Journal of Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

UFCN: a fully convolutional neural network for road extraction in RGB imagery acquired by remote sensing from an unmanned aerial vehicle

Ramesh Kestur
Shariq Farooq
Rameen Abdal
Emad Mehraj
Omkar Narasipura
Meenavathi Mudigere

SPIE.

Ramesh Kestur, Shariq Farooq, Rameen Abdal, Emad Mehraj, Omkar Narasipura, Meenavathi Mudigere, "UFCN: a fully convolutional neural network for road extraction in RGB imagery acquired by remote sensing from an unmanned aerial vehicle," *J. Appl. Remote Sens.* **12**(1), 016020 (2018), doi: 10.1117/1.JRS.12.016020.

UFCN: a fully convolutional neural network for road extraction in RGB imagery acquired by remote sensing from an unmanned aerial vehicle

Ramesh Kestur,[a,*] Shariq Farooq,[b] Rameen Abdal,[b] Emad Mehraj,[b] Omkar Narasipura,[c] and Meenavathi Mudigere[a]

[a]Bangalore Institute of Technology (BIT), Department of Electronics and Instrumentation Engineering, Bangalore, India

[b]National Institute of Technology (NIT), Department of Electronics and Communication Engineering, Srinagar, India

[c]Indian Institute of Science (IISc), Department of Aerospace Engineering, Bangalore, India

Abstract. Road extraction in imagery acquired by low altitude remote sensing (LARS) carried out using an unmanned aerial vehicle (UAV) is presented. LARS is carried out using a fixed wing UAV with a high spatial resolution vision spectrum (RGB) camera as the payload. Deep learning techniques, particularly fully convolutional network (FCN), are adopted to extract roads by dense semantic segmentation. The proposed model, UFCN (U-shaped FCN) is an FCN architecture, which is comprised of a stack of convolutions followed by corresponding stack of mirrored deconvolutions with the usage of skip connections in between for preserving the local information. The limited dataset (76 images and their ground truths) is subjected to real-time data augmentation during training phase to increase the size effectively. Classification performance is evaluated using precision, recall, accuracy, F1 score, and brier score parameters. The performance is compared with support vector machine (SVM) classifier, a one-dimensional convolutional neural network (1D-CNN) model, and a standard two-dimensional CNN (2D-CNN). The UFCN model outperforms the SVM, 1D-CNN, and 2D-CNN models across all the performance parameters. Further, the prediction time of the proposed UFCN model is comparable with SVM, 1D-CNN, and 2D-CNN models. [DOI: 10.1117/1.JRS.12.016020]

Keywords: image segmentation; computer vision; remote sensing; deep learning; road extraction; fully convolutional neural networks.

Paper 170726 received Aug. 22, 2017; accepted for publication Jan. 16, 2018; published online Feb. 13, 2018.

1 Introduction

Road extraction is a subject of considerable interest for information extraction from remote sensed images. Road extraction has several applications, such as city planning, traffic management and GPS navigation,[1,2] land cover, and land use analysis.[3] In the recent years, road transport infrastructure has received a significant boost in investments particularly in the developing economies of the Asia Pacific. The cumulative road infrastructure investment until 2025, in Asia Pacific, is estimated at 5 trillion USD.[4] The monitoring and construction operations of roads hold a significant cost. They are currently manual, which is inefficient and not a cost-effective approach. Remote sensing is used for road extraction and automatic detection of roads. Traditional remote sensing-based road extraction is carried out using satellite images, which have high spectral bandwidth such as hyperspectral and multispectral images.[2,5] Current application of satellite remote sensing-based road extraction is in city planning and land cover estimation studies, which are at a macro level, given the large swathes available

*Address all correspondence to: Ramesh Kestur, E-mail: rkestur@gmail.com

1931-3195/2018/$25.00

in satellite remote sensed images of high spectral resolution and relatively low spatial resolution. Classification methods in road extraction using hyperspectral and multispectral imagery primarily exploit the depth of spectral information[6] for road extraction tasks. During the last decade, remote sensing from unmanned aerial vehicles (UAVs) also called low altitude remote sensing (LARS) has shown accelerated usage in the field of agricultural science and research.[7,8] Moreover, other works on UAV imagery in the domains such as species classification,[9] line extraction, contour generation,[10] and shadow classification[11] have proved the potential of UAVs in such domains.

LARS provides an option for microlevel applications of road extraction, such as road construction monitoring. Road construction monitoring is dynamic, which requires acquisition of data over short time cycles and at a higher spatial detail. Another interesting microlevel application is in the determination of actual area available for cultivation in small holder farming. This is particularly relevant in the developing countries where agriculture is carried out in highly fragmented land patches. Fragmentation invariably leads to a decrease in effective cropping area by an increase in the land parcels[12] due to provisioning of land space for pathways and access. Land fragmentation directly impacts agricultural productivity. A one standard deviation increase in the level of fragmentation by Simpson index reduces productivity by 9.8%.[13]

Various traditional methods exist for road detection and extraction. In their work, Wang et al.[1] summarized and compared the methods used for road extraction in recent years. Song and Civco[14] compared performance of support vector machine (SVM) classifier against Gaussian maximum likelihood (GML) classifier for road extraction task and concluded SVM to be better than GML. Fuzzy shell clustering algorithms[15] have been applied for road vectorization. Some knowledge-based methods have also been proposed,[16] which use trackers using nonlinear curves such as parabola to model the trajectory of the road in an image. Neural networks have been used for road detection.[15] However, these methods are not adaptable but are susceptible to noise and discontinuities[1] including misclassification due to shadows and occlusion. The road segments can be occluded by trees, vehicles, buildings, and shadows, which pose a massive challenge. Moreover, inter-spectral distance between the road and surrounding scene, e.g., building rooftops, is sometimes minimal; hence, such scene-pixels can be analogous to road pixels, making some techniques render less accurate results.

Recently developed state-of-the-art convolutional neural networks (CNNs) are outperforming other models in various domains, such as object detection, dense semantic labeling, and image classification.[17,18] AlexNet[18] is one such notable model that was first discovered to popularize the convolutional networks in computer vision by winning the ImageNet ILSVRC challenge in 2012. More recently, fully convolutional network (FCN) proposed by Shelhamer et al.,[19] a type of CNN, has shown promising results in dense semantic segmentation. Particularly, in road segmentation, an FCN-based model has been explored on LIDAR data for road extraction.[20] A special type of FCN architecture called "U-net" is used for biomedical segmentation by Ronneberger et al.[21] U-net is a special type of FCN, where a convolutional network is followed by a deconvolutional network producing the output segmented image. Skip connections are used to reinforce the local information during deconvolution.

In this work, we propose an FCN-based approach for carrying out dense semantic segmentation to extract roads in UAV images using LARS. The proposed architecture is called U-shaped FCN (UFCN), primarily designed as a modification over the U-net model.[21] The UFCN is trained on a custom dataset of 76 LARS images and ground truths along with real-time data augmentation.

