01 Jan 2023

An Effective Transfer Learning Based Landmark Detection Framework For UAV-Based Aerial Imagery Of Urban Landscapes

Bishwas Praveen

Vineetha Menon

Tathagata Mukherjee

Bryan Mesmer

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/engman_syseng_facwork/895

Recommended Citation

B. Praveen et al., "An Effective Transfer Learning Based Landmark Detection Framework For UAV-Based Aerial Imagery Of Urban Landscapes," *Conference Proceedings - IEEE SOUTHEASTCON*, pp. 844 - 850, EDP Sciences; Société le Mathématiques Appmuquéco et Irameworkustriellco, Jan 2023.
The definitive version is available at https://doi.org/10.1109/SoutheastCon51012.2023.10115176

An Effective Transfer Learning Based Landmark Detection Framework for UAV-Based Aerial Imagery of Urban Landscapes

Bishwas Praveen
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL, USA
bp0052@uah.edu

Vineetha Menon
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL, USA
vineetha.menon@uah.edu

Tathagata Mukherjee
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL, USA
tathagata.mukherjee@uah.edu

Bryan Mesmer
Department of Industrial & Systems Engineering and Management
University of Alabama in Huntsville
Huntsville, AL, United States
bryan.mesmer@uah.edu

Sampson Gholston
Department of Industrial & Systems Engineering and Management
University of Alabama in Huntsville
Huntsville, AL, United States
sampson.gholston@uah.edu

Steven Corns
Department of Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, MO, USA
cornss@mst.edu

SoutheastCon 2023 | 978-1-6654-7611-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SoutheastCon51012.2023.10115176

***Abstract*— Aerial imagery captured through airborne sensors mounted on Unmanned Aerial Vehicles (UAVs), aircrafts, satellites, etc. in the form of RGB, LiDAR, multispectral or hyperspectral images provide a unique perspective for a variety of applications. These sensors capture high-resolution images that can be used for applications related to mapping, surveying, and monitoring of crops, infrastructure, and natural resources. Deep learning based algorithms are often the forerunners in facilitating practical solutions for such data-centric applications. Deep learning-based landmark detection is one such application which involves the use of deep learning algorithms to accurately identify and locate landmarks of interest in images captured through UAVs. This study proposes an efficient transfer learning method for feature extraction using a ResNet50 architecture, paired with a FasterRCNN object detection for an automated landmark detection framework. Additionally, a novel technique for hierarchical image annotation and synthetic sampling is also introduced to address the issue of class imbalance. Empirical results prove that our proposed approach outperforms other state-of-the-art landmark detection methodologies compared.**

***Index Terms*—landmark detection, transfer learning, ResNet50, FasterRCNN, deep learning, aerial imagery**

I. Introduction

Aerial imagery refers to images that are captured from a high altitude, often from air or space. These images can be captured through a variety of sources, including aircrafts, unmanned aerial vehicles (UAVs) and satellites. UAVs, also known as drones, are small aircraft that can be remotely piloted or operated autonomously and are equipped with sensors such as vision (RGB) cameras and LIDAR sensors. Satellites, on the other hand, are spacecraft that orbit the earth and are equipped with multi-sensors (vision, hyperspectral, multispectral) and cameras. Both UAVs and satellites can capture high-resolution images and data that can be used for a variety of applications, such as mapping, surveying, and monitoring crops [1], war zones [2], landscapes [3], infrastructure [4], and natural resources [5]. The use of aerial imagery captured through UAVs and satellites has become increasingly popular in recent years due to their ability to provide a unique perspective and access hard-to-reach areas.

Over the past decade, deep learning has had a significant impact on land cover surveillance and landmark detection [6]. Land cover surveillance and landmark detection involves monitoring, mapping and identifying significant landmarks on the land surface to assess its use and condition. Deep learning algorithms can effectively be used to analyze aerial imagery to automatically classify land cover types [7]–[9] and detect landmarks of interest present in them. In literature, there are several deep learning algorithms that can be specifically used for landmark detection or object detection in general, including, convolutional neural network (CNN) [10], region-based convolutional neural network (R-CNN) [11], Fast-RCNN which is an extension of R-CNN [12], You Look Only Once (YOLO) [13], single shot detector (SSD) [14]. In addition to these object detection algorithms, the feature extraction module at the backend of the overall architecture plays a significant role, not only aiding in extracting principal features which are cardinal to effectively distinguish objects of interest present in the input data but also affect the computational complexity

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-21-2-0266. We would also like to express our gratitude to Army DAC for their invaluable support throughout the completion of this project. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

generally measured in terms of overall execution time directly.

