Multisized Object Detection Using Spaceborne Optical Imagery

Muhammad Haroon, Muhammad Shahzad, and Muhammad Moazam Fraz

***Abstract*—This article addresses the highly challenging problem of vehicle detection from high-resolution remote sensing imagery by introducing a novel medium size annotated dataset named satellite imagery multivehicles dataset (SIMD) along with an adapted single pass deep multiscale object detection framework with the aim to detect multisized/type objects for catering above-ground perspective of vehicles. The dataset images are acquired from multiple locations in the EU/US regions available in the public Google Earth satellite imagery. Specifically, it comprises 5000 images of resolution 1024 × 768 and collectively contains 45 096 objects in 15 different classes of vehicles including cars, trucks, buses, long vehicles, various types of aircrafts, and boats. In the proposed architecture, we demonstrate the relevant modifications needed to translate the state-of-the-art object detection frameworks to solve the object detection problem from remote sensing imagery. The proposed architecture has been evaluated on SIMD and a public dataset VEDAI. The comparative analysis has been performed with existing off-the-shelf single-shot object detection models including YOLO and YOLT yielding superior performance measured with standard evaluation strategies. To ignite further research in this domain, the introduced SIMD dataset and the corresponding architecture is publicly available at this link: http://vision.seecs.edu.pk/simd.**

***Index Terms*—Aerial surveillance, aircraft detection, deep neural networks, satellite imagery, vehicle detection, YOLO.**

I. Introduction

MOVABLE object detection in aerial or satellite imagery is of great practical interest owing to its variety of applications in numerous fields including traffic monitoring, airport surveillance, parking lot analysis, search and rescue (SAR), determining transportation infrastructure, etc. However, the problem is highly challenging since such remote sensing images are acquired from high altitudes causing atmospheric distortions, illumination and viewpoint variations, partial occlusions, and clutter (especially in urban environments). Moreover, objects when viewed from an elevated platform (satellite, drone etc.) present a difficult to understand prospect and arbitrary orientation, which subsequently leads to suboptimal performance of machine (deep) learning models trained over datasets containing terrestrial object images. For example, a complex scene of a traffic intersection in Paris, France, is shown in Fig. 1 where one may wish to detect all types of vehicles from this image. The movable objects and vehicles present in this view are of varying sizes including small, medium, and large objects. These objects have wide variety in terms of object closeness and vacillating directions in comparison to normal ground based vertical images. Due to these variations in size, direction, object closeness, self-occlusion, and variety of multiobjects in a single satellite image, a traditional neural network trained on available datasets of ground-based images has limited potential to solve aerial perspective based object detection task.

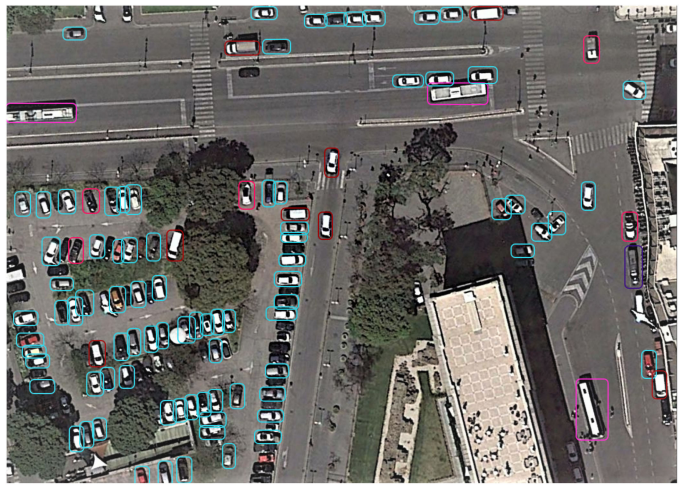


Fig. 1. Example from our dataset showing small vehicles in congested traffic shown at a signal crossing at City Center, Paris, France. It can be seen that objects are very small and occupy very less resolution in complete image.

Conventionally, the task of object detection relied on appearance-based handcrafted features encapsulating geometric and structural attributes pertaining to information related to color, texture, shape, etc. Later, these features are fed to a typical machine learning classifier such as support vector machine and random forests to detect the item of interest. Even though such techniques perform fairly well but are often constrained owing to the lack of representational abilities of handcrafted features. With recent state-of-the-art progression in neural network architectures, the deep-learning-based approaches provide much improved generalization power by encompassing several hidden layers to automatically learn high-level abstract image features relevant to the task at hand. Among different deep neural

Manuscript received April 4, 2020; revised May 30, 2020; accepted June 2, 2020. Date of publication June 5, 2020; date of current version June 18, 2020. This work was supported by NUST, Islamabad, Pakistan. *(Corresponding author: Muhammad Shahzad.)*

Muhammad Haroon and Muhammad Moazam Fraz are with the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan (e-mail: mharoon.mscs16seecs@seecs.edu.pk; moazam.fraz@seecs.edu.pk).

Muhammad Shahzad is with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan, and also with the Deep Learning Laboratory (DLL), National Center of Artificial Intelligence (NCAI), Islamabad 44000, Pakistan (e-mail: muhammad.shehzad@seecs.edu.pk).

Digital Object Identifier 10.1109/JSTARS.2020.3000317

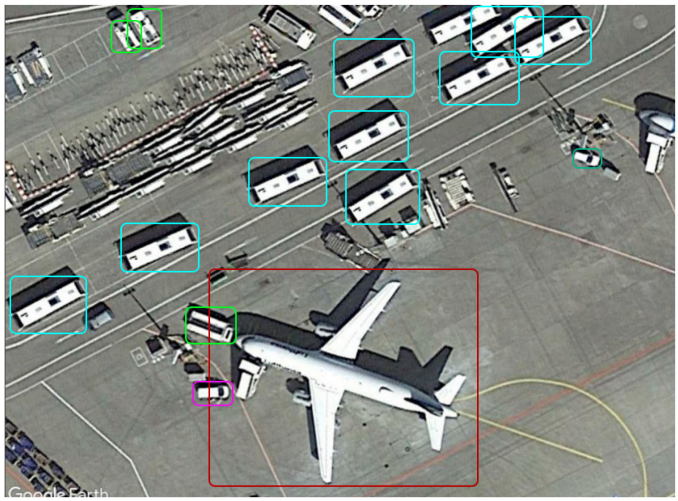


Fig. 2. Large-size airplanes, medium-size buses and small-size vehicles, all present in one satellite image taken at Charles de Gaulle Airport, Paris.

network-based architectures, the convolutional neural networks (CNNs) is the most prominent and state-of-the-art technique in processing images for vision related tasks. Lately, several notable deep learning based object detection architectures have been proposed.

For instance, Region-based CNNs (R-CNNs [1]) and its variants Fast R-CNN [2], Faster R-CNN [3], SSD [4], Retina Net [5], and YOLO [6] along with its extensions are some of the worth mentioning architectures that have achieved significant improvements on the large-scale benchmark ImageNet and COCO datasets. However, these object detection models cannot be directly applied to detect objects in aerial images owing to different viewing characteristics of aerial images. E.g., the objects of interest have monotonic appearance, i.e., the roads, flat parking, vehicle tops, and building roofs have no or very little texture (see Fig. 1). Moreover, in the context of movable object (vehicles) detection, there are different instances with varying sizes, i.e., varying proportion of size for different types of vehicles may be present in same image. As an example, Fig. 2, taken from Charles de Gaulle Airport in Paris, shows multisized vehicles present in one sample of satellite image. The relative size of vans and buses parked nearby a large sized aircraft is much smaller. Such complexities of size variations along with no (or homogeneous) texture information makes it highly challenging to learn distinctive features able to discriminate one object from another. Thus, to cope with these problems and enable effective feature learning, in addition to multiscale object detectors, diverse and large-scale annotated satellite/aerial datasets are essential which at present remains the major bottleneck of research in this domain.

A few datasets such as VEDAI [7], COWC [8], DOTA [9], NWPU-VHR10 [10], and DLR-3K [11] have been introduced using RGB images to enable detection of vehicles in satellite/aerial imagery. Although these datasets do have vehicle classes but are limited to only basic types of vehicles such as cars and other small vehicles. To solve for a variety of practical real-world applications, one must deal with more variations in vehicle classes/types in addition to challenges involving viewpoint variations and complex background owing to occlusions and clutter. Within this context, the motivation of this article is two-fold.

1) As a primary contribution, this article introduces satellite imagery multivehicles dataset (SIMD), which is a novel annotated dataset designed to address the task of multisized and multiclass/type vehicle detection in remote sensing images. Specifically, a large-scale dataset for vehicles detection with annotations in three commonly used formats has been presented which consists of 5000 satellite images of around 45 000 vehicular objects categorized in 15 dedicated classes. Such diversity of vehicle appearances will allow to make further progress in the field of automatic scene analysis, scene surveillance, and target detection.
2) Secondarily, the article benchmark the performance of some baseline algorithms and show their performance on the proposed dataset for the sake of comparison. Moreover, we have also presented results of an adapted single pass deep multiscale object detection framework to detect small, medium, and large size objects. The proposed architecture is thoroughly compared with the existing deep learning object detectors and achieved superior performance. The proposed model source code, configurations and dataset along with annotations is open to further research and shall be made available on Git Hub[1] upon publication.

