to $F1$. If a class is much more dominant than the other, $F1$ score may have a higher value. But *Kappa* score will be lower and give a warning about the validity of the model. So that we also used *Kappa* in the performance comparison.

$$Sp = \frac{TN}{TN + FP}, \tag{12}$$

$$P = \frac{TP}{TP + FP}, \tag{13}$$

$$R = \frac{TP}{TP + FN}, \tag{14}$$

$$F1 = \frac{2PR}{P + R}. \tag{15}$$

5 RESULTS

Firstly, we evaluate the effect of using DropBlock in Inception UNET-V2 to prove that it can improve the performance of segmentation. The ablation experiment given in Table 1 shows that using DropBlock improves the segmentation performance in all metrics for both training sets.

The performance evaluation of the proposed architecture is carried out with qualitative and quantitative analyzes. For qualitative evaluation, the estimated building segmentation results from sample images on the dataset are shown in the Figure 10. Red, yellow and green colors over Figure 10 indicate missed building pixels (FN), true detections (TP) and wrong estimations (FP), respectively. While the clearest results are obtained with the proposed architecture, details and exact borders are missed with the classical UNET and SA-UNET, as shown on the fourth row. In the third row in Figure 10, the image contains relatively smaller buildings and it is

	Method	**Sp**	***P***	***R***	***F1***	***Kappa***
test set-I	INC UNET-V2	0.9507	0.7923	0.8105	0.8009	0.7538
	INC UNET-V2 + DropBlock	**0.9555**	**0.8030**	**0.8264**	**0.8141**	**0.7710**
test set-II	INC UNET-V2	**0.9596**	0.6815	0.7574	0.7126	0.6691
	INC UNET-V2 + DropBlock	0.9539	**0.7513**	**0.8055**	**0.7764**	**0.7358**

Table 1. Ablation study for Dropout block

seen that INCSA-UNET segmentation result produces the least red color (missed buildings). It means that INCSA-UNET has lowest wrong detections between all methods as seen on first and second rows also. It is remarkable that the rightest building in the image in the first row can only be detected with the proposed method.

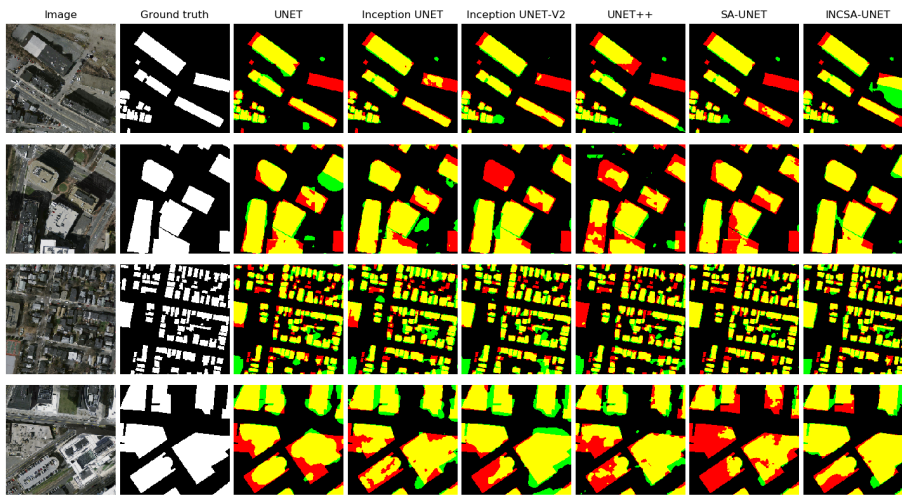

Figure 10. Annotations and segmentation results for sample image patches on Massachusetts dataset. Yellow: TP, Red: FN, Green: FP.

Tables 2 and 3 show the performance metrics obtained with UNET, Inception UNET, Inception UNET-V2, UNET++, SA-UNET and proposed INCSA-UNET. INCSA-UNET* represents the results of using channel attention in the proposed architecture. Even if using both attention mechanisms improve the performance, the experiments show that spatial attention is more effective. Quantitative performance comparison shows that best R, $F1$ and $Kappa$ results are obtained with the INCSA-UNET while SA-UNET has the best Sp and P value. Already, the best true detections ratio in qualitative results also supports the best R value of INCSA-

UNET. Quantitative results also show that even if SA-UNET has best P value, it has lower R value compared to UNET and Inception UNET. Inception blocks enhance the UNET performance and models including Inception blocks are also better compared to UNET++. Using DropBlock prevents the overfitting, spatial attention module enhances key areas and restrains irrelevant areas. In brief, it has been observed that combining Inception, DropBlock and spatial attention modules in an architecture gives good results. According to the results of the quantitative and qualitative evaluations, it is seen that INCSA-UNET architecture outperforms the other models for the most important metrics $F1$ and *Kappa* scores.