In our work, we propose the use of FasterRCNN object detection architecture for effective landmark detection with a pre-trained ResNet50 architecture on the backend for feature extraction of various landmark classes used in the dataset used for empirical analysis. The ResNet50 architecture on the backend is pre-trained on COCO17 dataset [15]. Additionally, as a part of preprocessing, we propose a novel hierarchical landmark annotation technique for the 'housing community' class identified in our dataset. In this hierarchical annotation technique, the first stage involves annotation of all the identified 'buildings' as a single class separately, which is followed by grouping together a cluster of buildings that are in spatial proximity together as a 'housing community' class. This aids the landmark detection algorithm in hierarchically learning to map a set of buildings which look alike and are present in a spatial cluster as a housing community. Finally, synthetic sampling of entities for four classes identified in our dataset ('housing community', 'football stadium', 'baseball stadium' and 'waterbody') is also introduced in this work to alleviate any class imbalance issues and significantly enhance the performance of our proposed technique.

The rest of the paper is structured as follows. Section II gives a detailed description of the feature extraction and landmark detection architecture proposed in our work. This is followed by a brief description of all the other methodologies used in our work for direct comparison with the proposed architecture for effectiveness in Section III. Section IV outlines the details of the dataset used for our experimentation and summarizes the effectiveness of our proposed landmark detection framework. Finally, Section V presents a synopses of our work and discusses future research directions.

II. Proposed System Architecture (FasterRCNN-ResNet50)

A. Image Annotation

The raw input tiles were pre-processed and broken down to image slices of size $(500 \times 500 \times 3)$ because training a neural network for landmark detection offers advantages such as automated faster training, reduced memory requirements, improved generalization, and produces superior results for generic target detection applications. The preprocessing of the raw image tiles is outlined in section IV-A. In this work, we are interested in 5 types of landmarks (classes) present in the captured data annotated as 'building', 'football stadium', 'baseball stadium', 'waterbody', and 'housing community'. The data annotation of the first 4 classes was straight forward. However, annotation of landmarks identified as housing community was complex due to its close association with the 'building' class. To combat this, we propose a hierarchical annotation methodology where the identified houses in a community are first included in the 'building' class. In the next step, all such buildings which are similar structurally and are in proximity to one another are amalgamated and annotated as a single 'housing community' entity. Hence, this form of hierarchical annotation can alleviate any ambiguity in landmark detection, especially in cases of buildings that are occluded or partially visible. It can also handle cases where multiple landmarks are closely linked or overlapping. The entire process of annotation was achieved through a software called LabelImg [16].

B. Synthetic Sampling

After the raw input image tiles were pre-processed and annotated for respective landmarks of interest, it was noticed that the ratio of the number of samples present in the 'building' class to the number of samples present in all the other four classes of interest was greater than $10:1$. This orchestrates the classical machine learning challenge of class imbalance issue which can lead to a bias in the model prediction towards the more prevalent classes. This form of biased data in turn leads to biased data learning which can lead to poor generalization of the AI/ML model and cause over-fitting. To mitigate this problem, 30 synthetic samples along with their vertical flipping and horizontal rotation based augmentations for each of the classes, 'building', 'football stadium', 'baseball stadium' and 'housing community' were introduced during the training to boost the generalization capability of the landmark detection framework used in our work. Examples of synthetic samples that were introduced for all the four classes are as shown in Fig. 2.

C. Feature Extraction and Landmark Detection

Here, the input annotated images are resized to a spatial resolution of (640×640) with a spectral dimension of 3 and are produced to a ResNet50 architecture pre-trained on COCO17 dataset for effective feature extraction. ResNet50 is a convolutions based architecture and has 50 layers, which allows it to learn complex and hierarchical features from the input images. This is important for object detection, as it allows the network to capture the details and nuances of the objects in the image. The resultant feature map that is generated as a result of feature extraction is produced as an input to the Region Proposal Network (RPN).

