Object Detection in Deep Surveillance

Narina Thakur

Bharati Vidyapeeth's College of Engineering New Delhi

Preeti Nagrath (✉ preeti.nagrath@bharativedyapeeth.edu)

Bharati Vidyapeeth's College of Engineering New Delhi <https://orcid.org/0000-0002-5863-3113>

Rachna Jain

Bharati Vidyapeeth's College of Engineering New Delhi

Dharmender Saini

Bharati Vidyapeeth's College of Engineering New Delhi

Nitika Sharma

Bharati Vidyapeeth's College of Engineering New Delhi

Jude Hemanth

Karunya Institute of Technology and Sciences

Research Article

Keywords: Object Detection, Surveillance, Computer vision, UAVs, Yolov5, Pedestrian Detection

Posted Date: November 3rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-901583/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Object Detection in Deep Surveillance

**Narina Thakur¹, Preeti Nagrath¹, Rachna Jain¹, Dharmender Saini¹, Nitika Sharma¹,
D. Jude Hemanth²**

¹Department of Computer Science and Engineering,
Bharati Vidyapeeth's College of Engineering, New Delhi, India

²Department of Electronics and Communication Engineering,
Karunya University, Coimbatore, Tamil Nadu, India

Abstract. Object detection is a key ability required by most computer visions and surveillance applications. Pedestrian detection is a key problem in surveillance, with several applications such as person identification, person count and tracking. The number of techniques to identifying pedestrians in images has gradually increased in recent years, even with the significant advances in the state-of-the-art deep neural network-based framework for object detection models. The research in the field of object detection and image classification has made a stride in the level of accuracy greater than 99% and the level of granularity. A powerful Object detector, specifically designed for high-end surveillance applications, is needed that will not only position the bounding box and label it but will also return their relative positions. The size of these bounding boxes can vary depending on the object and it interacts with the physical world. To address these requirements, an extensive evaluation of the state-of-the-art algorithms has been performed in this paper. The work presented in this paper performs detections on MOT20 dataset using various algorithms and testing on a custom dataset recorded in our organization premises using an Unmanned Aerial Vehicle (UAV). The experimental analysis has been performed on Faster-RCNN, SSD and YOLO models. The Yolov5 model is found to outperform all the other models with 61% precision and 44% of F measure value.

Keywords: Object Detection, Surveillance, Computer vision, UAVs, Yolov5, Pedestrian Detection

1 Introduction

Surveillance is a time-consuming task that entails collecting and analyzing massive amounts of visual evidence. Data collection is a major failure point of any monitoring scheme. Monitoring cameras (CCTV) can only see people within the sight line of the camera. Therefore, criminals may decide when or where they have not been tracked. In terms of preventing and tracking, standalone security cameras have drawbacks. The use of UAVs for surveillance becomes essential for the reliability of the system. In recent years, computer vision technology has developed at a fast and promising pace. Part of this success may be attributed to the introduction and application of Machine Learning and Deep Learning methods, while others can be attributed to the development of novel representations and templates for real-world computer vision challenges, or the development of effective solutions. [1], [2].

Object Detection is one field that has accomplished tremendous progression. Object detection is a problem with computer vision for identifying and finding an object in a picture frame. In computer vision, the more comprehensive testing areas include person detection, facial detection, and recognition of human activity, etc. [3]. The object detector is aimed at detecting all object instances of one or more groups of objects independent of the size, location, position, camera view [4], partial occlusions [5] and lighting conditions [6], [7].

Object detection is the initial task performed in many computer vision systems because it allows further information about the identified object and the scene to be obtained. Additional information can be gathered once an object instance has been identified, such as identifying the instance location, monitoring the object over an image set, and collecting additional information about the object, as well as determining the presence or placement of additional objects in the scene and better estimating additional scene information along with other contextual information. Human-Computer Interaction (HCI) [8], [9], [10], consumer electronics [11], robotics [12], transportation and surveillance [13], [14] etc. are some of the most popular applications of Object detection [15].

Processing time, robustness to occlusions, invariance to rotations, and identification under pose shifts are all different criteria for all these applications. Although several applications detect a single class of objects from a single view, some need to detect a single class from multiple views or multiple classes of objects from multiple views. Image detection and motion prediction are in general two main building blocks of video monitoring applications. The first and most critical step is object identification, which is directly determined by the contextual information. Since video data in surveillance applications comprises an overwhelming amount of unnecessary and redundant data, it must be compressed in the very early stages of data-processing.

A reliable person beings identification is important in visual security system for a wide range of applications, including suspicious incident detection [16], human motion characterization [17], traffic or crowd analysis, individual recognition, gender classification, and fall detection for the elderly[18]. Object detection has been accomplished through background subtraction, optical flow, and spatiotemporal filtration methods.