Methods	**Sp**	**P**	**R**	**F1**	**Kappa**
UNET	0.9473	0.7717	0.7798	0.7752	0.7223
INC UNET	0.9522	0.7862	0.7802	0.7823	0.7323
INC UNET-V2	0.9507	0.7923	0.8105	0.8009	0.7538
UNET++	0.9499	0.7597	0.7439	0.7494	0.6924
SA-UNET	**0.9684**	**0.8345**	0.7501	0.7875	0.7415
INCSA-UNET	0.9574	0.8097	**0.8184**	**0.8135**	**0.7706**
INCSA-UNET*	0.9600	0.8099	0.7895	0.7984	0.7529

Table 2. Performance comparison of INCSA-UNET and other methods on test set-I

Methods	**Sp**	**P**	**R**	**F1**	**Kappa**
UNET	0.9144	0.6605	0.7427	0.6959	0.6485
INC UNET	0.9277	0.6893	0.7262	0.7033	0.6585
INC UNET-V2	0.9236	0.6815	0.7574	0.7126	0.6691
UNET++	0.8879	0.6188	0.7346	0.6652	0.6118
SA-UNET	**0.9704**	0.7937	0.6967	0.7363	0.6947
INCSA-UNET	0.9575	0.7638	**0.8025**	**0.7819**	0.7430
INCSA-UNET*	0.9675	**0.7988**	0.7675	0.7813	**0.7440**

Table 3. Performance comparison of INCSA-UNET and other methods on test set-II

Number of trainable and total parameters for each architecture is also given in Table 4. Inception UNET model uses Inception blocks in both of encoding and decoding stages, so that it extremely increases the number of parameters. But Inception UNET-V2 applies Inception blocks only in encoding stage and number of filters are reduced to decrease the the number of total parameters. UNET has about 7.7 M total trainable parameters while SA-UNET and INCSA-UNET have 9 M and 13.8 M, respectively. Experiments show that INCSA-UNET has performance improvements up to about 3 % and 5 % for $F1$ scores and does not extremely increase the number of parameters.

Methods	Number of Parameters	
	Trainable	**Total**
UNET	7 760 385	7 760 385
INC UNET	67 155 905	67 174 497
INC UNET-V2	11 987 921	11 991 281
UNET++	37 648 292	37 698 666
SA-UNET	9 093 219	9 098 851
INCSA-UNET	13 879 517	13 882 877
INCSA-UNET*	13 313 787	13 318 683

Table 4. Number of trainable and total parameters

6 CONCLUSIONS

In this study, the UNET model, which is successful in segmentation problems, is used as a backbone network for building detection from aerial images. Proposed architecture is inspired by the success of Inception UNET architecture expanded with Inception blocks, and SA-UNET architecture using DropBlock and attention modules. The architecture applies attention blocks in the middle of the network. We have seen that two different attention modules used in the proposed architecture increase the performance, but spatial attention performs slightly better. Compared to the classical UNET and SA-UNET, the model has a good ability to learn buildings with different shapes and size. The experimental results demonstrate that spatial attention module and DropBlock are effective to focus on important features and prevent overfitting, respectively. Two level Inception approach in encoding stage also enhances the feature extraction in the proposed method. We conclude that INCSA-UNet is a general network and can be applied to different segmentation problems.

REFERENCES

[1] Jensen, J. R.—Cowen, D. C.: Remote Sensing of Urban/Suburban Infrastructure and Socio-Economic Attributes. Photogrammetric Engineering and Remote Sensing, Vol. 65, 1999, pp. 611–622.

[2] Yang, H. L.—Yuan, J.—Lunga, D.—Laverdiere, M.—Rose, A.—Bhaduri, B.: Building Extraction at Scale Using Convolutional Neural Network: Mapping of the United States. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 11, 2018, No. 8, pp. 2600–2614, doi: 10.1109/jstars.2018.2835377.

