Object Detection and Classification Algorithms using Deep Learning for Video Surveillance Applications

Mohana, H. V. Ravish Aradhya

Abstract: Object Classification is a principle task in image and video processing. It is exercised over a multitude of applications ranging from test and number classification to traffic surveillance. The primitive machine learning concepts had provided the pedestal for carrying out umber of image processing tasks. Nowadays requirement of detection algorithm is to work end to end and take less time to compute. Real-time detection and classification of objects from video provide the foundation for generating many kinds of analytical aspects such as the amount of traffic in a particular area over the years or the total population in an area. In practice, the task usually encounters slow processing of classification and detection or the occurrence of erroneous detection due to the incorporation of small and lightweight datasets. To overcome these issues, YOLO (You Only Look Once) based detection and classification approach (YOLOv2) for improving the computation and processing speed and at the same time efficiently identify the objects in the video. Classifier such as Haar cascade which uses Haar like features was primitively used for face detection. Moreover, due to the ever-increasing demand and scope of improvement in the existing fields, the primitive methods need a lot of upgradation. Neural Networks have made the tasks quite plain sailing. Right from the vanilla neural networks to Fast R-CNN and then Faster R-CNN, all models have contributed significantly in the domain of computer vision. This paper mainly focuses in detection and classification ranging from single class objects to multi class objects. The classification algorithm creates a bounding box for every class of objects for which it is trained, and generates an annotation describing the particular class of object. The Haar cascade classifier was trained on a batch of positive and negative samples which were later stitched together to form a vector file and finally form the xml file. On the other hand, COCO dataset used for implementing YOLOv2 and R-CNN algorithm due to the presence of pertained model in it. In addition, use of GPU (Graphics Processing Unit) to increase the computation speed and processes at 40 frames per second.

***Index Terms*: Object classification, detection, YOLOv2, Neural Network, Haar cascade Classifier, Mask R-CNN.**

I. INTRODUCTION

Humans look at an image and instantly process the objects in it and determine their locations due to the interlinked neurons of the brain. The human brain is very accurate in performing complex tasks such as identifying objects of similar attributes, in a very small amount of time. Just like the human interpretation, the world today requires fast and accurate algorithms to classify and detect various objects for many applications. These applications include pedestrian detection, vehicle counting, motion tracking, cancer cell detection and many more [22] [26]. For a human visual system, the perception of visual information is with apparent ease. In artificial intelligence, we face a huge amount of visual information and few useful techniques to process, understand and classify them. The process of object classification and detection workflow aims to classify objects, based on their features and attributes. As days have gone by, many approaches have been incorporated time to time for obtaining better results[4] [6] [7][11]. The object detection approaches have progressed from sliding window-based methods to single shot detection frameworks. The Convolutional Neural Network (CNN), in particular, has numerous applications such as facial recognition, as it achieved a large decrease in error rate but at the expense of speed and computation time[23]. The Region based Convolutional Network (R-CNN) uses the process of Selective Search process in order to detect the objects. The descendent of R-CNN, Fast R-CNN and, Faster R-CNN fixed the slow nature of CNN and R-CNN by making the training process end-to-end. YOLOv2 (You Only Look Once version 2) is an object detection technique in which the detection process is considered as a single backsliding problem which takes an input image and generates the confidence level of each object in the image. It is the descendent of primitive YOLO algorithm [2] [3].

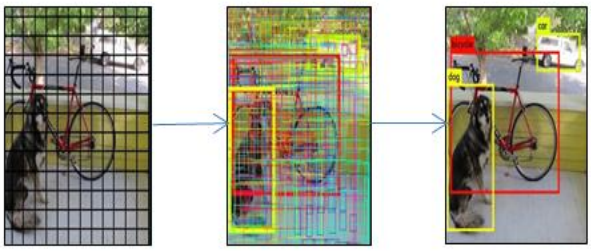


Fig. 1. YOLOv2 model regression.

Fig. 1 depicts a regression model wherein the image, which is given as input, is divided into grids, followed by formation of bounding boxes on all objects and finally detection of the objects as per requirement. The YOLOv2 detection algorithm finds its genesis to the open source deep learning framework known as Darknet.

