Superpixel-based Multiscale CNN Approach towards Multiclass Object Segmentation from UAV-captured Aerial Images

Tanmay Kumar Behera, *Member, IEEE,* Sambit Bakshi, *Senior Member, IEEE,* Michele Nappi, *Senior Member, IEEE,* Pankaj Kumar Sa, *Member, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) are promising remote sensors capable of reforming remote sensing applications. However, for artificial intelligence (AI)-guided tasks, like land cover mapping and ground-object mapping, most deep learning-based architectures fail to extract scale-invariant features, resulting in poor performance accuracy. In this context, the article proposes a superpixel-aided multiscale convolutional neural network (CNN) architecture to avoid misclassification in complex urban aerial images. The proposed framework is a two-tier deep learning-based segmentation architecture. In the first stage, a superpixel-based simple linear iterative cluster (SLIC) algorithm produces superpixel images with crucial contextual information. The second stage comprises a multiscale CNN architecture that uses these information-rich superpixel images to extract scale-invariant features for predicting the object class of each pixel. Two UAV-image-based aerial image datasets: *NITR-Drone* dataset and *urban drone dataset* (UDD) are considered to perform the experimentation. The proposed model outperforms the considered state-of-the-art methods with an intersection of union (IoU) of 76.39% and 86.85% on UDD and NITRDrone datasets, respectively. Experimentally obtained results prove that the proposed architecture performs superior by achieving better performance accuracy in complex and challenging scenarios.**

***Index Terms*—VHR, Deep Learning, Aerial Image, UAV, CNN, Mutliscale CNN, Superpixel, SLIC, Semantic Segmentation**

I. Introduction

OUR world has come a long way since the launch of the first satellite into space, and we are in an era fifty centuries ahead of it, significantly changing our daily lives. The technological advancement in space technology and remote sensing (RS) sector can be analyzed from the ever-growing number of operated satellites around the earth since 1957 till date. As per one statistic, a remarkable jump of $1,070$ satellites is noticed in 2019 to 2020 making the total number of satellites to $3,368$ [1].

A massive number of very high resolution (VHR) images are on a daily basis by these earth observation satellites such as the WorldView series, Landsat series, and RESOURCESAT [2], [3]. These captured images have been used to address many societal issues at a higher level through different RS applications. However, certain gaps in satellite-based RS applications make it difficult to go through the tropical regions, which are mostly covered by clouds [4]. This opens up a space for the new edge remote sensors in the form of UAVs that can truly improve the spatial, temporal, and spectral resolution of satellite-captured data at different scales. UAVs can help satellites overcome their limitations and accomplish particular tasks through real-time assessment and monitoring actions in different scenarios. These small devices have taken their usage to a whole different level, managing various issues of our day-to-day lives through several RS applications such as traffic management, urban management in smart cities, land cover classification, fishery management, forest area management, *etc.* at a lower scale as compared to the satellites.

Mostly, the images captured by the UAVs are of high resolution and provide a detailed view of a particular area in a scene. Among several image data acquisition tasks for UAV-based RS images, semantic segmentation is one of the emerging and challenging areas for computer vision researchers. Here, the task is to predict the pixel-level object class according to the semantic information represented by that pixel in the captured aerial image. Recent years have witnessed tremendous progress in deep learning-based approaches like CNNs, which have proved their significance in attending semantic segmentation tasks [5]–[7].

UAV-based aerial image analysis systems differ from satellite image analysis systems concerning their use cases and approaches to solving tasks in various application domains. Some of these applications include detecting objects, such as roads, buildings, vegetation, and vehicles that play a vital role in critical applications like military target identification and damage estimation and rescue operations in natural disasters [8], [9]. Therefore, developing a robust aerial image segmentation algorithm is needed for such critical tasks. However, several inherent challenges, such as image resolution, large field of view (FOV), and diversified and complex backgrounds, make the task more challenging (in UAV-inspired segmentation tasks). Many popular semantic segmentation frameworks designed for satellite-captured images are unsuitable for UAV-borne remote sensing image-based tasks. It is generally due to the specificity of UAV-captured RS images. Another area for improvement of these popular approaches is that the purpose of UAV-inspired remote sensing differs from satellite remote sensing. Satellite-borne RS images focus on object extraction and land cover analysis in a larger area. In contrast, UAV-

Manuscript received xxxx xx, 2022; revised xxxx xx, 2022.

Tanmay Kumar Behera, Sambit Bakshi, Pankaj Kumar Sa are with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India, 769008. (e-mail: tanmay.nitr@gmail.com, sambitbaksi@gmail.com, pankajsa@nitrkl.ac.in)

Michele Nappi is with the Department of Computer Science, University of Salerno, Italy (e-mail: mnappi@unisa.it)

borne RS images are meant to extract information at a smaller scale in a smaller area. Hence, these large numbers of high-resolution UAV remote sensing images aim to analyze the objects more accurately. It is because the UAV-borne remote sensing images possess richer contextual information to work on addressing the UAV RS-inspired tasks.

A. Motivation

1) Motivation of using superpixel algorithm: A group of pixels can be termed a superpixel, where the members of the superpixels share some common attributes compared to the non-members. As suggested by definition, the use of superpixel techniques is more beneficial for image segmentation tasks [10], [11]. Superpixel images have several advantages, such as reducing the computational cost by representing pixels inside a superpixel. Thus, they can be used to reduce the overhead incurred by the deep learning frameworks in terms of time and memory. Similarly, superpixels can extract essential regional features, which are more distinctive than the standard pixel-wise features used in several computer vision tasks. They are adaptive due to their shape and size, containing more local and spatial features [12]. Thus, having these features, superpixels can be generated at different scales, which can be used in multiscale-inspired applications with specific parameter settings [13].

2) Motivation of using a multiscale architecture: The traditional segmentation techniques could perform better due to their low generalization ability. Thus, developing a deep learning-based robust framework is essential to strengthening the aerial image segmentation process. However, certain underlying complications in these deep learning frameworks could lead to false classification. The issues lie within the process through which the image patches are fed to the architecture during training. The CNN architecture misses many high-level feature sets with strict image sizes, thereby losing crucial contextual information. These missing features are essential in multi-object semantic segmentation, especially in aerial images. The different flight heights of a UAV can create ambiguity for a model leading to poor generalization for several small-scale objects. It is where a multiscale sampling process can become a savior in extracting and gathering the spatial-level object features. Multiscale features are desirable to realize the abstraction of the image at different scales. Introducing a multiscaling process to the CNN framework can help it learn multiple heterogeneous scale-invariant features, which can lower the misclassification rate.

B. Contribution

In this work, we have proposed a multiscale CNN framework for UAV-captured images. In addition to this, the proposed approach benefits from the SLIC-inspired superpixel techniques to generate the superpixel images, which act as the input for the multiscale CNN architecture. Some of the major contributions of this work are summarized as follows:

- The proposed deep-learning framework is a two-staged architecture for aerial scene segmentation. The first stage uses the UAV-captured images to perform coarse-level segmentation using the SLIC superpixel technique to generate superpixel images. Superpixels carry more spatial information than normal pixels and provide a more compact and convenient representation. Hence, they are useful for computationally demanding applications.
- In the second stage, a multiscale CNN architecture is proposed to analyze the given superpixels for pixel-level classification. Here, the superpixel images are sampled at different scales to the multiscale module to extract the scale-invariant features to perform multiclass segmentation.
- The proposed model is evaluated over the two UAV-borne aerial image datasets to ensure the robustness of the proposed architecture in real-world settings.
- Moreover, the model is also evaluated by changing some important parameters to show its improved behavior with the superpixel and multiscale convolution to detect small-scale ground objects.

The rest of the manuscript is organized as follows: The existing semantic object segmentation approaches are discussed in Section II. Similarly, Section III presents the different methodologies used in the proposed approach, which is followed by Section IV. Section IV and Section V discuss the detailed structure of implementation and overview of the obtained results, respectively. Similarly, a discussion section is also added as Section VI. Finally, Section VII briefly describes the conclusion drawn from the whole article.

II. Related Work and Background Study

This section briefly discusses the different approaches proposed by the researchers in aerial scene understanding. The evolution of deep learning and multi-scale learning algorithms towards aerial scene understanding problems are discussed in this section.