The RPN takes the generated feature map as the input and generates a set of region proposals, or potential landmark locations, in the input image. The RPN works by using a sliding window approach to scan the input image and generate a set of region proposals at each window position. The region proposals are then passed through a convolutional neural network (CNN) that classifies each proposal as either an object or background and also predicts the boundaries of the object within the proposal. The RPN uses a set of predefined anchor boxes or reference boxes to process objects of different sizes and shapes. These anchor boxes are chosen to cover a range of aspect ratios and scales, and the RPN uses them to generate region proposals that are adapted to the objects in the image. The region proposal r_i is defined by its coordinates (x_i, y_i, w_i, h_i), where (x_i, y_i) is the center of the proposal, w_i is the width, and h_i is the height. The probability that the region contains an object is given by p_i, and the bounding box regression targets are given by $t_i = (t_{x_i}, t_{y_i}, t_{w_i}, t_{h_i})$. The probability p_i is calculated as the product of the probability

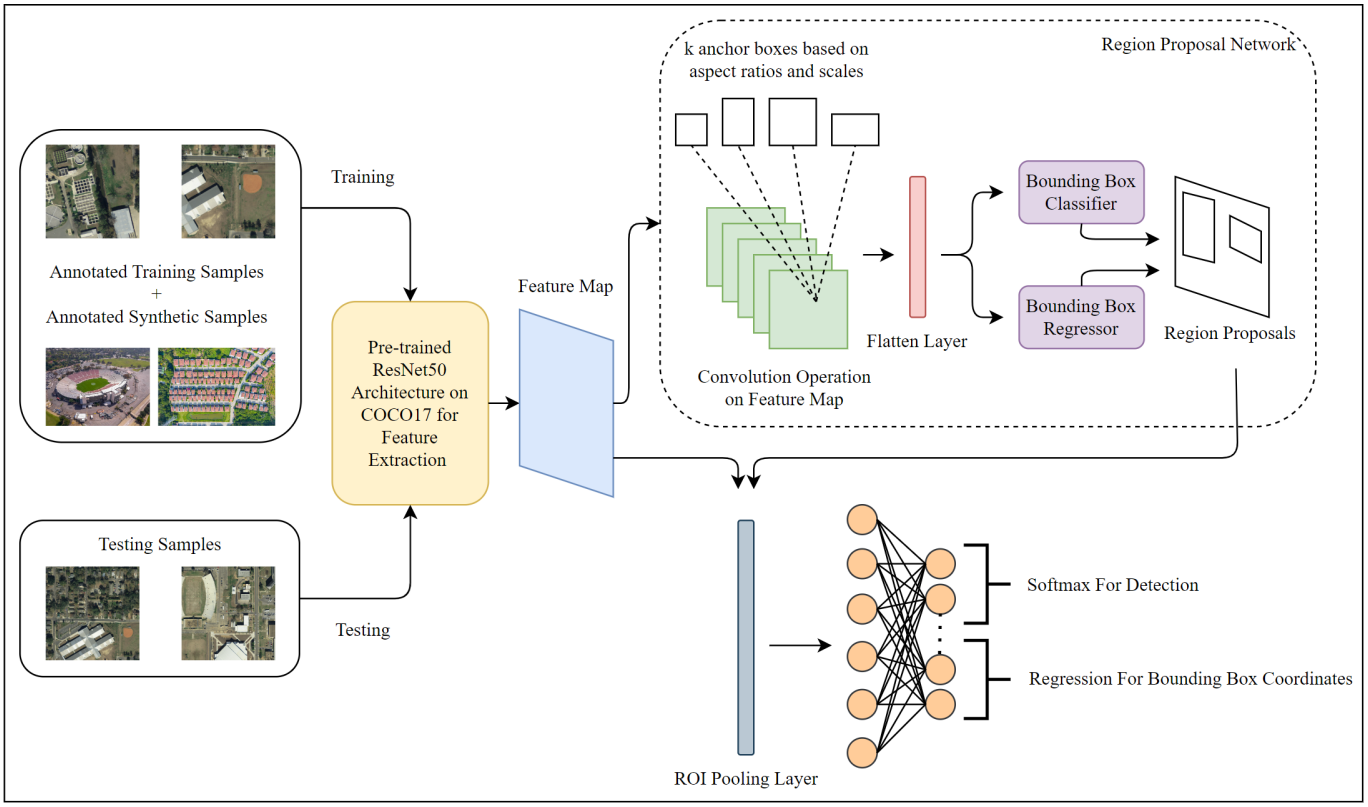


Fig. 1. The proposed pre-trained ResNet50 based feature extraction and FasterRCNN based landmark detection framework (FasterRCNN-ResNet50).

