Computers and Electronics in Agriculture 198 (2022) 107094

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Vegetation Extraction from UAV-based Aerial Images through Deep Learning

Tanmay Kumar Behera, Sambit Bakshi *, Pankaj Kumar Sa

Department of Computer Science & Engineering, National Institute of Technology Rourkela, Odisha 769008, India

ARTICLE INFO

Keywords:
Remote sensing
Semantic segmentation
Supervised learning
Deep learning
Convolutional Neural Network (CNN)
Urban mapping
Vegetation extraction
Unmanned Aerial Vehicle (UAV)

ABSTRACT

The panoptic aerial images of the earth's surface captured by satellites and unmanned aerial vehicles (UAVs) have great potential to support applications in various domains, especially robotics-based solutions for smart urban planning, terrain classification, vegetation detection, agriculture planning, and environmental surveillance. This article proposes a deep learning-based model influenced by the dense modules, which help preserve the network's feed-forward nature, thereby eliminating the vanishing gradient problem usually seen in deep state-of-the-art mechanisms. The proposed end-to-end convolutional neural network (CNN) architecture consists of contracting and symmetric expanding paths that precisely extract the global features to segment the vegetation class from the aerial image. The proposed architecture is evaluated on the two UAV image datasets: *Urban Drone Dataset (UDD)* and *NITRDrone Dataset*. It succeeds to achieve an intersection over union (IoU) of 74% and 84% on the UDD and NITRDrone datasets, respectively, thus demonstrating better performance accuracy than the state-of-the-art methods. The experimentally obtained results show that implementing the dense connections helps to significantly reduce the number of trainable parameters and improves the model's efficacy in addressing such multi-class segmentation problems, particularly in vegetation detection and road-line extraction.

1. Introduction

Recent years have witnessed drastic technological advancement in several areas, including space technology and spaceborne technology, and are still growing. These platforms have captured huge volumes of images that act as fuel for several applications. These overwhelming generated data make manual understanding difficult, thus requiring machine vision to be employed for automatic interpretation. Thanks to the advancement of machine learning, machine vision and computer vision that have expanded their capabilities to different application platforms. Image classifications and semantic segmentation are the core problems in computer vision tasks. They can be applied to various application domains, including urban management, traffic monitoring, land cover classification, and so on (Cheng et al., 2020). Not only image classification and semantic segmentation but also object detection in aerial images is also another problem that is quite important for fishery management, intelligent traffic management, etc. (Varma et al., 2019; Franke et al., 2013; Pan et al., 2018; Di et al., 2017; Xie et al., 2016).

The difference between the tasks of image classification and semantic segmentation is what they predict: an image or part of an image. In semantic segmentation, instead of predicting the labels per image (in image classification), the task is to determine the object of interest from the entire image. Hence, the basic/foremost approach that can come to one's mind is dividing the whole image into small blocks and then applying the convolutional neural networks (CNNs) to determine the blocks containing the object of interest. However, this process requires each block to run through CNN; thus, a single image has to go many times through the network to get the output, which is also coarse type. Therefore, a second approach is usually considered that allows more defined objects of interest and needs the model to run through each image.

The deep learning mechanisms, especially CNNs, have become an essential tool for addressing problems like image classification, object detection, and dynamic object understanding. Still, they face challenges considering complex, diversified scenes, light illuminations, shadows, etc. The prediction error may arise as two different objects may have similar appearances because of external factors such as light illumination or environmental change. These challenges need to be addressed in state-of-the-art mechanisms. Researchers have proposed many CNN models that are based on the mechanism of fully convolution neural

* Corresponding author.
E-mail addresses: bakshisambit@ieee.org, bakshisambit@nitrkl.ac.in (S. Bakshi).

https://doi.org/10.1016/j.compag.2022.107094
Received 2 December 2021; Received in revised form 20 May 2022; Accepted 24 May 2022
Available online 10 June 2022

networks (FCNs) (Long et al., 2015). These state-of-the-art mechanisms come with several significant advantages. They can extract the features very well at the local levels but suffer from attending global features, which can be lost moving deeper towards higher layers of the network. So these features are pretty essential to distinguish between the two look-alike objects and need to be restored while propagating through the network.

Similarly, the deeper networks may suffer from the problem of degradation that arises due to overfitting or the depth of the network for which gradient may disappear, and deeper layers face issues while training. Thus model may return erroneous outputs. To minimize this problem, a skip connection (Srivastava et al., 2015; He et al., 2016) mechanism has been introduced that creates a secondary path for the gradient to move along the deeper layers so that proper training of the model can be possible. Some of the proposed architecture, such as U-Net (Ronneberger et al., 2015), are proposed to address these issues in the FCNs specifically for medical images. It is an end-to-end, fully convolutional network that uses skip connections (He et al., 2016; Huang et al., 2017) to pass the local features to the deeper layer. Hence, they can combine high-resolution features better to predict the labels of the object of interest, thus playing a vital role in semantic scene parsing or segmentation.

Like semantic scene understanding (based on semantic segmentation), aerial scene parsing is one of the fundamental topics in computer vision, where the goal is to assign each pixel of an aerial image a categorical label. Thus, it provides a complete understanding of a particular scene by predicting each element's label and shape. In the past, satellite images were the only images considered for aerial scene parsing. They have been used in diversified applications, including agriculture mapping, archaeological survey, land cover classification, climate change observation, innovative city applications, disaster management, ecological survey, military applications, weather prediction, etc. However, with the evolution of UAV technology, this field possesses some new interests for different applications in robotics, automatic UAV flying, and sensing. Difficulty in aerial scene parsing techniques lies not only in scene and label variety but also in preparing the dataset. In the past few years, several attempts have been made by various computer vision research groups to build UAV-based datasets to satisfy the increased demand for the new age algorithms (Mundhenk et al., 2016; Xia et al., 2018; Hsieh et al., 2017; Barekatain et al., 2017; Robicquet et al., 2016). But these attempts mainly focus on supporting the ongoing research on object detection and tracking techniques from aerial images. A few drone-based aerial datasets like Aeroscapes (Nigam et al., 2018), UDD (Chen et al., 2018a), Semantic-DD (Semantic Drone Dataset, 2018) have been developed and made public in the interest to work on the field of semantic segmentation. Still, it possesses some limitations from the point of collection of datasets and labeling as these tasks take a severe amount of time. Moreover, the approaches toward satellite-based object extraction techniques use heuristic-reasoning-based and straight-line-based methods for multiclass segmentation problems where objects like road lines look like a straight line. Hence, these approaches are not suitable enough to perform the task in UAV-based segmentation problems. Similarly, the existing state-of-the-art mechanisms usually suffer from the model complexity in terms of their number of parameters, thus restricting their use in small-scale devices. These reasons motivate us to work in this problem direction. In this article, we aim to use the flexibility of the U-Net for aerial scene parsing and removing the underlying problems for a better understanding of different objects in a scene. By redefining the intermediate layers of the model and changing some of the stages, we aim to achieve better accuracy than state-of-the-art architectures.

This article discusses the use of the skip connection mechanism in terms of partially dense connections and residual connections to reuse the learned features from the previous layers, thereby increasing the feature space while reducing the number of parameters. Following are the main contributions of the paper:

1. We have proposed a deep CNN architecture inspired by skip connections (dense and residual connections) and the legendary U-Net. The internal stages of the architecture are replaced by partial dense modules that reduce the number of trainable parameters of the proposed architecture to a great extent.
2. Unlike U-Net, the proposed model does not use any pre-trained backbone for down-sampling of the input images to collect the required features during the encoding path. Instead, the proposed model is trained from scratch on the two considered UAV-image datasets to collect the native complex patterns to perform the segmentation task.
3. With the reduced number of trainable parameters, the proposed architecture also performs better in pixel-level classification by extracting the important features concering vegetation in a multiclass environment.