A. Traditional Approaches in Aerial Image Segmentation

Aerial images are the images of the earth captured from above it, where the space-borne remote sensors or satellites were the only option until the UAV-based technology pitched in this work to leverage the load incurred on a satellite at a lower scale. These devices have been widely used in various RS applications such as ground objects detection: cars, roads, buildings, trees, and pedestrians, which is an essential aspect of many projects viz. agriculture mapping, urban mapping, forest mapping, *etc.* The UAV images are a bit complicated compared to the satellite images due to the detailed and vast population of diversified objects, making the task more complex and challenging. Previously attempted research works by computer vision researchers are based on the rule descriptor influenced methods for object-level feature extraction, specifically in building extraction [14], road detection [15], [16]. However, due to poor generalization concerning aerial data, the hierarchical rule-based approaches miss out on several significant features. Conventional classifiers employed machine learning techniques that extract the local features from the input pixel intensities through simple arithmetic combinations [17], [18]. Researchers also proposed discriminating classifiers

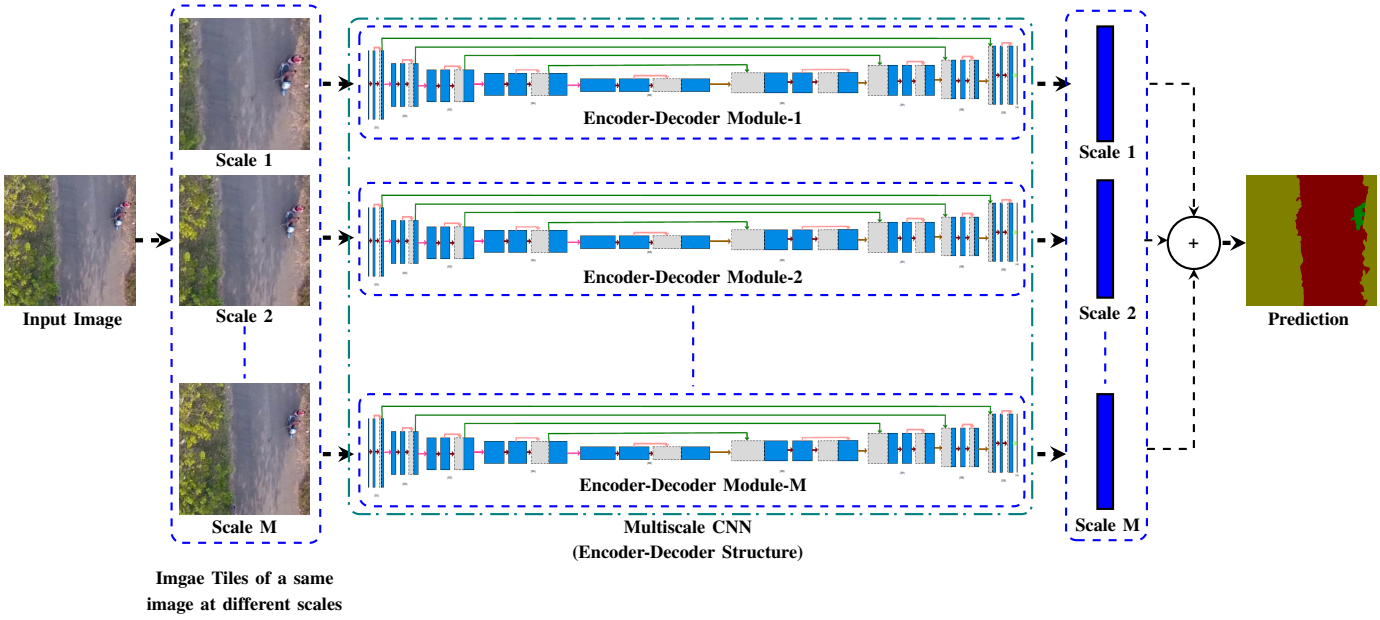

Figure 1. Architectural framework of multiscale CNN architecture

like boosting and random forest to evaluate the redundant local feature maps for training purposes [19]–[21]. In an aerial image segmentation problem, the global features are equally essential as local features [22]. In [23], authors have used marked point processes to build architectural models and road network topologies through probabilistic priors defined for global knowledge gain. Conditional random fields, also known as CRFs, are also used for object-level segmentation, and detection from the aerial images [24]. Similarly, Wang *et al.* [25] have proposed a fusion approach using a superpixel-based labeling technique and Markov random field towards aerial video segmentation.

B. Deep Learning Approaches in Aerial Image Segmentation

Unlike conventional machine-learning techniques, deep-learning algorithms have no requirement for feature definition steps. They learn the critical distinguishing features from an input dataset according to the provided task. These methodologies were proposed back in the 80's at a time when there was limited computing power and available training data [26], [27]. These algorithms announced their return with [28] in 2012 and managed to achieve impressive outcomes for the ImageNet challenge [29], creating hope in the research community with tons of opportunities. Several layers are stacked on one another in the proposed baseline models to learn and analyze the essential local-global feature sets from the input images. One of the crucial aspects of deep CNN architectures lies in its ability to parallelize both training and inference through GPUs.

CNN has started its journey with the image classification problem, and in a short period, they have been successfully able to address computer vision problems like object detection [30], tracking [31], and object-level segmentation [32]. The usage of convolution network frameworks has not been restricted to classical classification tasks but can also be noticed in aerial scene parsing using RS images [33]. A number of common RS tasks in this domain comprise buildings extraction [34], [35], road networks extraction [36]–[38], vegetation extraction [39]. Aerial scene understanding based on an encoder-decoder-based fully convolutional network (FCN) structure is proposed by [5], [40] that yields an explicit label image depicting the contexts associated with each pixel. Then the extracted feature maps propagate through an expansion module to up-sample the reduced image back to the original resolution. In [41], Xie *et al.* have proposed a multiscale densely-connected CNN architecture for remote sensing-based hyperspectral aerial image (HSAI) classification. Similarly, Fan *et al.* [42] have presented a superpixel-aided deep-sparse-representation technique to construct hierarchical architecture to understand HSAI context information. This gathered information (features) obtained from the multi-layered network is concatenated and trained by a support vector machine (SVM) classifier. Moreover, for small-scale applications, UAV usage is increasing, and collected data have been utilized in many crucial RS applications. Computer vision researchers [43], [44] have provided several solution approaches to address the existing issues using deep learning-based architectures. Authors have recommended a deep learning-based framework inspired by Fast R-CNN and Faster R-CNN for vehicle extraction from aerial images [45]. The two networks are combined to gather important feature space, which can be used to detect vehicles semantically. Moreover, datasets are the backbone of the success behind deep learning frameworks. A thorough and detailed analysis of the available UAV image datasets for computer vision researchers in conducting research towards UAV-inspired applications is presented in [46].

III. Proposed Methodology

The manuscript proposes a superpixel-aided multiscale deep learning framework that semantically segments the aerial im-

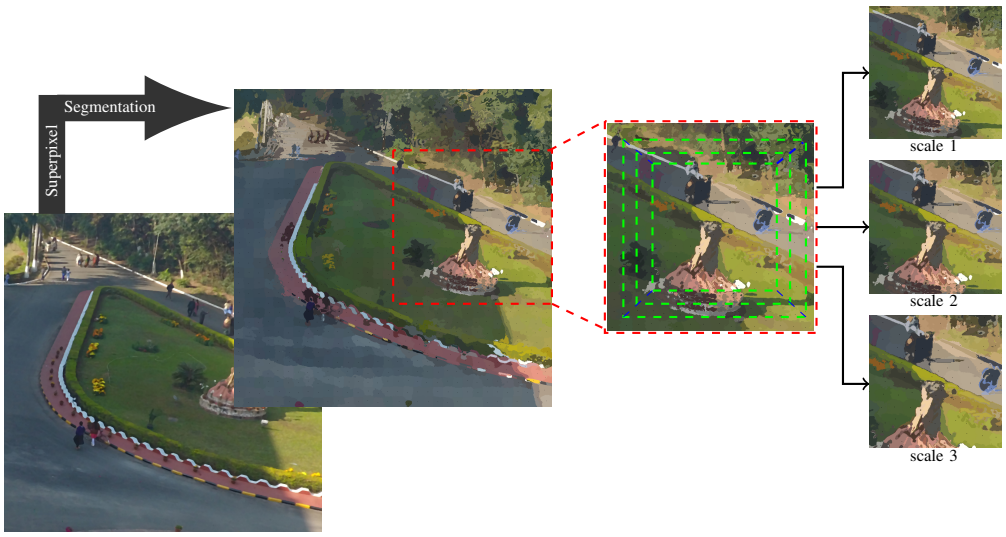

Figure 2. Generation of images at multiple scales from the superpixel images

ages captured by UAVs. This section discusses each module used in the proposed deep architecture.

A. Overview

The proposed framework consists of two modules: a superpixel module and a multiscale CNN module to work on extracting the scale-invariant features. In the backend of the architecture, the superpixel algorithm works to determine the essential scale-invariant features. As the first phase of the whole segmentation process, the superpixel technique narrows down the texture and color-based features. These extracted features are considered the input to the second phase of the proposed framework, where these superpixel images are used to produce the final segmentation map. The superpixel images help the deep learning architecture to be implemented quickly, reducing the overall training and validation/testing time (in most instances). The architectural overview is presented in Fig. 1. Each module of the proposed architecture is explained in the following subsections.

B. Superpixel Method

A number of pixels sharing common characteristics can be referred to as a superpixel. They can carry more information than simple pixels and provide a more convenient and compact representation that could be useful for computationally demanding applications. Some of these applications include medical imaging [47], object detection, scene segmentation, video surveillance, *etc.* Among the superpixel algorithms, Simple Linear Iterative Clustering, also known as SLIC, has been widely used [48], [49].