It is worth mentioning that the imagery acquired for dataset generation is primarily acquired from Google Earth platform which provides high-resolution RGB images. The inspiration of using RGB images (in contrast to other satellite imaging modalities such as multispectral/hyperspectral/radar) is because they provide more flexibility in terms of cross platform capability. With this, it is meant that the RGB dataset prepared using satellite images can still be used with an airborne platform having no spectral sensor. For the same reason, almost all the relevant existing vehicle detection datasets including DOTA, VEDAI, NWPU-VHR, DLR 3 K, and RSD-GOD exploit RGB high-resolution imagery only. Furthermore, another aspect of using RGB high-resolution imagery only is the fact that the primary users of any vehicle detection system are the law enforcement agencies which could use such a system for the purpose of automated traffic management and control, forensics, crime prevention, statistical analysis, etc. In this context, the use of Google Earth platform allows the flexibility to download high-resolution RGB images with predefined viewpoint and altitude making them suitable to acquire drone-like imagery and thus the annotations of the images can be used in a relatively more cross-platform independent manner.

II. Related Work

There are two aspects of existing studies done on vehicle detection in aerial imagery. First is the availability of correctly

[1][Online]. Available: Dataset GitHub: https://github.com/ihians/simd

annotated dataset of necessary objects as research in neural networks is highly dependent on it. Second is the transformation of existing or introduction of new fine-tuned neural network that could work on object detection in aerial domain. Following sections contain information about existing works in these both aspects.

A. Available Datasets

Object detection in aerial domain is relatively new in comparison to ground-based object detection, therefore, availability of large scale dataset such as MS COCO or ImageNet is sparse. However, many small scale dataset has been introduced which gives initial benchmarks for object detection in aerial domain. We introduced only few here to establish the concept of need of a new dataset.

DLR-3 K dataset introduced by Liu *et al.* [11] from *German Aerospace Center (DLR)* contains 20 high-resolution (5616 $\times$ 3744) images taken from 1000 m AGL using an airborne platform. This dataset contains only two vehicle classes, i.e., large vehicles for trucks and buses and small vehicles for cars and vans. A much popular *Cars Overhead With Context (COWC)* [8] dataset contains aerial images of vehicles taken at six different locations gives a kick start to small models. It also contains only two classes, i.e., cars and noncars with 32 716 instances and is ideal for binary classifiers targeting to perform car counting operations on a given aerial image.

Razakarivony and Jurie [7] introduced a small scale dataset *VEDAI* containing 11 dedicated vehicle classes and provides a basic level startup dataset with 512 and 1024 resolution images (RGB and infrared). The article is inclined toward automatic target recognition (ATR) tasks from an aerial perspective. Due to its well annotation and manageable size, VEDAI is widely cited across literature. Robicquet *et al.* [12] from Stanford University released a dataset of drone videos recorded in university campuses to depict human behavior. It contains a set of annotated videos in various classes out of which only three belong to the vehicles class including cars, golf carts, and buses. Carlet and Abayowa [13] introduced a hybrid dataset which contained UAV streams videos and images from static camera placed on a building facing a parking lot. The article presented experimental results from a later version of Redmon *et al.* [6]. Cheng *et al.* [10] introduced NWPU VHR-10 which contains 800 images in 10 classes of different nature out of which three classes belong to the vehicles (airplane, ship, vehicles) category. Liu *et al.* [14] introduced a dataset of ships containing 1070 images from Google Earth. Zhuang *et al.* [15] introduced a latest dataset which contains aerial imagery data of five classes that includes airport, helicopter, oil tank, plane, and warship.

DOTA introduced by Xia *et al.* [9] is one of the largest available dataset as per our knowledge which contains 188 282 instances in 2800 images of about 4000 pixels resolution. It has oriented bounding boxes annotation format which can help localizing objects after direction adjustment. However, DOTA covers only five generic vehicles types which include small vehicles, large vehicles, ships, planes, and helicopters. Yang *et al.* [16] introduced another data set *ITCVD* which was taken from an aeroplane flying over The Netherlands. The dataset contains 228 images of high resolution (5616 $\times$ 3744) containing 29 088 instances of general vehicles. Very few researchers focused on Infrared dataset generation, one such example is of Liu *et al.* [17] who generated custom dataset using FLIR TAU2 camera on board a UAV. Robinson and Zhang [18] proposed a CNN for wide area surveillance on their customized infrared dataset at PV Labs.

Existing abovementioned datasets comprises either very less number of vehicle classes or contains less number of instances, both the cases are inefficient for training of a robust classifier. Our dataset addresses this issue by introducing 15 dedicated vehicle classes with 45 K instances in normal resolution images. This contribution would definitely help in designing new networks as well as to evaluate existing architectures.

B. CNN Models for Aerial Object Detection

Initial spade work on object detection from aerial platform was presented by Joshua *et al.* [19]. The proposed algorithm (fast detection) eliminates background using Haris Corner detection and then applied random forest classifier to detect man made objects and vehicles using target clustering. Object detection using hand crafted features suffer in generalizing as discussed by Ren *et al.* [3] therefore, Liu and Mattyus [11] presented first time use of sliding window in two-stage detector with experimental results on cropped images of 48 $\times$ 48 and 48 $\times$ 28 resolution. Mundhenk *et al.* [8] also presented a neural network called ResCeption with Inception styled layers to count cars in one pass. Object-based image classification by construction of object adjacent graph has been introduced for satellite images [20].

Later, when Ren *et al.* [3] introduced region boxes (anchors) in Faster R-CNN, it reduced complexity and provided significant improvement in time complexity. Researchers then focused on single pass object detection such as Joseph *et al.* [6] introduced their model named *You Only Look Once (YOLO)* which use customized anchor boxes and it was widely accepted as fast and accurate model. Sakla *et al.* [21] presented a parametric configuration of Faster R-CNN in their article using VEDAI [7] dataset for performing end to end object detection. The authors adjusted training parameters to detect small objects from aerial imagery which were insufficient with default values. Carlet and Abayowa [13] and Xia *et al.* [9], both used enhanced version of YOLO [6] with default layers configuration to train CNNs their respective datasets. Liu *et al.* [22] also worked on object detection from infrared imagery in a similar way in their article. Tayara and Chong [23] used pyramid styled CNN built on multiple backbone networks including VGG-16, Resnet-50, and Resnet-101 stacked with feature maps for the purpose of object detection. Similar to this idea, Tayara *et al.* [24] presented a vehicle counting techniques from aerial imagery using fully convolutional regression network to regress vehicle spatial density map across all the image and then performed downsampling using VGG-16 network concatenated with skip connections to achieve the object localization. Similarly, Tang *et al.* [25] proposed a feed forward CNN (Oriented SSD) for oriented bounding box detection using VGG-16 as the backbone network.

To enable robust feature extraction, Wu *et al.* [26] proposed a novel object detection framework (ORSIm detector) which incorporates diverse spatial-frequency features extraction and learning, fast pyramid matching, and boosting strategy to obtain high-level and semantically meaningful features. Similarly, Wu *et al.* [27] also exploited the idea of feature boosting and employed it to aggregate channel features for discriminative feature representation useful to perform efficient geospatial object detection. Koga *et al.* [28] used the concept of hard example mining to train CNNs for improved object detection. Similarly, Tianyu *et al.* [29] also used hard negative example mining to employ hyper region proposal network in extracting vehicles with higher recall rate. Audebert *et al.* [30] used SegNet with semantic segmentation to generate segmented objects from imagery. This method provides results in segmented form instead of bounding box. Shao *et al.* [31] also used segmentation coupled with FCN to extract multilabel classes from satellite imagery.

Object tracking in satellite videos has been demonstrated in a recent study [32] by using a simple regression network. Yang *et al.* [33] built motion heat-map coupled with visual background extractor based motion detection, targeting urban planning from airborne videos. Extracting features using linear SVM from input image segments followed by a pass of sliding rectangular window to remove false positives is proposed by Ammour *et al.* [34]. A surveillance detection system [35] has been proposed for various fire detection, and SAR operation by using thermal information retrieved from infrared images human versus large fire patches.

