Multiple Object Detection on Surveillance Videos For Improving Accuracy Using Enhanced Faster R-CNN

[1]Divya G, [2]Manoj Kumar D S and [3]Shri Bharathi S V
[1&3]Department of Data Science and Business Systems, School of Computing, SRM Institute of Science and Technology, Kattankalathur Campus, Chennai, India.
[2]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, India.
[1]mailtodivya16@gmail.com, [2]manojkumards03@gmail.com, [3]shribharathi01@gmail.com

Correspondence should be addressed to Divya G : mailtodivya16@gmail.com.

Article Info
Journal of Machine and Comp uting (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202303042
Received 30 March 2023; Revised from 22 July 2023; Accepted 15 August 2023.
Available online 05 October 2023.

Abstract – Computer vision is a dynamic and rapidly evolving field within the broader domain of artificial intelligence. Within surveillance monitoring systems, one of the central tasks is object detection, which involves identifying and localizing objects of interest in video sequences to provide safety and security of the people. Detection of multiple objects is a challenging task in video sequences which interprets less accuracy and false Bounding box regression. In this paper, enhanced faster R-CNN model is proposed and trained to compute regional proposal through Convolutional layers on the different scene of the sequences in term of lighting, motion capture related to spatial analysis. These enhancements could encompass architectural improvements, novel training strategies, or the incorporation of additional data sources to improve the model's overall performance. Proposed model is experimented on pedestrian video gives an improved accuracy detection rate than single detector techniques.

Keywords – Faster R-CNN, Visual Geometry Group, MOT, Computer Vision, Regression.

I. INTRODUCTION

Computer vision as well as pattern recognition, very critical topic of research is Person detection, because it is applied widely in video surveillance, analysis of action and in driver assistance systems (ADAS). The pedestrian detection performance is still susceptible to huge difficulties in real-world applications because of variations in illumination, changing pose, occlusion, and deformation of human. Major problem of object-detection is Pedestrian detection i.e., rigid object detection or half-rigid object detection. As compared with the human detection, this detection absolutely different in terms of posture deformation, shooting angle and image resolution, in which the complex models like as explicit geometric modeling and multi-view modeling are used usually. Generally, the appearance of shape and aspect ratio are similar in some pedestrians and this in turn produces some difficulties like occlusion, size and illumination which are normally produced by shooting angle of camera and particular constraints of scene. Information of weak appearance is present in the camera which is distant from the pedestrians and therefore often it is recognized incorrectly as long-thin objects surrounded around it because of the self-occlusion in body crowd and profile.

Human operator observer forms the basis for surveillance systems at public and private areas. The walking or running person on the street is termed as pedestrian. The surveillance of video automatically is the most challenging task for detecting and tracking the activity of the suspicious pedestrian. For real time scene analysis, there is no appropriate solution for the learning-based methods in the real-time dynamic environment. This real time scene analysis is challenging task for obtaining the previous knowledge about each and every objects.

II. LITERATURE SURVEY

R-CNN (Region based Convolutional Neural Organization) based continuous framework was introduced which naturally identifies objects which may be found in an indoor climate [1]. A normal exactness of 74.33%, and the interim taken to distinguish objects per picture was 0.12 s was accomplished.3D object recognition strategy that utilizations relapsed descriptors of privately tested RGB-D patches for 6D vote projecting in streaming video was introduced [2]. A convolutional auto-encoder was utilized that has been prepared on a huge assortment of arbitrary nearby fixes. This technique conveys vigorous discovery results that contend with and outperform the best-in-class while being versatile in the quantity of articles.

A novel strategy for multi-class geospatial object discovery was introduced in utilizing just scene-level labels [3]. A couple shrewd scene-level closeness was utilized to learn discriminative convolutional loads by taking advantage of the shared data between scene sets. Point-wise scene-level labels to learn class-explicit actuation loads were used. Finally, articles can be distinguished by fragmenting the CAM. The exploratory outcomes exhibit that the profound organizations significantly outflank the cutting-edge strategies.

DFPN (Feature Pyramid Organization with DenseNet) technique was acquainted with beat the issue of class disarray in the Utilization pictures that it is difficult to be tackled by standard model which is the cutting edge object identification model FPN with RoI Align pooling. DFPN accomplishes top outcome with a Guide of 86.9% on USE test set in the wake of adjusting between the grouping misfortune and bouncing box relapse misfortune, which further develop focuses contrasted with benchmark model, and particularly erythrocyte's AP is enormously improved from 65.4% to 93.8%, showing class disarray has been fundamentally settled.

Convolutional neural network had introduced promising outcomes for object recognition by easing the requirement for human ability for physically handcrafting the highlights for extraction [4]. It permits the model to advance naturally by letting the neural organization to be prepared for huge scope picture information utilizing strong and hearty GPUs in an equal manner, and less computational time.

A viable powerful foundation displaying with quick profound learning grouping was introduced to provide an exact plan for human-creature identification from camera-trap pictures with jumbled moving articles [5]. Another square shrewd foundation model, called Main Flow Detection (MFD), to show the variety of the foundation of the camera-trap groupings and the forefront object proposition was created. Accuracy, precision examination of DCNN was performed to develop a quick profound learning order plan to arrange this area proposition into three classes: human, creatures, and foundation patches. The upgraded CNN had the option to decrease the order time by multiple times and keep up with high precision.