From the review of literature by the authors, it is observed that there is no evidence of extraction of roads in UAV imagery using FCN, more specifically the U-net-based architecture.

Four representative RGB images from test set are visualized for demonstrating extraction of roads. The performance of road extraction for the entire test dataset (33 images) using performance parameters is discussed Sec. 5. The road extraction performance is compared with SVM, 1D-CNN, and 2D-CNN approaches. Successful extraction by the UFCN model demonstrates the potential of using UAV imagery for road construction operations monitoring.

Rest of this article is organized as follows: the UAV image acquisition using LARS and description of the custom-built UAV deployed with its characteristics and ratings are discussed in Sec. 2. CNN and fully CNN are discussed in Secs. 3, 4, and 5 discuss the methodology,

performance evaluation, and comparisons with SVM, 1D-CNN, and 2D-CNN models. Finally, the conclusion is discussed in Sec. 6.

2 Unmanned Aerial Vehicles for Road Extraction

This section provides a review of the current work in road extraction using UAV imagery and a description of the UAV used for image acquisition in this work. Zhao et al.[22] presented an automatic approach to detect generic roads from a single UAV image. The proposed method uses stroke width transformation to identify where the roads probably are. Other features such as color and width are used to classify images using a convex contour segmentation model. Zhao et al.[23] used a graph cut-based approach in detection of road regions in UAV images. Kim[24] carried out road detection by learning from one example. The approach learned road structure from a single image and applied it to detect and localize a road from a new image. The road structure was defined as a structure having a strong vertical correlation.

From authors' review of literature, it can be observed that the road detection approaches in UAV imagery use the geometry and shape of road structure for detection. The proposed approach does not assume any shape and geometry features.

UAV imagery of road structures is acquired by LARS carried out using a UAV. UAVs used in LARs are either rotary wings or fixed wings. Rotary wings can take off vertically and they have the capacity to hover and carry out precision maneuvering. However, they have shorter flight duration and lower speeds, which require additional flights to survey a given road region thereby leading to an increase in costs and time. Fixed wing can be launched from a runway or hand launched. They can cover a larger area in a shorter time since they fly at faster speeds as compared with rotary wings making them more suitable for road inspection applications. Imaging sensors are important constituents of the payload. LIDARS are used for remote sensing; however, they are bulky and costly,[25] hence imaging sensors are constrained to vision spectrum (also popularly known as RGB) cameras due to their lower weight, size, and cost. In this work, UAV imagery is acquired by LARS using a fixed wing aircraft. The fixed wing UAV is a custom-built electrically powered UAV with a takeoff weight of 2.2 kg and hover flight time of 20 min. The payload is a GoPro 2 (GoPro HD Hero 2, San Mateo, California) camera.

3 Convolutional Neural Network and Fully Convolutional Network

Artificial neural network (ANN)[26] is a system of interconnected neurons. ANNs are generally used to model complicated functions for various tasks, such as classification and regression. The most basic ANN model is the multilayer perceptron (MLP), which is composed of at least three layers of one-dimensional (1-D) nodes, an input layer, one or more hidden layers, and an output layer.

A CNN[18] is a type of ANN in which, instead of a 1-D set of neurons constituting a layer in MLP, a convolutional block is used. The convolutional block is a series of convolutional layers. The convolution operation is the inner product of trainable filter (commonly a 3×3 square matrix) and the input while sliding and summing the entries at overlapping regions of the input. Nonlinear activation functions, such as rectified linear unit (ReLU) and sigmoid, are applied to the convolution operation outputs to produce activation maps. Pooling layers are used between convolutional layers, reducing the spatial dimensionality and complexity such that global features are extracted along with local features. After the convolutional layers, fully connected dense layers are appended to produce a vector of desired dimension, representing the output.

FCN is a modified CNN, where fully connected layers at the end are replaced by only convolutional layers™ introducing scale invariance and providing the ability to accept inputs of varying sizes to the network.[19] Also, CNNs with a fully connected output layer may end up prior learning of the locations in a scene,[27] resulting in loss of spatial-invariance inherent to convolutional layers. The loss of spatial invariance poses an issue in semantic segmentation.[19] Figure 1 shows the overview of the proposed UFCN model, which is a type of FCN. In this architecture, the reduction in spatial resolution due to the convolution and

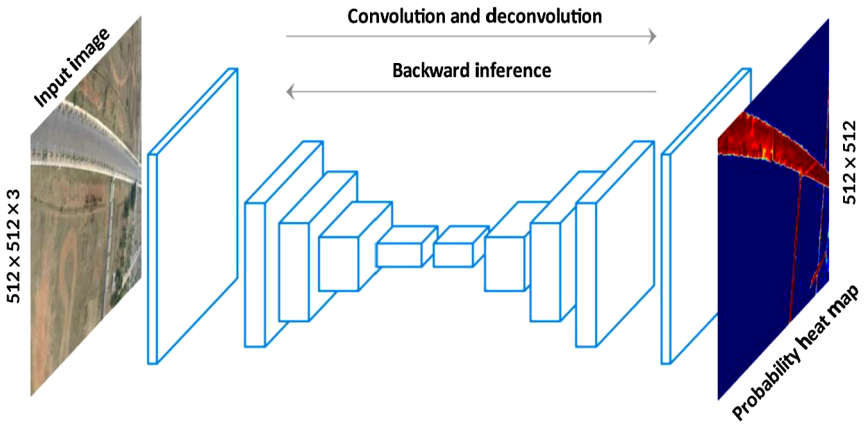

Fig. 1 Overview of the structure of proposed UFCN model.

pooling operations is restored using deconvolution. The complete working of the model is described in methodology section (Sec. 4).

4 Methodology

4.1 *Data Acquisition and Preparation*

The UAV was flown over an area that was covered with roads. A video was captured as the UAV flew over the road area at varying altitudes and angles. The aerial video was acquired for 4 min at 50 frames per second. An image extractor was used to extract RGB images from the videos. A total of 109 images acquired at various angles and altitudes were chosen as the dataset. About 70% of data (=76 images) were chosen as a training set and remaining 30% as a test set. Four representative images out of test set were chosen for detailed analysis and visualization. The original size of images is of high spatial dimension 1920×1080. Due to the computational constraints, the size is resized to 512×512. The ground truth of the aerial images was created manually using an image editor tool. The ground truth has two class labels, road and nonroad (or background) regions. Although preparing a perfect ground truth with no mislabeled pixel is a challenging task, we have made exhaustive attempts to ensure minimum errors in ground truth. The task of preparing the ground truths of all the 76 images for training and the test sets was given to multiple persons multiple times, and the best combination was chosen after evaluation on the basis of comprehensive visual judgment.