Drones mounted with cameras have been introduced and deployed to a variety of applications such as to detect anomalies, cover blind-spots, and identify trespassers during the night, etc. [19]. An automated and efficient object recognition plays an important role in the perception and study of the visual data obtained from drones, which could be further extended to civilian and military sectors. Researchers can capture aerial (bird view) photographs from an acceptable altitude using UAVs fitted with mounted cameras. The viewpoint of aerial images allows the presence of objects to squat, while aerial photos can provide more contextual details about the world from a wider viewing angle, the instances of objects can be identified by accident. The pedestrian UAV-based identification model is split into conventional and in-depth learning models [20]. The conventional manual detection model focuses on the nature of pedestrian methods for collecting functions. But most algorithms only consider the application environments. These extraction methods are problematic for complicated real scenarios, the target detection accuracy is too poor, and not ideal for more precise pedestrian detection. In current human identification benchmarks, crowd conditions are still significantly under - represented, to further complicate matters [21], [22]. The occlusal question of distinguishing people is also a long way from being answered in heavily populated areas.

This paper is aimed at constructing an autonomous pedestrian detection framework for city surveillance using a drone. The state-of-the-art deep learning algorithms have been trained on the

MOT20 dataset [23] and tested on the custom videos captured by the drone for pedestrian detection. The deep learning algorithms, helps in distinguishing the various objects on the streets that further can be used for detecting crimes like harassment, as well as for the video proof of street accidents, traffic violations etc. The designed drone quadcopter can hold the camera with a gimbal, servos to move the rotors, servos for control surfaces, Wi-Fi module for handling communications and any other miscellaneous tasks like long range video transmission module, video transmission signal repeater, MAV links for long range communication between the drone and the server.

YOLO deep learning architectures were used for both identification and classification purposes. Faster-RCNN, SSD, and YOLO algorithms were implemented in this research work, along with experimental findings from various field experiments and simulations.

The paper is further organized as: Section 2 presents related work discussing the trends, models and challenges in pedestrian detection. Section 3 details the methodology used to perform the training and testing of object detection models. The experiments performed as described in section 4 and the discussion on the corresponding results obtained is discussed in section 5 followed by conclusion in section 6.

2 Related work

This paper addresses Object detection of pedestrians in crowded scenarios. This work demonstrates an object detection system in a typical crowded scenario with a state-of-the-art MOT20 dataset in which sequences are shot from a single, static camera with a heightened viewpoint. We will apply the trained method to the challenging task of detecting and localizing other pedestrians from a moving camera mounted on a drone.

2.1 Pedestrian detection

Pedestrian detection is a major concern in computer vision, with numerous applications that have the potential to improve the quality of life. The range of strategies to detect pedestrians in images has gradually increased in recent years. Nevertheless, many data sets and multiple evaluation protocols are used that make precise comparisons difficult [24]. In this field of machine vision, publicly accessible benchmarks, the most common of which is the INRIA dataset, has led to stimulate interest and advancement [25]. Caltech Pedestrian Dataset [26], which is twice as large than the previously existing dataset provides richly annotated video, taken from a moving car, with complex low-resolution images and occluded persons. The Crowd Human dataset [27] is vast, well-annotated, and diverse. There are 470K individual instances in the dataset, with 22.6 persons per image and various types of occlusions, from the train and validation subsets. Each human case has a head bounding-box, a human visible-region bounding-box, and a human full-body bounding-box. The simple output of cutting-edge Crowd Human identification algorithms is provided. While significant progress has been achieved in pedestrian recognition [28], [29], identification in congested settings remains difficult. The conventional Non-Maximum Suppression (NMS) [30] has significant problems due to the severe occlusion of pedestrians. A low intersection-over-union (IoU) threshold indicates the lack of substantially overlapping pedestrians, whereas a larger one results in more false positives [31]. The pedestrian detection has been observed with various facets by the researchers such as moving objects-static camera and moving-object - moving-camera. General approach for moving-object-static-camera is background subtraction, whereas it is difficult to detect objects while the capturing device is in motion as the background consistently changes [32].

Convolution networks are made up of Convolution layers, Pooling layers, and a final component, which is a completely linked or expanded item that will be utilized for a certain job like as classification or detection.

Figure 1 Convolution Neural Network

The output of convolution is a two-dimensional matrix called feature map, which is generated by sliding filters all over the input picture.