Terrail *et al.* [36] proposed a modified pipeline of Faster RCNN with oriented and nonoriented detection showing experimental results on VEDAI dataset. Sommer *et al.* [37] also extended Faster RCNN model by applying deconvolutional module that upsampled low-dimensional features and combined them with features from shallow layers. Instance segmentation coupled with object detection has been introduced by Zamir *et al.* [38] for accurate object localization of dense scenes. Ding *et al.* [39] used a transformed Region of Interest (RoI) model by combining horizontal and oriented RoI with experimental results on DOTA. Zhuang *et al.* [15] proposed a single shot framework with multiscale fusion of coarse features. Li *et al.* [40] proposed a scale invariant CNN architecture which works on parallel multibranch with different receptive fields. Li *et al.* [41] also proposed a method for rotate-able detection. Chen *et al.* [42] proposed a hybrid deep neural network with multiple blocks of variable receptive field sizes or max-pooling field sizes, to enable extraction of variable-scale features to detect particularly smaller vehicles. Similarly, Zhang *et al.* [43] proposed multiscale feature pyramid network to fuse low resolution and weak features with high resolution features for robust object detection. Yang *et al.* [44] also presented a novel multicategory rotation detector for small, cluttered, and rotated objects which fuses multilayer attention based feature extraction network with effective anchor sampling to improve the sensitivity to smaller objects. Recently, Chen *et al.* [45] proposed a novel two-stage object spatial density building net (SDBN) where in first phase candidate regions are generated that are later refined in the later stage with meticulous heauristics. The accurate geometrical parameters of all objects are computed using the spatial density maps similar to [24] and achieved state-of-the-art performance in four different target maneuvering target datasets.

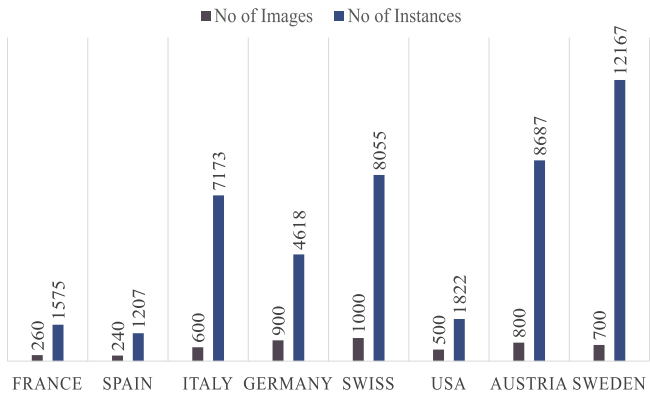


Fig. 3. Country wise collected images and annotated instances.

All this literature work focused either on generic vehicles detection (vehicle/nonvehicle) or counting of vehicles mostly for applications in traffic analysis. Whereas very few had entirely focused on detecting the type of vehicle in complex satellite imagery scene at multiscale level. Our work addresses this deficiency in a novel way by proposing an end-to-end solution covering from dataset availability to the network architecture design.

III. Dataset Description

A. Data Collection

A number of constraints are to be studied while collecting data such as lightening conditions, appropriate elevation, and area to be captured with presence of ample number of vehicles, etc. Downloading satellite imagery without these consideration may lead to data that is not useful for the task in hand. For example, many images present in VEDAI [7] are without any object present in them. All images available in our dataset are of same resolution (1024×768) in RGB format with an elevation of almost 500 to 1000 feet above ground level (AGL). Compute power required to process very high-resolution (VHR) images is many fold in comparison to less-resolution images. Therefore, instead of selecting very less images with high resolution as done in DLR3K [11] and ITCVD [16], we choose to select more images with low resolution as done in VEDAI [7] and NWPU-VHR10 [10]. The purpose of this methodology is to fasten the neural network model training and validation by using the less compute power. We envisaged to covert out dataset in array form and be made available as an API for fast prototyping and experiments. MNIST Hand Writing dataset is an example of array dataset which is available in Keras [46].

In existing datasets such as NWPU VHR-10 [10] and ITCVD [16], the imagery was captured from a single place with multiple shots of different scenes. This created tendency of biases in data and CNN trained on this may not perform well on evaluation data from other locations. To overcome this problem

Fig. 4. Data samples from our dataset comprising variety of instances including multiple classes. Also showing arbitrary angles and closeness of multitype vehicles in close vicinity.

our dataset has been collected from various European countries with multiple locations of city centers and airports. More emphasis was put to collect data from signal crossing, highways, parking lots, beaches, bus stands, workshops, lakes, airport tarmacs, and runways. Number of images and corresponding annotated instances from each country are shown in Fig. 3. To add diversity in the dataset, images have been collected in different weather conditions such as bright sunshine, afternoons, and few icy conditions. However, it is to be noted that class imbalance still exists in our dataset as cars are widely used vehicle type and are present in large quantity in overhead imagery of any city. This problem can be resolved by implementing some antibias

TABLE I
LIST OF OBJECT TYPES, COUNT, AND SIZE CATEGORIES OF OUR DATASET SIMD

ID	Class Name	Count	Size
1	Car	20,242	Small
2	Truck	2,800	Medium
3	Van	5,732	Small
4	Long Vehicle	1,622	Large
5	Bus	1,989	Medium
6	Airliner	969	Large
7	Propeller Aircraft	209	Large
8	Trainer Aircraft	631	Medium
9	Chartered Aircraft	641	Large
10	Fighter Aircraft	63	Medium
11	Others	791	Small
12	Stair Truck	446	Small
13	Pushback Truck	228	Small
14	Helicopter	64	Small
15	Boat	8,669	Small
	Total	45,096	-

function in the model for cars class. Few samples from our dataset are shown in Fig. 4 displaying variety of objects with annotated bounding boxes available in each image.

B. Class Selection

Existing datasets such as DOTA [9], VEDAI [7], COWC [8], ITCVD [16], and RSD-GOD [15] lack this ability as their focus remained on detection of static objects and these datasets include very less number of generic vehicle classes. We have gathered imagery as mentioned above and carefully decided the number and type of classes. We focused on type of moveable objects including vehicles, various types of airplanes, and boats. We therefore select classes that are solely considered as maneuvering objects.

Selection of vehicle classes is a tedious task as almost all vehicles resemble rectangular shapes. On the contrary, neural networks tend to extract features from shapes of objects present in input images and train on these patterns. We select a set of classes which are different in shape but can be categorized under the same vehicle category. For example cars, buses, and trucks look like rectangles, whereas airplanes and boats have different distinct shapes. While selecting classes, a careful approach has been adopted to select most common viewable object categories. List of 15 classes with the number of objects and different size categories are shown in Table I.

C. Data Annotation

Most of the current object detection models work on horizontal bounding boxes such as Faster R-CNN [3] and SSD [4] for object detection, therefore, we choose to annotate our dataset on the same method and images has been annotated in plain horizontal and vertical rectangles instead of oriented bounding boxes. By using this way, there is no need to do preprocessing work or realign orientations by the architecture prior to perform detection task. Data annotation has been done using Microsoft VOTT formatting tool with rectangular bounding boxes. There are following two reasons why horizontal rectangular annotation has been done on our dataset.

1) Nature of Satellite Data: In most of the cases downloaded data or streaming feed is coming from satellites which are north aligned, however, ground structures such as buildings, roads, airports, beaches, and road network built on Earth are not aligned to any preset direction. Furthermore, the moveable objects including all types of vehicles are never found in any specific direction. Yang *et al.* also highlighted the same challenge by Cheng *et al.* [10] in their article. Due to this reason, when localizing vehicles from satellite data, vehicles may be found in any direction additional to any size depending upon elevation of the aerial sensor. Oriented bounding boxes with respect to any fixed point (i.e., top-left) will not be able to aid detection algorithms much. Rather it may put extra processing on the classifier to first align the image to north followed by localizing the object. Our dataset therefore presents horizontal aligned annotations which could directly be used for detection work without any preprocessing.

2) Practical Applications: If we observe the flight path and heading of any aerial platform, we could easily identify that no explicit heading is being maintained by UAVs and drones. Instead, they follow arbitrary paths and imagery/video data received from these platforms does not align to any particular direction. Hence, object detection algorithms that work on this type of data have to follow the same rule. If we train our model on a dataset that has oriented bounding box, it will be more compute intensive for detection. Developing real-time applications hence require a classifier with a higher speed which is irrespective of orientation detection.

D. Annotation Formats

The aim of our dataset is to provide an opportunity to future researchers to focus on neural network designing instead of data cleaning, preprocessing, and resolution adjustments. Therefore, our dataset contains fixed resolution images of 1024×768 taken from 500 feet AGL. All images in our dataset contain the same resolution, divisible by 32 which is often required by most CNNs and objects marked in images are of small, medium, and large categories. We provide three standard annotation formats which are performing object detection via following state-of-the-art deep neural network architectures.

1) YOLO (You Only Look Once): This is the most common and easy to understand annotation format which can be described as (c, xi, yi, w, and h), where c is the object class starting from 0, (xi, yi) are the center of object and (w, h) are width and height, respectively. All these values are percentages to the actual image.
2) CNTK Faster RCNN Format: Developed by Microsoft for Faster RCNN, this format saves data in 02 XML files. First for bounding box information described as (x1, y1, x2, and y2) where xi and yi are exact coordinates in the xy domain. Second file contains the name of the class type written in plain, i.e., car, bus, truck, etc. Both files are saved with each single image in the same directory.