(a) Football Stadium (b) Baseball Stadium

(c) Housing Community (d) Waterbody

Fig. 2. Examples of synthetic samples for all the four classes introduced during training.

that the region contains an object, P_{obj}, and the intersection over union (IOU) between the ground truth bounding box g_i and the region proposal r_i as follows:

$$p_i = P_{\text{obj}} \cdot \text{IOU}(g_i, r_i) \tag{1}$$

The bounding box regression targets t_i are calculated as the difference between the ground truth bounding box g_i and the region proposal r_i, normalized by the size of the region proposal as:

$$t_i = \frac{g_i - r_i}{r_i} \tag{2}$$

The RPN uses a CNN to classify each region proposal as either an object or background, and to predict the bounding box regression targets. The CNN is trained to minimize the following loss function:

$$L = \frac{1}{N} \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda L_{\text{reg}}(t_i, t_i^*) \tag{3}$$

where L_{cls} is the classification loss, L_{reg} is the bounding box regression loss, p_i^* is the ground truth label for the region proposal, and t_i^* is the ground truth bounding box regression targets. The classification loss L_{cls} is typically the log loss, given by:

$$L_{\text{cls}}(p_i, p_i^*) = -(p_i^* \log(p_i) + (1 - p_i^*) \log(1 - p_i)) \tag{4}$$

The bounding box regression loss L_{reg} is typically the smooth L1 loss, given by:

$$L_{\text{reg}}(t_i, t_i^*) = \begin{cases} 0.5(t_{ij} - t_{ij}^*)^2 & \text{if } |t_{ij} - t_{ij}^*| < 1 \\ |t_{ij} - t_{ij}^*| - 0.5 & \text{otherwise} \end{cases} \tag{5}$$

where j indexes the components of the bounding box regression targets, t_{ij} is the predicted value, and t_{ij}^* is the ground truth value. The hyperparameter λ controls the relative importance of the classification loss and the bounding box regression loss. The mathematical representation of the working of a RPN is clearly explained from Eq. 1- 5.

After the RPN produces the region proposals, the output of the RPN, i.e., the region proposals are now together input to a Region Of Interest (ROI) pooling layer. ROI pooling works by dividing the region proposals into a set of fixed-size bins and then max pooling the activations within each bin. This has the effect of making the output of the CNN invariant to the size of the region proposals and allows the CNN to process region proposals of different sizes in a uniform way. To perform ROI pooling, the Faster-RCNN framework first takes the output of the ResNet50 feature extractor and applies a spatial transform to map it to the coordinate space of the region proposals. The transformed feature maps are then divided into a set of fixed-size bins and max pooled within each bin. The resulting pooled features are then fed into the detector, which processes them further to finally classify the objects and refine the boundary estimates using a fully connected neural network. The overall system architecture of the FasterRCNN-ResNet50 based feature extraction and landmark detection framework is as depicted in the Fig. 1.

III. Methodologies For Comparison

This section briefly discusses the two state-of-the-art landmark detection methodologies that are used for comparison in this work.

A. SSD-MobileNet

A single shot detector (SSD) is a light weighted object detection module that uses a single convolutions based network to predict both the bounding boxes and the respective class probabilities in an input image [14]. SSD strives towards making predictions with the aid of feature maps at multiple scales which results in successful identification of objects of various sizes in the given input. On the backend, SSD has the option of using multiple convolutional neural network based feature extractors. However, in our experimentation, to ensure the overall landmark detection module stays light-weight for UAVs, a MobileNet based feature extraction module is included [17]. It uses depthwise separable convolutions, which split the convolution operation into two separate stages: a depthwise convolution and a pointwise convolution. This reduces the number of trainable parameters by a great extent. It also ensures that the overall feature extraction module computationally efficient.

In our experimentation, the SSD model with MobileNet on the back end works by first resizing the input image to a spatial dimension of (320×320) and passing it through the MobileNet convolutional neural network to generate feature maps. These feature maps are then fed into several layers of convolutional and predictor layers, which predict the bounding boxes and class probabilities for the landmarks in the image. The bounding boxes are then filtered and refined using non-maximum suppression in the later stage to remove overlapping bounding boxes and ensure that each landmark is only detected once. Finally, the resulting bounding boxes and class probabilities for all the 5 classes in our dataset are used to identify and classify the landmarks in the image.

The hyperparameters for SSD-MobileNet were set as follows. The anchor box scales were defined to be 1.0 and 4.0 with aspect ratios of 0.5, 1.0 and 2.0. L2 regularization was used to reduce over-fitting and increase the generalization capability of the overall framework. The threshold for Intersection over Union (IOU) was set to 0.5 allowing a maximum of 100 predictions per input in this framework. The batch size was set to 4 and the entire network was trained for 35000 steps with a base learning rate of 0.8.