2.2 Detection methods

Object detection is a method for detecting and locating objects in an image or video. Using this type of identification and localization, object detection may be used to count objects in a scene, assess and monitor their precise locations, and accurately mark them. Two types of domain-specific image object detectors are widely used: two-stage detectors, such as Faster R-CNN [33], and single-stage detectors like YOLO and SSD are another choice. Single-stage detectors achieve a high degree of inference, whereas two-stage detectors have high object location and precision. SSD discretizes the output space of bounding boxes per function map position into a set of default boxes with varying aspect ratios and sizes. At prediction time, the network calculates scores for each object category's appearance in each default box, and then changes the box to best fit the object structure. In addition, the network uses predictions from numerous

feature maps with varied resolutions to seamlessly handle objects of various sizes. Since it fully removes proposal generation and subsequent pixel or feature resampling and encapsulates all computation in a single network, the SSD model is easy in comparison to methods that include object proposals. SSD is thus simple to train and easy to incorporate into systems that require a detection component. The Region Proposal Network (RPN) shares full-image convolution characteristics with the detection network, making regional proposals nearly cost-free. A RPN is a fully convolutional network that concurrently predicts object boundaries and objects at each location. The RPN is trained from start to finish to provide high-quality region proposals that Fast R-CNN may employ for detection. Object detection work in the past has repurposed classifiers to perform detection. YOLO, a modern method of target identification, presents it as a spatially segregated bounding box regression problem and associated class probabilities. In a single evaluation, a single neural network predicts bounding boxes and class probabilities directly from full images. Since a single network is the entire detection chain, end-to-end detection output can be well optimized.

2.3 Deep learning based multi-object detection overview

Despite major developments in object recognition focused on deep neural networks, focusing on small and occlusion targets remains a challenge. The researchers solved this problem by adding a cross-layer fusion multi-object detection and recognition algorithm based on Faster R-CNN [37]. The authors highlighted the problems and solutions for small object detection, as well as important deep learning approaches including fusing feature maps, integrating context detail, matching foreground-background examples, and producing enough positive examples. They looked at four study areas: basic object identification, face detection, aerial imaging object detection, and segmentation. In addition, this study evaluates the efficiency of various major

deep learning techniques for tiny object identification, including YOLOv3, Faster R-CNN, and SSD, utilizing three huge benchmark datasets of small objects. Although the detection precision of these deep learning approaches on small objects was poor, less than 0.4, their experimental findings indicate that Faster R-CNN worked better, followed by YOLOv3[38].

2.4 Object detection on UAV images

Object recognition algorithms based on deep learning systems have quickly become a popular way to process moving images captured by drones. In the field of deep learning, the increasing UAV industry dynamics and interest in possible applications like surveillance, visual navigation, identification of objects and sensor-based obstruction planning offer potential expectations. Object detection in images captured by Unmanned Aerial Vehicles (UAVs) is a daunting challenge in computer vision, owing to the difficulty in learning a well-trained object detection model for managing instances in UAV images of random direction, variations in various scales, unusual forms, etc. Researchers have developed a large-scale benchmark dataset, MOHR, aimed at conducting multi-scale object detection in UAV images with high resolution in order to promote object detection analysis and expand its applications in natural scenarios using UAVs [39]. The authors have conducted an extensive study of the state-of-the-art object detection algorithms focused on deep learning and have discussed the latest contributions of these algorithms to UAV datasets of low altitude. The document addresses the following: Faster RCNN, Cascade RCNN, R-FCN etc. in two stages, YOLO, and its derivatives in one stage, SSD, RetinaNet in CornerNet and Objects in advanced detection phases [40].

3 Methodology

This section comprises the preliminaries to carry out the research work including the dataset and the training algorithm. The methodology followed to conduct the research has been show in the flow chart discussed below:

Figure 2 Flowchart for methodology developed to execute the research work

3.1 Preliminaries

3.1.1 Dataset

The work has been carried out on two different datasets. Models have been tested on the state-of-the-art MOT20 dataset as well as the customized dataset captured through an assembled drone in campus premises. Both the datasets have been used for object detection purposes.

MOT20 dataset

The objective of the MOT Challenge was to create a consistent evaluation of different object tracking systems. Because pedestrians are well-studied in the tracking field and precise tracking and detection have great practical significance, the challenge focuses on multiple persons tracking. MOT15[41], MOT16[42], and MOT17 have all made significant contributions by offering a clean dataset and precise methodology for benchmarking multi-object trackers after their first release. The MOT20 dataset [43] has eight sequences, half of which is utilized for training and the other half for testing.

Custom drone-based testing dataset

The inferences have been recorded on the custom dataset created using the drone. The drone is a quadcopter integrated with a Wi-Fi enabled action camera SJ4000 supported and controlled by a gimbal, a GPS device, a Pixhawk flight controller, a telemetry, and a 5.8G 48CH transmitter for long range communication.

Figure 3 Sample shots from the custom drone-based dataset

This dataset includes 4 sequences of recorded videos comprising pedestrians in four different backgrounds. The sample images from the sequence recorded is given above in Figure 3.

3.1.2 Training algorithm

The training process for both datasets has been kept the same, as depicted in Figure 4 and detailed in the following algorithm.

Algorithm.