TABLE II
COMPARISON OF OUR DATASET WITH EXISTING DATASET CHARACTERISTICS

Dataset	Classes	Vehicles	Instances	Images	Resolution	Formats	Annotation
COWC [8]	1	1	32,716	53	2000 - 19000	1	one dot
UCAS-AOD [47]	2	2	14,595	1,510	1300 x 700	1	oriented
RSD-GOD [15]	5	3	40,990	18,187	300 ~600	1	horizontal
NWPU VHR-10 [10]	10	3	3,651	800	~1000	1	horizontal
DLR 3K [11]	10	2	14,235	20	5616 x 3744	1	oriented
VEDAI [7]	12	11	2,950	1,268	1024 x 1024	1	oriented
DOTA [9]	15	5	1,88,282	2,806	800 - 4000	1	oriented
SIMD (Proposed)	15	15	45,096	5,000	1024 x 768	3	horizontal

It is clearly shown that our introduced dataset SIMD contains maximum possible distinct vehicle classes with ample instances.

3) Tensor-flow Pascal VOC: This is an XML format which contains a single file for each image and contains complete image information in it. The first part contains image information such as file, folder, image resolution, depth, etc. The second part contains a repeating instances tag which contains the name and coordinates of each instance described as (xmin, xmax, ymin, ymax).

E. Characteristics of Data

As per our knowledge, we provide the largest repository of multiple vehicles dataset except DOTA [9] in comparison to other available datasets to extend research in the aerial domain. Our dataset contains 45 K instances of vehicles (only) types whereas DOTA contains five types of general vehicle objects mixed with other types. A comparative analysis of our dataset with known object detection datasets in aerial imagery is presented in Table II. It is clearly shown that our dataset addresses various major concerns including number of classes, annotation formats, and resolution. Following are few of the major characteristics of our data covering most concerns.

1) Location Diversity: One of the major concerns in available datasets such as Standford UAV, DOTA, and US AFRL dataset [9], [12], [13] is the monotonous of data objects as these images belong to one particular city or area thus generating almost same data multiple times. This does not suffice the objective to train generalized models which could work in all types of evaluations and test samples. Our dataset addresses this problem by providing data instances from a variety of locations. It contains images from 79 distinct locations mostly from European countries (France, Spain, Italy, Germany, Switzerland, Austria, Sweden) and the USA. To add further diversity in our data, we choose imagery from city centers, bus stops, signal crossings, highways, airports, beaches, and dry ports. This not only provides a different and difficult background for our dataset, but also includes variety in the shapes of vehicles from multiple countries such that a neural network trained on our dataset would be more generic and it must be less biased toward vehicle types of a specific area. A comprehensive detail of the number of objects taken from distinct locations with respect to countries is given in Fig. 3.

2) Object Size: When viewed from top (around 500 feet in our case) vehicles of varying sizes are visible. We collect our data in such a way that it contains objects of very small size such as of 23 pixels width as well as it also includes very large size objects of about 1000 pixels width and height. This made our dataset more diverse and usable to train generic models for small-, medium-, and large-sized objects. A CNN trained on our dataset has to face object size from 23 pixels to 1000 pixels.

3) Density of Objects: We have taken images with a dense number of objects per image. Various existing datasets such DLR3k [11] and VEDAI [7] have very less number of objects mostly from 1 to 10 instances per image. Our dataset contains instance count ranging from a single object to 100 objects per image and that too within the range of maximum resolution 1024. This gives a challenging task of fine tuning to the learning network.

IV. VEHICLE DETECTION NETWORK ARCHITECTURE

A. Brief Introduction to CNNs

CNN are specific types of neural networks which deal with image recognition and classification tasks. The correct composition of various components such as input data, output classes, loss function, and optimizer is referred to as CNN model. There are three very basic functions which a CNN has to perform. First is classification which is to identify the class to which a given object belongs. This is usually achieved by probability estimation of all classes and selecting the highest value. Second is localization which deals with the position of objects in an image. Last is the combining of output from the first two steps and providing bounding boxes around detected objects. The most difficult and important task among the above is classification. This task is performed by sliding fixed size windows across the image, extracting features from each window, calculating probability among classes, and finally combining same and adjacent windows to output localization and classification.

One class of frameworks is region-based networks which works on two passes by generation of proposed regions where objects are likely to be present followed by a CNN which classifies the class of object and performs localization. Object detection networks are not only to detect objects accurately, but need to be incredibly fast for real time object detection especially in video feeds which are recorded at least at 24 fps. Although Faster R-CNN [3] improves speed in comparison to existing work, it has limitations for real time applications. SDBN [45] also used a two phase approach to enhance accuracy. Another type of frameworks relying on unified single process pass emerged as single pass object detection. The initial concept

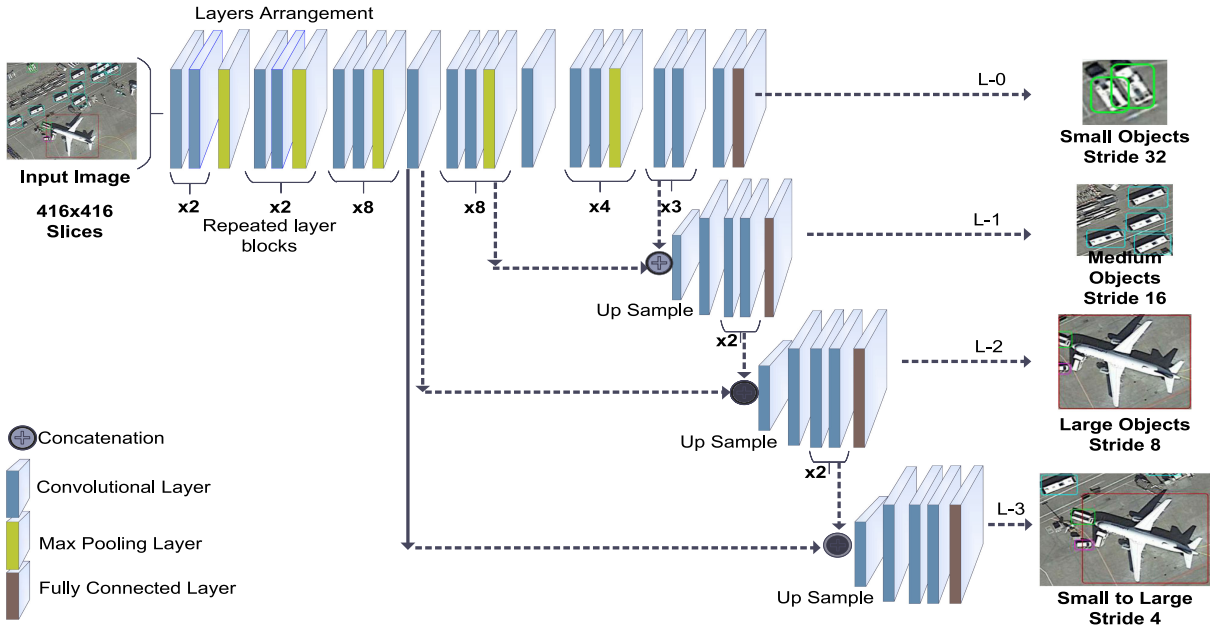


Fig. 5. Our proposed CNN architecture with details of layers arrangement, up sampling, concatenation, and detection levels. It is shown that our model detects multiscale objects from same input image in single pass using various multilevels.

was presented by Liu *et al.* in their article titled *SSD: Single Shot Multi-Box Detector* [4]. Here, the detector performs localization and classification of objects in a single pass over the complete image which reduces time significantly. YOLO implements semi SSD technique and performs object detection by putting a fixed cell grid over the image and allows each grid cell to detect objects. An enhanced version, YOLO v3 outperformed various existing methods in accuracy and speed [48].

B. Proposed Model

We propose a network model which falls under single pass object detection frameworks. As object detection from aerial imagery is a challenging task, therefore, we use deep learning to solve this problem. We used a C++ based (Darknet) implementation of YOLO. We performed our network designing and training using the same framework and demonstrated our results considering YOLO as our base network. The proposed network model is a fully convolutional network designed using convolutional, max pooling, and up sample layers. The proposed model contains 118 layers deep neural network with four levels of detection. The proposed model takes an RGB input image and performs multivehicle detection. The output is in the form of bounding boxes around detected objects of any size within the boundaries of the input image.

The existing model YOLO v3 is performing object detection at three levels on MS COCO dataset. While balancing between accuracy and speed, we proposed four levels detection as per the task at hand. The proposed model performance has been balanced to perform detection at four levels, i.e., L-0, L-1, L2, and L-3 to cater various object sizes. The detailed layers arrangements and description has been shown in Fig. 5. The proposed model is evaluated on VEDAI [7] and on our introduced dataset SIMD. Adding more levels of detection improved accuracy however declines speed which is shown later.