Object Identification through a Neighborhood and Worldwide strategy in light of dep neural network (Turf LGDRN) for saliency calculation was introduced [6]. Residual Network (ResNet-G) was prepared to gauge the conspicuousness of the striking article worldwide and separate numerous level nearby elements through another profound lingering organization (ResNet-L) to catch the neighborhood property of the remarkable item. profound remaining organization (ResNet-G) to gauge the noticeable quality of the remarkable item all around the world and concentrate numerous level nearby elements by means of another deep neural (ResNet-L) to catch the neighborhood property of the striking article.

III. PROPOSED SYSTEM

Proposed System focus on to improve the detection rate of objects (persons) in stride of video sequences. It aims to identify the persons in continuous sequence of frames. The faster R-CNN pertained model is enhanced to detect the persons [7].

Enhanced Faster RCNN Model

CNN forms the backbone of the proposed model, and it is fed with input image, which is started by Region Proposal Network (RPN) [8]. The first step is to resize input image with 600px as shortest side and more than 1000px is the longer side. H x W indicates the output features of CNN, which is usually smaller than the input image based on CNN strides. 16 is the network stride for VGG, ZF-Net (backbone networks), corresponding to the two points of 16 pixels separated from input image as shown in **Fig 1**.

The dimensions of the object present in an input image are learnt by the networks, "Anchors" are used for this location and size estimation. Objects which are possible in different ratios and sizes at this location is indicated by these anchors [9]. The possible anchors are 9 in number and it is represented in the below **Fig 2** with 3 various aspect ratios and sizes for a point A on the output feature map present in the input image. 128^2, 256^2, 512^2 are the 3 box scales and 1:1, 1:2 and 2:1 are the 3 aspect ratios used by the anchors in PASCAL challenge [10].

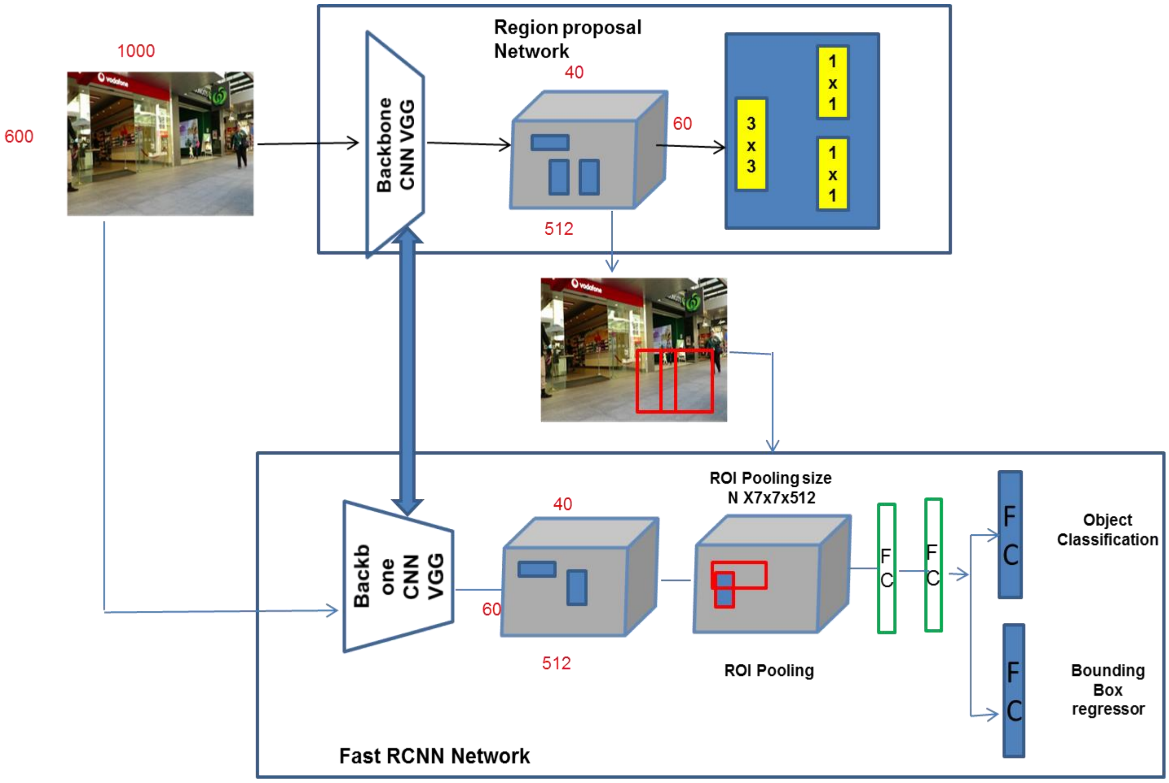


Fig 1. Object Detection using Enhanced Faster R CNN

The backbone feature map, as illustrated in **Fig 1**, is subjected to 3 x 3 convolution with 512 units to produce 512-d feature maps for each location. Two sister layers are then added: a 1 x 1 convolution layer for object classification with 18 units and a 1 x 1 convolution layer for bounding box regression with 36 units.