B. CenterNet-ResNet101

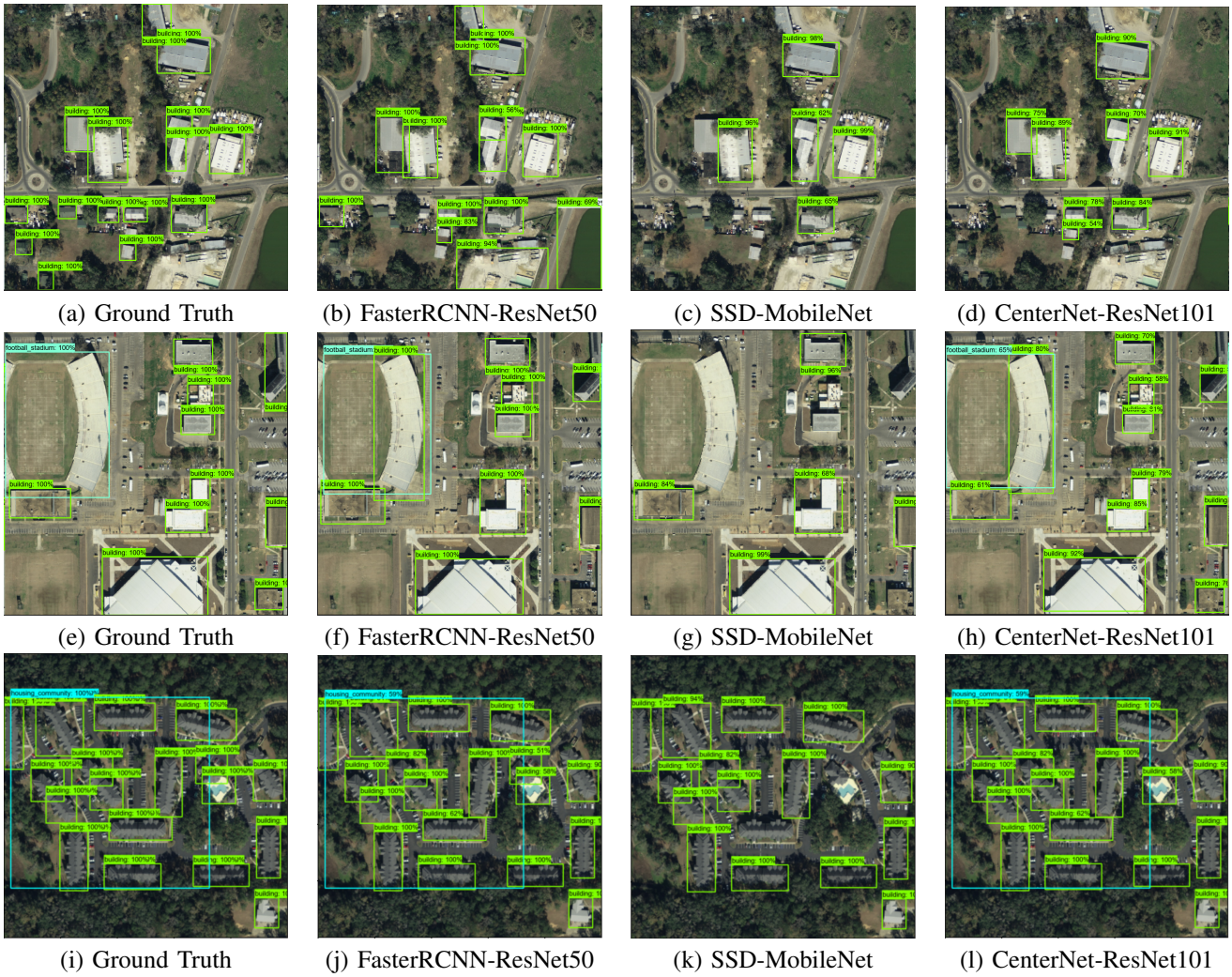
CenterNet is a single-stage object detection model that uses a convolution neural network to predict bounding boxes and class probabilities for landmarks identified in an image [18]. It is based on the idea of predicting the center point of an object, rather than predicting the bounding box directly. In our experiments, CenterNet model with a ResNet-101 backend works by first resizing the input image to a spatial dimension of (512×512) and then passing it through the ResNet-101 convolution neural network to generate feature maps. These feature maps are then fed into several layers of convolutional and predictor layers, which predict the center points and sizes of the bounding boxes for the landmarks detected in the image. The bounding boxes are then generated from the center points using a decoder network, which also predicts the class probabilities for each bounding box. The resulting bounding boxes and class probabilities are used to identify and classify the landmarks found in the image.