In the first part, the proposed model extracts features from the input image using stacked convolutional layers along multiple residual blocks just like ResNet structure. In each block of layers, we use two convolutional layers with incremental filters along with one residual layer. These blocks are then repeated in multiplicities of (1, 2, 8, 8, 4) separated by one convolutional layer each. The filters are gradually incremented from 32 filters at initial to 1024 filters at last block. In the second part, we made detection at four levels by getting input features extracting at different previous layers. For detection, our model concatenates previous level layers with the upsampled versions of the learned layers down the line. The detection layer at last is fully connected and produces the class probabilities. The details of the proposed multiscale architecture on each detection level is described here.

1) *Small Vehicle Detection:* Very small object detection is being done by features extracted from the first block of layers concatenated with layers from the second block. After concatenation, we add a set of four more convolutional layers to adjust resolutions. Densely packed small objects are detected at this level. Small object detection is done after up sampling the result added with two blocks of more layers. Detection results at this level are preserved to be used later at last level.
2) *Medium Vehicles Detection:* To detect medium sized objects, the model uses more convolutional layers from the upsampled version of L-0 block of layers. This block contains three sets of two convolutional layers each followed by a convolutional layer with 60 filters. Then, the resultant layer is concatenated with the extracted feature layer followed by the same set of four convolutional layers to adjust resolution.

3) *Large Vehicles Detection:* To detect large-sized objects, the model then uses more convolution layers from upsampled versions of L-1 Block layers. This block contains three sets of two convolutional layers each followed by a convolutional layer with 60 filters. Then, the resultant layer is concatenated with the extracted feature layer followed by the same set of four convolutional layers to adjust resolution. The previous and this block has the same number of layers, however, input from the feature layer is from different layers as shown in Fig. 5.
4) *Last Level Detection:* To further enhance accuracy we added a last level detection for any missing object. This level used features layer from the very first block and concatenated it with the upsampled version of the latest L-3 block. We added a block that contains three sets of two convolutional layers each followed by a convolutional layer with 60 filters. Detection is done at the last layer which covers all missing objects from small to large.

C. Experiments and Model Fine Tuning

We experiment with various types of methods and layers combinations to improve accuracy out of which very prominent are listed below.

1) Regenerated Anchor Boxes: Instead of using existing nine anchor boxes from our base network YOLO v3 [48], we increased anchor boxes and regenerated 12 using K-Means with 416 height and width. These regenerated anchor boxes were then used at four detection levels for training our model and it improved the accuracy. We use four sets of anchors boxes, three anchor boxes used at each detection level.

2) Learning Rate: Learning rate is one of the crucial parameters while training the network model. Instead of using a steep constant learning rate, we used a step by step learning rate of 0.0001 across multiple GPUs. We adjust our network model for improved fine tuning and enhanced model performance by increasing learning rate by a factor of 0.0001 after every 40 K iterations which yields better performance.

3) Training Batches: Training batches play an important role in fine tuning of network models. We therefore did not limit our training iterations to small numbers. In our case, 1 epoch is completed in 250 iterations. We learned that our model is fine tuned after 200 K iterations, however, we train proposed network architecture for 240 K batches. The training loss in terms of mAP for each batch is shown in Fig. 6.

4) Transfer Learning: The network model if already trained for some classes can be used for further training. It needs not to start learning from arbitrary weights and converge to some reasonable weight values. We used the same technique for training and used existing trained base models instead of starting from scratch. For this purpose, we use weight values of YOLO v3 trained on MS COCO dataset.

D. Training Details

We trained the proposed and base networks on NVIDIA GeForce GTX TITAN X/1060 GPUs and evaluated the dataset in two iterations. We used a smaller dataset introduced in [7]

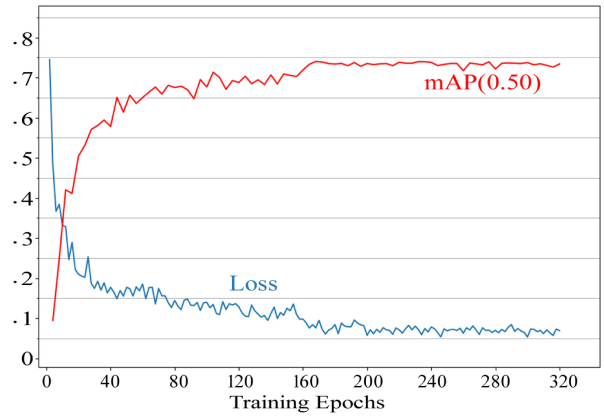


Fig. 6. Training loss and validation accuracy @mAP@0.50. *X*-axis shows number of epochs and *Y*-axis shows decimal values for training loss and validation accuracy. The training loss and accuracy became almost constant after 200 epochs.

TABLE III
COMPARISON OF DIFFERENT APS WHEN EXISTING MODEL APPLIED ON OUR CUSTOMIZED DATA SET

			AP Results	
ID	**Class Name**	**Size**	**YOLO v3**	**Proposed**
1	Car	Small	81.62	**84.75**
2	Truck	Small	58.05	**65.30**
3	Van	Small	71.10	**76.33**
4	Long Vehicle	Medium	59.68	**55.09**
5	Bus	Medium	47.29	**58.91**
6	Airliner	Large	66.67	**78.48**
7	Propeller Aircraft	Large	78.08	**84.03**
8	Trainer Aircraft	Medium	54.55	**75.00**
9	Chartered Aircraft	Large	**85.78**	76.81
10	Fighter Aircraft	Medium	0	**0.7**
11	Others	Small	15.01	**16.24**
12	Stair Truck	Small	**75.57**	61.27
13	Pushback Truck	Small	13.22	**24.46**
14	Helicopter	Small	54.55	**80.52**
15	Boat	Medium	**78.82**	75.31

There is a significant improvement in arbitrary shaped objects such as propeller aircraft and helicopter.

for evaluations of our proposed network in comparison to base network [48]. Similarly, we evaluated our customized dataset with base network and our proposed network. We used standard object detection evaluation metrics to assess performance of both the existing and proposed model. These metrics include precision recall curves (PRCs) which are shown in Fig. 7.

There are few experiments where we failed to achieve desired results, however we have list down the training loss, results, and reason of failure as of our knowledge. These experiments need to be more explored. For example, we tried to train the model on following already implemented CNN models in Darknet and based on the results, we designed our network. We also tried to add more layers than 118 in the model for further evaluation of our concept but after a certain limit, the systems performance started declining. All these experimental results are shown in Table III and Table IV.

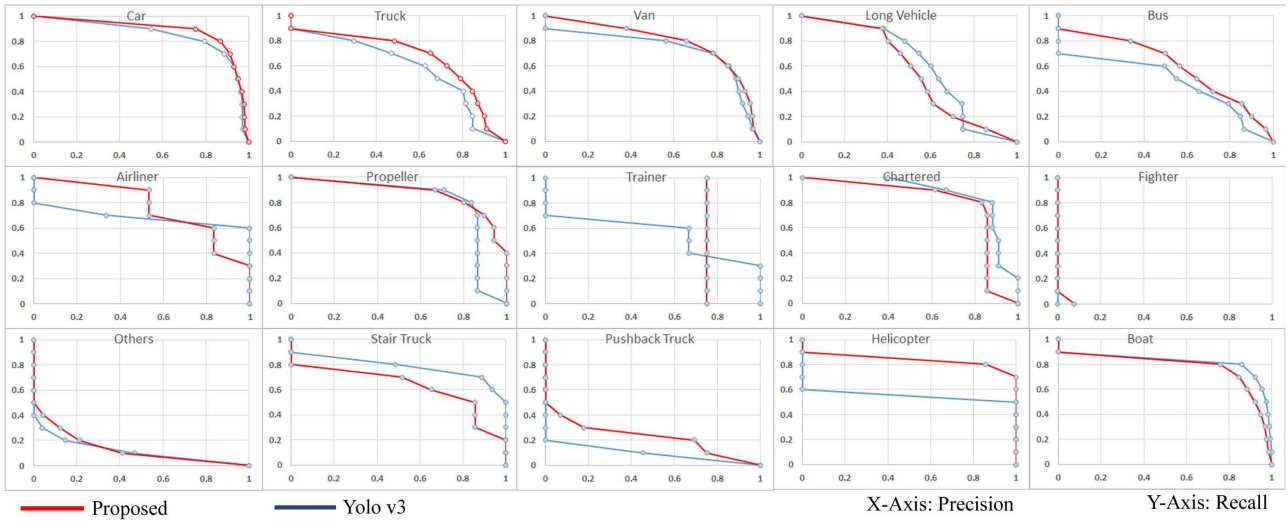


Fig. 7. PRCs of the proposed method and base model for following dedicated 15 vehicle types: car, van, truck, long vehicle, bus, airliner, propeller, trainer, chartered, fighter, others, stair truck, pushback truck, helicopter, and boat, respectively.