-
- Step1:** Create and activate Virtual Environment
 - Step2:** Install required packages
 - Step3:** Install protobuf and add to path
 - Step4:** Annotate image dataset and obtain xml files
 - Step5:** Partition the dataset into training and testing
 - Step6:** Convert xml to csv
 - Step7:** Create label map
 - Step8:** Create TfreCORDs
 - Step9:** Get the model and configure training pipeline
 - Step10:** Training the model
 - Step11:** Evaluate the model
 - Step12:** Export the trained model
 - Step13:** Get the inferences
-

Figure 4 Workflow for training and testing in Object Detection

3.2 Object Detection models

3.2.1 YOLO

YOLO is a shortened form of “You Only Look Once”. YOLO is the state-of-the-art object detection algorithm. Due to its fast computation and optimal results, it has become a standard way of detecting objects in the field of computer vision. Previously researchers used sliding window object detection then faster versions were worked upon such as R-CNN [6], fast R-CNN [7], faster R-CNN [8] and cascade R-CNN [9]. But in 2016 YOLO was invented which outperformed all the previous object detection algorithms [10].

For object detection it employs convolutional networks. On a single picture YOLO can recognize multiple objects. This indicates that YOLO locates the position of these objects on the image along with predicting classes of objects. The entire image is processed by a single Neural Network in YOLO. This neural network splits pictures into regions and creates probabilities for all regions. There after YOLO predicts the number and the probability of bounding boxes covering locations in the image and selects the best ones according to the probabilities. It is a unified approach of object localization, detection, and classification.

Figure 5 Yolov3 architecture

The 53 convolution layers of the YOLO v3 architecture are also referred to as Darknet-53 as shown in figure 5. But the original design has 53 layers for the detection task which provides for

106 architectural layers for YOLO v3. Detections are carried out at three levels - 82, 94 and 106. The most important aspects in the current edition of YOLO v3 are Residual Blocks, Skip Connections and Up-Samples. The batch normalizing layer and the Leaky ReLU activation function are followed by each convolution layer. There are no pooling layers, but the down-sample feature maps instead include extra convolutional layers with stride 2. All of them are scaled by the input network. And there is a possibility to experiment with keeping or not keeping aspect ratio by adjusting parameters when training and testing in original Darknet framework in Tensorflow, keras or any other framework. This enables the customization of the model. The detection is carried out at three distinct sizes and three different network locations. There are 82, 94 and 106 layers of distinct sites for detection. Network samples of the input picture at these different locations of the network by following factors: 32, 16, and 8. These three figures are termed network steps and demonstrate how the output at three different network locations is smaller than the network input. To generate YOLO v3 output, 1 x 1 detection kernels are used in these distinct network locations. The resultant feature maps will therefore have the same spatial size. The shape of the kernels is also determined with the following equation in terms of its depth.

$$d = b * (5 + c) \quad (1)$$

b in the equation represents the number of bounding boxes predicted by each cell of the produced feature map. YOLOv3 predicts 3 bounding boxes for every cell of these feature maps, thus b is equal to 3. Centre coordinates of bounding box, width and height of each bounding box has 5 + c attributes. For instance, YOLO version 3 trained on a COCO dataset that has 80 classes. Then, “c” is equal to 80 and the total number of attributes for each bounding box is 85. This incorporates one more dimension of 255 attributes of the bounding box given by equation 2.

$$d = 3 * (5 + 80) = 255 \quad (2)$$

For each cell in the feature map, YOLOv3 provides 3 bounding boxes. Every cell predicts an object through one of these bounding boxes if the object center belongs to its receptive field. When YOLOv3 is trained, it features a ground bounding box that detects an object. Therefore, it is important to identify the cells to which the bounding box belongs.

There are the following characteristics in each predicted bounding box: center coordinates, predicted width, predicted height; the scores for objects and the list of confidences for each class of bounding box.

The following stage is to extract probability between three predicted cell boundary boxes to determine that a given box has a certain class, calculate the product of the objective score and confidence list element wise, and find out the greatest probability of that class.

YOLO version 3 calculates offsets to preset anchors to predict real width and real height of bounding boxes. The log-space transform is another name for this offset. YOLO version 3 uses the sigmoid function to estimate the center coordinates of the bounding boxes. The following equations are used to calculate the width, height, and center coordinates of the predicted bounding box.

$$bx = \sigma(tx) + cx \quad (3)$$

$$by = \sigma(ty) + cy \quad (4)$$

$$bw = pw * etw \quad (5)$$

$$bh = ph * eth \quad (6)$$

The center coordinates, width and height of the bounding box are b_x , b_y , b_w , b_h , b_x and t_x , t_w and t_h are network outputs after training. YOLOv3 predicts anchor offsets, because during training it helps minimize unstable gradients.

The objectness score indicates the likelihood that a given cell is a center cell responsible for predicting a single object, with the proper bounding box containing the object within.