Generally, SLIC-based superpixel algorithms generate relatively uniform and compact superpixels based on the spatial and color proximity of pixels in an image plane. Five dimensional $(5D)$ [laβxy] space is utilized by this approach, where [laβ] represents the pixel color vector and $[xy]$ indicates the position of a pixel. Hence, it should be normalized so that the Euclidean distance can be employed in $5D$ space. Hence, the maximum spatial distance within a cluster should lie within a sampling interval, S, and can be represented as follows:

$$S = \sqrt{\frac{N}{K}} \tag{1}$$

where, N = Number of pixels in the input image
K = Number of Superpixels required
$\frac{N}{K}$ = Approximate area of a superpixel

The superpixel algorithm considers the desired number of superpixels of approximately equal sizes (K). The cluster centers C_k can be represented as $C_k = [l_k, a_k, b_k, x_k, y_k]$, where k varies between a range of 1 to K at a regular interval of S within a grid. The spatial extent of a superpixel is generally S^2 (approximate area of a superpixel). Thus, an assumption can be made corresponding to its cluster center that associated pixels fall within a region $2S \times 2S$ area around the superpixel head-on xy plane. Hence, the normalized distance (D_s) can be calculated as the sum of the lab color space distance $(d_{la\beta})$ and XY plane distance (d_{xy}) normalized by the grid interval S and is given as follows:

$$D_s = d_{la\beta} + (\frac{m}{S}) * d_{xy} \tag{2}$$

where, $dla\beta = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$
$dxy = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$
$m = Maximum\ Color\ Distance$

Like spatial distance, the color-related distance plays a crucial role in estimating the normalized distance (D_s) in SLIC algorithm. Estimating color distance is a complex job as the color-based distance may vary rapidly from cluster to image and image to image. Thus, to avoid such a problem, a constant m is introduced that controls the compactness of a superpixel. The higher the value of m, the more compact the cluster is. Reducing the compactness factor (m) (lied within $[5 - 40]$) gives us images that are more closely related to the original images keeping the relevant object features.

The Superpixel module acts as the first-level optimizer to transform complex aerial images into more compact-sized superpixel images. Pixels representing a single superpixel share similar visual attributes in a superpixel image. Thus, the superpixel images carry more information values than the usual ones. The UAV-based VHR aerial imageries are given as inputs to the superpixel module to produce the superpixel images using a SLIC-based algorithm. It is a linear-time algorithm and can generate superpixel images that are lightweight in terms of memory space, thus consuming less storage space. They can provide a compact and convenient representation of standard images, which can be very useful for computationally demanding applications that process RS images in a low-bandwidth environment. Further optimization takes place at the CNN module on these superpixel images.

C. Convolutional Neural Network

Convolutional neural networks (also known as CNNs/ConvNets) are enhanced neural networks most commonly applied to analyze visual images. The structure of CNN is distinctive; one convolutional layer stacks upon another, followed by a few pooling layers, and finally, a few Fully-connected layers (for image classification) or upsampling layers (for image segmentation). The convolutional layer is the core of a CNN, which extracts the high-level features through the local perception and weight-sharing mechanism of the kernels/filters. The pooling layer can be considered the backbone of CNN used as a stuffing layer of a sandwich between the two slices of convolutional layers. It is used to enhance efficiency and avoid over-fitting in training procedures. It downsamples the input feature map using a non-linear max function that reduces the number of parameters to be used for calculations in the following Convolutional layers. The deep architecture used in the proposed multiscale CNN (MCNN) approach is an encoder-decoder-based convolutional framework (also known as AerialSegNet [50]) that is composed of four stages:
(a) Contraction Path: The input RGB images get decomposed to provide spatial and temporal features through convolution operations.
(b) Dense modules: Each stage of the architecture contains densely connected modules to pass the learned feature maps to the follow-up stages to enhance the feature set without increasing the number of parameters.
(c) Bottleneck layer: At this stage, the extracted features from the contraction path are then fed to the decoder blocks in the expansion path.
(d) Expansion path: Here, the shrunken image (in the encoder path) is reshaped to its original shape to produce the desired segmented map through some deconvolution operation (using transpose convolution or bilinear interpolation techniques). The overview of the architecture can be seen from Fig. 1's middle blocks, where the combined use of dense connections and skip connections can be observed.

D. Multiscale module

The correctness and accuracy of the image segmentation model need to integrate pixel-level accuracy concerning multiscale context reasoning. Deep CNNs combine multiscale context feature maps based on consecutive pooling, and convolution layers reduce resolution [28]. Moreover, the dense/deeper layers require context information in addition to full resolution [51]. The input images can be downscaled and upscaled with proper interpolation technique to get the multiscaled resolution images Fig. 2. As mentioned in Fig. 1, these multiscale images were given as inputs to the corresponding CNN modules to obtain the scale-invariant feature sets. Each CNN framework processes an image scene with different scales extracting the multiscale feature maps, which are further aggregated to form a multiscale context feature map that can predict pixel-level object class. The aggregation process is performed under the resize and concatenation process to make the process simple. The process of aggregation can be understood from the following equations:

$$M_{img} = D_s + U_s + I_{img} \tag{3}$$

where,

$$D_s = ds_{f1}(I_{img}) + ds_{f2}(I_{img}) + ... + ds_{fm}(I_{img}) \tag{4}$$

$$U_s = us_{1/f1}(I_{img}) + us_{1/f2}(I_{img}) + ... + us_{1/fm}(I_{img}) \tag{5}$$

Here, ds, us, represent downsampling and upsampling of an input image, respectively. Similarly, f, I_{img} and M_{img} denote the scale factor used for downsampling or upsampling, input image and the obtained multiscale feature map, respectively.

In our experiment, we have used 512×512 image tiles as input, which are then upscaled and downscaled by a factor of 2 to get 256×256, 1024×1024 resolution images. All these three different resolution images are trained individually through the encoder-decoder CNN architecture to fetch the multiscale feature maps that decide the pixel class.

IV. Experimentation

In order to access the performance of the proposed ensemble superpixel-MCNN architecture, extensive experimentation has been conducted on the NITRDrone scene understanding dataset and is described in the section IV-A. Moreover, the proposed approach is compared to some of the chosen state-of-the-art methodologies of semantic segmentation tasks, viz. [5]–[7], [40], [52].

A. Data Description

To perform the experimentation, we have considered the following two datasets:

1) NITRDrone Dataset: The NITRDrone dataset [1] [53] is proposed and built on seeing the rising demand for UAV-based applications for scene understanding that uses semantic segmentation-based techniques. The dataset contains around 101 number of variable resolution of VHR images captured with the help of DJI Phantom 4 and DJI Mavic drone having ground sampling distance *(GSD)* of 0.025 *sq. cm/pixel*. The resolution of an image in the dataset can be any of the following $1280 \times 720, 4000 \times 3000, 4096 \times 2160$. A pixel can

[1] https://github.com/drone-vision/NITRDrone-Dataset

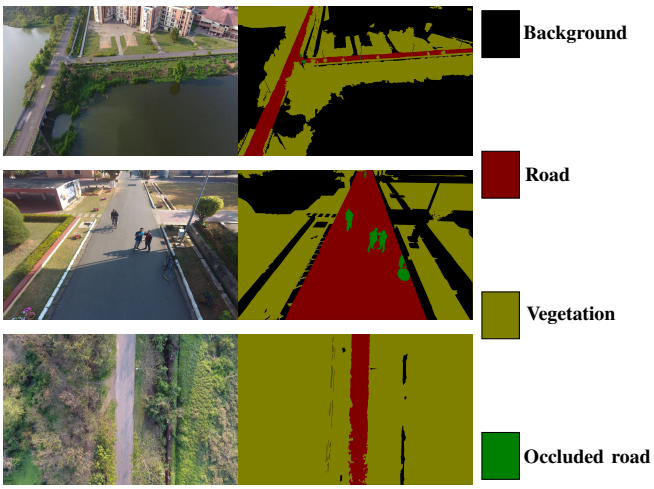

Figure 3. NITRDrone dataset containing UAV captured aerial outdoor images

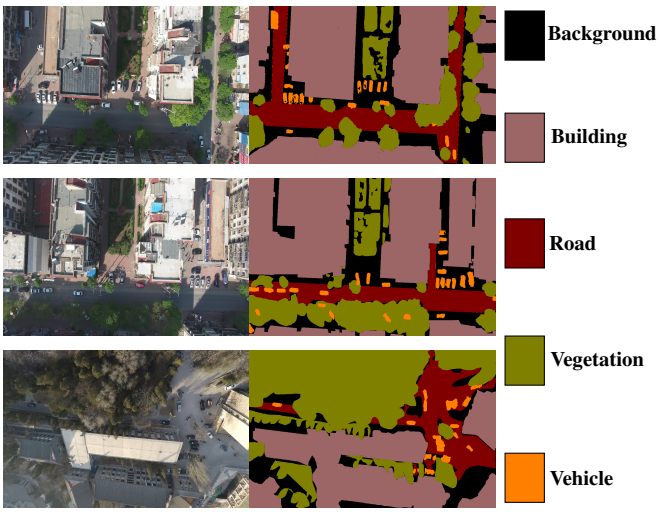

Figure 4. Urban Drone Dataset (UDD) containing UAV-captured aerial outdoor images (including five classes)

belong to any of the four different considered classes named "road", "vegetation", "occluded_road," and "_background_." Some of the sample images and their corresponding ground truths of the dataset are presented in Fig. 3.