The result of the 18 units in the classification branch is size (H, W, 18). This output is used to provide probability for whether an object is present at each place in the backbone feature map (size: H x W) at that position.The regression branch's 36 units produce an output with the dimensions (H, W, 36). The 4 regression coefficients of each of the 9 anchors for each point in the backbone feature map (size: H x W) are provided by this output. The coordinates of the anchors that contain objects are improved using these regression coefficients [11].

Training and Loss Function

40 x 60 positions equal 40*60*9 20k anchors altogether. To prevent them from adding to the loss, all anchors that cross the border are disregarded at train time. Approximately 6k anchors per image are now left.

The anchor has the highest IoU (Intersection over Union, a measure of overlap) with a groundtruth box, or the anchor has an IoU more than 0.7 with any groundtruth box. An anchor is deemed to be a "positive" sample if it meets either of these two conditions. Multiple anchors can receive positive labels from the same groundtruth box [12].

Every mini-batch of RPN made up with only one image. Because collecting every anchor out of the image could influence learning procedure towards negatives, 128 positives and negatives examples are chosen at random to make up the group, with extra negative samples being added if the proportion of positives is inadequate [13]. RPN's training loss is the multi-task loss, indicated as Equation (1).

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \quad (1)$$

In this case, i represent index value of anchors inside mini batches. Categorization losses $L_{cls}(p_i,p_i^*)$ to be log losses between 2 classifications. pi gives categorization branch out-coming scores. I as well as pi* would be labels (one / zero). Regress losses $L_{reg}(t_i,t_i^*)$ just active if anchor includes objects, i.e. ground truth p_i* equals one. The t_i indicates outcome estimation, comprises 4 variable [t_x, t_y, t_w^*, t_h^*]. Regression goal t_i*, determines in Equation 3.2

$$t_\mathrm{x}^* = (x^* - x_\mathrm{a})/w_\mathrm{a}, t_\mathrm{y}^* = (y^* - y_\mathrm{a})/h_\mathrm{a}, t_\mathrm{w}^* = \log(w^*/w_\mathrm{a}), t_\mathrm{h}^* = \log(h^*/h_\mathrm{a}) \qquad (2)$$

x, y, w, and h stand for the box's height and breadth, as well as the (x, y) coordinates of the box's center. The anchor box's and its matching groundtruth bounding box's coordinates are denoted by xa, x*. Different regressors that do not share weights are present in k (= 9) of the anchor boxes. Therefore, if the sample is positive, the regression loss for an anchor i is applied to the appropriate regressor [14]. The predicted positive anchor box may be applied the learnt regression output ti, and the x, y, w, and h parameters for the anticipated item proposal bounding box can be back-calculated using Equation 3.

$$t_\mathrm{x} = (x - x_\mathrm{a})/w_\mathrm{a}, t_\mathrm{v} = (y - y_\mathrm{a})/h_\mathrm{a}, t_\mathrm{w} = \log(w/w_\mathrm{a}), t_\mathrm{h} = \log(h/h_\mathrm{a}) \qquad (3)$$

Test time Details

To send in the item proposal bounding boxes at test time, the 20k anchors from each image undergo a number of post-processing procedures. The anchors are subjected to the regression coefficients for accurate localization [15]. This results in exact bounding boxes. The positions of each box are determined by their class scores. After that, a non-maximum suppression (NMS) with a threshold of 0.7 is used. All bounding boxes that have an IoU of more than 0.7 with another bounding box are deleted starting at the top.

Thus, for a set of overlapping boxes, the highest-scoring bounding box is kept. There are over 2000 ideas for each image. Cross-boundary bounding boxes are kept and cropped to the edge of the image. All 2k proposals from the RPN are used to train the Fast R-CNN detection pipeline using these object suggestions [16]. Only the Top N RPN suggestions are picked at test time for Fast R-CNN detection.

Methodology

Proposed Methodology includes the process of how the objects are detected using Enhanced faster R-CNN model. Relu activation function is used in fully connected layer to detect the various objects.

Each frames is process through VGG back bone and convolution layer. The methodology includes following steps for object detection.

- Regional proposal Network
- Anchor Boxes
- Generating Anchor Boxes
- Labelling Anchor Box in Training Data
- Assigning Ground Truth Box to Anchor Box
- Labelling offset and classes
- Predicting the bounding box.

Region Proposal Network

The region proposals are now should be trained and customized based on benchmark dataset which is used for detection task.

- Regional proposal is trained completely, so that it gives better results in detection. Thus, it produces better region proposals compared to traditional methods like Selective Search and Edge Boxes.
- The RPN processes each frame using the same convolutional layers used in the Fast R-CNN detection network.
- Due to sharing the same convolutional layers, the RPN and the Fast R-CNN can be merged into a single network. Thus, training is done only once

The final convolutional layer shared with the Fast R-CNN serves as the foundation for the RPN's work. The feature map is traversed using a sliding window based on a rectangle window of size nxn. Several potential region recommendations are generated for each window [17]. The "objectness score" will be used to filter these recommendations, thus they are not the final ones.

Anchor Boxes

Fig 2 represents process of generating anchor box of each frame or image. Each final convolution layer's feature map is run through a box of dimensions nxn, where n=3 for the VGG-16 network. K region proposals are generated for each box. Each suggestion is parametrized in accordance with an anchor box, a reference box. The anchor boxes' two parameters are.