Revised Manuscript Received on June 05, 2019

Mohana, Telecommunication Engineering, R. V. College of Engineering, Bangalore, India

H V Ravish Aradhya, Electronics & Communication Engineering, R. V. College of Engineering, Bangalore, India

The Darknet is based on GoogLeNet architecture. YOLOv2 is extremely fast and makes fewer background errors than traditional R-CNN approaches. YOLOv2 divides each image into a several grid boxes and each grid box predicts certain bounding boxes and associated confidence levels. The confidence levels reflect the precision of localization of the objects, regardless of the class. Most of the grids boxes and bounding boxes are removed accounting to fewer threshold values, leaving behind the particular class of objects, which it is trained to detect [13][14].

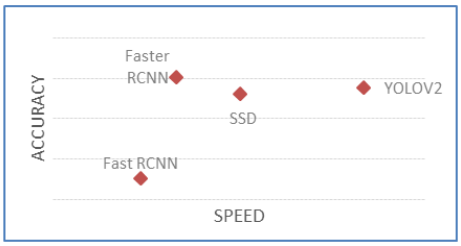


Fig. 2. Speed v/s Accuracy of detection algorithms.

Two main issues that are present in the traditional CNN algorithms mainly motivate the work. These are low accuracy rate and slow computation speed due to the absence of GPU. Fig. 2 shows the graph of speed versus accuracy of various detection algorithms. In this paper, focusing on the working and implementation of YOLOv2 detection algorithm by the YOLO9000 detection system and run it on the video records, which will predict the bounding boxes along with the annotations on the objects. This algorithm is implemented mainly using OpenCV library [27]. The process of object classification and detection workflow aims to classify objects, based on their features and attributes. The object detection approaches have progressed from sliding window-based methods to single shot detection frameworks. If the application demands classification and detection of single type of objects (single class), then one can opt for Haar-Cascade Classifier as one of the classifiers. It increases performance time for detecting a single class, but comes with limited usage where it can detect a single class and with precision. Neural networks such as CNN, R-CNN perform very well in situations, which demand multi class object classification and detection [1] [10].

II. DESIGN AND IMPEMENTATION

Casse-1: Implementation-using YOLO algorithm

Case 1 describes the overall design requirements and implementation of YOLO model on input images. In addition, how the model efficiently and accurately detects and classifies objects by implementing Anchor Boxes and CUDA environment [31]. Fig 3. Shows the flow diagram of YOLO model. YOLO model follows a certain flow method to analyze and detect the objects quickly. Firstly, it follows a regression model wherein it takes the input and derives the class probabilities. Secondly, it calculates the class specific confidence scores. Further, it compares the confidence score with the predefined threshold values to detect and classify the objects. If the confidence score is less than the threshold, value the algorithm does not detect that particular object.

A. Flow Diagram

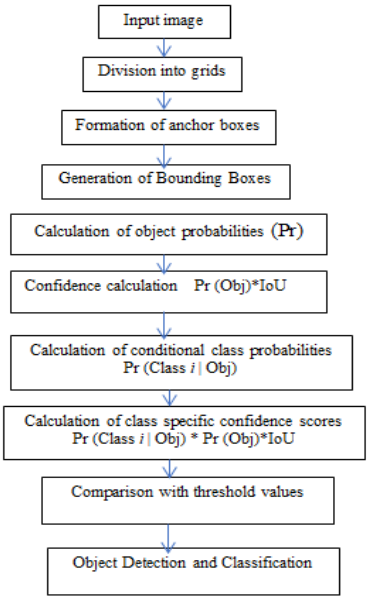


Fig 3. Flow diagram of YOLO Model.

B. Intersection over Union (IoU)

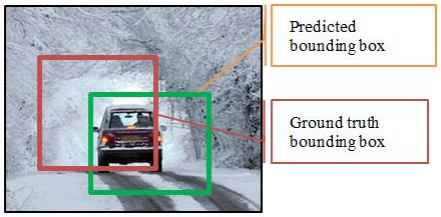


Fig 4. Predicted and ground truth bounding boxes.

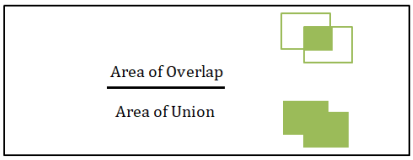
Intersection over Union is a gauging metric that is used to compute the precision of an object detector and classifier on a particular dataset. It consists of two evaluation metrics.