4.2 *UFCN Model Architecture*

Figure 2 shows the detailed architecture of the proposed UFCN model. The UFCN model is fully convolutional and thus can take input of any size.[19] For our dataset, the input image size is

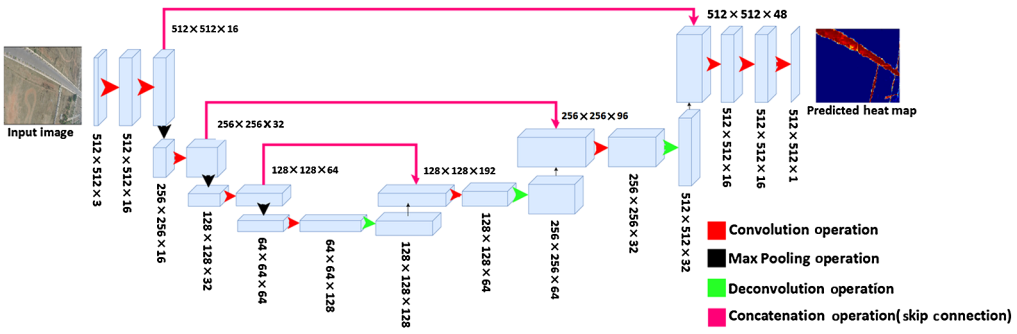

Fig. 2 The detailed architecture of the UFCN model.

$512 \times 512 \times 3$. The model takes $N \times N \times 3$ RGB image as input (here $N = 512$) and outputs the predicted segmentation map of the same size, where the input image is segmented into two regions (classes)—road and background.

A 3×3 kernel size is used across all convolutional layers with a stride of 1×1 and a padding type of "same." Padding type "same" ensures that the output feature map of the convolutional layer is of the same spatial size as that of input after a convolution operation. In every convolutional layer, the convolutions are followed by element-wise activation using ReLU denoted as $f(\cdot)$, which is given by

$$f(x) = \max(0, x). \tag{1}$$

The stack of convolutions is followed by a stack of mirrored deconvolutions resulting in a U-shaped network. Skip connections are used between the stacks. A skip connection is the concatenation operation of the filters in convolution layer with the corresponding mirrored deconvolution layer. Skip connections are used to reinforce the information of the primary layers for reconstruction during upsampling in the deconvolution layers. In this work, "deconvolution" refers to the operation of "transposed convolution" (not the mathematical inverse of the convolution) or "learned upsampling," where pixel values are not simply interpolated to obtain the image of higher spatial size but learned during training. Deconvolution can be seen as a convolution with the fractional stride.[19] The fractional stride here is a half stride (stride $= 1/2$).

As we go deeper into a convolutional network, the local information is lost due to successive convolutions, and the global information is attained.[19] This is because of the pooling operations, which follow the nonlinear activation and convolution operations (see Sec. 3). "Local information" here refers to the information regarding the locations/pixel positions of the predicted class. The convolutional network is exceptionally good when it comes to identifying "what" is in the given image, but due to local-information loss, it cannot accurately deduce "where" the predicted class is present.[19] In addition, depending upon the stride size, the pooling and convolution operations down sample the input image. Thus, successive convolutional layers (including pooling) may result in a very low spatial resolution output.

Shelhamer et al.[19] used a single skip layer to add all the skip information at the end of the network. However in the proposed UFCN architecture, a step-wise approach is used. Skip connections taken from a convolutional layer of size, say, $m \times m \times r$, is connected to the deconvolutional layer of corresponding size, $m \times m \times r'$ (see Fig. 2). Such a connection structure also avoids any information loss. U-net architecture proposed by Ronneberger et al.[21] uses cropping operations in skip layers, leading to inefficiency. The proposed UFCN architecture is designed so as to avoid any cropping and, hence, skip connections in UFCN are lossless. Further, an output image matrix, with the same spatial resolution (unlike the U-net model) of the input image matrix, is obtained with multifold feature maps as compared with the input image. A 1×1 convolutional layer with a sigmoid activation is appended at the end to reduce the number of feature maps to 1, resulting in the output of size $N \times N \times 1$, which is represented with just a $N \times N$ 2-D matrix with a value between 0 and 1 at every pixel position.

4.3 *Problem Formulation*

Let $\mathbf{X} = \mathbb{R}^{N \times N \times 3}$ be the space of $N \times N \times 3$ dimensional rank-3 tensors (or simply RGB color images), $\mathbf{Y} = \mathbb{R}^{N \times N}$ be the space of $N \times N$ dimensional rank-2 tensors (also called matrices), and $W_k \subset \mathbb{R}^{k \times k}$ be the set of learnable 2-D filters of size k (described in Sec. 4.2). In this work, we have chosen a fixed size for filter ($k = 3$) throughout the model. The proposed UFCN model can be thought of as a transformation $U_{W_k} : \mathbf{X} \rightarrow \mathbf{Y}$ with W_k as parameter for defining U. The learning objective is to choose W_k to minimize a loss function. We leave the intricate mathematics of the internal machinery of transformation U and the theory of learning of conditional distributions to the extensive literature available. Only a minimal mathematical formulation of the problem is presented. However, a visualization of the transformation at multiple convolutional levels is provided (Fig. 3).

As the spectral depth of these maps is too large to be visualized, element-wise-average feature map of each level in the UFCN model is presented. We see that earlier convolutional layers tend