- Scale
- Aspect Ratio

As the size of the anchors varies, reference anchors (also known as anchor boxes), which are employed in a single frame at a single scale, can provide scale-invariant object detectors. By doing this, extra photos or filters are avoided. To communicate features between the RPN and the Fast R-CNN detection network, multi-scale anchors are essential.

A feature vector (of length 512 for the VGG-16 net and 256 for the ZF net) is retrieved for each nxn region proposal. The following two fully connected sibling levels receive this vector:The objectness score for each region suggestion is produced by a binary classifier called cls, which is the first FC layer [18].

Reg, the second FC layer, returns a 4-D vector specifying the region's bounding box. There are two outputs on the first FC layer. First, the region is classified as a background, and then it is classified as an item.

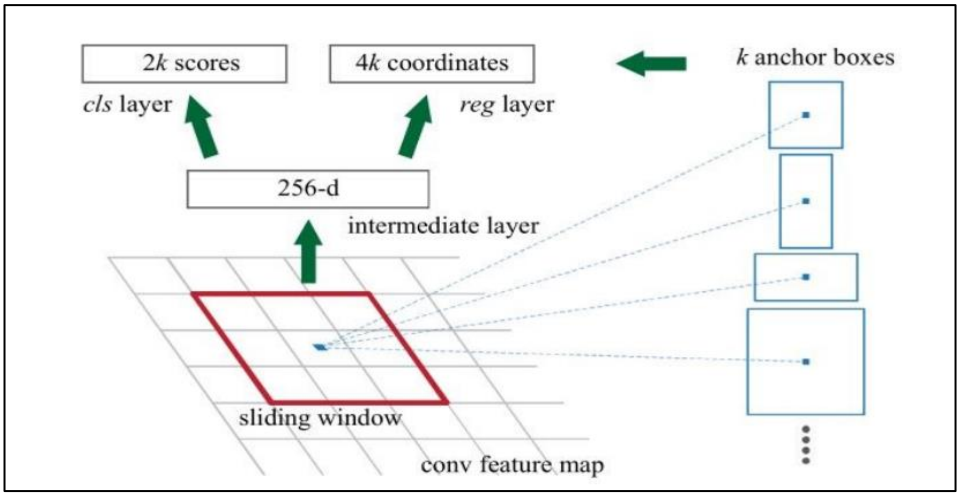


Fig 2. Anchor box Process

Generating Multiple Anchor Box

Assume the input image has a height and width of h and w, respectively. to design anchor boxes that are focused on each individual pixel in the image. The scale should be set to [s(0,1]Aspect ratio (the ratio of widths to heights) is s(0,1) and is r>0r>0. At that time, the anchor boxes' widths and heights are, respectively, wsr and hs/r and hs/r. Always keep in mind that an anchor box with known dimensions is determined when the center position is specified. Let's set a sequence of scales s1,...,sn, s1,...,sn and a progression of viewpoint proportions r1,...,rmr1,...,rm to create numerous anchor boxes with diverse forms. The information picture will include a total number of anchor boxes using all of these scales and perspective proportions, with each pixel serving as the center [19]. Because of these anchor boxes, the computing complexity of all the ground truth boxes can actually be too high. This function will return all the anchor boxes in Equation 4 once you specify the input image, a list of scales, and a list of aspect ratios.

$$(s1, r1), (s1, r2), \dots (s1, rm), (s2, r1), (s3, r1), \dots, (sn, r1) \tag{4}$$

n+m−1n+m−1 is the number of anchor boxes centered on the same pixel. It will produce a total of wh(n+m−1)wh(n+m−1) anchor boxes for the full input picture. If you provide the input picture, a listing of dimensions, and a listing of aspect ratios, this method would retrieve every one of the anchor boxes.

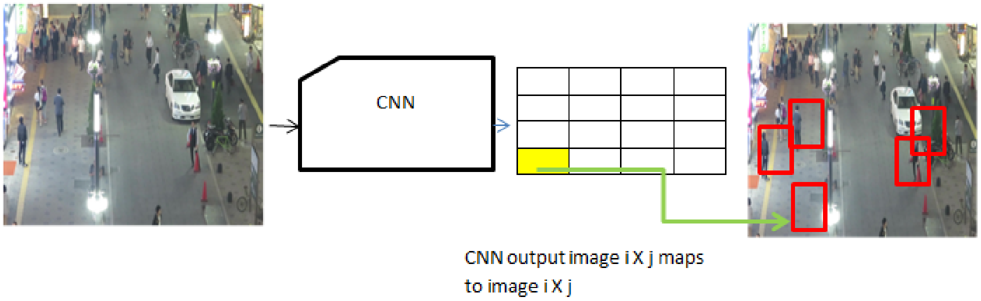


Fig 3. 8 X 8 Feature map of an Image

In **Fig 3** feature map of image is visualized and it is the outcome of the filtered image. Every position in outcome is the activation of neuron and outcome is gathered in the feature map.

Labelling Anchor Box in Training data

Consider each anchor box in a training dataset to be training data. Each anchor box needs a class label that corresponds to it and an offset label that indicates how far away the ground-truth bounding box is from the anchor box in order to train the model [20]. During the prediction, for each frame w multiple anchor boxes are generated, predict classes and offsets for all the anchor boxes, adjust their positions according to the predicted offsets to obtain the predicted bounding boxes, and finally only output those predicted bounding boxes that satisfy region of the image

Labels are included in an object detection training set that identify the classes of the remaining objects and the positions of the ground truth bounding boxes. Refer to the labeled position and class of the ground-truth bounding box that is assigned to the created anchor box in order to label it. Here, we outline an approach for tying anchor boxes to the closest ground-truth bounding boxes.