The ground truth bounding box: The hand labeled bounding box of a particular object in an image.

The predicted bounding box: The predicted bounding box from the detection and classification algorithm.

Fig. 4 depicts the hand labelled and the predicted bounding boxes. These help in determining the closest bounding box for a particular object.

IoU can also be defined as:



C. Anchor Box

The YOLOv2 model segments the input image into N×N grid cells. Each grid cell has the task of localizing the object if the midpoint of that object falls in a grid cell. But the grid cell approach can predict only a single object at a time.

If the midpoint of two objects coincides with each other, the detection algorithm will simply pick any one of the objects. To solve this issue, the concept of Anchor Boxes is used.

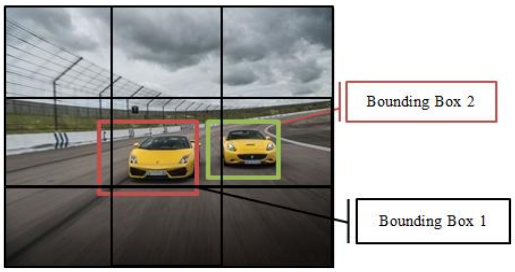


Fig 5. Generation of grid cells and bounding boxes.

In Fig. 5, it is observed that the image consists of two objects (cars) and for ease of explanation chosen N=3 for the number of grids. Here, the image is divided into 3 X 3 grid cells. If the classification and localization algorithm is trained to classify three sets of classes, namely car, person, motorcycle, then the output vector 'Y' of the neural net can be defined as a matrix of 8 possible elements.

$$\begin{bmatrix} P \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix} \quad (1)$$

Equation 1 describes the attributes of an object in an input image. 'P' defines the presence of object in the grid cell which can take values either 0 or 1, 'bx' and 'by' defines the coordinates of the midpoint of the object in a particular grid, 'bh' defines the percentage height of the bounding box of the total height of the grid cell, 'bw' defines the percentage width of the total width of the grid cell and c1, c2 and c3 defines the classes namely person, car and motorcycle. The target volume output will be of order 3 X 3 X 8, where 8 is the number of dimensions defined for this particular classification example.

Consider now a case where two objects share the same midpoint. In this situation, the approach of Anchor Boxes is implemented.

Fig 6. Implementation of Anchor Boxes [8].

In Fig. 6, two objects are sharing the same midpoint. The system of objects now generates an output variable 'Y' as a matrix of 16 elements.

$$Y = \begin{bmatrix} P \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \\ --- \\ P \\ bx' \\ by' \\ bh' \\ bw' \\ c1 \\ c2 \\ c3 \end{bmatrix} \quad (2)$$

Equation 2 depicts the possible attributes of the system of objects where the new parameters bx', by', bh', bw' are the bounding box parameters of the second object. The ground truth bounding box associated with a particular object is compared with the Anchor Boxes, and the IoU is determined. The object whose IoU metric is maximum will be coded and detected. For instance, in Figure 5 if car is to be detected, then the output variable Y will take the values as shown in equation 3.

$$Y = \begin{bmatrix} 0 \\ D.C \\ D.C \\ D.C \\ D.C \\ D.C \\ D.C \\ D.C \\ --- \\ 1 \\ bx' \\ by' \\ bh' \\ bw' \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (3)$$

Don't care

Detection of car

The YOLOv2 algorithm requires a specific set of platforms as it extensively uses the GPU of the system. These platforms readily increase the speed and performance of the algorithm. For a Windows based system, the algorithm requires the help of Microsoft Visual C++ platform

D. CUDA (Compute Unified Device Architecture)

CUDA is a computing platform created by Nvidia in order to perform general purpose computing on the Graphics Processing Unit (GPU). It works extensively with programming languages like C /C++. The slow speed of a CPU is a serious hindrance to productivity for any image processing computation. CUDA has built-in features that enable it to associate a series of threads to each pixel so that speed is uncompromised.