The hyperparameters for CenterNet-ResNet101 were set as follows. No explicit definition of anchor box scales and aspect ratios were made here and the pre-defined defaults were used. No regularization has been included in the experimentation related to this approach. The threshold for Intersection over Union (IOU) was set to 0.5 allowing a maximum of 100 predictions per input in this framework. The batch size was set to 4 and the entire network was trained for 25000 steps with a base learning rate of 0.001.

IV. Experimental Results

A. Data Description

A Light Detection and Ranging (LIDAR) sensor was mounted onto a drone and flown over the city of Tallahassee in Florida (USA). The raw input data is in the form of image tiles and the raw input data is in the shape of $(M \times N \times D)$, where $(M \times N)$ denote the spatial resolution of the captured



(a) Ground Truth (b) FasterRCNN-ResNet50 (c) SSD-MobileNet (d) CenterNet-ResNet101

(e) Ground Truth (f) FasterRCNN-ResNet50 (g) SSD-MobileNet (h) CenterNet-ResNet101

(i) Ground Truth (j) FasterRCNN-ResNet50 (k) SSD-MobileNet (l) CenterNet-ResNet101

Fig. 3. Test images demonstrating the results of all the landmark detection frameworks along with their ground truths with annotation, where ground truth (a) covers Buildings, (e) Football stadium and buildings, and (i) shows Housing community and buildings classes.

image tiles, and D denotes the number of spectral bands. In our case, the spatial resolution of the captured image tiles were (5000×5000) with a spectral dimension of 4. The additional fourth band present in the captured image tiles represents the LIDAR band which encodes the depth information present in the image captured. However, since the focus of this work is to perform landmark detection on aerial imagery which are three channeled (RGB), the LIDAR band was eliminated from the raw image tiles. Thus, each of the raw input image tiles were now in the shape of $(5000 \times 5000 \times 3)$.

In the next step, 9 tiles out of all the the raw data collected indicated the presence of urban landscape were handpicked for further processing. Each of these 9 high resolution image tiles were now split and saved as 100 images each of size $(500 \times 500 \times 3)$. We then eliminated any areas in our dataset that did not cover our targets-of-interest or landmarks we were looking for, which includes, 5 classes, namely, 'building', 'football stadium', 'baseball stadium', 'waterbody', and 'housing community'. The remaining dataset consisted of 393 images of shape $(500 \times 500 \times 3)$ was used for training our proposed landmark detection framework. For testing, 30 randomly selected images with urban landscape from the same region were chosen and each of those images were of shape $(1000 \times 1000 \times 3)$. Finally, both training and testing images created in our work were annotated with 5 classes, namely, 'building', 'football stadium', 'baseball stadium', 'waterbody', and 'housing community' using LabelImg [16].

B. Experimental Setup and Hyper-Parameter Tuning

This section covers the hyper-parameters used to train and validate our proposed FasterRCNN-ResNet50 architecture based landmark detection framework. Firstly, the original input data which is of size $(500 \times 500 \times 3)$ after pre-processing, is reshaped to size $(640 \times 640 \times 3)$ before passing it onto a ResNet50 based feature map generator. Later, within the region proposal network, the anchor box scales were set to 0.25, 0.5, 1.0 and 2.0 with aspect ratios 0.5, 1.0 and 2.0. L2 regularization has been used in this framework to effectively amplify the generalization capability of our landmark detec-

TABLE I
COMPARISON OF AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) (IN PERCENTAGE %) FOR VARIOUS RANGES OF IOU FOR DIFFERENT LANDMARK DETECTION METHODOLOGIES

Metric (Average Precision/Recall)	Area	Max Detections	FasterRCNN--ResNet50	SSD--MobileNet	CenterNet--ResNet101
AP @ IOU = 0.50 : 0.95	All	100	44.60	39.40	46.30
AP @ IOU = 0.50	All	100	75.80	47.72	74.20
AP @ IOU = 0.75	All	100	51.30	42.80	49.50
AP @ IOU = 0.50 : 0.95	Medium	100	22.00	6.40	38.20
AP @ IOU = 0.50 : 0.95	Large	100	47.90	41.80	47.50
AR @ IOU = 0.50 : 0.95	All	1	38.50	34.40	36.30
AR @ IOU = 0.50 : 0.95	All	10	55.60	51.50	56.60
AR @ IOU = 0.50 : 0.95	All	100	58.00	59.20	58.50
AR @ IOU = 0.50 : 0.95	Medium	100	28.40	15.70	52.70
AR @ IOU = 0.50 : 0.95	Large	100	61.80	61.80	59.20