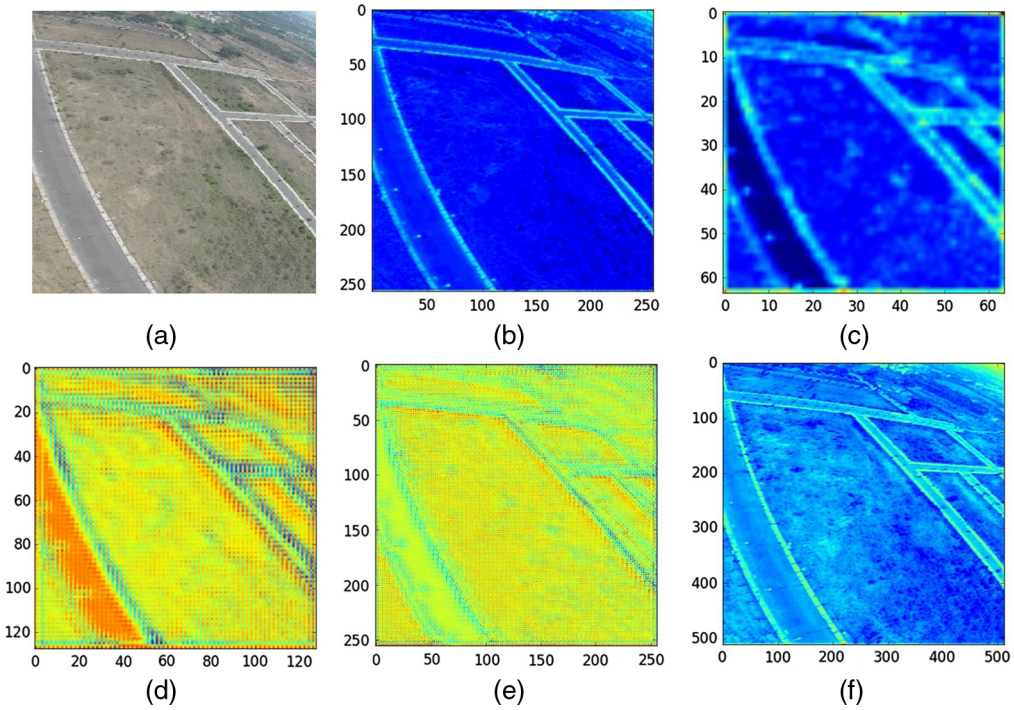

Fig. 3 Feature maps (activation maps) extracted by corresponding filters (w_k) in different layers of the trained UFCN model: (a) input image ($512 \times 512 \times 3$), (b) averaged feature map at level two of the convolutional network (originally: $256 \times 256 \times 32$), (c) averaged feature map at level four of the convolutional network (originally: $64 \times 64 \times 128$), (d) averaged feature map at level first of the deconvolutional network after merging operation (originally: $128 \times 128 \times 192$), (e) averaged feature map at level two of the deconvolutional network after merging operation (originally: $256 \times 256 \times 96$), and (f) averaged feature map at level three of the deconvolutional network after merging operation (originally: $512 \times 512 \times 48$).

to extract edges [see Fig. 3(b)] and deeper convolutional layers are of low resolution [see Fig. 3(c)]. An increase in the resolution via deconvolutional layers and skip connections, discussed in Sec. 4.2, can be observed as we go from Figs. 3(c) to 3(d) through Fig. 3(f).

We interpret the output $P \in \mathbf{Y}$ of the transformation U as the "prediction probability matrix" $P = [p_{ij}]$, where p_{ij} is the probability of the pixel at position (i, j) belonging to road. To produce this matrix, the last activation used in the proposed UFCN model is an element-wise sigmoid activation, which is $\sigma(\cdot)$ given by

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

and hence every element p_{ij} in the output lies between 0 and 1.

The corresponding probabilities for the background can be obtained as $b_{ij} = 1 - p_{ij}$. We, therefore, formulate the subsequent discussion in terms of P matrix only. The "prediction probability matrix" P is thresholded to obtain the final output, involving assigning of the pixel to road or the background.

Let $M \in \mathbf{X}$ be the input image, $T : \mathbf{Y} \to \mathbf{Y}$ represents the thresholding operator and $S \in \mathbf{Y}$ be the matrix obtained after thresholding then

$$P = U_{W_k}(M), \tag{3}$$

$$S = T(P), \tag{4}$$

such that $s_{ij} = \text{nint}(p_{ij})$, where $\text{nint}(\cdot)$ is the nearest-integer function, which in this case, maps the probabilities to integers 0 or 1, whichever is "nearer." The thresholding operation maps the

probabilities to the maximum probability class since the operation nint$(\cdot)$ results in matrix S of the form, which is given by

$$s_{ij} = \begin{cases} 1 & \text{if } p_{ij} > b_{ij} \\ 0 & \text{if } p_{ij} \leq b_{ij} \end{cases}. \tag{5}$$

The final output after thresholding, S, is thus a binary matrix in which "1" occurs at the pixel positions where the road is predicted and 0 otherwise. This thresholded matrix S is also termed as "threshold image" from now on.

The prediction probability matrix P output from the model is visualized as a heat map, which is an RGB image where the likelihood of a pixel belonging to the road (i.e., p_{ij}) is indicated by a color on a blue-to-red scale. Some output heat maps are shown in Fig. 7 (see Sec. 5).

4.4 *Objective and Learning*

The objective is to choose W_k such that given any input image $M \in \mathbf{X}$ should produce a segmentation map $T[U_{W_k}(M)] \in \mathbf{Y}$, in which road and background are differentiated with "1" at road pixels and "0" otherwise. A subset of images is selected from a set of aerial images obtained from the UAV described in Sec. 2. The subset is chosen to include a variety of aerial images of roads taken at various perspectives and altitudes to thoroughly train the model. The subset is denoted as $\mathcal{M}$ whose cardinality is $n(\mathcal{M}) = 76$. The model is trained via supervised learning approach with segmentation maps (ground truth or label) corresponding to all the images in $\mathcal{M}$. Thus, a ground truth $G \in \mathbf{Y}$ is a 2-D binary matrix $G = [g_{ij}]$ representing the actual map of the aerial image, which is given by

$$g_{ij} = \begin{cases} 1 & \text{if pixel at (i, j) belongs to road} \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

The set of such binary ground truths corresponding to aerial images in $\mathcal{M}$ is represented as $\mathcal{G}$, where $\mathcal{G} \subset \mathbf{Y}$. The set of pairs $\mathcal{D} = \{(M, G) : M \in \mathcal{M}, G \in \mathcal{G}\}$ is referred to as dataset. Our original dataset is not large enough (only 76 aerial images along with ground truths); therefore, real-time data augmentation is done while training. Data augmentation is a technique of generating more images (and corresponding ground truths) by doing random affine transformations on images such as rotation by an arbitrary angle, vertical flip, horizontal flip, and so on. Real-time data augmentation implies that these affine transformations are done while training (and not in advance) to save memory and time. This implies that new pairs are generated and included in $\mathcal{D}$ in real time and are immediately used as training examples. Hence, final cardinality of $\mathcal{D}$ depends on the duration of training phase. Ronneberger et al.[21] showed that U-structured networks can actually be trained end to end on very few examples, making strong use of data augmentation more efficiently.

Let $\mathcal{P} = \{P : P = U_{W_k}(M) \; \forall \; M \in \mathcal{M}\}$. We define loss function or cost function $\mathcal{L} : \mathcal{P} \times \mathcal{G} \rightarrow \mathbb{R}$ to be the binary cross entropy (BCE) evaluated between the ground truth (G) and the prediction probability matrix (P). The cross entropy gives the measure of similarity between the two probability distributions, here probability distributions of g_{ij} and p_{ij} in particular. The equation for this loss function is given as

$$\mathcal{L} = -\sum_i \sum_j [g_{ij} \log(p_{ij}) + (1 - g_{ij}). \log(1 - p_{ij})]. \tag{7}$$

In Eq. (7), g_{ij} are predefined and p_{ij} are obtained from Eq. (3). It should be noted that cross entropies corresponding to every pixel (which follow its own distribution $p_{ij} \sim F_{ij}$) are added to obtain the final loss. The minimization of loss function is done using "adam" optimizer,[28] which is a method for stochastic optimization. The required gradients are obtained using the back-propagation algorithm.[29]

The UFCN model processes the whole of the test image to predict every single pixel such that the neighborhood information of a pixel is incorporated during prediction. The pixel-wise prediction machine-learning models (e.g., SVM) and 1D-CNN in particular, do not take

neighborhood information in consideration. This means that probability distributions F_{ij} such that $p_{ij} \sim F_{ij}$ are estimated independently of each other in the traditional models, where contextual information is not utilized. Some works[30,31] take into account the context information in one way or the other, e.g., including texture features using methods such as wavelet transforms, gray-level co-occurrence matrix (GLCM), or window-wise prediction but are inefficient.[19,27] Practically, neighboring distributions do affect each other and estimation of F_{ij} may depend on a contextual window in the image. UFCN model estimates each F_{ij} taking whole of the image M as the contextual window. FCNs exploit the contextual information in a more effective way than traditional approaches.[19]

4.5 *Experiments*

For the purpose of comparison, we employ three different models, namely SVM, 1D-CNN, and 2D-CNN, for pixel-wise classification and compare the results with the proposed UFCN model. Section 5 later shows how the results of a classical model (here SVM), 1D-CNN, and 2D-CNN models are comparable.