The CUDA environment supports heterogeneous programming that involves a host, that primarily works on the CPU and a device, that consists of the graphics card interfaced with the GPU. The host and the device work hand in hand to improve the workflow and computation speed. The host is responsible for allocating share in memory for the program variables, and the device improves the speed of the computation. In YOLOv2, performance is the main criteria and to achieve a non-dispensable output, CUDA environment plays a vital role. In real time scenario, reduction of noise and redundancy from the objects that are being detected and classified is important. The CUDA environment incorporates a library cuDNN, which provides GPU accelerated functionality in Deep Neural Networks. This environment speeds up the process of smoothening (reduction of noise) and edge detection by manifold due to the associated thread approach and the device-host paradigm.

E. YOLO 9000

YOLO 9000 is a significant improvement to the original object detection system (YOLO). The earlier version of YOLO gave a speed of 35 FPS or 22ms/image. It also was behind in terms of accuracy when compared to other methods like RCNN and Fast RCNN. There is, therefore, the need to make the original YOLO version better and faster. There is also a need to improve the accuracy. General object detectors pre train on ImageNet dataset on 224*224 and then resize the network to 448*448. Later, they fine-tune the model on detection. This version however trains a bit longer upon resizing before fine-tuning. This increases the mean average precision (mAP) by 3.5%. The earlier YOLO version used the bounding box technique wherein the coordinates of the X, Y, width and height were obtained. This version uses the anchor box technique, which calculates the offsets for images. The offsets are calculated from candidate boxes or the reference points [30].

YOLO 9000 has also brought the concept of multi scale training into limelight. General object detection systems train at a single aspect ratio (448*448). On the contrary, the new version resizes the network randomly during the training process on a bunch of different scales. The detector is trained on image scales from 320*320 to 608*608. We, therefore, get a network we can resize at test time to a bunch of different sizes and without changing weights that we have trained. We can run the detector at different scales, which gives us a trade-off between performance (speed) and accuracy.

There is a dearth of training data in the data detection models. Hence, there is a need for data augmentation. This is also called the Joint training method. The Common Objects in Context (COCO) dataset is used as the detection data set. Although, the COCO dataset can detect multiple objects in an image accurately, however it is confined to only 80 classes. We use the ImageNet as our classification dataset for the model that has 22000 classes. ImageNet, however, labels only one object that is in focus and not on multiple objects in the image. We, therefore, combine the detection and classification datasets and then use backpropagation technique to determine the exact location and class of the image [5]. This result in better performance, more accuracy, and low latency compared to the earlier version of YOLO [22].

Casse-2: Implementation-using RCNN algorithm

Case 2 describes the design and implementation of algorithms such as Haar- cascade classifier for single class detection and mask R-CNN for multi class detection.

F. Haar-Cascade Classifier

The Haar cascade classifier depends on a calculation proposed by Viola and Jones. Haar-Like feature is a rectangular basic element that is utilized as an information highlight for cascaded classifier.

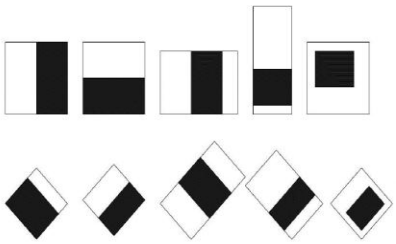


Fig 7. Different kinds of detectors based on Haar-Like feature[15].

Fig. 7 shows the various kinds of detectors based on Haar-Like feature. It contains different type of filters based on Haar-Like feature. By applying every one of these filters into one special area of the image, the pixel sums under white areas are subtracted from the pixel sums under the black areas. That is the weight of white and black area can be considered as "1" and "-1", respectively. A Haar like feature contemplates neighboring rectangular districts at a specific territory in an acknowledgment window, totals up the pixel powers in each unmistakable edge and figures the difference between these aggregates. This refinement is then used to arrange subsections of a picture. The basic segments of Haar-course are Integral pictures, Adaboost calculation and Cascaded classifier[28].

Integral images: The underlying step of the Viola-Jones algorithm's estimation is to change the data image into an essential image. Fig. 8 depicts the integral image.

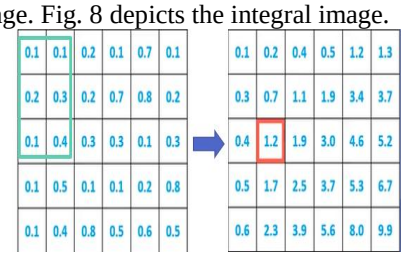


Fig.8. Integral image [5].