2) Urban Drone Dataset (UDD): The UDD is a UAV-based image dataset that was proposed by Chen *et al.* towards semantic segmentation problems in computer vision [54]. The dataset is collected by a UAV DJI Phantom 4 operated at an altitude of 60m to 100m. The considered resolution for each image in the dataset is either 3000×4000 or 4096×2160. This dataset has been divided into three types, UDD-3, UDD-5, and UDD-6, that have three, four, and five classes, respectively, of which we have considered UDD-4, on which the proposed model is implemented and validated. As mentioned, UDD-4 has four-pixel classes named vegetation, buildings, roads, vehicles, and others (denoted for the rest of the object in a scene other than the mentioned classes). The dataset comprises of two sets training sets and a validation set consisting of 160 images and 45 frames, respectively. Sample images and the masks are shown in Fig. 4.

B. System Setup

In the proposed architecture, the first stage is meant for the SLIC superpixel algorithm to produce superpixel images. These superpixel images are considered as inputs for the second stage and are sampled at different scales to multiple deep CNN frameworks, which are trained to extract the required features for further classification of the pixels into one of the four classes in NITRDrone dataset and one of the five classes in UDD. The flow of operations to perform the experimentation is presented in Fig. 5. The implementation and validation of the proposed architecture are carried out on the datasets mentioned above and compared with the benchmark and peer-reviewed state-of-the-art methods. All the considered models are implemented with the help of the deep learning library PyTorch [2] [55] and are trained with NVIDIA TITAN V graphics card having $12GB$ of GPU memory.

C. Dataset Pre-processing

The proposed architecture is evaluated on the semantic drone datasets NITRDrone dataset [53] and UDD [54]. The resolution of the images of the considered datasets is of different sizes, such as $1280 \times 720, 4000 \times 3000, 4096 \times 2160$. Hence, we apply a sliding window technique with a constant stride that works over these images to extract the image tiles of 576×576 from both datasets. Through this operation, we are able to generate around $3,470$ number of images from the NITRDrone dataset and $3,500$ number of images from the UDD dataset. Out of the total number of images extracted from the NITRDrone dataset, we have considered $2,590,880$ images as training and testing sets, respectively. Similarly, for the UDD, $3,100$ images are considered for training the model, and the rest 400 images are equally divided among validation and testing set.

D. Pre-processing with SLIC

It is the first phase of segmentation in our proposed architecture. The image tiles produced by the sliding window are fed to this module. One of the popular superpixel algorithms, SLIC, is applied to produce semi-segmented superpixel images. There are two important parameters of SLIC algorithms: N and m representing the number of superpixels in a superpixel image and compactness control parameter, respectively. They play a vital role in preserving the natural properties of the ground objects. We have considered different combinations of N and m to find out the best combination with which we can apply the SLIC algorithm on the raw input images that preserve the integral properties of the objects to be segmented. The value of N and m are initialized to certain constant values as $N = [500, 1000]$ and $m = [5, 15, 25, 35]$. Thus, eight types of superpixel images are generated from this module, which will be the inputs for the next stage of CNN implementation.

[2]https://pytorch.org/docs/stable/index.html.

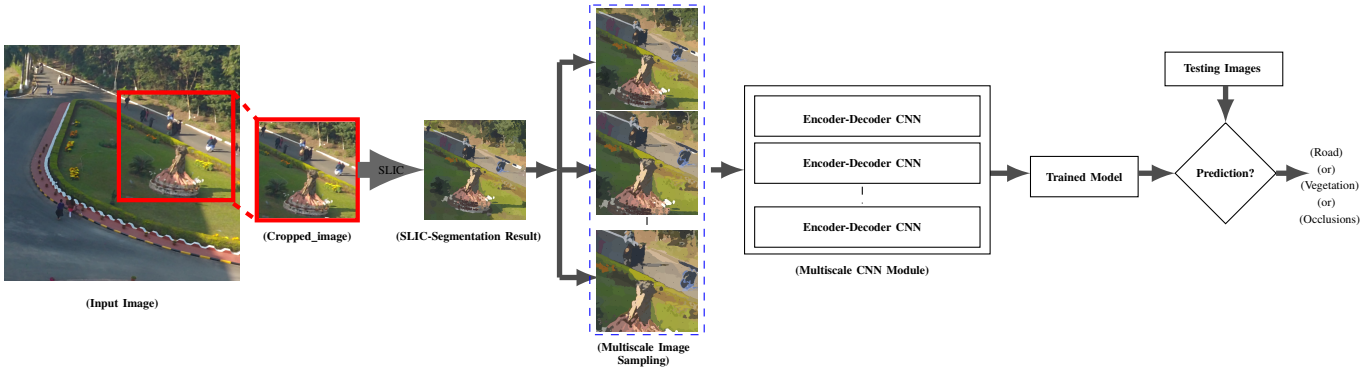

Figure 5. Flow diagram of the experimentation

E. Training

1) Input Pre-processing: The images from the superpixel module can be denoted as Image X. These Image X are collected at the CNN module, where they have to pass through a simple pre-processing step before considering for training. Image X are then down-sampled from 576×576 to 512×512 (can be denoted as Image Y) through random cropping or center cropping techniques. These cropped images (Image Y) are used by the deep CNN framework to train individual models. The resolution of the input and target images for the proposed architectures and state-of-the-art methods remains the same. The only difference lies in the type of images considered in both cases: the proposed approach uses superpixel aerial images, whereas the state-of-the-art models use stock aerial imageries.

2) Target Pre-processing: The corresponding target or the masks need to be down-scaled to 512×512 as per the input images described in the previous section. However, the training is performed with the one-hot coded masks. The one-hot coded target images are then color-coded with different colors for each object class for better visualization. These RGB color-coded masks are presented in Fig. 3 and 4.

3) Implementation Details: Adaptive moment estimation (Adam) [56] is considered the optimizer, and Cross-Entropy (CE) is used as a loss function. The learning rate is initialized to $5e-3$, whereas momentum and batch size are initialized to 0.9, and 3, respectively. A weight decay of 0.002 is introduced to handle the problem of over-fitting in the long run during training. ReLU activation function [57] is employed to improve the convergence and accuracy of the network. Moreover, two types of augmentation techniques are applied to the superpixel images making the number of images stand double, thus extending the training process. The learning rate is set to be decreased after every 30 epochs by a factor of 0.002 to maintain the regularization. The training operation continues until the learning rate reaches 10^{-20}. After 450 epochs, the proposed architecture seems to converge, which can be observed through minor changes in loss and accuracy.

F. Loss Function

The choice of the loss function is vital in carrying out neural network-based optimization. The loss-weighting scheme of the network architecture targets the interior pixels and the border of the segmented objects. The Cross-Entropy (CE) loss, also known as logarithmic or logistic loss, is chosen to train the baseline models. The predicted class probability is compared with the truly desired class output 0 or 1. The corresponding loss/score of the corresponding pixel class is obtained to check for the deviation from the actual (true) value. And as a penalization, the weights will travel backward to re-correct the same for a better understanding of the object feature map. SoftMax differential function (S_i) is also used with CE, which aims at minimizing the loss during training, i.e., smaller the loss value better the model. Cross-entropy can be defined as follows:

$$L_{CE} = -\sum_{i=1}^{n} T_i \log(S_i) \tag{6}$$

where, T_i and S_i are the truth value $\in [0, 1]$ and the $SoftMax$ Probability for i^{th} class, respectively.