TABLE IV
PRECISION RESULTS OF VARIOUS EXPERIMENTS PERFORMED USING OTHER NETWORK ARCHITECTURES

	Experimented Networks					
Class	**Mini-3L**	**YOLOv2**	**YOLT** [49]	**YOLOv3** [48]	**Initial-4L**	**Proposed**
Car	53.04	74.07	84.51	81.62	50.82	**84.75**
Truck	44.32	**69.45**	60.64	58.05	60.76	65.30
Van	47.69	72.06	73.50	71.10	59.08	**76.33**
Long Vehicle	45.3	**64.13**	59.88	59.68	55.88	55.09
Bus	44.19	**69.36**	49.03	47.29	60.97	58.91
Airliner	1.56	77.69	73.30	66.67	77.81	**78.48**
Propeller Aircraft	44.78	60.35	66.66	78.08	73.00	**84.03**
Trainer Aircraft	60.61	68.00	50.00	54.55	58.01	**75.00**
Chartered Aircraft	31.42	79.70	73.74	**85.78**	82.55	76.81
Fighter Aircraft	**22.73**	0.57	0.91	0	12.12	0.7
Others	9.62	15.62	**19.13**	15.01	13.46	16.24
Stair Truck	27.2	69.97	52.17	**75.57**	57.44	61.27
Pushback Truck	6.66	12.07	21.49	13.22	18.61	**24.46**
Helicopter	76.92	23.36	17.43	54.55	76.14	**80.52**
Boat	45.98	55.46	**88.11**	78.82	61.36	75.31
mAP@0.50	37.47	54.12	52.70	56.00	54.53	**60.88**

The maximum AP achieved by models are shown in bold. The last column contains AP values of the proposed model.

V. EXPERIMENTAL EVALUATIONS

We evaluated the proposed network on NVIDIA GeForce GTX TITAN X/1060 GPUs with our dataset in two iterations. This shows that our model was able to detect objects from top view when evaluated on unseen images. We used a smaller dataset introduced in [7] for evaluations of our proposed network in comparison to the base network. Similarly, we evaluated our customized dataset with base network and our proposed network. We used PRCs to assess performance of both the existing and proposed model. Fig. 7 displays the PRCs for each class of our dataset. The average running time (seconds) is shown in Table V. The running time of our model is far better in comparison to RICNN [10] on similar dataset.

TABLE V
RUNNING TIME COMPARISON OF FIVE ARCHITECTURES CALCULATED ON SAME IMAGE WITH 100 INSTANCES IN IT

Architecture	**Average running time(seconds)**
RICNN [10]	8.77
YOLO v2	0.012268
YOLT [49]	0.015898
YOLO v3 [48]	0.023561
Initial-4L	0.026461
Proposed	0.025386

A. Network Evaluations

We took the base network and trained it on a smaller dataset VEDAI [7] for 190 K iterations initially. The results gave us

TABLE VI
COMPARISON OF RESULTS OF DIFFERENT DATASETS WITH EXISTING ARCHITECTURES AND OUR PROPOSED NETWORK

Datasets	mAP%	F1	IoU	Architecture
DLR 3K	71.40	-	-	YOLO [41]
VEDAI	41.34	0.58	43.71	YOLO v3 [48]
VEDAI	42.31	0.60	51.48	Proposed
UCAS-AOD	76.15	-	-	YOLO v2 [9]
NWPU VHR-10	72.63	-	-	Cheng et al. [10]
DOTA	60.51	-	-	YOLO v2 [9]
DOTA	68.16	-	-	Azimi et al. [50]
SIMD	56.00	0.77	68.14	YOLO v3 [48]
SIMD	60.88	0.74	56.09	Proposed

It is clearly shown that aerial datasets with classes more than 10 have reached mAP under 70%.

mAP@0.50 in range of 50%–55%, therefore, we started adding more layers while carefully observing the accuracy. Once our network design was finalized, it was then we introduced our customized dataset to our proposed network. Our model gave accuracy mAP@0.50 in range of 55%–60% maximum with half the number of instances of our dataset. However, when trained finally on a maximum number of instances for 240 K iterations (960 epochs), our model accuracy jumped in the range of 60%–65%. Table VI shows the results of various network architectures applied on known datasets. It is clearly shown that accuracy improved on datasets with less classes such as DLR 3K [11] and UCAS-AOD [47]. But accuracy dropped significantly when the same model applied to datasets with more than 10 classes such as VEDAI [7] and DOTA [9]. The complete PRCs of each class for proposed and base network is shown in Fig. 7.

B. Dataset Evaluations

One of the most common limitations in existing datasets is the less number of classes. Most of the datasets have 5 to 10 classes with a limited number of instances which are very less compared to other computer vision datasets such as MS COCO, ImageNet, etc. These datasets also have very less annotation formats which limits the number experiments which could be performed. Hence, we provide our network evaluations on VEDAI [7] which is closer to our problem. As our network model finalized, the customized dataset gave us an accuracy jump from 55% to 60.88%. However, the model stops further training after 200 K iterations and remains oscillating on the same accuracy. We also evaluated our results based on size of objects (small, medium, large). The accuracy of YOLOv3 [48] and our proposed architecture on two datasets, i.e., VEDAI and our customized dataset (SIMD) is shown in Table VII. It is shown that our proposed architecture worked equally well on all sizes of objects.

C. Qualitative Evaluations

Finally, we evaluated our model on unseen images from the last part of our dataset. We take best performing learned weights and use them for extracting qualitative evaluations with 0.25% threshold. The few samples from detection results are shown in Fig. 8. The detection results show the vehicles detection of all types which includes small cars, small boats, medium trucks, small and large aircrafts in various complex scenarios. It is clearly shown that our learned model localizes and identifies vehicle objects very efficiently on multiscales.

TABLE VII
ACCURACY MEASURES BASED ON SIZE OF OBJECTS BOTH FOR YOLO AND OUR PROPOSED ARCHITECTURE ON VEDAI AND SIMD DATASETS

	VEDAI			SIMD		
	S	M	L	S	M	L
Base Model	51.28	32.77	71.01	52.73	48.07	76.84
Proposed	46.39	38.68	80.24	58.41	53.01	79.78

Small (S), medium (M), and large (L).

To further evaluate our model, we also use our trained model on vehicle detection from drone imagery with image slices of various drone videos. Last two rows of Fig. 8 contains four images showing results of our model. It is clearly visible that our trained model can be applied to any other aerial imagery source including drone, airplane, etc. Hence, our trained model is fully capable to be used in real time applications.

D. Ablation Study

There are few experiments where we failed to achieve desired results, however these need to be more explored. The first experiment was performed on YOLOv2 which has 30 layers composition in total and detection at the last level as described by the authors. This is the small network designed initially by authors to work on ground-based imagery. We tried to train this model on our dataset but it could not perform very well on small objects. The AP values of small and complex objects are very less than larger objects such as airlines. This model has a maximum AP value of 74.07 that is on a higher frequent object (car). We also tried the Mini-3 L classifier which was designed for low computing devices such as mobiles and tablets. Mini-3 L layers composition has detection at 3 levels but has a total of 30 layers. We tried to train this model but it also could not perform very well on small objects. The AP values of small and complex objects are very less except for large objects such as trainer aircraft.

We tried a four-level detection method which has an equal interval of loop back layers. This model takes input from four various levels and combines the input to later learned layers and produces results accordingly. This method is suitable for ground-based imagery but it could not perform well on aerial imagery. As of our knowledge, no other author has implemented this model. As shown in Table IV, this model achieved maximum accuracy on the airliner which was a fixed shape object in our dataset. We applied stand configuration of YOLO v3 [48] on DOTA [9] dataset which produced very less @mAP 3.78%. Then, the same model is applied on our dataset SIMD which produces mixed values of AP. We tried a five-level detection method with equal intervals of loop back layers. This model takes input from five