The rest of the paper is organized as follows: Section 2 discusses the existing research works in the field of semantic segmentation and aerial scene parsing. A detailed overview of the proposed deep architecture is presented in Section 3. Section 4 describes the experimental results and a detailed comparison with other state-of-the-art methodologies. Finally, the conclusion drawn from this research work has been depicted in Section 5, along with the future directions.

2. Related work

In this section, we have briefly discussed various approaches towards the topic of aerial scene parsing. The classical approaches, along with CNN-based architectures, are taken into account to understand the evolution of the methods that the researchers have proposed to address the problem of aerial scene understanding.

2.1. Image segmentation and classical methodologies

Aerial data can be interpreted as the data of the earth that is collected from above the earth. Satellites have played a crucial role in obtaining information for many years and still serve human goods. The unmanned flying objects, known as UAVs, are also employed for many tasks, which the satellite does, but on a lower scale. These devices capture images of the earth's surface, which are the prime sources for urban mapping, agriculture mapping, forest mapping, etc., that include tasks like ground-level object detection like roads, buildings, trees, cars, pedestrians, bikers, and many more. The aerial images are complex due to the enormous population of objects in a single aerial scene making the semantic segmentation task more challenging. Early works attempted by researchers used rule descriptor-based methods to extract the object features, mainly in the case of building extraction (Mohan and Nevatia, 1989) and road extraction (Fischler et al., 1987; Stilla, 1995). However, poor generalization of the aerial data creates limitations for the hierarchical rule-based approaches; even hard-coded experts also miss much significant evidence.

Machine learning is employed to learn classification rules from the given data. Generally, conventional classifiers are given the raw pixel intensities as input to extract the local features through simple arithmetic combinations (Leung and Malik, 2001; Schmid, 2001). Discriminating classifiers such as Boosting and Random forests are also used to compute the redundant set of local feature maps that can be used for training (Viola and Jones, 2001; Fröhlich et al., 2013; Tokarczyk et al., 2015).

From the perspective of aerial image segmentation, global features are as important as local features. Authors in Chai et al. (2013) and Ortner et al. (2007) have used the concept of marked point processes (MPPs) to design and model building-road network topologies through probabilistic priors defined to extract the global knowledge. MPPs count on object primitives such as lines and rectangles to match the input image data via sampling. Graphical models provide a good amount of

flexibility while modeling such complex problems, but on the other hand, they may lead to complicated optimization issues. Similarly, methodologies such as conditional random fields (CRFs) are also used for object extraction and segmentation problems in computer vision (Wegner et al., 2013; Wegner et al., 2015).

As modeling high-level correlation becomes a difficult task, many researchers attempted to improve the local evidence by finding more discriminating feature spaces (Dalla Mura et al., 2010). The resultant feature/evidence vectors are inputs to standard classifiers such as decision trees and support vector machines to deduce the probabilities per object class. A lot of efforts have been invested in achieving a reduced feature space to a discriminative subset (Rezaei et al., 2012; Schwartz et al., 2009).

2.2. Deep learning and aerial image segmentation

Deep learning approaches don't need a feature definition step; instead, they learn the essential discriminating features from the input raw images according to the given task. These algorithms were developed in the 80's Fukushima and Miyake (1982) and LeCun et al. (1989) when they suffered from limited computing power and training data. However, their return was announced by Krizhevsky et al. (2017) in 2012 through achieving impressive results in the ImageNet challenge (Russakovsky et al., 2015). A large number of layers (often varies from 10 *to* $>$ 100) in the state-of-the-art networks help in learning and analyzing the local and global feature maps from raw input image data. Moreover, the training and inference in the deep CNN architectures can be parallelizable through GPUs, leading to a low execution time. Similarly, in Behera et al. (2021), the authors have provided a detailed review of the existing UAV-image-based datasets to support the ongoing research work in UAV-image-based applications.

CNN's journey from image classification to semantic segmentation is quite remarkable and a quick one (Farabet et al., 2013; Cheng et al., 2018). The use of CNNs is not only limited to ground-level scene parsing but can also be seen in aerial scene parsing that uses remote sensing images (Zhu et al., 2017). Common tasks in this field include building footprint extraction (Van Etten, 2018), road extraction (Zhang et al., 2018; Bastani et al., 2018) and vegetation extraction (Kattenborn et al., 2021). Our approach toward aerial image parsing is based on an encoder-decoder-inspired, fully convolutional structure (Long et al., 2015; Ronneberger et al., 2015) that yields a spatially explicit label image that represents the context related to each pixel. It then propagates through the expansion module to up-sampling back to the resolution as the original one. Recently, UAVs have been widely used as remote sensors, and the images collected by them have been used in many small-scale applications. Researchers have come forward to contribute to this field by addressing the existing problems through their proposed architectures (Gibril et al., 2021; Pandey and Jain, 2021; Zhou et al., 2017).

In this article, we propose one such deep learning-based architecture that semantically segments the UAV images, thereby mainly detecting vegetation, along with other object classes like roads and buildings, to show the behavior of the proposed model in a multiclass segmentation environment. The main intention is to reduce the number of parameters of the deep architecture, maintaining the overall prediction accuracy. Our model benefits from skip connections in the form of partial dense blocks and residual connections to retain the flow of feature maps and gradients from the lower layers to the higher layers. The following section talks about the proposed approach.

3. Proposed work

This section presents different stages of the proposed deep convolutional architecture that extracts the essential features to segment various pixels of an aerial image. The architecture is an end-to-end, fully connected CNN that benefits from the two important architectures, i.e., U-Net and DenseNet (especially dense connections).

3.1. Rationale for new architecture

For the same level of accuracy, the deeper networks are usually considered over the corresponding shallower counterparts because of their ability to learn a new and more abstract representation of the provided input through a deep representation. However, these come at the cost of a common problem known as the "degradation problem," where the model's performance degrades as the architecture's load increases. One of the common reasons behind such a problem could be overfitting or the depth of the model that may cause the disappearance of the gradients or exploding gradient. Therefore, the concept of "skip connections" has been introduced to mitigate such problems. It ensures smooth gradient passes to tackle the degradation problem and feature reusability in many models.

3.2. Architectural overview

In a semantic segmentation task, it is crucial to get the low-level feature maps while retaining the high-level semantic details (Long et al., 2015; Ronneberger et al., 2015). U-Net (Ronneberger et al., 2015) was developed to address the underlying issues in the field of medical image segmentation. However, its use cases are beyond medical image applications and are a popular and successful image segmentation methodology in several fields, including satellite-based aerial segmentation, robotics scene understanding, etc. The architecture of U-Net helps in propagating the finer low-level features to the corresponding upper layers that, in turn, return a detailed, crisp segmented output of the corresponding input image. It has the advantages of training with fewer samples while retaining accuracy through the powerful usage of data augmentation, thereby making it a benchmark model for semantic scene parsing. However, certain apparent drawbacks of the U-Net architecture need to be addressed. One such drawback is that learning may slow down in the middle layers in the case of deeper networks, which results in the situation where the network may ignore some of the intermediate layers representing abstract features. This situation may also arise due to the dilution of gradients from the output of an architecture resulting in slower learning in far removed weights. Another challenge commonly encountered in UAV-based aerial scene parsing is the presence of diversified objects in a small space. These complexities in an aerial scene may create an issue in obtaining low-level feature sets requiring finer details. Our proposed approach tries to mitigate these observed problems in the existing architectures by implementing dense and residual connections.