Assigning Ground Truth Box to Anchor Box

Model describe an algorithm for assigning closest ground-truth bounding boxes to anchor boxes. Given a image, assume that the anchor boxes are A1,A2,… ,An and the ground-truth bounding boxes are B1,B2,… ,Bnb , where na ≥ nb . Let's characterize a network X∈R na ×nb , where component xij in the ith line and jth section is the IoU of the anchor box simulated Ai with box value Bj . Calculation comprises by accompanying advances:Track down the biggest component in grid X and indicate its line and section lists as i1 and j1 , separately. Later, at that point, the ground-truth jumping box Bj1 is relegated to the anchor box Ai1 . This is very natural in light of the fact that Ai1 and Bj1 are the nearest among every one of the sets of anchor boxes and ground-truth jumping boxes. After the principal task, dispose of the relative multitude of components in the i1th line and the j1th segment in lattice X. The biggest of the leftover components in grid X and indicate its line and segment lists as i2 and j2 , separately. We appoint ground-truth bouncing box Bj2 to secure box Ai2 and dispose of the relative multitude of components in the i2th line and the j2th segment in lattice X .

Now, components in two lines and two sections in grid X have been disposed of. We continue until all components in nb segments in lattice X are disposed of. Right now, we have relegated a ground-truth jumping box to every one of nb anchor boxes. Just navigate through the excess na‒nb anchor boxes. For instance, given any anchor box computer based intelligence , find the ground-truth bouncing box Bj with the biggest IoU with simulated intelligence all through the ith line of grid X , and allot Bj to artificial intelligence provided that this IoU is more noteworthy than a predefined edge.

To outline the above calculation utilizing a substantial model. As displayed in Fig. 3.4 (left), expecting that the greatest worth in framework X is x23 , we allot the ground-truth jumping box B3 to the anchor box A2. Then, at that point, we dispose of the multitude of components in line two and section three of the framework, find the biggest x71 in the leftover components (concealed region), and allocate the ground-truth jumping box B1 to the anchor box A7 . Then, as displayed in Figure 3.4 (center), dispose of the relative multitude of components in line seven and section one of the lattice, find the biggest x54 in the excess components (concealed region), and dole out the ground-truth bouncing box B4 to the anchor box A5 . At long last, as displayed in Figure 3.5 (right), dispose of the relative multitude of components in line five and segment four of the framework, find the biggest x_{92} in the leftover components (concealed region), and relegate the ground-truth jumping box B2 to the anchor box A9 . From that point forward, to navigate through the leftover anchor boxes A1,A3,A4,A6,A8 and decide if to allot them ground-truth jumping boxes as indicated by the limit.

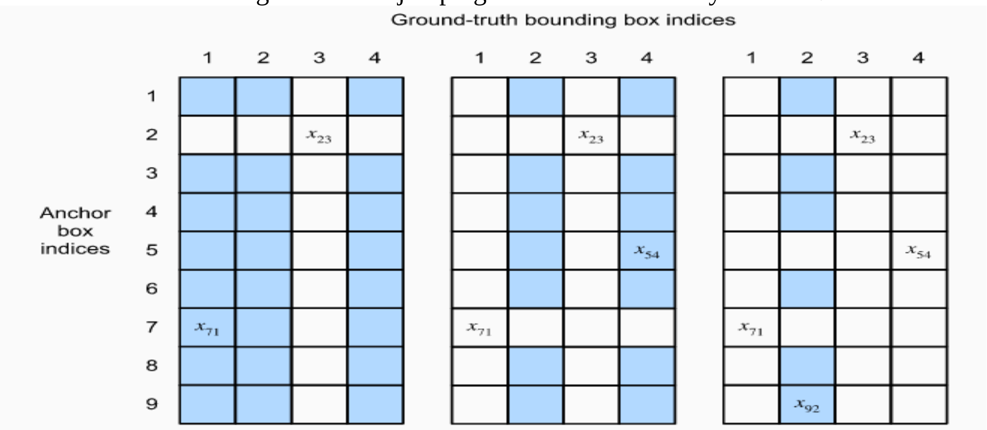


Fig 4. Indices of ground truth bounding box

Labelling offset and Classes

To indicate each anchor box's class and offset. Assume that a ground-truth bounding box BB is given to an anchor box AA. On the one hand, the BB class will be assigned to the anchor box AA. The anchor box AA's offset, on the other hand, will be labeled. according to the relative positions of the central coordinates of the boxes BB and AA and their respective sizes. Given the dataset's various boxes' placements and sizes, Given that AA and BB have respective central coordinates of (xa,ya)(xa,ya) and (xb,yb)(xb,yb), widths of wawa and wbwb, and heights of haha and hbhb. Identify the offset of AA as per Equation 7.

$$\left(\frac{\frac{x_b-x_a}{w_a}-\mu_x}{\sigma_x},\frac{\frac{y_b-y_a}{h_a}-\mu_y}{\sigma_y},\frac{\log\frac{w_b}{w_a}-\mu_w}{\sigma_w},\frac{\log\frac{h_b}{h_a}-\mu_h}{\sigma_h}\right) \tag{7}$$

Default value of constants are $\mu_x = \mu_y = \mu_w = \mu_h = 0, \sigma_x = \sigma_y = 0.1, \sigma_w = \sigma_h = 0.2$. This transformation is implemented below in the offset boxes function. Softmax activation is used as classifier to classify classes and bound box.