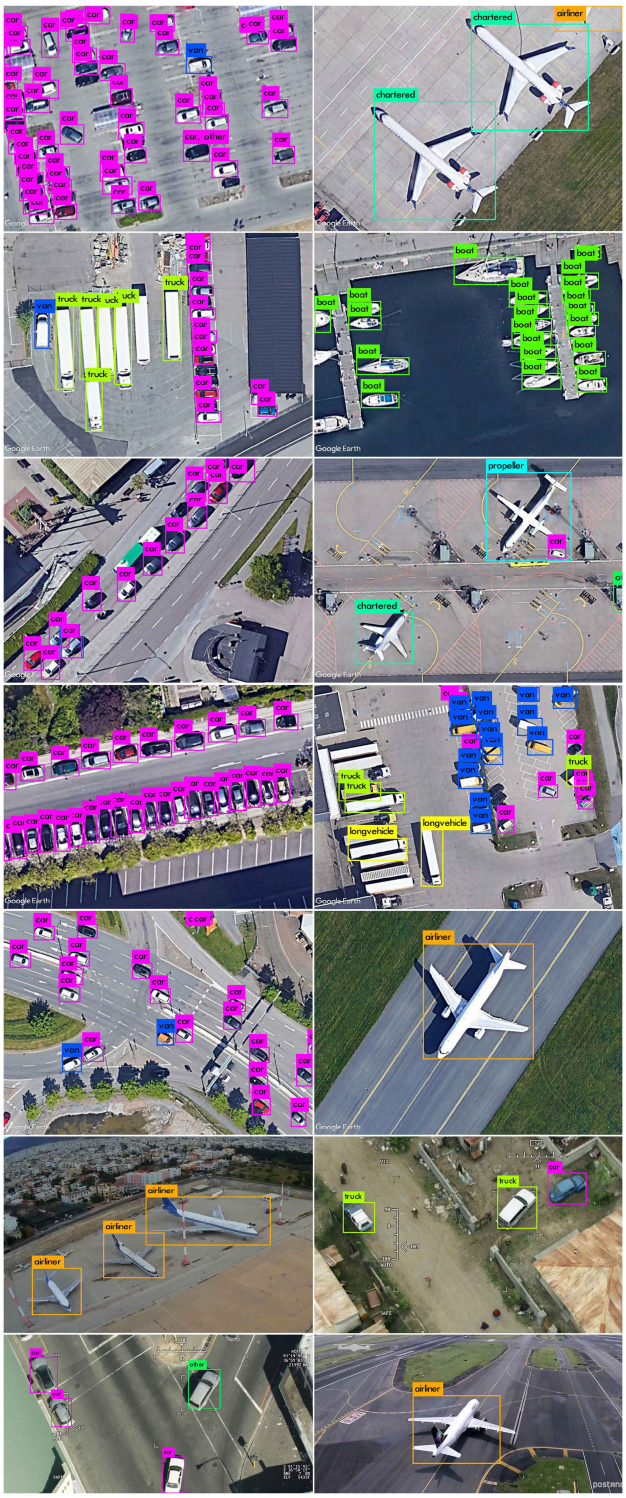


Fig. 8. Detection results on proposed model trained on our dataset. It includes all variety of objects from small, medium, and large. Last two rows showing results on drone imagery.

various levels and combines this input to later learned layers and produces results accordingly. This model could not achieve promising results in any class in comparison to existing work.

One of the reasons that the existing model performs well on few objects is the shape of the object. For example the shape of objects remains constant in a few classes across multiple instances such as aircrafts, long vehicles, and boats. But it could not perform well on arbitrary shaped objects where the shape of objects varies in different regions such as various types of cars, trucks, and medium sized aircrafts. Another reason for low performance on aerial images was the variety of backgrounds in images when seen from top. Each object when viewed from top has a different background. The roads are colored normally dark which confuse the network to decide object boundaries, whereas tarmacs (in case of aircrafts) are light colored which confuses make difficult for models to detect light colored aircraft detection. It is clearly shown in AP values of Table. IV that existing models work better in few cases but worst on all remaining classes. Whereas our proposed model achieved AP values more than 70% on prominent classes. Furthermore, Fig. 9 presents the comparative results which qualitatively shows that the proposed model detects the multisized objects both in simple and complex scenarios.

VI. Discussion

We have discussed various issues and problems occurred during the conception, designing, and implementation of the proposed method. We have deduced these discussion points based on our attempts and experiments which could be further explored in research of object detection in the aerial domain.

A. Varying Multisized Objects

If we analyze the results shown in Table IV, it is clear that existing models such as YOLOv2, Mini-3 L, and Initial-4 L have achieved maximum AP values on objects that are larger in size. It means that these classifiers are good at detecting large objects but at the same time were unable to detect small objects. To detect small objects in addition to large objects, we added another level of detection and demonstrated one such example by adding multilevel detection layers. It is imperative that in aerial imagery multisized objects will be present. The size of objects will vary as the elevation of the image or video capturing sensor is highly fluctuating. A classifier must be able to detect very small objects when viewed from higher elevation and also to detect same objects from very less elevation. Hence, a classifier must be able to detect multisized objects efficiently. Our research presented here can be considered as an entry to this area of research which has capacity to be further explored.

B. Preprocessing and Data Augmentation

It is shown in Table IV that our proposed network has AP value 0.7 on Fighter aircraft which has very few instances in our dataset. It can be deduced from here that the classifier is unable to train itself with less number of instances. A classifier must be fed with a balanced dataset to work effectively. The reason for less AP values of few classes is expected due to this class imbalance problem. The datasets already introduced and even our own dataset contains a higher number of most frequent objects such as cars in comparison to other classes such as fighter aircraft and helicopters. Data augmentation and prepossessing

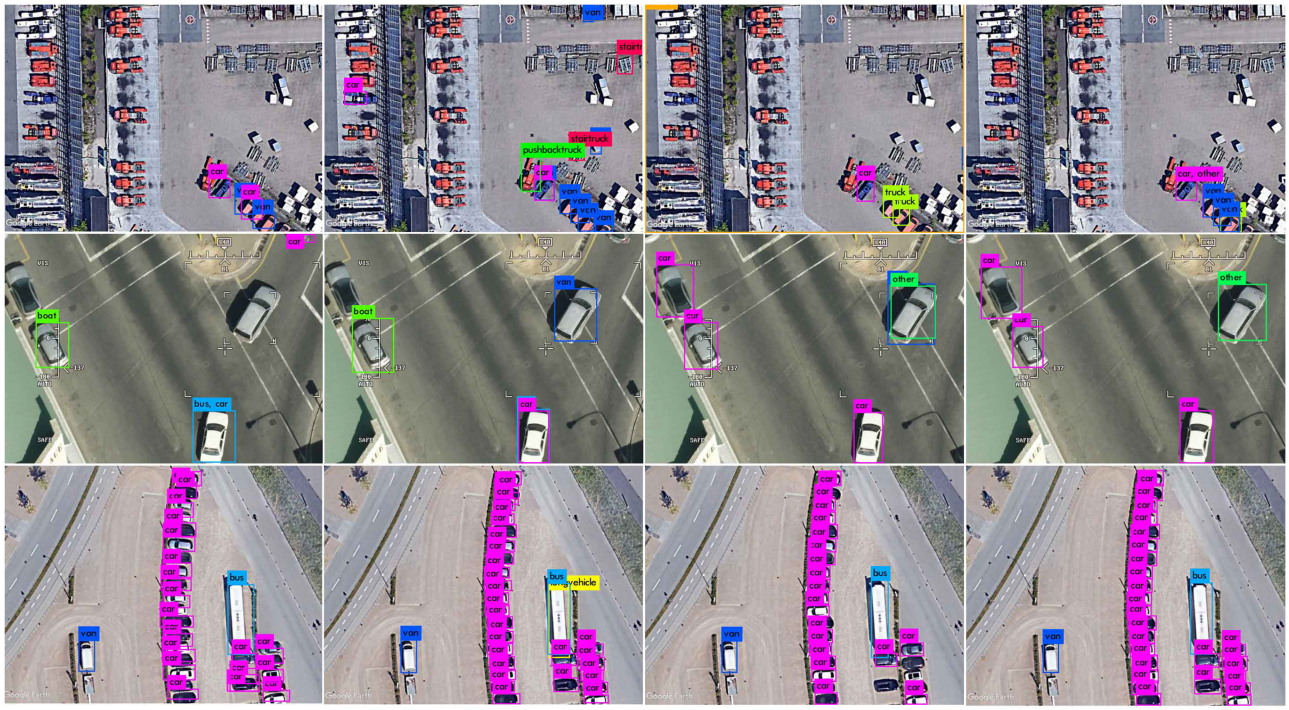


Fig. 9. Qualitative results are shown of four network models on same images. It is clearly shown that proposed model detects object accurately in simple as well as complex images. (i) YOLO v2. (ii) YOLT. (iii) YOLO v3. (iv) Proposed.

of aerial imagery, hence contains research prospects which need to be explored to expand and formalize available images dataset.

C. Fine-Grained Classification

One of the main reasons for less accuracy in some cases is wrong classification in our case. A car looks similar to a van, hence, the network mistakenly detects many cars as vans and vice versa. Merging multiple datasets to one by increasing the number of classes and instances may be one possible solution. However, this may require more resources to process and train the network.

D. Deeper Networks

We have tried to add more layers and detection levels, however the results were not encouraging. As shown in Table IV, Initial-4 L could not work efficiently on small objects even with more layers and levels of detection. So we suggest that adding more layers will probably not improve accuracy of the system. Some other mechanisms and methods may be explored. One example of two-phase SDBN detection framework is demonstrated in [45] where first pass generates candidate regions and second pass verifies detects object categories. Such methods need to be evaluated on accuracy versus speed on real-time applications.

E. Multi/Hyperspectral Imagery

Finally, it is worth mentioning that many studies rely on using multi/hyper spectral imagery for improved feature extraction and have demonstrated very good results particularly to solve the problem of land cover classification [41], [51] –[53]. Although these techniques are aimed at segmenting out stationary objects (such as buildings, roads, vegetation, etc.), a possible future study could be to use them for nonstationary (i.e., moveable) object detection.