The proposed architecture tries to take advantage of the skip connections mechanism in the form of residual connections and dense modules. The base model is a U-Net (Ronneberger et al., 2015) one, where the dense modules partially replace the internal stages; thereby, feature reuse can be done from the earlier CNN layers within a stage of the network. The obtained features are then propagated to the advancing stages. Additionally, the learned features from each stage are also directly passed to the corresponding deconvolution stages. Thus, in both ways, the reusability of the extracted features from the previous layers is done. The flow diagram of the proposed architecture, i.e., AerialSegNet, is given in Fig. 1. The proposed architecture is an encoder-decoder based FCN architecture and can be viewed as a combination of the following levels:

a. Contraction path: It decomposes the input RGB images through convolution operation so as to get the spatial and temporal features.
b. Dense Modules: At each stage of the architecture, we have used a few partially densely connected modules so that the previously learned features can be concatenated to enhance the number of features they use in the next stage.

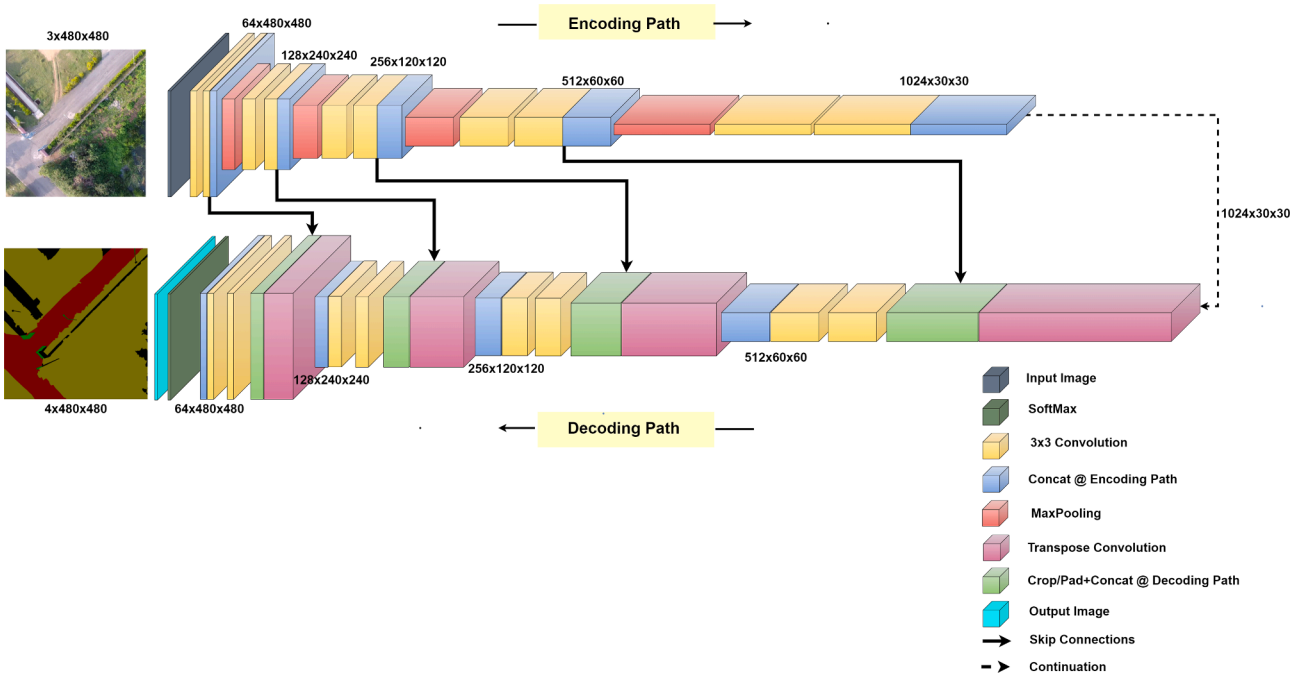

Fig. 1. Detailed overview of the proposed AerialSegNet.

c. Bottleneck stage: This is the last stage of the contraction operation, after which the network will expand to generate the original sized image. It collects the features obtained from the contraction path and transfers them back to the deconvolution blocks of the expansion path.

d. Expansion path: Here, the deconvolution operations are performed on the features obtained from the encoder/contraction path to produce the same sized segmented image as the input one.

e. Finally, a "SoftMax" activation function is used at the end of the expansion path to normalize the output of the architecture to a probability distribution over predicted output classes.

3.3. Stages

As presented above, the proposed model consists of five stages and are described as follows:

3.3.1. Contraction path

As mentioned earlier, the proposed framework is encoder-decoder-based architecture, i.e., an encoding or contracting path that reduces the feature maps of the input image. It follows the typical CNN architecture that contains a series of convolutional operations. This is nothing but repeated applications of two 3×3 or 5×5 convolutions (padded convolutions), each followed by batch normalization (Ioffe and Szegedy, 2015), ReLu (rectified linear unit) (Agarap, 2018), and a 2×2, which can be called one CNN layer. Finally, it is followed by 2×2 Max Pooling operation of stride 2 to compress the image to extract the essential features. Both the convolution and max pooling operations are used to down-sampling the input images.

3.3.2. Dense modules

In general, CNN connects the previous layer output to the subsequent layer output, and so on. On the contrary, in dense blocks (from DenseNet), each layer accepts inputs from its preceding layers and forwards its outputs to all of its subsequent layers. A few cues from the dense block are borrowed and introduced to the proposed architecture due to its compelling advantages in feature reusability and parameter minimization. The proposed architecture exploits the power of dense modules through feature reuse, thus yielding a partial dense structure within each stage of the network that is easy to train because of its efficiency in terms of the number of trainable parameters. One of the most significant advantages of the dense network is the variation of inputs the layers usually get from the previous layers that help it improve its accuracy. A comparison of the number of parameters in the proposed model with the state-of-art methods is mentioned in Table 1. At each stage of the contraction/expansion path of architecture, the outputs of the convolutional layers are passed to the subsequent layers in that particular stage. The flow of each stage is presented in Fig. 1 and Table 2.

3.3.3. Bottleneck layer

This is the layer that acts as a bridge or mediator between the contraction path and expansion path. Usually, the bottleneck layer tries to extract the most helpful information from the feature maps of the contraction path and passes the same to the expansion layers. This layer consists of two 3×3 CNN layers followed by a concatenation operation to create a partial dense module. The output of the bottleneck layer then presents to the next layer for re-creating the image back as the input.

3.3.4. Expansion path

The core of the architecture lies at the expansion stage. Identical to the contraction path, there are several blocks in the expansion path. Each module consists of one 2×2 transpose convolution layer with a stride 2 concatenated to the residual connection from the mirror layer of

Table 1
Comparison of Number of Trainable Parameters.

State-of-the-art	Parameters Lower the better
U-Net_ResNet-18 (Ronneberger et al., 2015)	15.50*M*
U-Net_ResNet-50 (Ronneberger et al., 2015)	36.49*M*
FCN-8s (Long et al., 2015)	136*M*
FCN-16s (Long et al., 2015)	134*M*
FCN-32s (Long et al., 2015)	134*M*
FC_DenseNet-103 (Jégou et al., 2017)	9.42*M*
SegNet (Badrinarayanan et al., 2017)	29.44*M*
DeepLab_V3_plus_Xception (Chen et al., 2018b)	41.3*M*
AerialSegNet (proposed)	11.76*M*

M = Million.

Table 2
Network structure of the Proposed Architecture (AerialSegNet).