To validate the importance of skip connections and learned upsampling, we use a standard 2D-CNN model, which is devoid of skip connections and deconvolutional layers, and is similar in architecture to UFCN upto final max pooling operation (see Fig. 2). Feature maps of the final Max Pooling operation are upsampled to the size of input using repetitive upsampling, which is followed by a single-node network-in-network layer[32] (or mlpconv layer) with sigmoid activation, producing a 2-D matrix similar to P [see Eq. (3)]. 2D-CNN is trained on the same data used for UFCN taking "binary crossentropy" as the loss function.

SVMs have been traditionally explored for the tasks of image classification[33] and road extraction.[1] We also evaluate the SVM model to compare the performance. A pixel-wise SVM classifier is designed and trained across 8192 pixels chosen from the adequate regions in the training set. A nonlinear kernel, Gaussian radial basis function is chosen for the classification task. The parameters of C (cost of classification) and γ (free parameter of the Gaussian radial basis function) were empirically fine tuned to values 1000 and 0.001, respectively.

1D-CNNs have been used as classifiers and feature extractors.[34,35] 1D-CNN-based approaches have also been used in hyperspectral image classifications.[6] The essence of "one dimensional" in such networks can be visualized as subsequent convolution operations on layers of vectors making the feature maps strictly 1-D.[36] The input to this model is a vector of red, green, blue (RGB) and hue, saturation, and value values of pixels. Throughout the model, stride of 1, kernel size of 1×3, and 64 filters are used in each convolutional layer. To involve every possible combination into convolution, RGB values are again appended at the end of vector. A fully connected dense layer is appended and is followed by a single node. ReLU activation is used for convolutional layers and fully connected layer, sigmoid is used for output node. Also, dropout of 0.2 is employed in the fully connected layer to avoid overfitting. The weights of filters are optimized via backpropagation, minimizing the "binary cross-entropy" function. For an

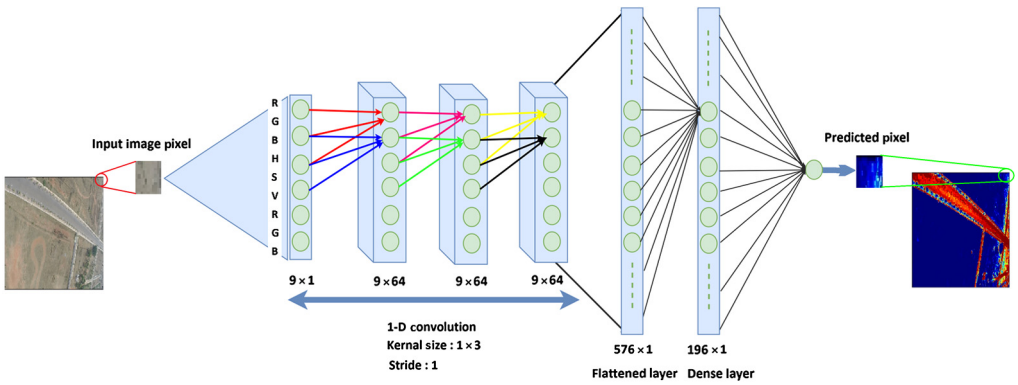

Fig. 4 Architecture of the 1D-CNN model.

equitable comparison, the training data are prepared from the same regions on which SVM classifier is trained. Figure 4 shows the detailed architecture of 1D-CNN model.

4.6 *Performance Evaluation*

In this work, we use standard measures for evaluating the semantic segmentation task, namely precision, recall, F1-score, and accuracy.[19,37] Brier score[38] is also used to evaluate predicted heat maps. Let **D** be the set of pixels on which metrics are to be calculated. We evaluate confusion matrix[39] (2×2), for the pixel-wise binary classification, where n_{ij} is the i, j entry of the matrix. Let n_{00}, n_{01}, n_{10}, and n_{11} denote the number of true negative, false positive, false negative, and true positives, respectively. Let N be the total number of pixels in **D**, T_i be the actual label of the i'th pixel (1 for road and 0 otherwise), and p_i be the predicted probability of road. We compute the following:

- Precision (P): $n_{11}/(n_{11} + n_{01})$
- Recall (R): $n_{11}/(n_{11} + n_{10})$
- F1 score: $2 \cdot P \cdot R/(P + R)$
- Accuracy (A): $(n_{00} + n_{11})/N$
- Brier score (BS): $\frac{1}{N}\sum_{i=1}^{N}(p_i - T_i)^2$

The model was implemented in the Keras framework (python) and was trained for 11 hours on GeForce GTX 780 TI GPU.

5 Results

In this section, we demonstrate the performance of the UFCN model and evaluate the results against SVM, 1D-CNN, and 2D-CNN models. Figure 5 visualizes the performance and