Prediction of Bounding Box

Several anchor boxes are constructed for the image during prediction, and each one predicts classes and offsets. As a result, an anchor box with its expected offset yields a predicted bounding box. offset inverse function returns the anticipated bounding box coordinates after applying inverse offset transformations on anchors and offset predictions as inputs.

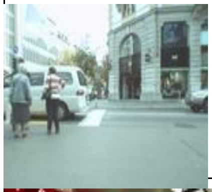
When there are numerous anchor boxes, it is possible for numerous predicted bounding boxes to surround the same item that are similar (and have a large amount of overlap). Merging comparable predicted bounding boxes that belong to the same item using non-maximum suppression (NMS) simplifies the result.

Dataset Description

MOT (Multiple object Tracking) data set is used to test the model for detection of objects in stride of frames. Each sample has minimum of 700 frames extracted from video sequences. Each video has 30 Frame per second, length of the video, description of the videos are given in the table 3.1. 188076 number of Annotated pedestrian per frame is used for training data set and 11297 number of annotated pedestrian is used as test data. .

Each MOT sample has strides of video sequences with COCO Label instance to identify various objects. Aim to detect pedestrian in the video sequences. Each object is generated with detection rate and sequence number. Sort Algorithm are used to track the pedestrians and sorted in the folder for sequences of pedestrian detected frames. Each object is identified by the bounding box which van be easy track to identify the person in the video sequences. This dataset contains scenes happened in different places with different situations.

Table 1. Dataset Description of MOT

Sample	Name	FPS	Resolution	Length	Description
	MOT17-04-DPM	30	1920x1080	1050 (00:35)	Pedestrian street at night, elevated viewpoint
	MOT17-05-DPM	14	640x480	837 (01:00)	Street scene from a moving platform
	MOT17-09-DPM	30	1920x1080	525 (00:18)	A pedestrian street scene filmed from a low angle.

	MOT17-10-DPM	30	1920x1080	654 (00:22)	A pedestrian scene filmed at night by a moving camera
	MOT17-11-DPM	30	1920x1080	900 (00:30)	Forward moving camera in a busy shopping mall
	MOT17-13-DPM	25	1920x1080	750 (00:30)	Filmed from a bus on a busy intersection

IV. RESULTS AND DISCUSSION

Object (persons) is detected in MOT 17-07 pedestrian data set. **Fig 5** represents sample frame before detection and **Fig 6** represents after detection.

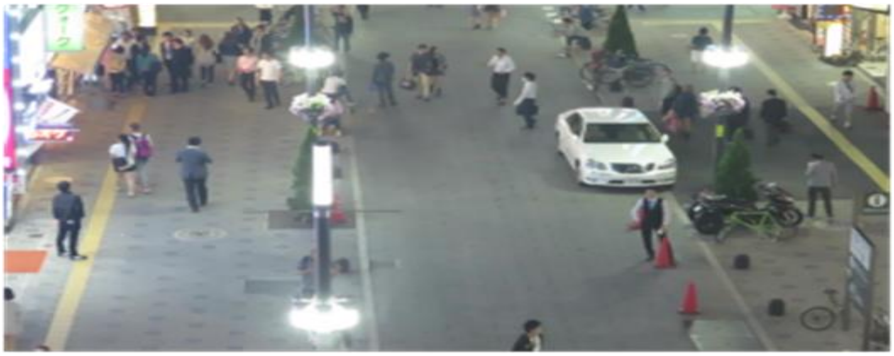


Fig 5. Before detection on Sample Frame

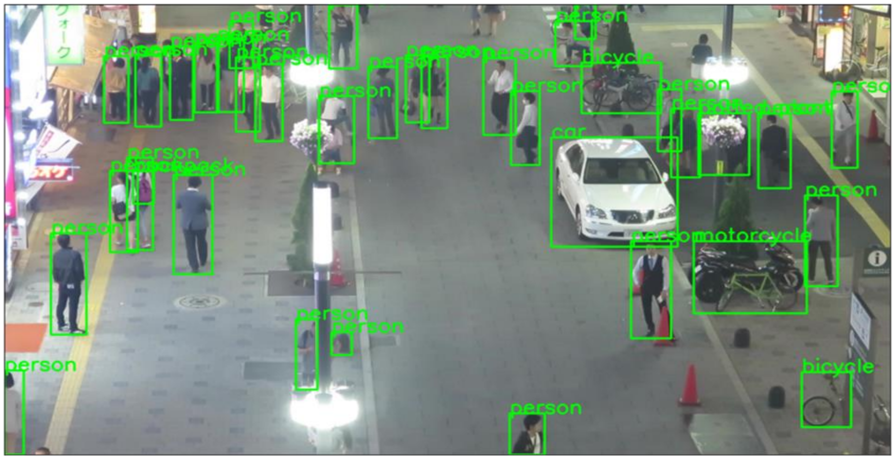


Fig 6. After Detection on sample frame 1

Sample frame of another situation of Multiple object tracking (MOT 17-03) as shown in **Fig 7** and **Fig 8** Before detection of sample frame 2 and after detection of sample frame 2.