TABLE II
COMPARISON OF CLASS-WISE AVERAGE PRECISION @ IOU = 0.50 IN PERCENTAGE (%) FOR ALL THE LANDMARK DETECTION METHODOLOGIES

Class Name	Metric : Average Precision @ IOU = 0.5 in percentage (%)		
	FasterRCNN-ResNet50	SSD-MobileNet	CenterNet-ResNet101
Building	9.24	3.41	6.07
Housing Community	50.00	50.78	100.00
Football Stadium	100.00	50.00	100.00
Waterbody	100.00	78.84	61.11
Baseball Stadium	66.67	55.55	79.16

TABLE III
OVERALL TRAINING AND EVALUATION TIME (IN MINUTES) FOR ALL THE LANDMARK DETECTION METHODOLOGIES

Metric	FasterRCNN-ResNet50	SSD-MobileNet	CenterNet-ResNet101
Overall Training Time	244.86	67.02	306.24
Overall Evaluation Time	0.005	0.0048	0.005

tion framework FasterRCNN-ResNet50. Additionally, the loss function used to minimize the classifier error is log loss and the bounding box regression loss used in this framework is the smooth L1 loss which are denoted by Eq. 4 and Eq. 5 respectively. The IOU threshold used at the end of the landmark detection framework to eliminate insignificant detections was set to 0.5 allowing a maximum of 100 detections per input image. The FasterRCNN-ResNet50 framework was trained for 40000 steps with a batch size of 4 and a base learning rate of 0.039 to optimize the landmark detection capabilities of this framework. The hyper-parameters used to train all the other landmark detection systems that have been discussed in our work have been clearly documented in section III. Two evaluation metrics have been used in our work to quantify and compare the performance of all the landmark detection techniques discussed. The first is the '*coco_api_detection_metric*' which produces the average precision and average recall for various ranges of IOU starting from 0.5 through 0.9, and the second is the '*pascal_voc_detection_metric*' which produces the class-wise average precision at 0.5 IOU for all the classes present in our dataset. Finally, all the experiments related to this work were conducted on a workstation with Intel(R) Core(TM) i7-7700 CPU with 32GB memory and a NVIDIA GeForce GTX 1060 GPU with 6GB memory.

C. Discussion

In this work, we validate the effectiveness of our proposed feature extraction and landmark detection technique FasterRCNN-ResNet50, with respect to two other well-known landmark detection methodologies, namely, SSD-MobileNet and CenterNet-ResNet101. It can be clearly noted from Table I that FasterRCNN-ResNet50 approach performed exceptionally well when compared to the other landmark detection frameworks discussed in our work. The FasterRCNN-ResNet50 approach produced an average precision of 75.80% when the IOU parameter was set to 0.5, and produced an average recall of 58.00% for an IOU range of 0.5 through 0.95 $(0.5 : 0.95)$, irrespective of the size of various landmarks present in the input data. Additionally, the landmark detection performance of FasterRCNN-ResNet50 can be qualitatively (visually) confirmed from Fig. 3 where the images in the left-most column show the ground truth annotations on few of the images used for validation and the other images denote the actual landmarks identified by the proposed FasterRCNN-ResNet50 architecture during validation along with other state-of-the art techniques discussed in our work. It can be verified from these images that the proposed FasterRCNN-ResNet50 based landmark detection system performs exceedingly well in identification of various landmarks of interest in our work

when compared. Table II shows the class-wise average precision when IOU was set to 0.5 for all the classes present in our dataset. It can be observed that the results produced by our proposed approach overshadows the performance of all other techniques in comparison.

It is also worthwhile mentioning that the CenterNet-ResNet101, which is used as a methodology of comparison with FasterRCNN-ResNet50 performs extremely well in many of the above mentioned cases. However, the overall training time necessary to optimize the performance CenterNet-ResNet101 architecture, as observed in Table III, is much higher when compared to all the other methodologies discussed in our work, which makes it computationally very expensive. With the goal of having a good trade-off between performance and overall computational complexity, FasterRCNN-ResNet50 turns out to be the ideal approach in our case.