Haar classifier compares how close the real scenario is to the ideal case. Ideally, the value of Haar feature is 1.

Δ = dark-white

$$\Delta = \frac{1}{n}\sum I_1(x) - \frac{1}{n}\sum I_2(x) \quad (4)$$

Adaboost Algorithm: It is a machine learning algorithm used to develop classifier through weights.

. A weak classifier is numerically depicted as

$$h(x,f,p,\theta) = \begin{cases} 1, & pf(x) > p\theta \\ 0 \ otherwise \end{cases} \quad (5)$$

Cascaded Classifier: Cascaded classifier is used for fast rejection of error windows and improving the processing speed. In every node of trees there is a non-vehicle branching, it means that the image will not be vehicle. By this technique, the false negative rate is at least.

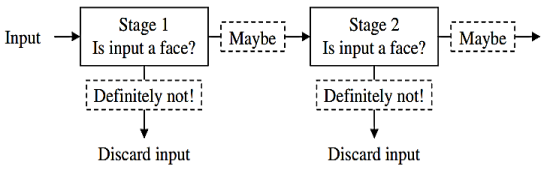


Fig 9. Cascaded classifier [8].

Fig. 9 describes the stages of cascaded classifier. In a solitary stage classifier, one would ordinarily acknowledge false negatives with the end goal to decrease the false positive rate. Implementation of Haar cascade for single class object detection:

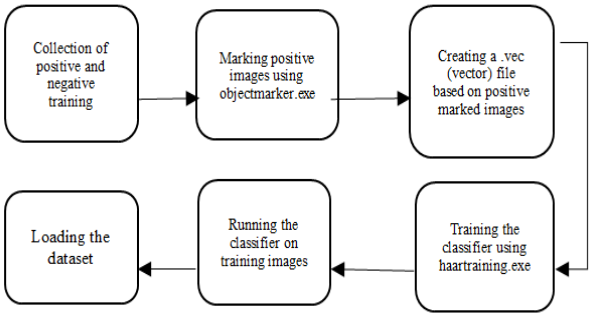


Fig 10. Block diagram of training process in Haar classifier.

Fig.10 describes training process in Haar cascade classifier. To prepare the distinctive phases of the cascade classifier the AdaBoost algorithm requires to be sustained with positive precedents – that is, images of vehicles. It is taken from Stanford Car Dataset. This Cars dataset contains 16,185 images of 196 classes of cars. In this paper, we have taken 2000 positive images of cars.

G. Mask Region based Convolutional Networks

Mask R-CNN has been the new condition of workmanship as far as occasion division. It is a significant neural framework planned to handle event division issue in machine learning or PC vision. Figuratively speaking, it can seclude unmistakable questions in a picture or a video. One can give it a picture; it gives question ricocheting boxes, classes and veils. Mask R-CNN, widens Faster R-CNN by including a branch for anticipating division veils on each Region of Interest (RoI), in parallel with the current branch for classification and hopping box backslide. The mask branch is a little FCN associated with each ROIs, envisioning a division cover in a pixel-to pixel way. On a basic level Mask R-CNN is a natural extension of Faster R-CNN, yet building up the mask branch suitably is fundamental for good results [20]. Most importantly, Faster RCNN was not planned for pixel-to-pixel course of action between system data sources and yields. This is most clear in how RoI Pool the genuine focus action for dealing with cases, performs coarse spatial quantization for highlight extraction. To fix the misalignment, a fundamental, non-quantization layer, called RoIAlign, that reliably spares right spatial regions [9].

Architecture of Mask RCNN: The architecture mainly comprises of Faster Region based convolutional code (Fast R-CNN) and FCN (fully connected network) as shown in Fig. 11. Both approaches collectively give rise to a seamless approach known as Instance segmentation, which is the building block of Mask R-CNN. The Faster RCNN is proposed to do the purpose of bounding box object detection and the FCN is used to form the masks around each mask. The first step to Mask R-CNN is implementation of R-CNN. It is an approach to bounding box object detection, where it creates a number of regions or Region of Interest (RoI). The next step is the incorporation of a better version of R-CNN, which is Faster R-CNN. It incorporates an attention mechanism using a Regional Proposed Network (RPN) as shown in Fig. 12. It performs the action of bounding box in two stages as shown in Fig. 13. First, determining the RoI using the RPN protocol. Second for each RoI, the class labels of each RoI are determined. This is done using RoI pooling.