VII. Conclusion

In this article, we first introduced a dataset of horizontal annotated satellite images of handful instances around 45 096 with the name of SIMD. Second, a one way forward pass object detection framework has been introduced which work excellent on object detection task on aerial imagery. Our proposed architecture use features extracted at initial levels to detect objects at for different scales to cater object sizes from small to large. Our proposed architecture predict object types and bounding boxes on two datasets: VEDAI [7] and SIMD. We believe that our trained model can be used for real time surveillance applications as demonstrated by Amanatiadis *et al.* [35] who proposed a surveillance detection system for various fire detection, and SAR operations. In future, we look forward to use our research work in development of real-time applications such as ATR, vehicle presence detection, traffic flow management, parcel delivery system, and autonomous drone landing.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 580–587.

[2] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 21–37.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2980–2988.

[6] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 779–788.

[7] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Representation*, vol. 34, pp. 187–203, 2016.

[8] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 785–800.

[9] G.-S. Xia *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3974–3983.

[10] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[11] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[12] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vision*, 2016, vol. 9912, pp. 549–565.

[13] J. Carlet and B. Abayowa, "Fast vehicle detection in aerial imagery," 2017, *arXiv:1709.08666*.

[14] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, Aug. 2016.

[15] S. Zhuang, P. Wang, B. Jiang, G. Wang, and C. Wang, "A single shot framework with multi-scale feature fusion for geospatial object detection," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 594.

[16] M. Yang, W. Liao, X. Li, Y. Cao, and B. Rosenhahn, "Vehicle detection in aerial images," *Photogrammetric Eng. Remote Sens.*, vol. 85, pp. 297–304, 2019.

[17] X. Liu, T. Yang, and J. Li, "Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network," *Electronics*, vol. 7, no. 6, 2018. [Online]. Available: https://www.mdpi.com/2079-9292/7/6/78

[18] G. Robinson and D. Zhang, "Object detection in wide area aerial surveillance imagery with deep convolutional networks," 2016. [Online]. Available: https://www.semanticscholar.org/paper/Object-Detection-in-Wide-Area-Aerial-Surveillance-Robinson/9651176a393302d9343a24beae4d7e6cb45b4e2b

[19] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2065–2070.

[20] B. Zou, X. Xu, and L. Zhang, "Object-based classification of polsar images based on spatial and semantic features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 609–619, 2020.

[21] W. Sakla, G. Konjevod, and T. N. Mundhenk, "Deep multi-modal vehicle detection in aerial ISR imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2017, pp. 916–923.

[22] X. Liu, T. Yang, and J. Li, "Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network," *Electronics*, vol. 7, no. 6, 2018, Art. no. 78.

[23] H. Tayara and K. Chong, "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network," *Sensors*, vol. 18, 2018, Art. no. 3341.

[24] H. Tayara, K. G. Soo, and K. T. Chong, "Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network," *IEEE Access*, vol. 6, pp. 2220–2230, 2017.

[25] T. Tang, S. Zhou, Z. Deng, L. Lei, and H. Zou, "Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1170.

[26] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "Orsim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.

[27] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.

[28] Y. Koga, H. Miyazaki, and R. Shibasaki, "A CNN-based method of vehicle detection from aerial images using hard example mining," *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 124.

[29] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors*, vol. 17, no. 2, 2017, Art. no. 336.

[30] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sens.*, vol. 9, no. 4, 2017, Art. no. 368.

[31] Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng, "Multilabel remote sensing image retrieval based on fully convolutional network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 318–328, 2020.

[32] Z. Hu, D. Yang, K. Zhang, and Z. Chen, "Object tracking in satellite videos based on convolutional regression network with appearance and motion features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 783–793, 2020.

[33] T. Yang *et al.*, "Small moving vehicle detection in a satellite video of an urban area," *Sensors*, vol. 16, no. 9, 2016, Art. no. 1528.

[34] N. Ammour, H. Hichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, pp. 1–15, 2017.

[35] A. Amanatiadis, L. Bampis, E. G. Karakasis, A. Gasteratos, and G. Sirakoulis, "Realtime surveillance detection system for mediumaltitude longendurance unmanned aerial vehicles," *Concurrency Comput., Pract. Experience*, vol. 30, pp. 1–14, 2017.

[36] J. Terrail and F. Jurie, "Faster RER-CNN: Application to the detection of vehicles in aerial images," 2017, *arXiv:1809.07628*.

[37] L. Sommer, A. Schumann, T. Schuchert, and J. Beyerer, "Multi feature deconvolutional faster R-CNN for precise vehicle detection in aerial imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2018, pp. 635–642.

[38] S. W. Zamir *et al.*, "ISAID: A large-scale dataset for instance segmentation in aerial images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2019, pp. 1–10.

[39] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning ROI transformer for detecting oriented objects in aerial images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2018, pp. 2849–2858.

[40] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2019, pp. 6054–6063.

[41] Q. Li, L. Mou, Q. Xu, Y. Zhang, and X. X. Zhu, "R3-net: A deep network for multioriented vehicle detection in aerial images and videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5028–5042, Jul. 2019.

[42] X. Chen, S. Xiang, C. Liu, and C. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[43] X. Zhang *et al.*, "Geospatial object detection on high resolution remote sensing imagery based on double multi-scale feature pyramid network," *Remote Sens.*, vol. 11, 2019, Art. no. 755.

[44] X. Yang *et al.*, "SCRDET: Towards more robust detection for small, cluttered and rotated objects," in *Proc. Int. Conf. Comput. Vision*, 2019, pp. 8232–8241.

[45] X. Chen, J. Lin, S. Xiang, and C.-H. Pan, "Detecting maneuvering target accurately based on a two-phase approach from remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 5, pp. 849–853, May 2020.

[46] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io

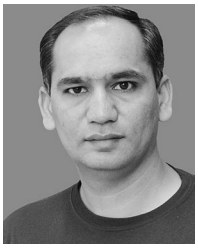
[47] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 3735–3739.

[48] J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," 2018, *arXiv:1804.02767*.

[49] A. V. Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*.

[50] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, "Towards multi-class object detection in unconstrained remote sensing imagery," in *Asian Conference of Computer Vision*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Berlin, Germany: Springer, 2019, pp. 150–165.

[51] D. Hong, N. Yokoya, J. Chanussot, and X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.
[52] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "CoSpace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.
[53] B. Rasti *et al.*, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep," *IEEE Geosci. Remote Sens. Mag.*, to be published, doi: 10.1109/MGRS.2020.2979764.



Muhammad Haroon received the bachelor's degree in computer science from Hajvery University, Lahore, Pakistan, in 2003, and the master's degree in computer science from the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan, in 2020, respectively. He is currently working toward the master's thesis research using deep learning and computer vision in the domain of intelligent surveillance systems.

He has more than 12 years' experience of software development in multiple technologies. As Team Lead, he is currently working on embedded systems with satellite imagery. The area of his research is aerial surveillance where he is focusing on problem of vehicle detection and identification from spaceborne and aerial platforms.

Muhammad Shahzad received the B.E. degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2004, the M.Sc. degree in autonomous systems (robotics) from the Bonn Rhein Sieg University of Applied Sciences, Sankt Augustin, Germany, in 2011, and the Ph.D. degree in radar remote sensing and image analysis from the Department of Signal Processing in Earth Observation (SiPEO), Technische Universität München (TUM), Munich, Germany, in 2016. His Ph.D. topic was automatic 3-D reconstruction of objects from point clouds retrieved from spaceborne synthetic-aperture-radar (SAR) image stacks. He has also attended twice two-weeks professional thermography training course at Infrared Training Center (ITC), North Billerica, MA, USA, in 2005 and 2007.

He has also worked as a Guest Scientist with the Institute for Computer Graphics and Vision, Technical University of Graz, Austria, from November 2015 to January 2016. Since October 2016, he has been working as an Assistant Professor with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan. His research interests include processing both unstructured/structured 3-D point clouds, optical RGBD images, and very high-resolution radar data.

Muhammad Moazam Fraz received the B.S. and M.S. degrees in software engineering, in 2008 and 2003, respectively, and the Ph.D. degree in computer science from the Faculty of Science Engineering and Computing, Kingston University London, London, U.K., in 2013.

After completing the Ph.D. degree, he has worked as a Postdoc Research Fellow with Kingston University in collaboration with St George's University of London and U.K. BioBank on development of an automated software tool for epidemiologists to quantify and measure retinal vessels morphology and size; determine the width ratio of arteries and veins as well as the vessel tortuosity index on very large datasets, to enable them to link systemic and cardiovascular disease to the retinal vessel characteristics. Since June 2014, he is working as an Assistant Professor with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan. His research area include the application of machine learning/computer vision techniques for diagnostic retinal image analysis. He has started his career in 2003 as a Software Development Engineer with Elixir Technologies Corporation, a California based software Company. He has served with Elixir until 2010 at various roles and capacities including software developer, development manager, and program manager.

Dr. Moazam Fraz was the recipient of two Gold Medals for "Best Graduate Award" and "Securing Top Position in the batch." He is a PMI (www.pmi.org) certified Project Management Professional. Besides, he also remained Rutherford Visiting Fellow at The Alan Turing Institute, United Kingdom during 2018–2019. His Ph.D. thesis was nominated for IET excellence Award 2013.