G. Tasks and metrics

The primary objective of the proposed SLIC-M-CNN framework is scene parsing and segments the aerial images as per the given number of objects (four for the NITRDrone dataset and five for the UDD). In order to analyze the architecture's performance, both quantitative and qualitative results play vital roles. Widely acceptable performance metrics, such as precision, recall, F-Score, the intersection of union, and overall accuracy, are used to examine the performance of the proposed framework. These metrics can be formulated as per the Eqs. 7- 11.

$$P = \frac{TP}{(TP + FP)} \tag{7}$$

$$R = \frac{TP}{(TP + FN)} \tag{8}$$

$$\begin{aligned} IoU &= \frac{Overlapping\ Area}{Area\ of\ Union} \\ &= \frac{TP}{(TP + FP + FN)} \end{aligned} \tag{9}$$

Table I
COMPARISON OF PERFORMANCE EVALUATION OF VARIOUS STATE-OF-THE-ART MECHANISMS ON SUPERPIXEL IMAGES OF NITRDRONE DATASET

	mPrecision	mRecall	mIoU	mFscore	mAcc
500_5	0.86	0.867	0.811	0.849	0.962
500_15	0.81	0.873	0.779	0.807	0.973
500_25	0.85	0.858	0.801	0.841	0.961
500_35	0.847	0858	0.797	0.839	0.958
1000_5	**0.888**	**0.944**	**0.861**	**0.896**	0.981
1000_15	0.87	0.910	0.829	0.876	0.978
1000_25	0.861	0.907	0.826	0.865	0.977
1000_35	0.857	0.891	0.819	0.857	**0.983**

Table II
COMPARISON OF PERFORMANCE EVALUATION OF VARIOUS STATE-OF-THE-ART MECHANISMS ON SUPERPIXEL IMAGES OF UDD

	mPrecision	mRecall	mIoU	mFscore	mAcc
500_5	0.807	0.792	0.747	0.777	0.989
500_15	0.832	0.785	0.734	0.767	0.988
500_25	0.810	0.842	0.757	0.784	0.990
500_35	0.826	0.837	0.767	0.791	0.993
1000_5	0.811	**0.872**	**0.772**	**0.803**	**0.994**
1000_15	0.7961	0.8022	0.7347	0.7676	0.988
1000_25	**0.875**	0.849	0.767	0.789	0.993
1000_35	0.840	0.472	0.379	0.409	0.734

$$\begin{aligned} F &= \frac{Overlapping\ Area}{Pixel\ count\ in\ both\ GT\ and\ PR} \\ &= \frac{2PR}{(R+P)} \\ &= \frac{2TP}{(2TP+FP+FN)} \end{aligned} \tag{10}$$

$$A = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{11}$$

Here, P, R, IoU, F, and A represent precision, recall, dice score, intersection over union, and overall accuracy, respectively. Similarly, TP and TN stand for True Positive and True Negative, respectively, which can be explained by the number of predicted pixels belonging to the same class as the ground truth. Additionally, FP and FN denote False Positive and False Negative, respectively.

V. EXPERIMENTAL RESULTS AND OBSERVATION

This section presents the obtained results from the proposed model through our experiment. It also discusses an extensive comparison of these results with state-of-the-art methodologies. It highlights the improvements achieved through the proposed architecture in semantically segmenting the object classes from the UAV images.

A. Observation

As discussed earlier, the SLIC-based superpixel algorithm works based on two core parameters: N and m to decide the number of superpixels in a superpixel image and a scale variation parameter, thus playing an important role in estimating the size of a superpixel, respectively. The value of m falls within a range of [5, 35]. The greater the value of m, the more compact the cluster.

In the experiment, we have considered m as $[5, 15, 25, 35]$. That means when $m = 5$, each image patch (superpixel) in the superpixel image is of size 5×5. Small scale patches ($m = 5$) expose the features inside a superpixel efficiently. In contrast, the enormous value ($m = 25/35$)works at the border regions of an object to distinguish it from the others. It helps the training block be exposed to meaningful, distinguished features to learn about the object it needs to segment. However, to prove the efficiency, every possible combination of m and N is considered. The produced superpixel images are then given as input to the CNN block, and the outcomes are listed in the Tables I and II.

The resulting superpixel images from the SLIC module are accepted as inputs at the CNN module for the training operation, and the trained model is used on the validation/test set to obtain the results. At the CNN module, the superpixel images are gone through M number of CNN architectures ($M =$ No. of multiscale images) constitute the multiscale CNN/ConvNet architecture. The main idea of using a multiscale ConvNet (M_CNN) architecture (Fig 1) is to have a vast feature space of the ground objects at different scales sampled through a random multi-sampling process. The ConvNet module in the M-CNN architecture is an encoder-decoder architecture inspired by the skip connection mechanisms (dense module within a stage) that passes the previously learned parameters in the encoder stage to the following equivalent decoder stage. This architecture can determine the edge-level object features and the imbalanced occlusion class objects. These features are crucial from the perspective of a segmentation task, as even a few pixels' misclassifications may affect the accuracy of the architecture.

B. State-of-the-art Comparison

The proposed model is also compared with the state-of-the-art methodologies based on the evaluation matrices described in the previous section. The baseline models are validated with the raw input images considered for multiscale CNN and Aerial SegNet architectures. The training process for these models is performed for around 450 epochs till the convergence occurs. The obtained results from the experiments are listed in Table III and also in Fig. 6.

As shown in Fig. 6, it can be clear that among the state-of-the-art models, U-Net [40], FCN-32s [5], and DeepLab-plus-exception [7] manage to performs well to segment the vegetation and road class pixels thus achieving a reasonable $75-80\%$ IoU score. However, they have failed to capture the outlines of different class objects resulting in a slight drop in accuracy. This is where the multiscale feature fusion technique looks useful in aggregating the scale-invariant features that help fetch the missing feature sets, thus improving the accuracy. Hence, it can be concluded from Tables III and IV that the proposed model can perform better than the existing methods, such as [5], [40] in terms of segmenting the pixel class and achieving a smoother boundary of the objects. Moreover, along with the performance measures like F_Score and IoU, we have also considered precision and recall. It can be observed from Table III that from the mean precision (mPrecision)

Table III
OVERALL PERFORMANCE EVALUATION OF VARIOUS STATE-OF-THE-ART MECHANISMS ON NITRDRONE DATASET

	mPrecision	mRecall	mIoU	mFscore	mAccuracy
FCN-8s [5]	0.58	0.35	0.12	0.14	0.55
FCN-16s [5]	0.80	0.82	0.71	0.76	0.93
FCN-32s [5]	0.81	0.83	0.76	0.81	0.95
FC_DenseNet-103 [52]	0.81	0.79	0.68	0.74	0.90
SegNet [6]	0.84	0.74	0.68	0.76	0.92
DEEPLAB_V3_PLUS_XCEPTION [7]	**0.91**	0.79	0.74	0.83	0.96
U-NET "ResNet-18" [40]	0.84	0.90	0.80	0.85	0.98
AerialSegNet	0.88	0.91	0.84	0.88	0.98
MCNN_AerialSegNet	0.878	0.932	0.862	0.897	**0.981**
Superpixel_MCNN_AerialSegNet (1000_5)	0.888	**0.944**	**0.865**	**0.898**	**0.981**