The objectness score represents the probability of presence of object inside the bounding box while the probability of detected object belonging to a certain class is represented by the class confidences. The objectness score (P_o) can be mathematically represented as the below equation:

$$P_o = P_{object} * IoU = \sigma(t_o) \quad (7)$$

Where IoU denotes the intersection over union of the predicted and ground truth bounding boxes, and P_{object} is the expected probability that the bounding box includes the object. The output is sent into a sigmoid function, which returns values between 0 and 1.

In this study, YOLOv5 is used for multi object tracking. It is state of the art and the latest version of the YOLO object detection algorithms. It beats all the previous YOLO versions and has set the benchmark for object detection very high.

3.2.2 SSD

SSD is the abbreviated term for Single Shot Multibox Detector, single shot means that various tasks like localization of the object and classification of the object is done in a single forward pass of the given network, it came initially in 2016 and trained on the various datasets such as COCO and PascalVOC, on these datasets SSD gave promising score of 74% mAP (mean Average Precision) at 59 frames per second. In Single Shot Detector, the detector classifies the

and location loss is also the measure of the bounding box, but it calculated the distance between bounding box and ground truth of the object.

$$MultiboxLoss = ConfidenceLoss + \alpha * LocationLoss \quad (8)$$

Alpha is the term which helps to balance the location loss, it is basically a parameter value which most optimally reduces the loss function.

In multibox, researchers created priors (or anchors in Faster-RCNN). These priors assist manage the Intersection over Union (IoU) ratio by providing pre-computed bounding boxes that closely match the original ground truth boxes. It is believed that value of IoU should be greater than 0.5 and to gain this results prior plays an important role, these priors are used in the replacement of starting the predictions with random coordinates.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (9)$$

In recent years many more features are added to the SSD to make it more strong performer, first the priors of SSD are fixed, in SSD these priors are chosen manually, second, now L1-norm is used in the SSD to calculate the location loss, it is not precise as L2-norm but it can give promising results, third, classification techniques is used in the SSD, for each bounding box there is a prediction class present.

3.2.3 Faster RCNN

Faster RCNN represented in figure 7 is a convolutional neural network-based object identification architecture developed by Ross Girshick, Shaoqing Ren, Kaiming He, and Jian Sun in 2015. Faster RCNN is made up of three steps. The convolution layers are the first step. Filters

are trained in these layers to extract relevant features from an image; for example, to obtain important features for a human face, the filters will be trained to learn forms and colors that only appear in the human face.

Figure 7 Faster RCNN architecture [45]

The identified features continue to the next step of the Region Proposal Network (RPN). RPN is a tiny neural network that strides over the convolution layers' final feature map to predict the existence of an object and its bounding box. A region proposal network's (RPN) output is a collection of boxes/proposals that will be analyzed by a classifier and regressor to check for the presence of objects. RPN anticipates whether an anchor will be in the background or foreground, then refines the anchor accordingly. The final stage predicts classes and bounding boxes. This stage comprises of another fully connected neural network is utilized to predict object class and bounding boxes using the regions provided by the RPN as an input. The loss function of the Regressor is given by

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} smooth_{L_1}(t_i^u - v_i), \quad (10)$$

in which

$$smooth_{L_1}(x) = \begin{cases} 0.5 x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}, \quad (11)$$

The RPN's total loss is made up of both the classification and regression losses.

Following RPN, regions of various sizes is proposed. CNN feature maps of varying sizes correspond to different sized regions. It's not simple to put up an effective structure for working on features of various sizes. By reducing the feature maps to the same size, Region of Interest Pooling can make the problem easier to solve. Unlike Max-Pooling, which has a constant size, ROI Pooling divides the input feature map into a set number (let's say k) of nearly equal areas before applying Max-Pooling to each of them. As a result, regardless of the size of the input, the ROI Pooling output is always k.

4 Experiment and analysis

4.1 Benchmark and Evaluation Methodologies used in validation

For evaluation, Precision and Recall were used. Precision and Recall are the widely used evaluation metrics for classification problems. They are used to check the accuracy of the model. Precision is the fraction of correctly classified instances among the total number of instances. On the other hand, Recall is the fraction of correctly classified instances among the total number of instances in that class. The following equations for Precision and Recall respectively can be derived from a confusion matrix.

$$PPV = TP / (TP + FP) \quad (12)$$

$$TPR = TP / (TP + FN) \quad (13)$$

where PPV stands for Positive Predictive Value or Precision, TPR stands for True Positive Rate or Recall.

True Positive is abbreviated as TP, False Positive as FP, and FN stands for False Negative.