	Stages	Operational Layers	Kernel Size | # of Filters	Stride	Output Size (C × H × W)
Input					3 × 480 × 480
Contraction Path	Stage-1	Conv-1	3 × 3 | 32	1	32 × 480 × 480
		Conv-2	3 × 3 | 32	1	32 × 480 × 480
		Concatenation-1	—| 64	—	64 × 480 × 480
		MaxPool-1	2 × 2 | 1	2	64 × 240 × 240
	Stage-2	Conv-3	3 × 3 | 64	1	64 × 240 × 240
		Conv-4	3 × 3 | 64	1	64 × 240 × 240
		Concatenation-2	—| 128	—	128 × 240 × 240
		MaxPool-2	2 × 2 | 1	2	128 × 120 × 120
	Stage-3	Conv-5	3 × 3 | 128	1	128 × 120 × 120
		Conv-6	3 × 3 | 128	1	128 × 120 × 120
		Concatenation-3	—| 256	—	256 × 120 × 120
		MaxPool-3	2 × 2 | 1	2	256 × 60 × 60
	Stage-4	Conv-7	3 × 3 | 256	1	256 × 60 × 60
		Conv-8	3 × 3 | 256	1	256 × 60 × 60
		Concatenation-4	—| 512	—	512 × 60 × 60
		MaxPool-4	2 × 2 | 1	2	512 × 30 × 30
Bottleneck Layer	Stage-5	Conv-9	3 × 3 | 512	1	512 × 30 × 30
		Conv-8	3 × 3 | 512	1	512 × 30 × 30
		Concatenation-5	—| 1024	—	1024 × 30 × 30
Expansion Path		UpPool-1	2 × 2 | 1	2	1024 × 60 × 60
		Concatenation-6	—| 1536	—	1536 × 60 × 60
		Conv-10	3 × 3 | 256	1	256 × 60 × 60
		Conv-11	3 × 3 | 256	1	256 × 60 × 60
		Concatenation-7	—| 512	—	512 × 60 × 60
	Stage-7	UpPool-2	2 × 2 | 1	2	512 × 120 × 120
		Concatenation-8	—| 768	—	768 × 120 × 120
		Conv-12	3 × 3 | 128	1	128 × 120 × 120
		Conv-13	3 × 3 | 128	1	128 × 120 × 120
		Concatenation-9	—| 256	—	256 × 120 × 120
	Stage-8	UpPool-3	2 × 2 | 1	2	256 × 240 × 240
		Concatenation-10	—| 384	—	384 × 240 × 240
		Conv-14	3 × 3 | 64	1	64 × 240 × 240
		Conv-15	3 × 3 | 64	1	64 × 240 × 240
		Concatenation-11	—| 128	—	128 × 240 × 240
	Stage-9	UpPool-4	2 × 2 | 1	2	128 × 480 × 480
		Concatenation-12	—| 192	—	192 × 480 × 480
		Conv-14	3 × 3 | 32	1	32 × 480 × 480
		Conv-15	3 × 3 | 32	1	32 × 480 × 480
		Concatenation-11	—| 64	—	64 × 480 × 480
Final Layer	Stage-10	Conv-16	1 × 1 | M	1	M × 480 × 480

*Conv-N is a three step procedure, includes Convolution operation followed by a BatchNormalization and ReLU layer, *M=# of object classes in a dataset.

the contraction path that is immediately followed by two 3 × 3 convolutional layers and a concatenation operation for the two CNN layers. After each module, the number of filters used in convolutional layers becomes half to maintain symmetry. However, appending residual inputs from the corresponding contraction layers ensure that the learned features from the contraction path will be reused while reconstructing the image. Just like the architecture of U-Net (Ronneberger et al., 2015), the number of blocks in the contraction path is the same as that of the expansion path. Finally, the resultant map passes via a 3 × 3 convolution layer followed by a "SoftMax" activation function to produce the probability distribution over the predicted output classes, which can then be used to generate the desired segmentation map.

3.4. Loss function

The choice of loss function plays a critical role while carrying out any neural network-based optimization. The loss-weighting scheme used by the network architecture targets the interior pixels and the border of the segmented object. Thus, the resultant segmentation map will contain finer details at the edge lines. The Categorical Cross-Entropy (CCE) loss, also known as logarithmic or logistic loss, is selected to train the baseline models. The predicted class probability is compared with the truly desired class output of 0 or 1. The corresponding loss/score is obtained to penalize the probability based on how far it deviates from the actual expected value. Softmax differential function (S_i) is also used with CCE, which aims to minimize the loss during training, i.e., the smaller the loss value the better the model. Cross-entropy can be defined as per Eq. 1:

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

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

4. Experiments

To illustrate the efficiency and accuracy of the proposed model, we test it on two UAV-based aerial image segmentation datasets: NITR-Drone dataset[1] and Urban Drone Dataset (UDD).[2] The obtained results are compared with other state-of-the-art methodologies, including U-Net (Ronneberger et al., 2015), FCN-8s (Long et al., 2015), FCN_DenseNet-103 (Jégou et al., 2017), SegNet (Badrinarayanan et al., 2017). The experimental flow is presented in Fig. 2.

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

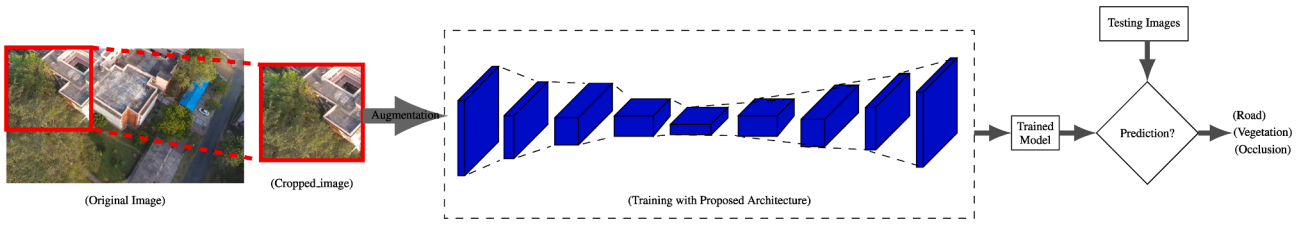

Fig. 2. Flow diagram for the experimentation.

4.1. Dataset

The two datasets: the NITRDrone dataset and the Urban Drone dataset, have been considered to validate the proposed model. This subsection briefly presents the details of the two mentioned datasets.

4.1.1. NITRDrone dataset

The NITRDrone dataset (NITRDrone Dataset, 2021; Behera et al., 2022) consists of 101 frames of training and a validation set. These images have been acquired by employing *DJI Phantom* 4 and *DJI Mavic Mini* drones. The images are of different resolutions ranging from 720×1280 to 3000×4000 collected from various corners of the NITR campus in different lighting conditions. These drone images were captured while operating at an altitude of $5-80$ *meters*, which leads to an approximate ground sampling distance *(GSD)* of 0.025 *sq.cm/pixel*. The dataset contains several ground objects, of which four classes are considered: road, vegetation, occluded road, and background. Here, the class "occluded road" represents the pixels covered by any movable objects (except vegetation and buildings), such as pedestrians, bicyclists, and cars, which come on the way while viewing the road from the viewpoint of a UAV. The dataset is divided into 71 and 30 as training and testing set out of the total images. From the total of 101 full-sized annotated images, we have managed to generate $2,452$ and 510 images to be considered for training and testing, respectively for performing the experimentation. Two types of augmentation techniques, such as center cropping and random cropping (of constant size), are applied to the images of the training set. In other words, a total of $4,950$, 480×480 images have been used to train the model, and the rest 510 images are used to test the trained model. A few Sample images of the dataset are presented in Fig. 3.

4.1.2. Urban Drone Dataset (UDD)

The UDD is a UAV-based image dataset that was proposed by Chen et al. for semantic segmentation problems in computer vision (Chen et al., 2018a). The dataset is collected by a *DJI Phantom* 4 operated at an altitude of 60 meters to 100 meters. 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 with three, five, and six classes, respectively from which we have considered UDD- 5 on which the proposed model is implemented and validated. UDD- 5 has five-pixel classes named vegetation, buildings, road, vehicles, and others (denoted for the rest of the objects in a scene other than the mentioned classes). The dataset comprises two sets training set and a validation set consisting of 160 images and 45 frames, respectively. Out of the above-mentioned full-sized training images, a total $3,156$ (training set) and 888 (validation set) 512×512 images are generated on which the experiment is performed. Sample images and the masks are shown in Fig. 4.