[3] Liow, Y. T.—Pavlidis, T.: Use of Shadows for Extracting Buildings in Aerial Images. Computer Vision, Graphics, and Image Processing, Vol. 49, 1990, No. 2, pp. 242–277, doi: 10.1016/0734-189X(90)90139-M.

[4] Sirmacek, B.—Unsalan, C.: Building Detection from Aerial Images Using Invariant Color Features and Shadow Information. 2008 23rd International Symposium on Computer and Information Sciences, 2008, pp. 1–5, doi: 10.1109/iscis.2008.4717854.

[5] Ok, A. O.—Senaras, C.—Yuksel, B.: Automated Detection of Arbitrarily Shaped Buildings in Complex Environments from Monocular VHR Optical Satellite Imagery. IEEE Transactions on Geoscience and Remote Sensing, Vol. 51, 2013, No. 3, pp. 1701–1717, doi: 10.1109/tgrs.2012.2207123.

[6] Benedek, C.—Descombes, X.—Zerubia, J.: Building Detection in a Single Remotely Sensed Image with a Point Process of Rectangles. 20th International Conference on Pattern Recognition, 2010, pp. 1417–1420, doi: 10.1109/icpr.2010.350.

[7] Ok, A. O.: Automated Detection of Buildings from Single VHR Multispectral Images Using Shadow Information and Graph Cuts. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 86, 2013, pp. 21–40, doi: 10.1016/j.isprsjprs.2013.09.004.

[8] Sumer, E.—Turker, M.: An Adaptive Fuzzy-Genetic Algorithm Approach for Building Detection Using High-Resolution Satellite Images. Computers, Environment and Urban Systems, Vol. 39, 2013, pp. 48–62, doi: 10.1016/j.compenvurbsys.2013.01.004.

[9] Inglada, J.: Automatic Recognition of Man-Made Objects in High Resolution Optical Remote Sensing Images by SVM Classification of Geometric Image Features. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 62, 2007, No. 3, pp. 236–248, doi: 10.1016/j.isprsjprs.2007.05.011.

[10] Shaker, I. F.—Abd-Elrahman, A.—Abdel-Gawad, A. K.—Sherief, M. A.: Building Extraction from High Resolution Space Images in High Density Residential Areas in the Great Cairo Region. Remote Sensing, Vol. 3, 2011, No. 4, pp. 781–791, doi: 10.3390/rs3040781.

[11] Boonpook, W.—Tan, Y.—Ye, Y.—Torteeka, P.—Torsri, K.—Dong, S.: A Deep Learning Approach on Building Detection from Unmanned Aerial Vehicle-Based Images in Riverbank Monitoring. Sensors, Vol. 18, 2018, No. 11, Art. No. 3921, doi: 10.3390/s18113921.

[12] Liu, J.—Gong, M.—Qin, K.—Zhang, P.: A Deep Convolutional Coupling Network for Change Detection Based on Heterogeneous Optical and Radar Images. IEEE Transactions on Neural Networks and Learning Systems, Vol. 29, 2018, No. 3, pp. 545–559, doi: 10.1109/tnnls.2016.2636227.

[13] Chen, Y.—Jiang, H.—Li, C.—Jia, X.—Ghamisi, P.: Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. IEEE Transactions on Geoscience and Remote Sensing, Vol. 54, 2016, No. 10, pp. 6232–6251, doi: 10.1109/TGRS.2016.2584107.

[14] Ševo, I.—Avramović, A.: Convolutional Neural Network Based Automatic Object Detection on Aerial Images. IEEE Geoscience and Remote Sensing Letters, Vol. 13, 2016, No. 5, pp. 740–744, doi: 10.1109/LGRS.2016.2542358.

[15] Ghiasi, G.—Lin, T. Y.—Le, Q. V.: DropBlock: A Regularization Method for Convolutional Networks. 2018, arXiv: 1810.12890.

[16] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.):

Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[17] Long, J.—Shelhamer, E.—Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.

[18] Ronneberger, O.—Fischer, P.—Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (Eds.): Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer, Cham, Lecture Notes in Computer Science, Vol. 9351, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[19] Guo, C.—Szemenyei, M.—Wang, W.—Chen, B.—Fan, C.: SA-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation. 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 1236–1242, doi: 10.1109/ICPR48806.2021.9413346.