In the YOLO model, loss for each box prediction consists of three loss functions - Coordinate or box loss, Objectness loss, and Classification loss. Coordinate or box loss is due to the incorrect positioning of the bounding box. The loss of objectness is caused by an incorrect box-object IoU prediction. The deviations from properly predicting '0' for wrong classes and '1' for correct classes in a specific box generate classification loss. The network is trained to predict a correct IoU using the objectness loss term, while the coordinate loss is used to train the network to predict a better bounding box. All the accurate bounding boxes in each spatial cell contribute to coordinate and classification loss but in case of objectness loss, all the box predictions whether accurate or inaccurate contribute towards it. Objectness indicates the coverage of the object by the bounding box therefore bounding boxes with very small objectness are discarded.

4.2 Experimental Setup

For experimental analysis, MOT20 dataset was used. Various models were applied on the dataset using pre-trained weights. The system specifications were T4 15 GB GPU, 13 GB RAM, 25 GB disk space and xeon dual core 2.2 GHz CPU.

The experiments are conducted using Tensorflow custom object detection API.

4.2.1 SSD

To experiment with Single Shot Multi box detector, SSD MoboleNet v2 was used. MobileNet is a mobile and embedded vision-focused deep neural network architecture. The training of data was conducted in 50,000 steps having batch size of 10 and image size 320 x 320 with L2 regularizer having weight 4×10^{-5} . The SSD model yields training loss of 30 % while testing loss is 25%. The sample output is shown in figure 8 and the plots are shown in figure 9. With the

given configuration SSD has shown larger fluctuations in precision vs epoch, recall vs epoch curves with a gradual increase in precision to 34% and recall to 33%. The mAP value has reached upto 48% which finally yields the F measure to 34%.

Figure 8 Sample output of detections by SSD MobileNet model trained on MOT20 dataset

Figure 9 Evaluation metrics vs epochs

4.2.2 Faster RCNN

Inception Resnet model was used, for faster RCNN detector. Inception-ResNet is a CNN, trained on the ImageNet database in more than one million pictures. The network is 164 deep layers, allowing pictures to be classified in 1000 types of objects. The training of data was conducted in 50,000 steps having a batch size of 1 and image size 800 x 1333 with L2 regularized to zero

weight. The model invigorated the training loss of 91.6% and testing loss of 64%. Precision and recall values yielded by the model are 18% and 27% respectively. The sample output is shown in figure 10 and the plots are shown in figure 11. The mAP value measures 22.5% and the F measure to 22%.

Figure 10 Sample output of Faster RCNN Inception-Resnet model

Figure 11 Evaluation metrics vs epochs

4.2.3 YOLOv5

YOLOv5 contains the simple functionalities for test time augmentation (TTA), model ensemble, hyper-parameter evolution, and exportation to ONNX, CoreML, and TFLite, and is a compound-sized object recognition model family trained in the COCO dataset. YOLOv5 was custom trained

on MOT20 dataset which gave precision of 61% and recall 35%. The mAP value evaluated by the model is 0.09 and the F measure is calculated as 0.44. the metrics can be visualized in figure 12.

Figure 12 YOLOv5 results on state of the art MOT20 dataset

4.2.4 Modified YOLOv5

YOLOv5 model was modified for experimental purpose by tuning hyperparameters such as learning rate to 1×10^{-5} and beta value to 0.9. However, this slightly mitigated the performance of YOLOv5. The results shown in figure 13 depict values of precision and recall as 44% and 11.93% respectively. mAP value was observed as 0.12 and F measure is calculated to 19%.

Figure 13 Modified Yolov5 results on state of the art MOT20 dataset

4 Results and Discussion

Figure 9, 11, 12 and 13 displays the results using SSD, Faster RCNN, YOLO version 5 and Modified version of Yolov5 models respectively on MOT 20 dataset. These figures display the graphs of loss functions i.e., box or coordinate loss, objectness loss and classification loss along with the precision and recall curves.

Larger fluctuations in precision, recall and mAP curves are observed in SSD model which is smoother and steeper in all the other models. A tradeoff is observed in Precision and recall values, therefore F1 measure is calculated to evaluate and analyze the results. Yolov5 models, both original and the modified model have shown high precision values i.e., 0.61 and 0.45 respectively, in precision vs epoch curves. Other than these graphs, there are mAP, mean averages with IoU thresholds 0.5 and 0.95 curves which also follow the same pattern as the precision and recall curves. However, Mean Average Precision values are lesser in Yolov5 and

maximum value of 0.48 is seen in SSD. When mAP curves are observed then besides SSD possessing maximum value, its graph is rather highly fluctuating with frequent spikes, which is smooth in all the other models. Therefore, based on precision, recall and mAP values it becomes difficult to conclude and declare the best performance of any model. To evaluate these models, we chose F measure metric to draw the comparison. The F1 score is the harmonic mean of precision and recall given by equation 14.

$$F1\ score = 2 \cdot \frac{(Precision * Recall)}{Precision + Recall} \quad (14)$$

Thus, from the F1 score values observed, it can be stated that Yolov5 model trained on MOT20 dataset has outperformed rest of the models with 0.44 F1 score followed by SSD, Faster RCNN and modified Yolov5. The comparative analysis of all the above figures is depicted in table 1.