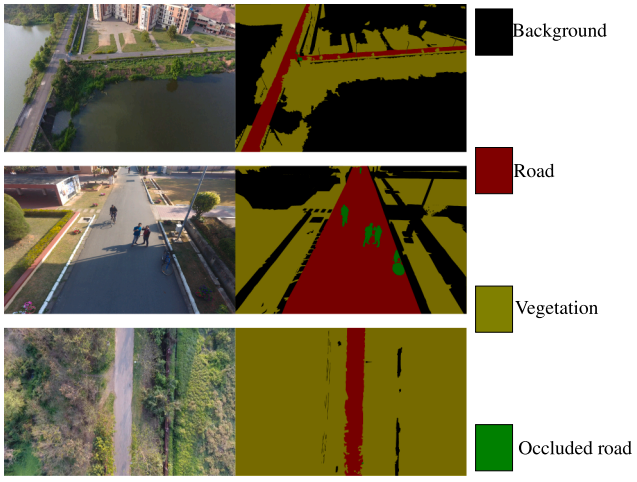

Fig. 3. NITRDrone dataset containing UAV captured aerial outdoor images (includes four classes).

4.2. System setup

The proposed architecture is implemented and validated with some benchmark models in the semantic segmentation research field. The architecture is implemented in PyTorch[3] (PyTorch Documents, 2016) and trained with a NVIDIA TITAN V graphics card with 12*GB* GPU memory. The weights are initialized with random weights as the training is done from scratch. The weights for the batch normalization and ReLU have been initialized from a standard normal distribution.

4.3. Training

Training is an integral part of implementing the proposed deep learning model, especially when everything is being done from scratch.

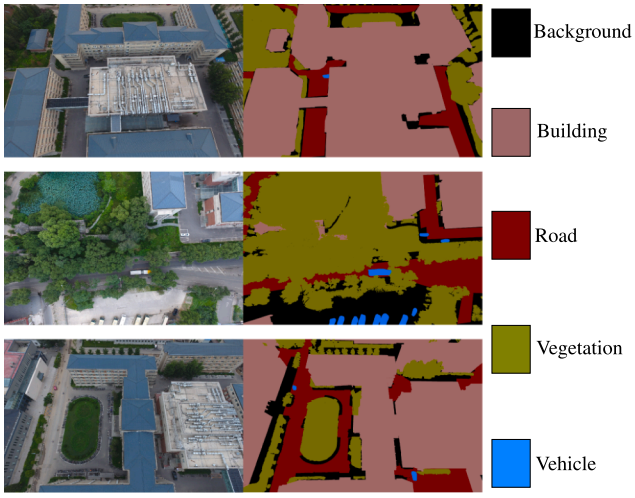

Fig. 4. Urban Drone Dataset (UDD) containing UAV-captured aerial outdoor images (includes five classes).

[3] https://github.com/pytorch/pytorch.

4.3.1. Pre-processing

The captured images/input images are of higher resolution and can not be considered input directly to the network for training because of the computational limitation. Hence the input images are cropped into smaller tiles and can be denoted as **Image X**. The **Image X** is then pre-processed to get the **Image Y** that is being trained by the deep neural network to extract the underlying features. The proposed model is designed to accept any size or resolution of input images. However, the images are of different sizes, such as $720 \times 1280, 2160 \times 4096$, *and* 3000×4000 . Hence they need to be pre-processed before considering as input for the architecture. These images are cropped to get tiles of 512×512 (Image X) with some overlapping. Finally, the input images are down-sampled with a resolution of 480×480 (Image Y) for the training to keep a constant batch size throughout our experiment.

4.3.2. Target pre-processing

As the input images are downscaled to 480×480 , the corresponding target or the masks need to be scaled within the input images' range. But the training is done with the one-hot code masks. The one-hot coded target images are then color-coded with different colors for different classes or object pixels to have segmented images. This step is included in the whole process to effectively visualize the different object pixels. The identical RGB color code is followed as that of the RGB mask, which is presented in Figs. 3 and 4.

4.3.3. Implementation details

Adaptive moment estimation (Adam) (Kingma et al., 2015) is used as the optimizer, whereas Categorical Cross-Entropy (CCE) is used as the loss function and is presented in Eq. 1. The learning rate is initialized to 0.005 with a batch size set to four, while momentum is initialized to 0.9 . Weight decay of 0.0002 is employed to handle the problem of overfitting (if it occurs) in the network. Moreover, we have also employed two types of augmentation techniques in the training image set: center-cropping and random cropping to handle such scenarios. During training, after each 25 epochs, the learning rate is made to be decreased by a factor of 0.2 to maintain regularization. The training process continued until the learning rate reached 10^{-20}. While training, the proposed architecture is converged post 300 epochs, after which there are hardly any changes that can be noticed in loss and accuracy.

4.4. Tasks and metrics

The objective of our model is to parse the scene and segment the input aerial images based on the number of given objects that need to be considered, as in our case, it is four (NITRDrone) and five (UDD). The architecture needs to be validated numerically as well as graphically, and to evaluate the proposed method; we have used some of the widely used metrics such as precision (P), recall (R), f-score (F), the intersection of union (IoU), overall accuracy (A). These metrics can be computed as follows:

$$Precision\ \left(P\right) = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall\ \left(R\right) = \frac{TP}{(TP + FN)} \tag{3}$$

$$\begin{aligned} Intersection\ over\ Union\ (IoU) &= \frac{Area\ of\ Overlap}{Area\ of\ Union} \\ &= \frac{TP}{(TP + FP + FN)} \end{aligned} \tag{4}$$

$$\begin{aligned} F - Score\ (F) &= \frac{Area\ of\ Overlap}{Count\ of\ pixels\ in\ both\ GT\ and\ PR} \\ &= \frac{2PR}{(R + P)} \\ &= \frac{2TP}{(2TP + FP + FN)} \end{aligned} \tag{5}$$

$$Pixel\ Accuracy\ \left(A\right) = \frac{TP + FP}{(TP + TN + FP + FN)} \tag{6}$$

Here, TP, TN, FP, and FN denote the count of pixels that are true positives, count of pixels that are true negatives, count of pixels that are false positives, and count of pixels that are false negatives, respectively. Similarly, GT and PR in Eq. 5 represent an aerial image's ground truth mask and predicted mask.

4.5. Results and discussion

This Section discusses the obtained results of the proposed AerialSegNet model over a few state-of-the-art methodologies. We have also highlighted the improvements that have been made concerning the number of parameters in Section 3.

4.5.1. Observation

After training, the trained model is tested on the validation set, and the obtained results are presented in Tables 3 and 4. The implementation is performed on two UAV-captured datasets: one has four object classes, and the other has five object classes. Our primary objective is to analyze the proposed model's performance concerning the vegetation class in a multiclass segmentation environment. As per the results obtained from training, it is visible that the proposed architecture learns the desired features to segment the target object classes. One of the crucial aspects of using dense modules in the architecture is to reduce the number of parameters, thereby reducing the computational complexity. At the same time, we are also required to have smoother decision boundaries. The architecture detects the border of the objects quite well and can be seen in Figs. 5 and 6. At the beginning of the training, the proposed model failed to detect the occlusion class objects (when trained with the NITRDrone dataset); however, it recovered nicely as the training progressed. A similar case can be seen in UDD for vehicle class, which is a minority class, and the model invested a reasonable amount of time in accessing the crucial underlying features for this particular class. Moreover, we have also observed certain limitations that the model suffers from and are described as follows: the model's performance is reduced in low light as it cannot individualize among similar-looking objects (e.g., the road looks similar to the roof sometimes because of the tar-covered sheet). Under these circumstances, the model has coarsely annotated the object classes without proper boundaries, and the predicted images look faded. Similarly, while training with the UDD image dataset, the proposed model takes sufficient time to pile up the feature sets. The model can be seen getting less fuzzy boundaries of the objects than other considered baseline models. Hence, our model performs better than the other state-of-the-art mechanisms in classifying the pixel labels under these circumstances.