Fig 7. Before Detection of sample frame 2

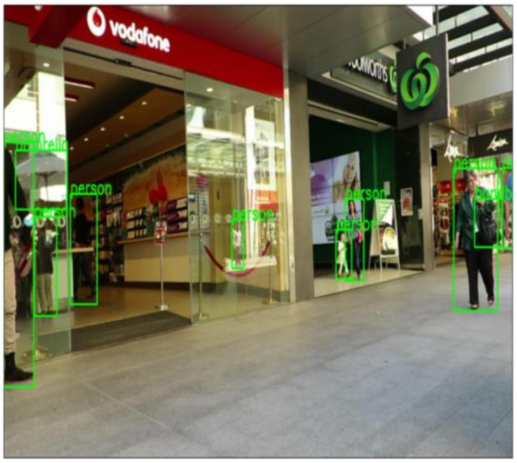


Fig 8. After Detection of sample frame 2

Intersection Over Union (IoU)

IoU overlapped measurement serves as guiding concept in all cutting-edge metrics. It is described precisely as the overlap of the detecting frame and the corresponding ground truth box. IoU is calculated by dividing the amount of overlapping in between bounding boxes box and ground truth through the size of their combination. Intersection over Union score greater than 0.5 is subjected as "good" prediction. Detection rate is based on the Area of union and Area of intersection. IOU is metric is used for detection of objects as shown in **Fig 9.**

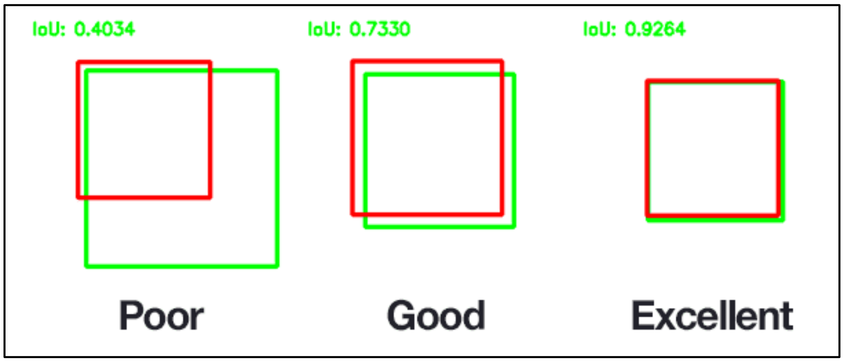


Fig 9. Representation of IOU

Table 2 represents results of detected person of sample 10 images of detection score and bounding size. Label 1 denotes persons. Bounding size value differs from different types of objects. Each Person detected score is with an average of 99.88%. Average accuracy detection rate is calculated on 500 sample frames is 97.33 %. Bounding box size is determined by breath and height of object enclosed in various frames.

MOT dataset is test with existing YOLO model (Yolo only look performance) compared with proposed model From **Table 3** shows the detection score of sample 10 frames using enhanced faster R-CNN model and existing YOLO model. Detection rate of proposed model achieves rate of 99.88 % on sample frames 10 and Yolo achieves 89% of detection rate. Proposed model outperforms well compared to the existing detector models.

Table 2. Detection rate of person with BBox size

Images Number of' Person' Object	Detection score (%)	Bounding box size
1	0.9979	"[1430.85, 447.09,1521.92, 644.14]"
2	0.9976	"[1836.79,431.41,1881.55,632.48]"
3	0.9766	"[1329.65, 552.97,1390.70,631.14],"
4	0.9356	"[1879.07,430.29,1918.32,636.26],"
5	0.9212	"[1678.66,470.33,1717.98,585.08],"
6	0.9023	"[1430.85,447.09,1521.92,644.14]"
7	0.9992	"[1792.53, 397.00,1894.42,668.64]"
8	0.9991	"[1679.89,471.85,1722.43,583.33"
9	0.9983	"[1879.53,428.75,1917.36 635.16"
10	0.9967	"[1319.04,562.11,1378.29,639.17]"

Table 3. Performance metrics of Detection

Images Number of' Person' Object	Detection score (%) using Proposed Model	Detection score (%) using Yolo
1	0.9979	0.9123
2	0.9976	0.8923
3	0.9766	0.8999
4	0.9356	0.9123
5	0.9212	0.9343
6	0.9023	0.7533
7	0.9992	0.7899
8	0.9991	0.8332
9	0.9983	0.8444
10	0.9967	0.9343

Enhanced faster R-CNN achieves 98 % on 100 input frames. When 200 input frames fed in to the model. It gives 95 % on 200 frames, 96 % on 300 frames, and 97 % on 400 frames and finally it achieves 97.33 % on 500 frames as shown in **Table 4.**

Table 4. Comparison metrics of different detectors-Accuracy

Number of frames	CNN (%)	YOLO (%)	Enhanced proposed faster R-CNN (%)
100	80	83	98.1
200	81	84.5	94.9
300	82.5	74.33	96
400	83.5	81	97
500	79.9	82.44	97.33

V. CONCLUSION

This Paper discusses detection of objects (persons) in video sequences. MOT(Multiple object Tracking) dataset is used to detect the objects using Enhanced Faster R-CNN pre-trained model. It detects the objects at the accuracy rate of 97.33 % carried out on 500 test frames. Enhanced Faster R-CNN outperforms well compared to CNN and YOLO detector models. Overall, the findings in the statement indicate that the Enhanced Faster R-CNN model is a promising solution for object detection in video sequences, particularly for the task of detecting persons. The reported accuracy rate and comparison with other models highlight the model's effectiveness, offering potential benefits for a wide range of applications in computer vision and beyond.