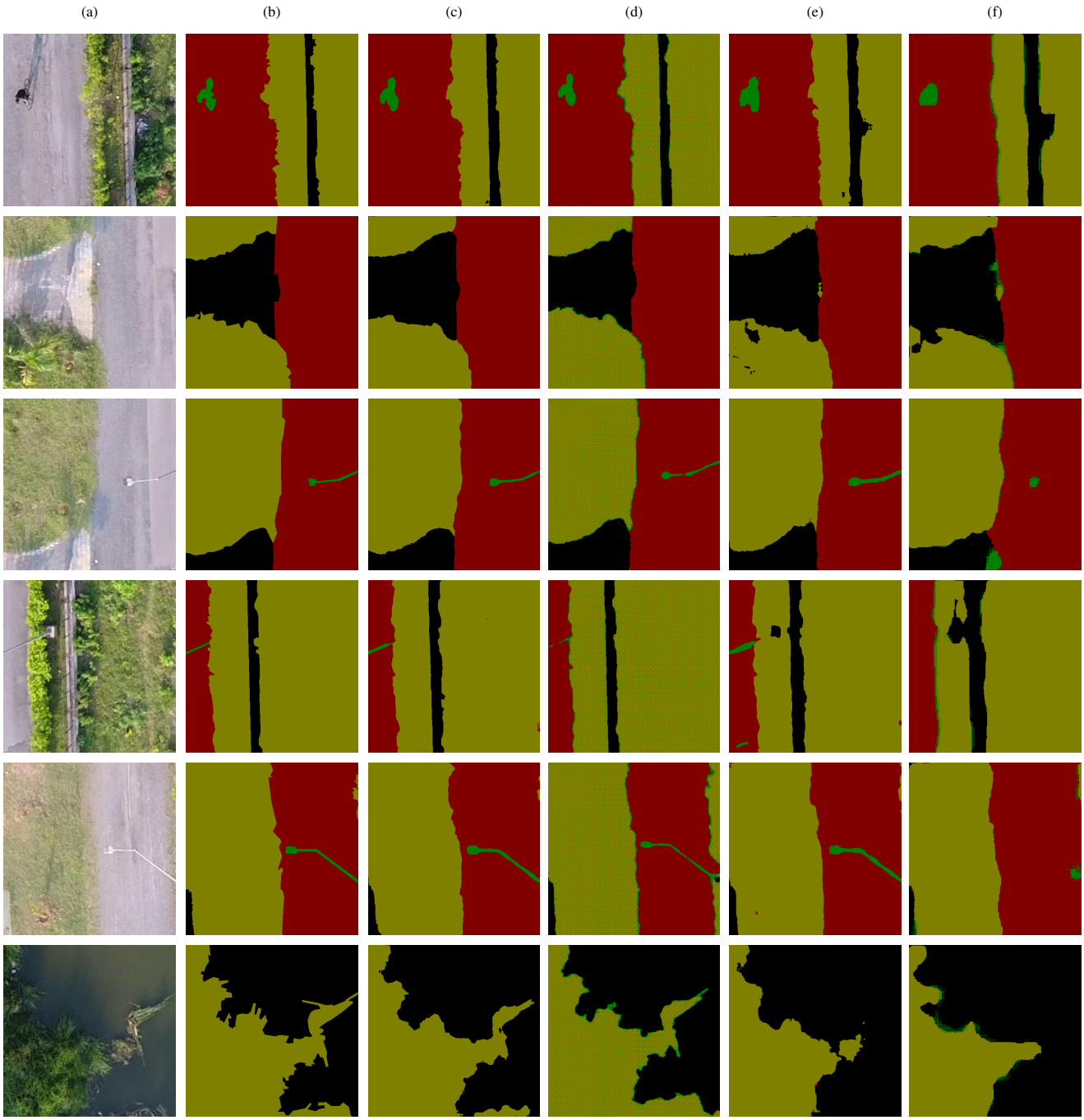

Figure 6. Prediction of the state-of-the-art mechanisms. Left to right: (a) UAV-based aerial images, (b) labeled mask of the corresponding images, (c) AerialSegNet, (d) Superpixel_MCNN_AerialSegNet (proposed), (e) UNet [40], and (f) FCN-16s [5]. Color coding of the semantic classes matches Fig. 3

Table IV
OVERALL PERFORMANCE EVALUATION OF VARIOUS STATE-OF-THE-ART MECHANISMS ON UDD

	mPrecision	mRecall	mIoU	mFscore	mAccuracy
FCN-8s [5]	0.58	0.33	0.116	0.14	0.49
FCN-16s [5]	0.72	0.71	0.60	0.68	0.94
FCN-32s [5]	0.66	0.53	0.54	0.60	0.92
FC_DenseNet-103 [52]	0.55	0.62	0.52	0.59	0.92
SegNet [6]	0.59	0.63	0.54	0.61	0.94
DEEPLAB_V3_PLUS_XCEPTION [7]	0.75	0.73	0.65	0.72	0.95
U-NET "ResNet-18" [40]	0.785	0.82	0.72	0.80	0.96
AerialSegNet	0.80	0.84	0.739	0.82	0.96
MCNN_AerialSegNet	**0.82**	0.85	0.767	0.819	0.96
Superpixel_MCNN_AerialSegNet (1000_5)	0.811	**0.872**	**0.772**	**0.823**	**0.994**

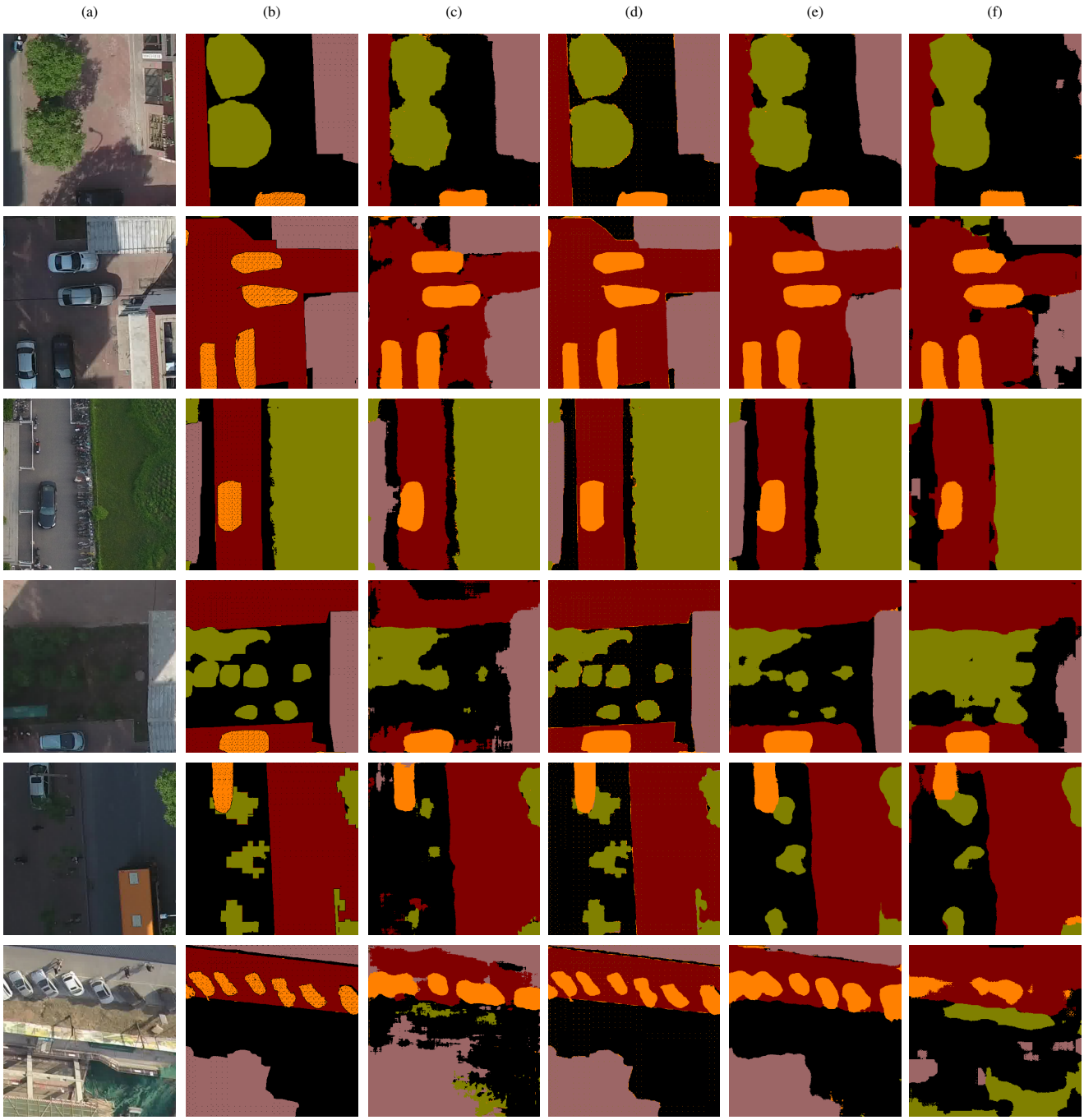

Figure 7. Prediction of the state-of-the-art mechanisms on UDD. Left to right: (a) UAV-based aerial images, (b) labeled mask of the corresponding images, (c) AerialSegNet, (d) Superpixel_MCNN_AerialSegNet (proposed), (e) UNet [40], and (f) FCN-16s [5]. Color coding of the semantic classes matches Fig. 4

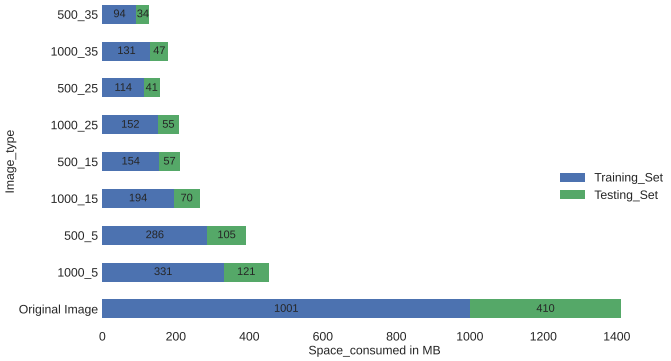

Figure 8. Variation of storage space consumed by the generated superpixel images of NITRDrone dataset

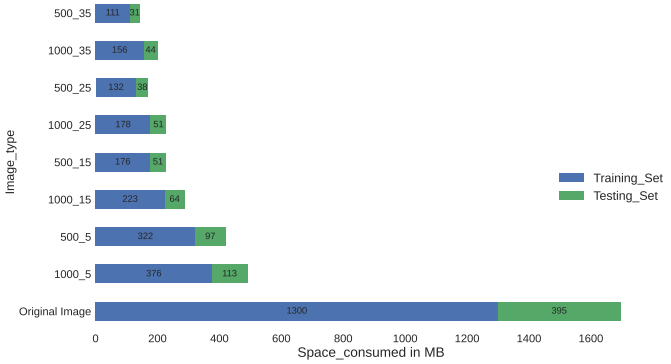

Figure 9. Variation of storage space consumed by the generated superpixel images of UDD

point of view, the DeepLab_V3+Xception [7] performs better (on the NITRDrone dataset) in terms of precision score than the proposed architecture. However, there is a miss, and it can be explained by seeing the table that there is a massive gap between the recall and precision score, which is entirely unacceptable from the perspective of a semantic segmentation task. At this point, the proposed framework acts superior maintaining a descent of true positives and true negatives as can be judged based on the obtained scores mentioned in Table III. Similarly, a comparison of improvement achieved through the proposed architecture is also presented in Table V.