4.5.2. State-of-the-art comparison

The proposed model performs superior to the existing methods (Long et al., 2015; Ronneberger et al., 2015; Badrinarayanan et al., 2017) for all the evaluation metrics as presented in Eqs. (2)–(6). It has also been observed from Table 3 that one of the considered baseline models, i.e., DeepLab_V3 + Xception (Chen et al., 2018b), performs better than our proposed architecture by achieving a higher precision score. Hence, it implies a lower false-positive rate (in DeepLab_V3 + Xception) than the proposed approach while working on an imbalanced dataset like the NITRDrone dataset, where two object classes: road and vegetation,

Table 3
Overall performance evaluation of various state-of-the-art mechanisms on NITRDrone dataset.

	mPrecision	mRecall	mIoU	mFscore	mAccuracy	time/valImage
FCN-8s (Long et al., 2015)	0.58	0.35	0.12	0.14	0.55	43.1 ms
FCN-16s (Long et al., 2015)	0.81	0.83	0.76	0.81	0.95	39.2 ms
FCN-32s (Long et al., 2015)	0.80	0.82	0.71	0.76	0.93	39.2 ms
FC_DenseNet-103 (Jégou et al., 2017)	0.81	0.79	0.68	0.74	0.90	16.7 ms
SegNet (Badrinarayanan et al., 2017)	0.84	0.74	0.68	0.76	0.92	25.4 ms
DeepLab_V3_plus_Xception (Chen et al., 2018b)	**0.91**	0.79	0.74	0.83	0.96	35.2 ms
U-Net "ResNet-18" (Ronneberger et al., 2015)	0.84	0.90	0.80	0.85	**0.98**	22.5 ms
U-Net "ResNet-50" (Ronneberger et al., 2015)	0.85	0.90	0.81	0.87	**0.98**	23.6 ms
AerialSegNet (proposed)	0.88	**0.91**	**0.84**	**0.88**	**0.98**	19.6 ms

mPrecision = mean Precision, *msec* = milliseconds.

Table 4
Overall performance evaluation of various state-of-the-art mechanisms on Urban Drone dataset (UDD).

	mPrecision	mRecall	mIoU	mFscore	mAccuracy	time/valImage
FCN-8s (Long et al., 2015)	0.58	0.33	0.116	0.14	0.49	47.2 ms
FCN-16s (Long et al., 2015)	0.72	0.71	0.60	0.68	0.94	43.1 ms
FCN-32s (Long et al., 2015)	0.66	0.53	0.54	0.60	0.92	42.8 ms
FC_DenseNet-103 (Jégou et al., 2017)	0.55	0.62	0.52	0.59	0.92	20.8 ms
SegNet (Badrinarayanan et al., 2017)	0.59	0.63	0.54	0.61	0.94	29.2 ms
DeepLab_V3_plus_Xception (Chen et al., 2018b)	0.75	0.73	0.65	0.72	0.95	39.8 ms
U-Net "ResNet-18" (Ronneberger et al., 2015)	0.785	0.82	0.72	0.80	**0.96**	23.7 ms
U-Net "ResNet-50" (Ronneberger et al., 2015)	0.79	0.83	0.728	0.81	**0.96**	26.8 ms
AerialSegNet (proposed)	**0.80**	**0.84**	**0.739**	**0.82**	**0.96**	22.7 ms

mPrecision = mean Precision, *msec* = milliseconds.

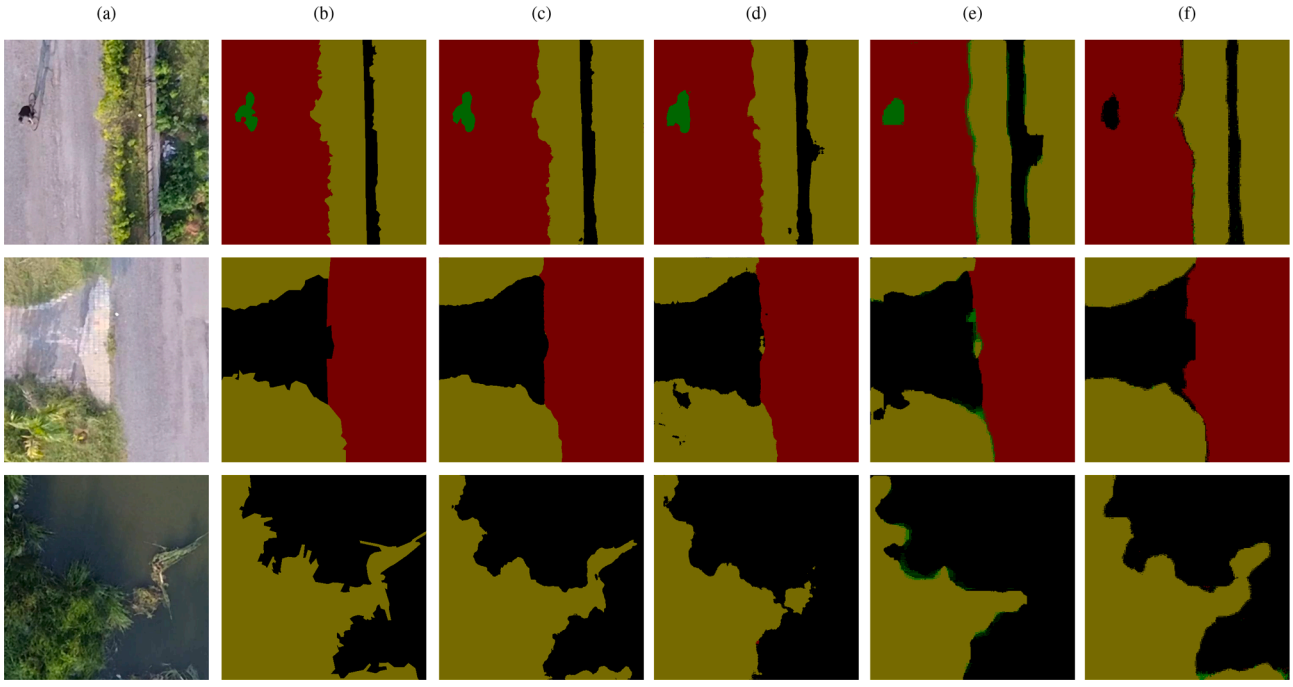

Fig. 5. Prediction of the state-of-the-art mechanisms. Left to right: (a) UAV-based aerial images, (b) labelled mask of the corresponding images, (c) Proposed AerialSegNet, (d) UNet (Ronneberger et al., 2015), (e) FCN-16s (Long et al., 2015), and (f) FCN-32s (Long et al., 2015). The color coding of the semantic classes matches Fig. 3.

dominate over the others. It might also be due to repeated predictions on certain types of images. However, it can be seen that there is a higher gap between precision and recall in DeepLab_V 3 (as compared to the proposed architecture), which is undesirable in a semantic segmentation task. This is where the proposed model becomes a superior one maintaining a reasonable true positive and negative rate with a low 11.76 million parameters as compared to (≈ 41 million) of DeepLab_V3. Hence, the proposed model becomes a better choice for real-time segmentation applications. As per the parameters of the model concern, the number of parameters required by our proposed architecture is less than all of the state-of-the-art methods except FC_DenseNet- 103 (Jégou et al., 2017), which takes 9.42 million parameters (almost 2 million less than