The excerpts from the tested footages of the custom dataset are given below in figure 14 showing precise person detection.

Figure 14 Sample shots from the test case run on custom drone-based dataset showing person detection

Table 1 Comparative analysis of state-of-the-art models SSD, Faster RCNN, YOLOv5 and Modified YOLOv5 based on evaluation metrics

Model	Training Loss	Testing Loss	Precision	Recall	mAP	F1 Score
SSD	0.3	0.25	0.34	0.33	0.48	0.34
Faster RCNN	0.916	0.64	0.18	0.27	0.225	0.22
YOLOv5	0.015	0.15	0.61	0.35	0.09	0.44
Modified YOLOv5	0.01713	0.1712	0.4487	0.1193	0.12	0.19

5 Conclusion

The deep learning methods assist UAV based surveillance system to proffer real-time video surveillance, rally round to amass useful crime evidence, and helps to prevent and reduce chances of theft. It enables to automate the monitoring of high-risk regions with precision and provides coverage to even remotest regions where human surveillance is exigent. Pedestrian detection, which offers vital information for the semantics of video evidence, is the significant and critical tasks of every astute video surveillance system. Thus, pedestrian detection is one of the most imperative categories of object detection. An effort has been made to formulate the understanding of deep learning methods for pedestrian detection through this paper. The work presented in this paper has been carried out to understand the performance of state-of-the-art deep learning models for object detection to perform surveillance. The models are trained on MOT20 dataset and tested on custom dataset. It was observed that YOLOv5 has outperformed all the other models with highest F1 score of 0.44 while the best mAP values 0.48 are obtained from SSD Mobile Net model.

Acknowledgement

This study is sponsored by a grant from the Indian government's Department of Science and Technology, which was awarded in response to a CFP made under the Interdisciplinary Cyber Physical Systems (ICPS) Program, DST/ICPS/CPS-Individual/2018/181 (G).

Compliance with ethical standards

- I. Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.
- II. Funding details: Research project was sponsored by DST but there is no provision for the manuscript publication fund.
- III. Conflict of interest: The authors declare that they have no conflict of interest.
- IV. Informed Consent: All authors have checked and agreed the submission.

Authorship contributions

The article conceptualization was done by N.T., P.N., R.J. and N.T., P.N., R.J., N.S. have contributed to technical, conceptual content and architectural design. The methodology was proposed by N.T., P.N., R.J. and J.H.. Software feasibility was validated and tested by N.T., P.N., R.J. and N.S.. Further more the formal analysis with Investigation was performed by N.T., R.J. The data curation for the experiments was done by P.N. and N.S.. Writing-Original Draft Preparation was done by P.N., and N.S. along with Writing-Review & Editing performed by N.T., R.J.. The project was visualized by N.S., under the supervision of P.N., R.J. and N.T.. D.S, J.H contributed to guidance and counseling on the writing of the paper. Project administration was done by D.S..

References

1. Christian Szegedy, Alexander Toshev, and Dumitru Erhan. 2013. Deep neural networks for object detection. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 2553–2561.
2. Piccinini, P., Prati, A. and Cucchiara, R., 2012. Real-time object detection and localization with SIFT-based clustering. *Image and Vision Computing*, 30(8), pp.573-587.
3. Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), pp.3212-3232.

4. Kundu, A., Krishna, K.M. and Sivaswamy, J., 2009, October. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4306-4312). IEEE.
5. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J. and Marín-Jiménez, M.J., 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), pp.2280-2292.
6. Tan, X. and Triggs, B., 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6), pp.1635-1650.
7. Fasel, I., Fortenberry, B. and Movellan, J., 2005. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, 98(1), pp.182-210.
8. Jaimes, A. and Sebe, N., 2007. Multimodal human-computer interaction: A survey. *Computer vision and image understanding*, 108(1-2), pp.116-134.
9. Majaranta, P. and Bulling, A., 2014. Eye tracking and eye-based human-computer interaction. In *Advances in physiological computing* (pp. 39-65). Springer, London.
10. Galarza, E.E., Egas, F.D., Silva, F.M., Velasco, P.M. and Galarza, E.D., 2018, January. Real time driver drowsiness detection based on driver's face image behavior using a system of human computer interaction implemented in a smartphone. In *International Conference on Information Technology & Systems* (pp. 563-572). Springer, Cham.
11. Lemley, J., Bazrafkan, S. and Corcoran, P., 2017. Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision. *IEEE Consumer Electronics Magazine*, 6(2), pp.48-56.
12. Luo, R.C. and Lai, C.C., 2013. Multisensor fusion-based concurrent environment mapping and moving object detection for intelligent service robotics. *IEEE transactions on industrial electronics*, 61(8), pp.4043-4051.
13. Wang, Y., Zhang, D., Liu, Y., Dai, B. and Lee, L.H., 2019. Enhancing transportation systems via deep learning: A survey. *Transportation research part C: emerging technologies*, 99, pp.144-163.