Future Enhancement

Enhancing multiple object detection on surveillance videos using Enhanced Faster R-CNN or similar models can benefit from various strategies and advancements. Improving accuracy in surveillance scenarios is crucial for reliable security and analysis. Here are some future enhancements and approaches to consider:

Expanding the dataset used for training to include a wider variety of surveillance scenarios, lighting conditions, weather conditions, and camera angles can help improve the model's robustness. Integrating online object tracking algorithms with the detection model to improve tracking accuracy over time.

These enhancements represent a multifaceted approach to improving the accuracy of multiple object detection in surveillance videos.

Data Availability

The Data used to support the findings of this study will be shared upon request.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding was received to assist with the preparation of this manuscript.

Ethics Approval and Consent to Participate

The research has consent for Ethical Approval and Consent to participate.

Competing Interests

There are no competing interests.

References

[1] B. Geluvaraj, P. M. Satwik, and T. A. Ashok Kumar, “The Future of Cybersecurity: Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Cyberspace,” Lecture Notes on Data Engineering and Communications Technologies, pp. 739–747, Sep. 2018, doi: 10.1007/978-981-10-8681-6_67.

[2] G. Nguyen et al., “Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey,” Artificial Intelligence Review, vol. 52, no. 1, pp. 77–124, Jan. 2019, doi: 10.1007/s10462-018-09679-z.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” Pattern Recognition Letters, vol. 119, pp. 3–11, Mar. 2019, doi: 10.1016/j.patrec.2018.02.010.

[4] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, Nov. 2019, doi: 10.1007/s10994-019-05855-6.

[5] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” Cluster Computing, vol. 22, no. S1, pp. 949–961, Sep. 2017, doi: 10.1007/s10586-017-1117-8.

[6] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep Reinforcement Learning: A Brief Survey,” IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/msp.2017.2743240.

[7] S. Krig, “Feature Learning and Deep Learning Architecture Survey,” Computer Vision Metrics, pp. 375–514, 2016, doi: 10.1007/978-3-319-33762-3_10.

[8] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, vol. 234, pp. 11–26, Apr. 2017, doi: 10.1016/j.neucom.2016.12.038.

[9] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data,” Information Fusion, vol. 42, pp. 146–157, Jul. 2018, doi: 10.1016/j.inffus.2017.10.006.

[10] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning,” Archives of Computational Methods in Engineering, vol. 27, no. 4, pp. 1071–1092, Jun. 2019, doi: 10.1007/s11831-019-09344-w.

[11] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, vol. 234, pp. 11–26, Apr. 2017, doi: 10.1016/j.neucom.2016.12.038.

[12] J. Zhai, S. Zhang, J. Chen, and Q. He, “Autoencoder and Its Various Variants,” 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2018, doi: 10.1109/smc.2018.00080.

[13] R. Thirukovalluru, S. Dixit, R. K. Sevakula, N. K. Verma, and A. Salour, “Generating feature sets for fault diagnosis using denoising stacked auto-encoder,” 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Jun. 2016, doi: 10.1109/icphm.2016.7542865.

[14] L. Wen, L. Gao, and X. Li, “A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 136–144, Jan. 2019, doi: 10.1109/tsmc.2017.2754287.

[15] E. Q. Wu, G.-R. Zhou, L.-M. Zhu, C.-F. Wei, H. Ren, and R. S. F. Sheng, “Rotated Sphere Haar Wavelet and Deep Contractive Auto-Encoder Network With Fuzzy Gaussian SVM for Pilot’s Pupil Center Detection,” IEEE Transactions on Cybernetics, vol. 51, no. 1, pp. 332–345, Jan. 2021, doi: 10.1109/tcyb.2018.2886012.

[16] Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” Artificial Intelligence Review, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/s10462-020-09825-6.

[17] D.-T. Hoang and H.-J. Kang, “A survey on Deep Learning based bearing fault diagnosis,” Neurocomputing, vol. 335, pp. 327–335, Mar. 2019, doi: 10.1016/j.neucom.2018.06.078.

[18] B. Shiva Prakash, K. V. Sanjeev, R. Prakash, and K. Chandrasekaran, “A Survey on Recurrent Neural Network Architectures for Sequential Learning,” Soft Computing for Problem Solving, pp. 57–66, Oct. 2018, doi: 10.1007/978-981-13-1595-4_5

[19] M. Pak and S. Kim, “A review of deep learning in image recognition,” 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Aug. 2017, doi: 10.1109/caipt.2017.8320684.

[20] N. Das, E. Hussain, and L. B. Mahanta, “Automated classification of cells into multiple classes in epithelial tissue of oral squamous cell carcinoma using transfer learning and convolutional neural network,” Neural Networks, vol. 128, pp. 47–60, Aug. 2020, doi: 10.1016/j.neunet.2020.05.003.