Additionally, it can also be observed from Table II and Figs. 3b, 3c, and 3d that all landmark detection methodologies faced difficulties in identifying landmarks under the 'building' category in our dataset, due to their varying small spatial sizes. Small landmarks may be difficult to be detected if the resolution of the input images is too low. In which case, these landmarks may appear as just a few pixels in the image, which may not contain enough information for the model to make an accurate prediction. Another possibility is occlusion. Small landmarks of interest may be difficult to detect if they are occluded by other objects or due to shadow effects in the image. This is because the model may not be able to see the entire object, which can make them difficult to identify. Albeit all the methodologies discussed in our work may not identify small buildings as expected, our proposed system FasterRCNN-ResNet50 did perform much better in this scenario comparatively.

V. Conclusion

In this work, a novel transfer learning framework using FasterRCNN with a ResNet50 architecture based feature extractor technique for landmark detection was introduced for UAV-based aerial imagery of urban landscapes. ResNet50 on the backend acts as an efficient feature extractor that is capable of extracting unique landmark features for landmark detection tasks, which often requires to learn and recognize a wide variety of different landmark classes and their variations. FasterRCNN on the other hand is fine-tuned in our work to be efficient in terms of both computation and memory. This allows the combination of FasterRCNN with ResNet50 to achieve an excellent trade-off between computational complexity and performance. When compared to other two-staged landmark detection methodologies discussed in our work, namely, SSD-MobileNet and CenterNet-ResNet101, our approach produced exceptional landmark detection results while being robust in limited training sample scenarios, thus paving path for new research direction in landmark detection for sensor-based aerial imagery.

References

[1] P. M. Taufiq, S. Kim, S. Ozawa, T. Ohkawa, Y. Chona, H. Tsuji, and N. Murakami, "Deep learning-based object detection for crop monitoring in soybean fields," International Joint Conference on Neural Networks (IJCNN), pp. 1–7, 2020.

[2] M. M. Anwar, M. K. Arrofi, G. Jati, F. Arifin, M. N. Kurniawan, P. Mursanto, and W. Jatmiko, "Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance," International conference on advanced computer science and information systems (ICACSIS), pp. 161–166, 2013.

[3] Z. Xin, L. Han, L. Han, and L. Zhu, "How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery?," Remote Sensing, vol. 12, no. 3, pp. 417, 2020.

[4] B. Zhengwei, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Infrastructure-based object detection and tracking for cooperative driving automation: A survey," IEEE Intelligent Vehicles Symposium (IV), pp. 1366–1373, 2022.

[5] W. S. David, J. T. Finn, and J. Finn, "Remote sensing imagery for natural resources monitoring: a guide for first-time users", Columbia University Press, 1996.

[6] G. Jan, M. Knapik, and B. Cyganek, "An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance," Integrated Computer-Aided Engineering, vol. 28, no. 3, pp. 221–235, 2021.

[7] B. Praveen, and V. Menon, "HYPER-VIT: A novel light-weighted visual transformer-based supervised classification framework for hyperspectral remote sensing applications," 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–5, 2022.

[8] B. Praveen, and V. Menon, "Dual-Branch-AttentionNet: A Novel Deep-Learning-Based Spatial-Spectral Attention Methodology for Hyperspectral Data Analysis," Remote Sensing, vol. 14, no. 15, pp. 3644, 2022.

[9] B. Praveen and V. Menon, "A bidirectional deep-learning-based spectral attention mechanism for hyperspectral data classification," Remote Sensing, vol. 14, no. 1, pp. 217, 2022

[10] G. Jiuxiang, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, et. al, "Recent advances in convolutional neural networks," Pattern recognition, vol. 77, pp. 354–377, 2018.

[11] G. Ross, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 1, pp. 142–158, 2015.

[12] G. Ross, "Fast r-cnn," IEEE international conference on computer vision, pp. 1440-1448, 2015.

[13] R. Joseph, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

[14] L. Wei, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," European conference on computer vision, pp. 21–37, 2016.

[15] T. Lin, M. Maire, s. J. Belongie, L. D. Bourdev, R. B. Girshick, and J. Hays, "Microsoft COCO: Common Objects in Context," CoRR [Internet], 2014, Available from: http://arxiv.org/abs/1405.0312.

[16] Tzutalin, "LabelImg," Free Software: MIT License, 2015.

[17] G. H. Andrew, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[18] D. Kaiwen, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," IEEE/CVF international conference on computer vision, pp. 6569–6578, 2019.

[19] T. Mingxing, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," IEEE/CVF conference on computer vision and pattern recognition, pp. 10781–10790, 2020.