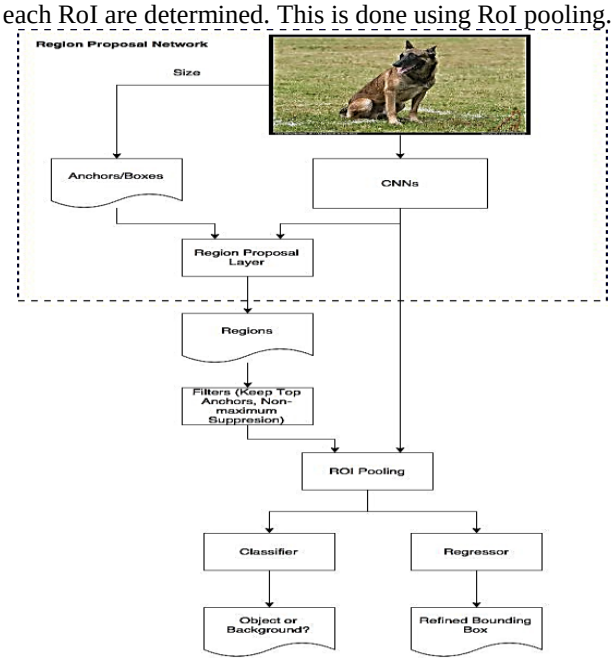


Fig 11. The architecture of Mask R-CNN [9].

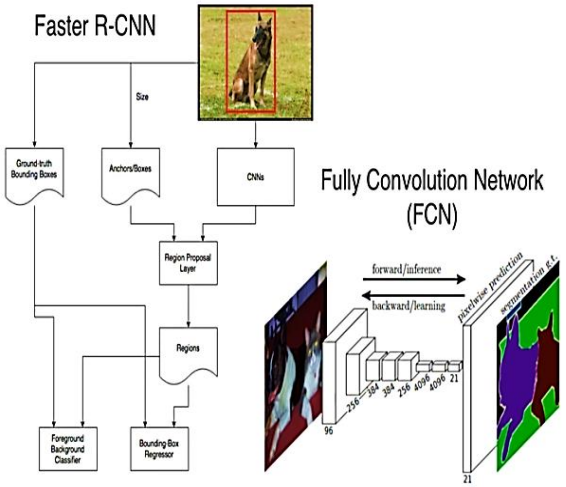


Fig 12. Use of Region Proposed Network (RPN) [24].

The Faster R-CNN has a disadvantage of data loss because of quantized stride used in pooling process. This involves the incorporation of max pooling function. Consider a RoI of 17 x 17 and it need to map it into a space of 7 x 7, the required stride will be 2.42, which is an ambiguous and meaningless value as shown in Fig.14. To counteract this stride value is quantized to 2. In doing so only the top 14x14pixels will be considered and the remaining points will be lost [24].

To address the problem RoI align is used as shown in Fig. 15. In the stride, value is not quantized. Rather each cell is divided into a 2x2 bin, that creates four regions. Each sub cell is then pooled

through bilinear interpolation leading to four values per cell. The final cell value is then computer through either a sum or average of the four sub-cell values.