VI. Discussion

To provide a better comprehensive comparison of the proposed approach, various experimental observations corresponding to external factors, such as space complexity, are also noted, which are discussed in this section.

A. Parameter Comparison

The number of learnable parameters plays a crucial role in accessing the performance of the model in terms of speed and memory. Hence, a comparative study is presented in Table VI. It can be observed that state-of-the-art models, except a few, such as FC_DensenNet-103 [52], AerialSegNet [50], and UNet [40] having ResNet-18 as the backbone. However, the proposed architecture overcomes the underlined issues of these

Table V
Comparative Results of top three best performers with the NITRDrone dataset and UDD

NITRDrone Dataset		
Architecture	mIoU	mF_Score
U-Net "ResNet-18"	0.80	0.85
AerialSegNet	0.84	0.88
Superpixel_MCNN_AerialSegNet	**0.865**	**0.898**
improvement	*0.025*	*0.018*
UDD		
U-Net "ResNet-18"	0.72	0.80
AerialSegNet	0.739	0.82
Superpixel_MCNN_AerialSegNet	**0.772**	**0.823**
improvement	*0.033*	*0.03*

$mIoU$ = mean IoU, $mFScore$ = mean F_Score

Table VI
Comparison of Number of Trainable Parameters

State-of-the-art methods	**Parameters**
	Lower the better
U-Net_ResNet-18 [40]	$15.50M$
FCN-8s [5]	$136M$
FCN-16s [5]	$134M$
FCN-32s [5]	$134M$
FC_DenseNet-103 [52]	$9.42M$
SegNet [6]	$29.44M$
DeepLab_V3_plus_Xception [7]	$41.3M$
AerialSegNet [50]	$11.76M$
Superpixel_MCNN_AerialSegNet (proposed)	$32.43M$

M=Million, LW=Light Weight

baselines, achieving better performance accuracy while having less trainable parameters than most of the considered baselines. Therefore, it can be deployed on various edge-end devices (like UAVs), where memory and computing power are constraints. In the next subsection, we discuss the optimization that has been achieved through the use of superpixel.

B. Space Efficiency

The superpixel technique provides a partially segmented image that helps the CNN module extract the object-level features while reducing the space complexity. As per our study, images with smaller m performs slightly better than others due to their feature preservation properties that are pretty close to a natural image (with meaningful information. From the space consumption point of view, it is pretty clear that the superpixel images with the highest space complexity are also 60% lesser than the original images while performing better or equal than with the original images. A bar graph representing the space consumption of all the considered images is presented in Fig. 8 and Fig. 9. Similarly, among the superpixel images considered for the experiment, space consumption (spc) can be arranged in a decreasing order like $spc(5) > spc(15) > spc(25) > spc(35)$. Considering an example, if $N = 1000$, then $spc(1000\_5) > spc(1000\_15) > spc(1000\_25) > spc(1000\_35)$. This space complexity matters a lot for the proposed approach to get implemented over IoT and network, as transferring the superpixel images (over the network) would require a low-bandwidth connection making bandwidth available for the other network operations. Thus, IoT-based remote sensing applications can be benefited from the proposed architecture.

C. Observed Limitations

The current study has a few limitations, which are presented below. Under low-light conditions, the model performs poorly (at the beginning of the training) in segmenting similar-looking objects. For example, the road surface may look similar to the rooftop (the tar-covered sheet), creating confusion for the model and leading to low accuracy in terms of IoU. Similarly, for the minor class objects, such as the occlusion class in the NITRDrone dataset and vehicle class in the UDD, the model invests a lot of time in obtaining the required feature maps before correctly classifying the minor object class pixels. Moreover, one more limitation can also be seen corresponding to the increased number of trainable parameters due to the multiple CNN modules. This may bring CNN architectural issues. In future work, a few cues, as presented in [58], can be considered to develop a multiscale CNN architecture, and its effectiveness can be verified.

VII. Conclusion

This article presents a superpixel-based multiscale CNN framework to address UAV aerial image-based semantic segmentation problems. The first-level segmentation is achieved using the SLIC superpixel algorithm that produces superpixel images from the input UAV images, which are the input for the CNN architecture for final segmentation. The proposed CNN architecture collectively uses the strength of skip connections and the multiscale context aggregation strategy to extract the crucial scale-invariant features that can uniquely classify a pixel as per the object classes. The multiscale CNN module is good at extracting scale-invariant features that are essential from a UAV imagery point of view, as the same ground objects may look small or large as per the operating height of UAV. Further, the proposed architecture is evaluated (on the NITRDrone and Urban drone dataset) and compared with the state-of-the-art methods. The experimentally obtained results prove the superiority of the ensemble framework (of the superpixel technique and the deep multiscale architecture) in segmenting the UAV aerial images. Moreover, the proposed architecture provides a robust solution towards semantic segmentation for object classes like road, vehicle, and vegetation, which the other considered state-of-the-art methodologies failed to do. The proposed approach can be integrated with the robotics-based AI solution to provide intelligent road extraction and vegetation detection through panoptic aerial imageries of UAVs. Similarly, the proposed approach can be combined with IoT and cloud concepts to actively analyze critical operations, such as disaster management and carrying out surveys.

As an extension to this work, different superpixel techniques, such as SLICO and SEEDS, may be tested to have a better-performing superpixel technique for road and vegetation extraction from aerial images. Similarly, the proposed architecture can also be implemented in a simulated IoT environment to demonstrate the efficiency of this approach in managing low-bandwidth operations.

Acknowledgments

This research is supported by the following projects:

1) Project titled "Deep learning applications for computer vision task" funded by NITROAA with support of Lenovo P920 and Dell Inception 7820 workstation and NVIDIA Corporation with support of NVIDIA Titan V and Quadro RTX 8000 GPU.
2) Project titled "Applications of Drone Vision using Deep Learning" funded by Technical Education Quality Improvement Programme (referred to as TEQIP-III), National Project Implementation Unit, Government of India.

References

[1] Statista, "Number of active satellites," https://www.statista.com/statistics/897719/number-of-active-satellites-by-year/#:~:text=This%20statistic%20illustrates%20the%20number,2%2C298%20active%20satellites%20in%202019.

[2] WorldView Series, "Earth Observation Satellites," https://earthdata.nasa.gov/worldview.

[3] Landsat Series, "Earth Observation Satellites," https://www.nasa.gov/mission_pages/landsat/overview/index.html.

[4] A.-V. Emilien, C. Thomas, and H. Thomas, "UAV & Satellite Synergies for Optical Remote Sensing Applications: A Literature Review," *Science of Remote Sensing*, p. 100019, 2021, doi: 10.1016/j.srs.2021.100019.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017, doi: 10.1109/TPAMI.2016.2644615.

[7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818, doi: 10.1007/978-3-030-01234-2_49.

[8] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[9] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A Review of Semantic Segmentation using Deep Neural Networks," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 87–93, 2018, doi: 10.1007/s13735-017-0141-z.

[10] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task Low-rank Affinity Pursuit for Image Segmentation," in *2011 International Conference on Computer Vision*, 2011, pp. 2439–2446, doi: 10.1109/ICCV.2011.6126528.

[11] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using Superpixels: A Bipartite Graph Partitioning Approach," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 789–796, doi: 10.1109/CVPR.2012.6247750.

[12] H. Tong, F. Tong, W. Zhou, and Y. Zhang, "Purifying SLIC Superpixels to Optimize Superpixel-based Classification of High Spatial Resolution Remote Sensing Image," *Remote Sensing*, vol. 11, no. 22, p. 2627, 2019, doi: 10.3390/rs11222627.

[13] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002, doi: 10.1109/34.1000236.

[14] R. Mohan and R. Nevatia, "Using Perceptual Organization to Extract 3D Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 11, pp. 1121–1139, 1989, doi: 10.1109/34.42852.

[15] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique," in *Readings in Computer Vision*. Elsevier, 1987, pp. 741–752, doi: 10.1016/B978-0-08-051581-6.50071-4.

[16] U. Stilla, "Map-aided Structural Analysis of Aerial Images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 50, no. 4, pp. 3–10, 1995, doi: 10.1016/0924-2716(95)98232-O.

[17] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001, doi: 10.1023/A:1011126920638.

[18] C. Schmid, "Constructing Models for Content-based Image Retrieval," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2001, pp. II–II, doi: 10.1109/CVPR.2001.990922.

[19] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–I, doi: 10.1109/CVPR.2001.990517.

[20] B. Fröhlich, E. Bach, I. Walde, S. Hese, C. Schmullius, and J. Denzler, "Land Cover Classification of Satellite Images using Contextual Information," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3, no. W1, 2013, doi: 10.5194/isprsannals-II-3-W1-1-2013.