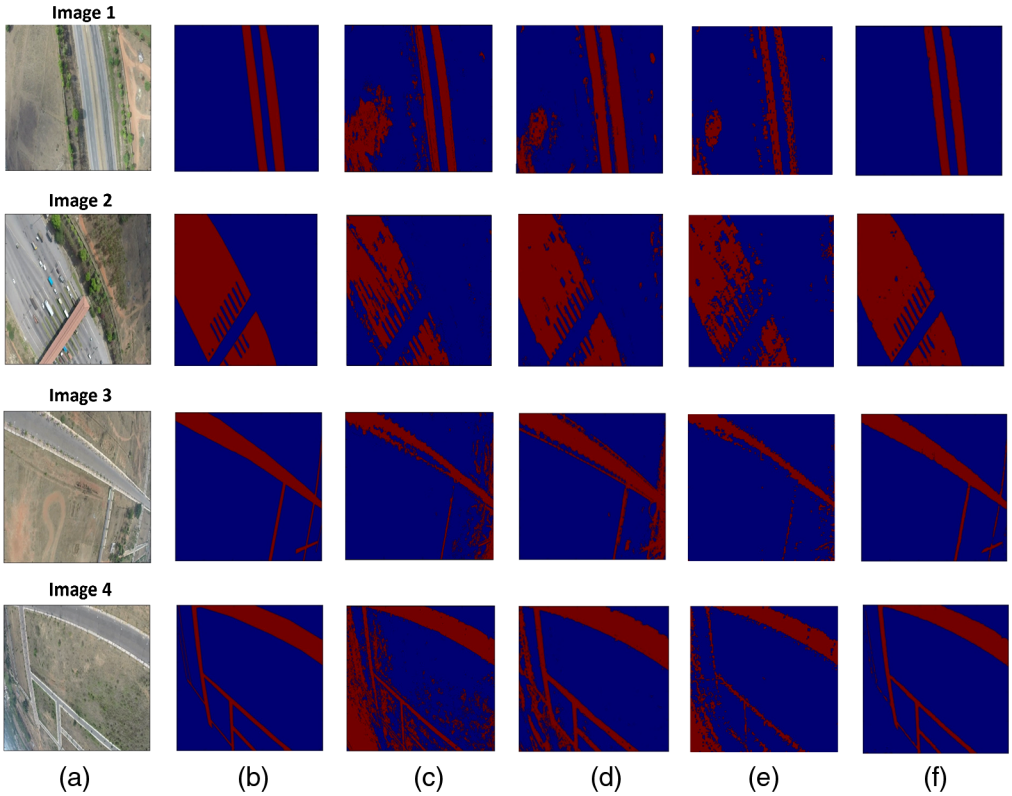

Fig. 5 Visual interpretation of the performance of models across four representative images in order: (a) input image, (b) ground truth, (c) SVM model threshold image, (d) 1D-CNN model threshold image, (e) 2D-CNN model threshold image, and (f) UFCN model threshold image.

Table 1 Detailed performance evaluation across four representative images.

Test data (**D**)	Model	Precision	Recall	F1 score	Accuracy	BS	Prediction time (s)
Image 1	UFCN	0.988	0.965	0.976	0.994	0.0046	1.90
	SVM	0.468	0.862	0.607	0.854	0.1325	7.05
	1D-CNN	0.567	0.988	0.720	0.900	0.0818	42.02
	2D-CNN	0.668	0.722	0.694	0.917	0.0565	1.05
Image 2	UFCN	0.989	0.977	0.983	0.987	0.0100	1.92
	SVM	0.979	0.737	0.841	0.894	0.1182	7.44
	1D-CNN	0.933	0.912	0.923	0.942	0.0488	43.10
	2D-CNN	0.938	0.734	0.824	0.880	0.0984	1.11
Image 3	UFCN	0.978	0.953	0.965	0.990	0.0073	1.99
	SVM	0.671	0.732	0.701	0.912	0.0783	8.45
	1D-CNN	0.633	0.960	0.763	0.916	0.0649	45.05
	2D-CNN	0.866	0.582	0.697	0.929	0.0530	1.07
Image 4	UFCN	0.973	0.961	0.967	0.989	0.0082	2.01
	SVM	0.411	0.904	0.566	0.768	0.1948	7.79
	1D-CNN	0.582	0.993	0.734	0.880	0.1021	45.34
	2D-CNN	0.614	0.727	0.666	0.879	0.0958	1.13

comparisons of the proposed UFCN method. We use "threshold image" of the trained SVM, 1D-CNN, 2D-CNN, and UFCN models across the four images to compare the performance of each model. The test images in Fig. 5 chosen for demonstration are taken by the UAV camera at different angles, terrains, and during different times of the day to evaluate the models more explicitly. Some of the factors that cause misclassification in the task of road extraction such as occlusion (see the blunt edges captured in Figs. 5 "Image 3" and 5 "Image 4" due the camera's curvature) and similar background texture (see Figs. 5 "Image 1" and 5 "Image 2") have been incorporated in the input images with an attempt to review the robustness of the models. Image 1 is a road stretch with two lanes separated by a median. Figure 5 shows Image 2, a toll junction where the road expands to multiple lanes without medians. Image 3 and Image 4 are roads in a residential locality with edge markers and multiple intersections.

Table 1 tabulates the performance parameters and prediction times evaluated on the correspondingly obtained results of SVM, 1D-CNN, 2D-CNN, and UFCN models across the four images shown in Fig. 5. The threshold image predicted by the three models is evaluated against the ground truths to calculate precision, recall, F1 score, and accuracy. The heat maps of the predicted images are evaluated against the ground truths to calculate the BSs. Going by the results of Fig. 5 and evaluation of Table 1, clearly by visual interpretation and taking any of the performance parameter in consideration, the UFCN model outperforms the SVM, 1D-CNN, and 2D-CNN models. Precision performance of the SVM model dips to as low as 0.411 and 1D-CNN model performance on accuracy is in the range of 0.567 to 0.933, producing much variance. Similar trends can be seen with 2D-CNN model across all the performance parameters, which are attributed to simple upsampling causing pixelation (discussed in Sec. 4.5). Comparatively, performance of UFCN model is consistent with much less variance across the four images. The consistent behavior is evident from the precision and recall of UFCN model being in the range of 0.973 to 0.988 and 0.961 to 0.971, respectively.

As is also evident from Table 1, UFCN and 2D-CNN models are faster and stable in prediction than 1D-CNN and SVM models. The faster output rate of 2D-CNN model is attributed to

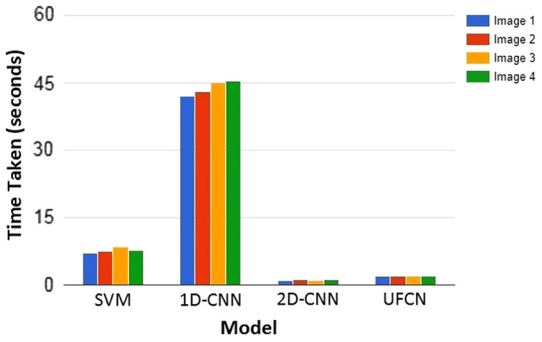

Fig. 6 Visualization of prediction time.

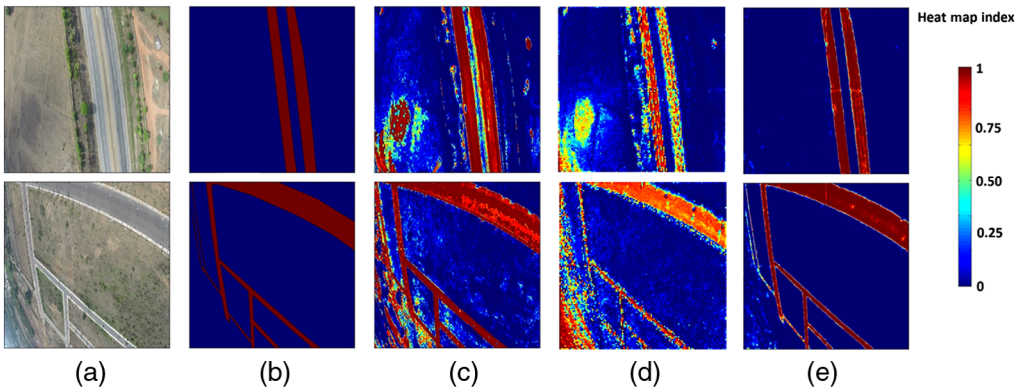

Fig. 7 Heat maps of 1D-CNN, 2D-CNN, and UFCN models for performance comparison: (a) input image, (b) ground truth, (c) output heat map of 1D-CNN model, (d) output heat map of 2D-CNN model, and (e) output heat map of UFCN model.

less complex architecture than the UFCN model. The prediction techniques used in 1D-CNN and SVM models are based on conventional machine learning following a pixel-by-pixel prediction strategy. The UFCN and 2D-CNN models predict the whole image (a matrix of pixels) at once. The simultaneous labeling of all the pixels is evident from the architecture of the model as discussed in Secs. 4 and 4.5. Here, SVM model is faster than 1D-CNN model with average prediction times of 7.68 and 43.88 s, respectively. This is mainly due to comparatively much computationally intensive architecture of the 1D-CNN model. In Fig. 6, the variation in prediction times of 1D-CNN and SVM models is noticeable and hence is inconsistent. The UFCN and 2D-CNN models show consistency across all images.