[20] Guo, C.—Szemenyei, M.—Pei, Y.—Yi, Y.—Zhou, W.: SD-Unet: A Structured Dropout U-Net for Retinal Vessel Segmentation. 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), 2019, pp. 439–444, doi: 10.1109/bibe.2019.00085.

[21] Zhao, P.—Zhang, J.—Fang, W.—Deng, S.: SCAU-Net: Spatial-Channel Attention U-Net for Gland Segmentation. Frontiers in Bioengineering and Biotechnology, Vol. 8, 2020, Art. No. 670, doi: 10.3389/fbioe.2020.00670.

[22] He, K.—Zhang, X.—Ren, S.—Sun, J.: Deep Residual Learning for Image Recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.

[23] Xu, Y.—Wu, L.—Xie, Z.—Chen, Z.: Building Extraction in Very High Resolution Remote Sensing Imagery Using Deep Learning and Guided Filters. Remote Sensing, Vol. 10, 2018, No. 1, Art. No. 144, doi: 10.3390/rs10010144.

[24] Szegedy, C.—Liu, W.—Jia, Y.—Sermanet, P.—Reed, S.—Anguelov, D.—Erhan, D.—Vanhoucke, V.—Rabinovich, A.: Going Deeper with Convolutions. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, doi: 10.1109/cvpr.2015.7298594.

[25] Delibasoglu, I.—Cetin, M.: Improved U-Nets with Inception Blocks for Building Detection. Journal of Applied Remote Sensing, Vol. 14, 2020, No. 4, Art. No. 044512, doi: 10.1117/1.jrs.14.044512.

[26] Zhou, Z.—Siddiquee, M. M. R.—Tajbakhsh, N.—Liang, J.: Unet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Stoyanov, D. et al. (Eds.): Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA 2018, ML-CDS 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11045, 2018, pp. 3–11, doi: 10.1007/978-3-030-00889-5_1.

[27] Delibaşoğlu, İ.—Çetin, M.: Building Segmentation with Inception-Unet and Classical Methods. 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1–4, doi: 10.1109/siu49456.2020.9302155.

[28] Liu, Y.—Piramanayagam, S.—Monteiro, S. T.—Saber, E.: Dense Semantic Labeling of Very-High-Resolution Aerial Imagery and LiDAR with Fully-Convolutional Neural Networks and Higher-Order CRFs. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1561–1570, doi: 10.1109/cvprw.2017.200.

[29] Pan, X.—Gao, L.—Marinoni, A.—Zhang, B.—Yang, F.—Gamba, P.: Semantic Labeling of High Resolution Aerial Imagery and LiDAR Data with Fine Segmentation Network. Remote Sensing, Vol. 10, 2018, No. 5, Art. No. 743, doi: 10.3390/rs10050743.

[30] Maggiori, E.—Charpiat, G.—Tarabalka, Y.—Alliez, P.: Recurrent Neural Networks to Correct Satellite Image Classification Maps. IEEE Transactions on Geoscience and Remote Sensing, Vol. 55, 2017, No. 9, pp. 4962–4971, doi: 10.1109/tgrs.2017.2697453.

[31] Ghiasi, G.—Lin, T. Y.—Le, Q. V.: NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7029–7038, doi: 10.1109/cvpr.2019.00720.

[32] Huang, Z.—Fang, Y.—Huang, H.—Xu, X.—Wang, J.—Lai, X.: Automatic Retinal Vessel Segmentation Based on an Improved U-Net Approach. Scientific Programming, Vol. 2021, 2021, Art. No. 5520407, doi: 10.1155/2021/5520407.

[33] Mnih, V.: Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto, Canada, 2013.

[34] McHugh, M. L.: Interrater Reliability: The Kappa Statistic. Biochemia Medica, Vol. 22, 2012, No. 3, pp. 276–282, doi: 10.11613/bm.2012.031.

Ibrahim Delibaşoğlu is Research Assistant at the Sakarya University, Turkey. He received his M.Sc. and Ph.D. degrees in computer science from the Yalova University, Turkey in 2013 and 2020, respectively. His current research interests include deep learning, object segmentation, motion detection and hyperspectral image analysis.