14. Liu, W., Zhang, M., Luo, Z. and Cai, Y., 2017. An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors. *IEEE Access*, 5, pp.24417-24425.
15. Pathak, A.R., Pandey, M. and Rautaray, S., 2018. Application of deep learning for object detection. *Procedia computer science*, 132, pp.1706-1717.
16. Saikia, S., Fidalgo, E., Alegre, E. and Fernández-Robles, L., 2017, September. Object detection for crime scene evidence analysis using deep learning. In *International Conference on Image Analysis and Processing* (pp. 14-24). Springer, Cham.
17. Butepage, J., Black, M.J., Kragic, D. and Kjellstrom, H., 2017. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6158-6166).
18. Lu, N., Wu, Y., Feng, L. and Song, J., 2018. Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data. *IEEE journal of biomedical and health informatics*, 23(1), pp.314-323.
19. Erfani, S.M., Rajasegarar, S., Karunasekera, S. and Leckie, C., 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58, pp.121-134.
20. De Smedt, F., Hulens, D. and Goedemé, T., 2015. On-board real-time tracking of pedestrians on a UAV. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1-8).
21. Zhang, C., Li, H., Wang, X. and Yang, X., 2015. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 833-841).
22. Hassner, T., Itcher, Y. and Kliper-Gross, O., 2012, June. Violent flows: Real-time detection of violent crowd behavior. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-6). IEEE.

23. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K. and Leal-Taixé, L., 2020. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
24. Dollar, P., Wojek, C., Schiele, B. and Perona, P., 2011. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4), pp.743-761.
25. Wojek, C. and Schiele, B., 2008, June. A performance evaluation of single and multi-feature people detection. In *Joint Pattern Recognition Symposium* (pp. 82-91). Springer, Berlin, Heidelberg.
26. Dollár, P., Wojek, C., Schiele, B. and Perona, P., 2009, June. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 304-311). IEEE.
27. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X. and Sun, J., 2018. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*.
28. Lin, C., Lu, J., Wang, G. and Zhou, J., 2020. Graininess-aware deep feature learning for robust pedestrian detection. *IEEE transactions on image processing*, 29, pp.3820-3834.
29. Zhang, Z., Gao, J., Mao, J., Liu, Y., Anguelov, D. and Li, C., 2020. STINet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11346-11355).
30. Neubeck, A. and Van Gool, L., 2006, August. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)* (Vol. 3, pp. 850-855). IEEE.
31. Huang, X., Ge, Z., Jie, Z. and Yoshie, O., 2020. NMS by representative region: Towards crowded pedestrian detection by proposal pairing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10750-10759).
32. Khalifa, A.B., Alouani, I., Mahjoub, M.A. and Amara, N.E.B., 2020. Pedestrian detection using a moving camera: A novel framework for foreground detection. *Cognitive Systems Research*, 60, pp.77-96.

33. Ferryman, J. and Shahrokni, A., 2009, December. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance* (pp. 1-6). IEEE.
34. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C., Liu, X. and Ma, W., 2018. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (pp. 0-0).
35. Zhang, S., Benenson, R. and Schiele, B., 2017. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3213-3221).
36. Liu, W., Luo, W., Lian, D. and Gao, S., 2018. Future frame prediction for anomaly detection—a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6536-6545).
- Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.
37. Li, C.J., Qu, Z., Wang, S.Y. and Liu, L., 2021. A Method of Cross-layer Fusion Multi-object Detection and Recognition Based on Improved Faster R-CNN Model in Complex Traffic Environment. *Pattern Recognition Letters*.
38. Zhang, H., Sun, M., Li, Q., Liu, L., Liu, M. and Ji, Y., 2021. An empirical study of multi-scale object detection in high resolution UAV images. *Neurocomputing*, 421, pp.173-182.
39. Liu, Y., Sun, P., Wergeles, N. and Shang, Y., 2021. A Survey and Performance Evaluation of Deep Learning Methods for Small Object Detection. *Expert Systems with Applications*, p.114602.
40. Mittal, P., Sharma, A. and Singh, R., 2020. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image and Vision Computing*, p.104046.
41. Leal-Taixé, L., Milan, A., Reid, I., Roth, S. and Schindler, K., 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*.

42. Milan, A., Leal-Taixé, L., Reid, I., Roth, S. and Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
43. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K. and Leal-Taixé, L., 2020. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
44. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
45. Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L. and Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 145, pp.3-22.