In Fig. 7, heat maps of the UFCN, 1D-CNN, and 2D-CNN models are visually compared. UFCN model shows a higher confidence in prediction and predicts the whole background with much less probability of it belonging to road. The heat maps of 1D-CNN and 2D-CNN models are highly grained, depicting a low confidence behavior before the thresholding is performed. This behavior is visualized by the results compared with heat map index provided. Most of the pixels predicted by UFCN model fall in the upper range of 0.70 to 1.00 for the road and lower range of 0.00 to 0.30 for the background. 1D-CNN and 2D-CNN models clearly show dispersed range of values across the index, which is undesirable.

To further examine the consistency of models across diverse data, we evaluate the entire test set (see Table 2) to verify the overall performance of the proposed UFCN model. The metrics are calculated on the set **D** containing the pixels of all the 33 test images. For visualization, precision–recall scatter plot and box plots of the above-mentioned performance parameters are shown in Figs. 8 and 9, respectively. In Fig. 8, it is clear that the UFCN model is highly consistent than its SVM, 1D-CNN, and 2D-CNN counterparts. The cluster of points plotted by the UFCN model across all test data is closer to coordinates (1,1), which is desirable. SVM, 1D-CNN, and 2D-CNN models show the plots scattered over the space.

Table 2 Performance evaluation across whole test set.

Test data (**D**)	Model	Precision	Recall	F1 score	Accuracy	BS
33 images	UFCN	0.925	0.868	0.896	0.952	0.0378
	SVM	0.805	0.759	0.780	0.899	0.0856
	1D-CNN	0.720	0.938	0.814	0.898	0.0851
	2D-CNN	0.869	0.748	0.804	0.914	0.0665

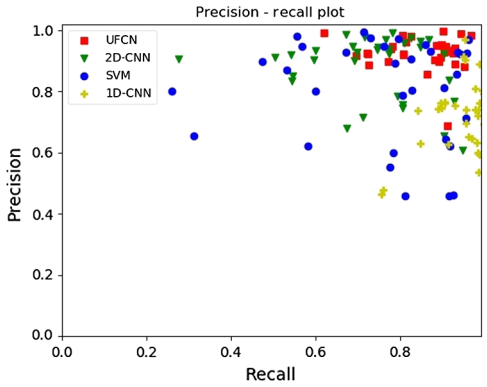

Fig. 8 Precision versus recall plot for performance evaluation of UFCN, SVM, 1D-CNN, and 2D-CNN models.

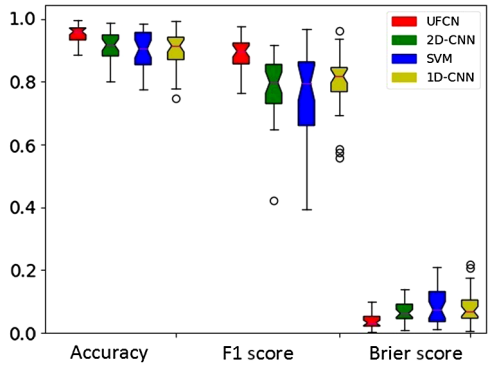

Fig. 9 Performance visualization of the models using box plots of F1 score, accuracy, and BS.

Box plots of the other three performance parameters (F1 score, accuracy, and BS) are presented in Fig. 9. These visualizations represent the performance data of models in Table 2. It is quite evident that the UFCN model shows far less variation than SVM, 1D-CNN, and 2D-CNN models across all the parameters.

6 Conclusion

The task of the extraction of roads in RGB images acquired through LARS carried out by a UAV was successfully accomplished. UFCN, a U-shaped FCN was used to carry out dense semantic segmentation for the road extraction. A set of 76 LARS images were used along with real-time data augmentation to train the UFCN model. The results compared with SVM, 1D-CNN, and 2D-CNN illustrates that UFCN outperforming all the models. Also, the prediction time of the UFCN model is fast and stable compared with SVM and 1D-CNN models, making it an

excellent choice for near real-time data acquisition operations. This work also demonstrates the potential of using LARS and the associated deep learning architecture for use in the microlevel remote sensing applications, such as road construction monitoring and determining effective crop area in fragmented land holdings. The results are promising; however, the work can be extended to apply the proposed UFCN architecture for road extraction in mosaic UAV images covering a large area.

Disclosures

There is no conflict of interest.

Acknowledgments

We express our gratitude to Gautham Anand and Chandrashekar from the UAV Lab at the Aerospace Engineering Department of IISc Bangalore for the UAV platform and the UAV flight campaign for data acquisition.

References

1. W. Wang et al., "A review of road extraction from remote sensing images," *J. Traffic Transp. Eng.* **3**(3), 271–282 (2016).
2. J. Wang et al., "A new approach to urban road extraction using high-resolution aerial image," *ISPRS Int. J. Geo-Inf.* **5**(7), 114 (2016).
3. M. Li et al., "Region-based urban road extraction from VHR satellite images using binary partition tree," *Int. J. Appl. Earth Obs. Geoinf.* **44**, 217–225 (2016).
4. S. Julian, W. Marie-Claude, and H. Graeme, "Assessing the global transport infrastructure market: outlook to 2025," PwC, with research by Oxford Economics (2015), https://www.pwc.com/gx/en/transportation-logistics/pdf/assessing-global-transport-infrastructure-market.pdf (18 February 2017).
5. F. Hu et al., "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery in surveying, mapping and remote sensing," *Remote Sens.* **7**, 14680–14707 (2015).
6. W. Li et al., "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.* **55**(2), 844–853 (2017).
7. J. Senthilnath et al., "Detection of tomatoes using spectral-spatial methods in remotely sensed RGB images captured by UAV," *Biosyst. Eng.* **146**, 16–32 (2016).
8. M. M. Saberioon et al., "Assessment of rice leaf chlorophyll content using visible bands at different growth stages at both the leaf and canopy scale," *Int. J. Appl. Earth Obs. Geoinf.* **2**, 277–281 (2008).
9. B. Lu and Y. He, "Species classification using Unmanned Aerial Vehicle (UAV)-acquired high spatial resolution imagery in a heterogeneous grassland," *ISPRS J. Photogramm. Remote Sens.* **128**, 73–85 (2017).
10. S. Crommelinck et al., "Review of automatic feature extraction from high-resolution optical sensor data for UAV-based cadastral mapping," *Remote Sens.* **8**(8), 689 (2016).
11. A. S. Milas et al., "Different colours of shadows: classification of UAV images," *Int. J. Remote Sens.* **38**(8–10), 3084–3100 (2017).
12. "All India report on input survey 2011–2012," Department of Agriculture, Cooperation and Farmers Welfare Ministry of Agriculture and Farmers Welfare, New Delhi (2016), http://agcensus.nic.in/document/is2011/reports/all_india_report_2011_12.pdf (25 March 2017).
13. K. Deininger et al., "Does land fragmentation increase the cost of cultivation? Evidence from India," *J. Dev. Stud.* **53**(1), 82–98 (2017).
14. M. Song and D. Civco, "Road extraction using SVM and image segmentation," *Photogramm. Eng. Remote Sens.* **70**(12), 1365–1371 (2004).