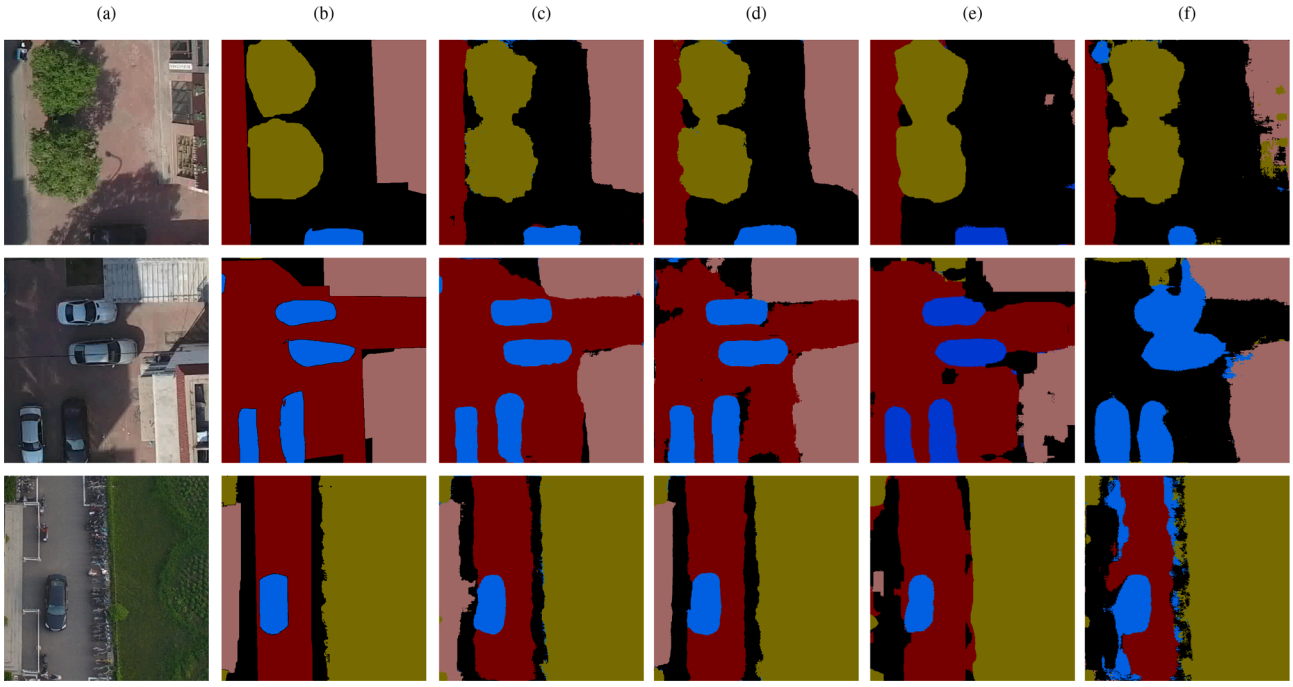

Fig. 6. Prediction of the state-of-the-art mechanisms. Left to right: (a) UAV-based aerial images, (b) labelled mask of the corresponding images, (c) Proposed AerialSegNet, (d) UNet (Ronneberger et al., 2015), (e) FCN-16s (Long et al., 2015), and (f) FCN-32s (Long et al., 2015). The color coding of the semantic classes matches Fig. 4.

that of ours). But our model performs way better than it. Similarly, if we will look at the three fully convolutional network-influenced models, such as FCNs (Long et al., 2015), which have more than 130 million parameters. In that case, still, they are unable to fetch the critical feature space required for a crisp prediction in the validation set. One of the classes, "occluded road," is remained undetected on prediction by FCN-32s, and a similar case can be seen with respect to UDD as well, where two classes, "road" and "vehicle," are not appropriately detected by the model in many cases leading to low accuracy. Moreover, the proposed model uses 13 times fewer parameters than FCN models and achieves better results. The proposed framework achieves a prediction speed of 19.6 milliseconds (msecs) and 22.6 msecs per validation image of the NITRDrone and UDD dataset, respectively, which is faster than the other considered state-of-the-art mechanisms except for FC-DenseNet architecture.

5. Conclusions

This article presents a deep learning framework to address pixel-based classification problems from the UAV-based aerial images. The proposed network combines the strength of skip connections (in the form of the partial dense modules and residual connections) and encoder-decoder-based deep architecture. The dense connections in the network at each stage help in feature reuse, while skip connections help propagate the gradient and lower-order feature maps to the higher-order mirror layers. This approach not only eases the process of training but also enables the design simple, rather a robust network with fewer trainable parameters. We evaluate our architecture on two publicly available UAV image datasets, namely the NITRDrone dataset and UDD, and compare the obtained results to state-of-the-art methods. The experimentally obtained results imply that our proposed architecture performs better quantitatively and qualitatively than other standard segmentation approaches. The model can extract the implicit features to segment the object classes, such as road and vegetation. Thus, it can be applied to robotics-based solutions for road extraction, vegetation detection, and crop field extraction through panoptic aerial view imageries of the UAV.

Abbreviations

CCE	:	Categorical Cross Entropy
CNN	:	Convolutional Neural Network
CRF	:	Conditional Random Fields
FCN	:	Fully Convolutional Network
FN	:	False Negative
FP	:	False Positive
GSD	:	Ground Sampling Distance
GT	:	Ground Truth
IoU	:	Intersection of Union
ReLU	:	Rectified Linear Unit
RS	:	Remote Sensing
TN	:	True Negative
TP	:	True Positive
UAV	:	Unmanned Aerial Vehicle
UDD	:	Urban Drone Dataset

Declaration of Competing Interest

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

Acknowledgment

This research is supported by the following projects:

a. 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.

b. 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

Agarap, A.F., 2018. Deep learning using rectified linear units (ReLu). arXiv preprint arXiv:180308375.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Image Process. 39 (12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Barekatain, M., Martí, M., Shih, H.F., Murray, S., Nakayama, K., Matsuo, Y., Prendinger, H., 2017. Okutama-action: an aerial view video dataset for concurrent human action detection. In: 1st Joint BMTT-PETS Workshop on Tracking and Surveillance, CVPR. IEEE, pp. 1–8. https://doi.org/10.1109/CVPRW.2017.267.

Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., DeWitt, D., 2018. RoadTracer: automatic extraction of road networks from aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4720–4728. https://doi.org/10.1109/CVPR.2018.00496.

Behera, T.K., Bakshi, S., Sa, P.K., 2021. Aerial Data Aiding Smart Societal Reformation: Current Applications and Path Ahead. IEEE IT Profess. 23 (3), 82–88. https://doi.org/10.1109/MITP.2020.3020433.

Behera, T.K., Bakshi, S., Sa, P.K., Nappi, M., Castiglione, A., Vijayakumar, P., Gupta, B., 2022. The NITRDrone Dataset to address the Challenges for Road Extraction from Aerial Images. J. Signal Process. Syst.

Chai, D., Förstner, W., Lafarge, F., 2013. Recovering line-networks in images by junction-point processes. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1894–1901. https://doi.org/10.1109/CVPR.2013.247.

Chen, Y., Wang, Y., Lu, P., Chen, Y., Wang, G., 2018a. Large-Scale Structure from Motion with Semantic Constraints of Aerial Images. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, pp. 347–359. https://doi.org/10.1007/978-3-030-03398-9-30.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018b. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp. 801–818. https://doi.org/10.1007/978-3-030-01234-2_49.

Cheng, G., Yang, C., Yao, X., Guo, L., Han, J., 2018. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. IEEE Trans. Geosci. Remote Sens. 56 (5), 2811–2821. https://doi.org/10.1109/TGRS.2017.2783902.

Cheng, G., Xie, X., Han, J., Guo, L., Xia, G.S., 2020. Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. IEEE J. Sel. Top. Appl. Earth. Obs. Remote Sens. 13, 3735–3756. https://doi.org/10.1109/JSTARS.2020.3005403.

Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L., 2010. Morphological Attribute Profiles for the Analysis of Very High Resolution Images. IEEE Trans. Geosci. Remote Sens. 48 (10), 3747–3762. https://doi.org/10.1109/TGRS.2010.2048116.

Di, S., Zhang, H., Li, C.G., Mei, X., Prokhorov, D., Ling, H., 2017. Cross-Domain Traffic Scene Understanding: A Dense Correspondence-Based Transfer Learning Approach. IEEE Trans. Intell. Transp. Syst. 19 (3), 745–757. https://doi.org/10.1109/TITS.2017.2702012.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning Hierarchical Features for Scene Labeling. IEEE Trans. Image Process. 35 (8), 1915–1929. https://doi.org/10.1109/TPAMI.2012.231.

Fischler, M.A., Tenenbaum, J.M., Wolf, H.C., 1987. In: Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. Readings in Computer Vision. Elsevier, pp. 741–752. https://doi.org/10.1016/B978-0-08-051581-6.50071-4.

Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R., 2013. Making Bertha see. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 214–221. https://doi.org/10.1109/ICCVW.2013.36.

Fröhlich, B., Bach, E., Walde, I., Hese, S., Schmullius, C., Denzler, J., 2013. Land Cover Classification of Satellite Images using Contextual Information. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 3 (W1) https://doi.org/10.5194/isprsannals-II-3-W1-1-2013.

Fukushima, K., Miyake, S., 1982. Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and Cooperation in Neural Nets. Springer, pp. 267–285. https://doi.org/10.1007/978-3-642-46466-9_18.

Gibril, M.B.A., Shafri, H.Z.M., Shanableh, A., Al-Ruzouq, R., Wayayok, A., Hashim, S.J., 2021. Deep Convolutional Neural Network for Large-Scale Date Palm Tree Mapping from UAV-Based Images. Remote Sens. 13 (14), 2787. https://doi.org/10.3390/rs13142787.

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

Hsieh, M.R., Lin, Y.L., Hsu, W.H., 2017. Drone-Based Object Counting by Spatially Regularized Regional Proposal Network. In: The IEEE International Conference on Computer Vision (ICCV). IEEE. Doi: 10.1109/ICCV.2017.446.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. doi:10.1109/CVPR.2017.243.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: International conference on machine learning. PMLR, pp. 448–456. doi:10.5555/3045118.3045167.

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

Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S., 2021. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 173, 24–49. https://doi.org/10.1016/j.isprsjprs.2020.12.010.

Kingma, D.P., Adam Ba, J., 2015. A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, 2015. ICLR. URL: http://arxiv.org/abs/1412.6980 (accessed: 2022-04-02).

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. https://doi.org/10.1145/3065386.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L. D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput. 1 (4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541.

Leung, T., Malik, J., 2001. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. Int. J. Comput. Vis. 43 (1), 29–44. https://doi.org/10.1023/A:1011126920638.

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

Mohan, R., Nevatia, R., 1989. Using perceptual organization to extract 3D structures. IEEE Trans. Image Process. 11 (11), 1121–1139. https://doi.org/10.1109/34.42852.

Mundhenk, T.N., Konjevod, G., Sakla, W.A., Boakye, K., 2016. A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning. In: European Conference on Computer Vision (ECCV). Springer, pp. 785–800. doi: 10.1007/978-3-319-46487-948.

Nigam, I., Huang, C., Ramanan, D., 2018. Ensemble Knowledge Transfer for Semantic Segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 1499–1508. doi:10.1109/WACV.2018.00168.

NITRDrone Dataset, 2021. URL: https://github.com/drone-vision/NITRDrone-Dataset.

Ortner, M., Descombes, X., Zerubia, J., 2007. Building Outline Extraction from Digital Elevation Models Using Marked Point Processes. Int. J. Comput. Vis. 72 (2), 107–132. https://doi.org/10.1007/s11263-005-5033-7.

Pandey, A., Jain, K., 2021. An intelligent system for crop identification and classification from UAV images using conjugated dense convolutional neural network. Comput. Electr. Agric. 106–543. https://doi.org/10.1016/j.compag.2021.106543.

Pan, X., Shi, J., Luo, P., Wang, X., Tang, X., 2018. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In: Thirty-Second AAAI Conference on Artificial Intelligence. AAAI Press, pp. 7276–7283. doi:10.5555/3504035.3504926.

PyTorch Documents, 2016. URL: https://pytorch.org/docs/stable/index.html (accessed: 2022-03-31).

Rezaei, Y., Mobasheri, M.R., Zoej, M.J.V., Schaepman, M.E., 2012. Endmember Extraction Using a Combination of Orthogonal Projection and Genetic Algorithm. IEEE Geosci. Remote Sens. Lett. 9 (2), 161–165. https://doi.org/10.1109/LGRS.2011.2162936.

Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S., 2016. Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In: European Conference on Computer Vision (ECCV). Springer, pp. 549–565. doi:10.1007/978-3-319-46484-8-33.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Int. J. Comput. Vis. 115 (3), 211–252. https://doi.org/10.1007/s11263-015-0816-y.

Schmid, C., 2001. 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, pp. II–II. doi:10.1109/CVPR.2001.990922.

Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S., 2009. Human detection using partial least squares analysis. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 24–31. doi:10.1109/ICCV.2009.5459205.

Semantic Drone Dataset, 2018. URL: https://www.tugraz.at/index.php?id=22387 (accessed: 2022-03-31).

Srivastava, R.K., Greff, K., Schmidhuber, 2015. J. Highway Networks. URL: http://arxiv.org/abs/1505.00387. arXiv preprint arXiv:150500387.

Stilla, U., 1995. Map-aided structural analysis of aerial images. ISPRS J. Photogramm. Remote Sens. 50 (4), 3–10. https://doi.org/10.1016/0924-2716(95)98232-O.

Tokarczyk, P., Wegner, J.D., Walk, S., Schindler, K., 2015. Features, Color Spaces, and Boosting: New Insights on Semantic Classification of Remote Sensing Images. IEEE Trans. Geosci. Remote Sens. 53 (1), 280–295. https://doi.org/10.1109/TGRS.2014.2321423.

Van Etten, A., 2018. You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery. arXiv preprint arXiv:180509512. URL: https://arxiv.org/abs/1805.09512.

Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., Jawahar, C., 2019. IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 1743–1751. doi:10.1109/WACV.2019.00190.

Viola, P., Jones, M., 2001. 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, pp. I–I. doi:10.1109/CVPR.2001.990517.

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

Wegner, J.D., Montoya-Zegarra, J.A., Schindler, K., 2015. Road networks as collections of minimum cost paths. ISPRS J. Photogramm. Remote Sens. 108, 128–137. https://doi.org/10.1016/j.isprsjprs.2015.07.002.

Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. DOTA: a large-scale dataset for object detection in aerial images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3974–3983. https://doi.org/10.1109/CVPR.2018.00418.

Xie, J., Kiefel, M., Sun, M., Geiger, A., 2016. Semantic instance annotation of street scenes by 3D to 2D label transfer. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3688–3697. https://doi.org/10.1109/CVPR.2016.401.

Zhang, Z., Liu, Q., Wang, Y., 2018. Road Extraction by Deep Residual U-Net. IEEE Geosci. Remote Sens. Lett. 15 (5), 749–753. https://doi.org/10.1109/LGRS.2018.2802944.

Zhou, H., Kong, H., Wei, L., Creighton, D., Nahavandi, S., 2017. On Detecting Road Regions in a Single UAV Image. IEEE Trans. Intell. Transp. Syst. 18 (7), 1713–1722. https://doi.org/10.1109/TITS.2016.2622280.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geosci. Remote Sens. Mag. 5 (4), 8–36. https://doi.org/10.1109/MGRS.2017.2762307.