[21] P. Tokarczyk, J. D. Wegner, S. Walk, and K. Schindler, "Features, Color Spaces, and Boosting: New Insights on Semantic Classification of Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 280–295, 2015, doi: 10.1109/TGRS.2014.2321423.

[22] D. Chai, W. Förstner, and F. Lafarge, "Recovering Line-Networks in Images by Junction-Point Processes," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1894–1901, doi: 10.1109/CVPR.2013.247.

[23] M. Ortner, X. Descombes, and J. Zerubia, "Building Outline Extraction from Digital Elevation Models Using Marked Point Processes," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 107–132, 2007, doi: 10.1007/s11263-005-5033-7.

[24] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler, "A Higher-Order CRF Model for Road Network Extraction," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1698–1705, doi: 10.1109/CVPR.2013.222.

[25] Y. Wang, W. Ding, B. Zhang, H. Li, and S. Liu, "Superpixel Labeling Priors and MRF for Aerial Video Segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2590–2603, 2020, doi: 10.1109/TCSVT.2019.2919482.

[26] K. Fukushima and S. Miyake, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition ," in *Competition and Cooperation in Neural Nets*. Springer, 1982, pp. 267–285, doi: 10.1007/978-3-642-46466-9_18.

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[30] N. Attari, F. Ofli, M. Awad, J. Lucas, and S. Chawla, "Nazr-CNN: Fine-Grained Classification of UAV Imagery for Damage Assessment," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 50–59, doi: 10.1109/DSAA.2017.72.

[31] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The Unmanned Aerial Vehicle Benchmark: Object detection and Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 370–386, doi: 10.1007/978-3-030-01249-6_23.

[32] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013, doi: 10.1109/TPAMI.2012.231.

[33] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017, doi: 10.1109/MGRS.2017.2762307.

[34] A. Van Etten, "You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery," *arXiv preprint arXiv:1805.09512*, 2018.

[35] M. Dai, J. Hu, J. Zhuang, and E. Zheng, "A Transformer-Based Feature Segmentation and Region Alignment Method For UAV-View Geo-Localization," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021, doi: 10.1109/TCSVT.2021.3135013.

[36] Z. Zhang, Q. Liu, and Y. Wang, "Road Extraction by Deep Residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018, doi: 10.1109/LGRS.2018.2802944.

[37] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt, "RoadTracer: Automatic Extraction of Road Networks from Aerial Images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4720–4728, doi: 10.1109/CVPR.2018.00496.

[38] T. K. Behera, P. K. Sa, M. Nappi, and S. Bakshi, "Satellite IoT Based Road Extraction from VHR Images Through Superpixel-CNN Architecture," *Big Data Research*, vol. 30, p. 100334, 2022, doi: 10.1016/j.bdr.2022.100334.

[39] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 173, pp. 24–49, 2021, doi: 10.1016/j.isprsjprs.2020.12.010.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[41] J. Xie, N. He, L. Fang, and P. Ghamisi, "Multiscale Densely-Connected Fusion Networks for Hyperspectral Images Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 246–259, 2021, doi: 10.1109/TCSVT.2020.2975566.

[42] J. Fan, T. Chen, and S. Lu, "Superpixel Guided Deep-Sparse-Representation Learning for Hyperspectral Image Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3163–3173, 2018, doi: 10.1109/TCSVT.2017.2746684.

[43] M. B. A. Gibril, H. Z. M. Shafri, A. Shanableh, R. Al-Ruzouq, A. Wayayok, and S. J. Hashim, "Deep Convolutional Neural Network for Large-Scale Date Palm Tree Mapping from UAV-Based Images," *Remote Sensing*, vol. 13, no. 14, p. 2787, 2021, doi: 10.3390/rs13142787.

[44] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "On Detecting Road Regions in a Single UAV Image," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1713–1722, 2017, doi: 10.1109/TITS.2016.2622280.

[45] L. Sommer, T. Schuchert, and J. Beyerer, "Comprehensive Analysis of Deep Learning-Based Vehicle Detection in Aerial Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2733–2747, 2019, doi: 10.1109/TCSVT.2018.2874396.

[46] T. K. Behera, S. Bakshi, and P. K. Sa, "Aerial Data Aiding Smart Societal Reformation: Current Applications and Path Ahead," *IEEE IT Professional*, vol. 23, no. 3, pp. 82–88, 2021, doi: 10.1109/MITP.2020.3020433.

[47] T. K. Behera, M. A. Khan, and S. Bakshi, "Brain MR Image Classification Using Superpixel-Based Deep Transfer Learning," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–11, 2022, doi: 10.1109/JBHI.2022.3216270.

[48] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk *et al.*, "SLIC Superpixels. Ecole Polytechnique Fédéral de Lausssanne (EPFL)," *Tech. Rep, 149300*, 2010.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012, doi: 10.1109/TPAMI.2012.120.

[50] T. K. Behera, S. Bakshi, and P. K. Sa, "Vegetation Extraction from UAV-based Aerial Images through Deep Learning," *Computers and Electronics in Agriculture*, vol. 198, p. 107094, 2022, doi: 10.1016/j.compag.2022.107094.

[51] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[52] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 11–19, doi: 10.1109/CVPRW.2017.156.

[53] T. K. Behera, S. Bakshi, P. K. Sa, M. Nappi, A. Castiglione, P. Vijayakumar, and B. B. Gupta, "The NITRDrone Dataset to Address the Challenges for Road Extraction from Aerial Images," *Journal of Signal Processing Systems*, pp. 1–13, 2022, doi: 10.1007/s11265-022-01777-0.

[54] Y. Chen, Y. Wang, P. Lu, Y. Chen, and G. Wang, "Large-Scale Structure from Motion with Semantic Constraints of Aerial Images," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2018, pp. 347–359, doi: 10.1007/978-3-030-03398-9_30.

[55] PyTorch Docs, "PyTorch Documents," https://pytorch.org/docs/stable/index.html.

[56] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, Y. Bengio and Y. LeCun, Eds., 2015, http://arxiv.org/abs/1412.6980.
[57] ReLu, "Relu Activation Function," https://www.tinymind.com/learn/terms/relu.
[58] T. K. Behera, S. Bakshi, and P. K. Sa, "A Lightweight Deep Learning Architecture for Vegetation Segmentation using UAV-captured Aerial Images," *Sustainable Computing: Informatics and Systems*, 2022, doi: 10.1016/j.suscom.2022.100841.

Tanmay Kumar Behera is currently working on his doctoral research in Computer Science and Engineering at National Institute of Technology Rourkela, India. His research interest includes machine learning, computer vision, drone vision, medical image processing, and visual surveillance. He received the M.Tech. degree in computer science from Veer Surendra Sai University of Technology, Burla, India. He is a Member of IEEE.

Sambit Bakshi is currently with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India. His area of interest includes surveillance, biometric security, and digital forensics. He presently serves as associate editor of IEEE Systems Journal, IEEE IT Professional magazine, Innovations in Systems and Software Engineering Springer: A NASA Journal, Expert Systems with Applications, Engineering Applications of Artificial Intelligence, Image and Vision Computing, Multimedia Systems, and Multimedia Tools and Applications. He served as Associate Editor of IEEE Access, Plos One, and Expert Systems in the past. He is a senior member of IEEE. He is Founding Chair of IEEE Rourkela Subsection. He presently serves as a member of IEEE Computational Intelligence Society Young Professionals Subcommittee since 2020 (liaison for IEEE Young Professional for the year 2022). He previously served as the vice-chair for IEEE Computational Intelligence Society Technical Committee for Intelligent Systems Applications for the year 2019. He is a member of Professional Activities Committee of IEEE Region 10 for the year 2022. He has published widely in more than 100 journals and conferences.

Michele Nappi received the Laurea degree (cum Laude)in computer science from the University of Salerno, Italy, in 1991, the M.Sc. degree in information and communication technology from I.I.A.S.S."E.R. Caian-iello", Vietri Sul Mare, Salerno, and the Ph.D. degree in applied mathematics and computer science from the University of Padova, Italy. He is currently a full professor of computer science with the University of Salerno. He is also a Team Leader of the Biometric and image Processing Lab (BIPLAB). His research interests include multibiometric systems, pattern recognition, image processing, compression and indexing, multimedia databases, human–computer interaction, and VR/AR. He has co-authored more than 190 papers in international conference, peer-review journals, and book chapters in these fields.He was a member of IAPR. He received several international awards for scientific and research activities. He was the President of the Italian Chapter of the IEEEBiometrics Council, from 2015 to 2017.

Pankaj Kumar Sa received the Ph.D. degree in computer science from NIT Rourkela in 2010. He is currently an Associate Professor with the CSE Department, NIT Rourkela. He has authored or co-authored a number of research articles in various journals, conferences, and book chapters. His research interests include computer vision, biometrics, and visual surveillance. He is a member of the CSI. He has co-investigated some R&D projects funded by SERB, PXE, DeitY, and ISRO. He has been conferred with various prestigious awards and honors.