15. M. Mokhtarzade, M. J. V. Zoej, and H. Ebadi, "Automatic road extraction from high resolution satellite images using neural networks, texture analysis, fuzzy clustering and genetic algorithms," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **37**, 549–556 (2008).
16. J. Shen et al., "Knowledge-based road extraction from high resolution remotely sensed imagery," in *Congress on Image and Signal Processing (CISP)*, Vol. 4, pp. 608–612 (2008).
17. S. Tang and Y. Yuan, "Object detection based on convolutional neural network," Stanford University (2015), http://cs231n.stanford.edu/reports/2015/pdfs/CS231n_final_writeup_sjtang.pdf.
18. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. of the 25th Int. Conf. on Neural Information Processing Systems*, Vol. 1, pp. 1097–1105, Curran Associates Inc. (2012).
19. E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017).
20. L. Caltagirone et al., "Fast LIDAR-based road detection using fully convolutional neural networks," in *IEEE Intelligent Vehicles Symp. (IV)*, pp. 1019–1024 (2017).
21. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241 (2015).
22. H. Zhou et al., "On detecting road regions in a single UAV image," *IEEE Trans. Intell. Transp. Syst.* **18**(7), 1713–1722 (2017).
23. H. Zhou et al., "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Trans. Intell. Transp. Syst.* **16**(1), 297–309 (2015).
24. Z. Kim, "Realtime road detection by learning from one example," in *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTIONS)*, Vol. 1, pp. 455–460, IEEE (2005).
25. M. A. Lefsky et al., "Lidar remote sensing for ecosystem studies: lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists," *BioScience* **52**(1), 19–30 (2002).
26. T.-H. Kim, "Pattern recognition using artificial neural network: a review," in *Int. Conf. on Information Security and Assurance*, pp. 138–148, Springer (2010).
27. E. Maggiori et al., "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.* **55**(2), 645–657 (2017).
28. J. Ngiam et al., "On optimization methods for deep learning," in *Proc. of the 28th Int. Conf. on Machine Learning (ICML)*, pp. 265–272 (2011).
29. R. Hecht-Nielsen et al., "Theory of the backpropagation neural network," *Neural Networks* **1**(Suppl. 1), 445–448 (1988).
30. V. Mnih and G. E. Hinton, *Learning to Detect Roads in High-Resolution Aerial Images*, pp. 210–223, Springer, Berlin, Heidelberg (2010).
31. Y. Hu, C. X. Zhao, and H. N. Wang, "Directional analysis of texture images using gray level co-occurrence matrix," in *Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA)*, Vol. 2, pp. 277–281, IEEE (2008).
32. Y. Pang et al., "Convolution in convolution for network in network," *IEEE Trans. Neural Networks Learn. Syst.* **PP**(99), 1–11 (2017).
33. O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Networks* **10**(5), 1055–1064 (1999).
34. O. Abdel-Hamid et al., "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.* **22**, 1533–1545 (2014).
35. D. Zeng et al., "Relation classification via convolutional deep neural network," in *Proc. of COLING, the 25th Int. Conf. on Computational Linguistics: Technical Papers*, pp. 2335–2344, Dublin City University and Association for Computational Linguistics, Dublin, Ireland (2014).
36. W.-T. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*, Short Papers, Vol. 2, pp. 643–648, Association for Computational Linguistics, Baltimore, Maryland (2014).

37. H. Li et al., "A benchmark for semantic image segmentation," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1–6, IEEE (2013).
38. J. W. Richards et al., "Construction of a calibrated probabilistic classification catalog: application to 50k variable sources in the all-sky automated survey," *Astrophys. J. Suppl. Ser.* **203**(2), 32 (2012).
39. T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.* **27**(8), 861–874 (2006).

Ramesh Kestur received his BE degree in electronics engineering from Bangalore University in 1995, his ME degree in electronics and communication from UVCE, Bangalore, in 2015. He has 17 years of work experience in telecom product development while he worked as a group project manager in product engineering division at Infosys Technologies, Bangalore. Currently, he is a PhD student at BIT, Bangalore. His research interests include image processing, machine learning, and CNNs.

Shariq Farooq is an undergraduate student of electronics and communication engineering at National Institute of Technology (NIT), Srinagar, India. He works as a machine learning research intern at Computational Intelligence (CINT) Lab, Department of Aerospace, Indian Institute of Science (IISc) Bangalore, India. His areas of interest for research include machine learning, convolutional networks, and memory networks.

Rameen Abdal is a BTech degree student at electronics and communication engineering (ECE) department, National Institute of Technology (NIT) Srinagar, India. He is a machine learning research intern at Computational Intelligence (CINT) Lab, Department of Aerospace, Indian Institute of Science (IISc) Bangalore, India. He is mainly engaged in the research areas of deep learning, pattern recognition, and computer vision.

Emad Mehraj is an undergraduate student of electronics and communication engineering at National Institute of Technology (NIT), Srinagar, India. He is a machine learning research intern at Computational Intelligence (CINT) Lab, Department of Aerospace, Indian Institute of Science (IISc) Bangalore, India. His current research interests include machine learning, neural networks, and computer vision.

Omkar Narasipura received his BE degree in mechanical engineering from the UVCE, Bangalore in 1985, his MSc (eng) degree in aerospace engineering from Indian Institute of Science in 1992, and his PhD in aerospace engineering from the Indian Institute of Science (IISc), Bangalore, in 1999. He is a chief research scientist at the Department of Aerospace Engineering, IISc, Bangalore. His research interests include biomechanics, helicopter dynamics, fuzzy logic, image processing, neural networks, and parallel computing.

Meenavathi Mudigere received her BE degree in electronics and communication engineering from Mysore University in 1989, her ME degree in digital techniques and instrumentation from University of Indore, Madhya Pradesh 1994, and her PhD in electronics and communication engineering from Dr. MGR University, Chennai, in 2010. She is a professor and head at the Department of Electronics and Instrumentation Engineering, BIT, Bangalore. Her research interests include image processing, signal processing, neural networks, and fuzzy logic.