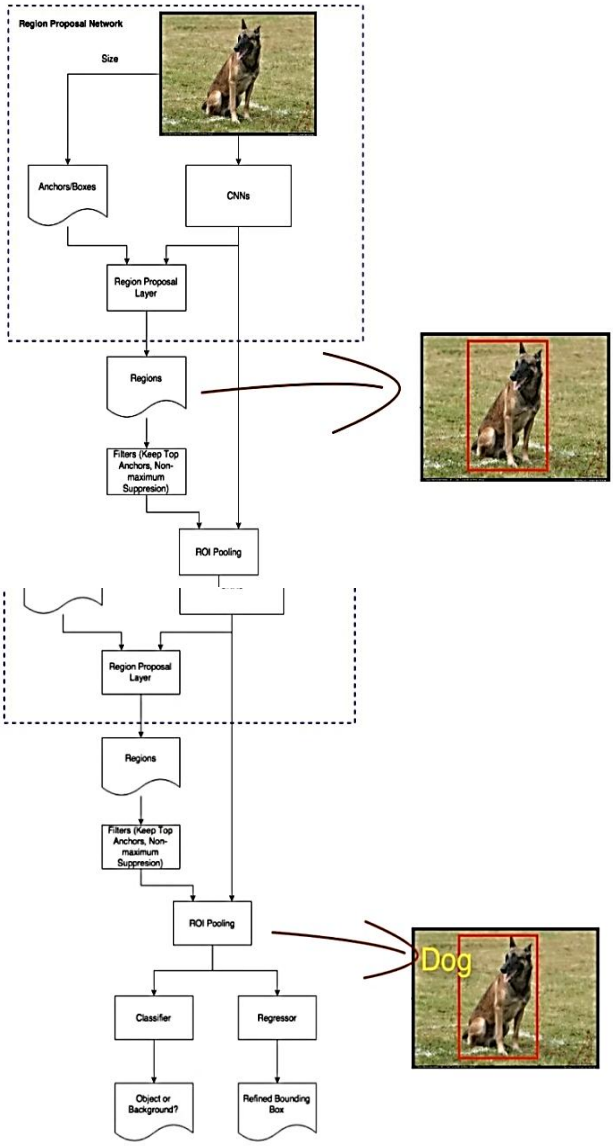


Fig 13. Determination of RoI and class label [24].

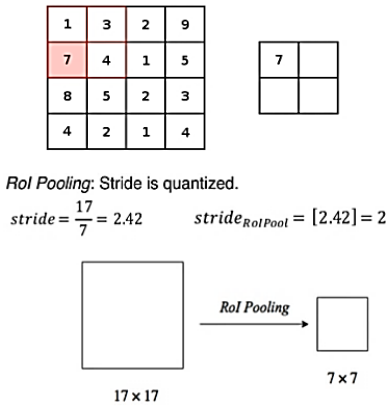


Fig 14. Max pooling and quantization of stride.

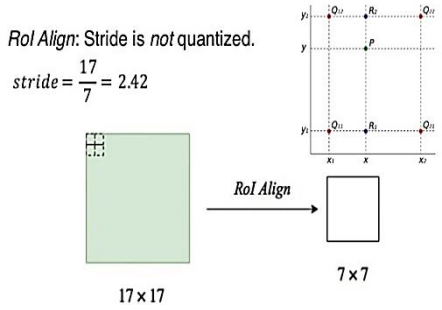


Fig 15. RoI Align operation.

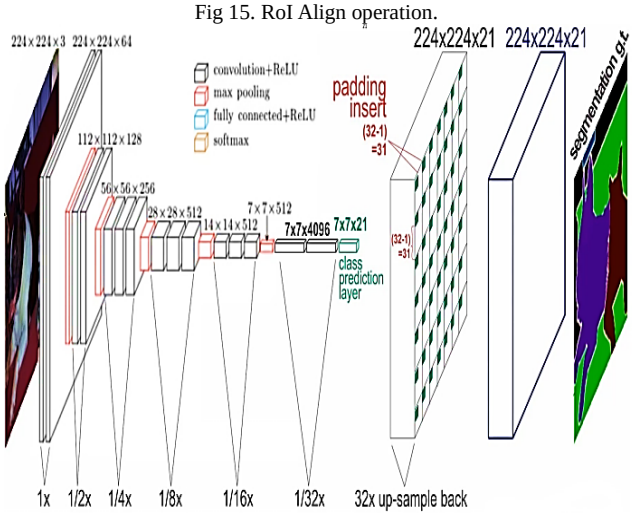


Fig 16. A detailed Fully Connected Layer architecture [12].

The formation of mask is the next step after the bounding box prediction. The Fully Connected Network (FCN) is used for generating the masks. FCN are designed naturally in a pixel to pixel alignment. A number of FCN layers are incorporated in every RoI for generating deterministic masks. As shown in Fig.16, the first stage is of same dimension as that if the input image. The last stage has a dimension reduced by a factor of 1/32 due to the activation and pooling operations on the feature maps. So, at the end we have to up sample the feature map (tensor) so that we obtain a map of same dimension as that of the input. After the class prediction layer, the tensors are up sampled by zero padding and performing normal convolutions .

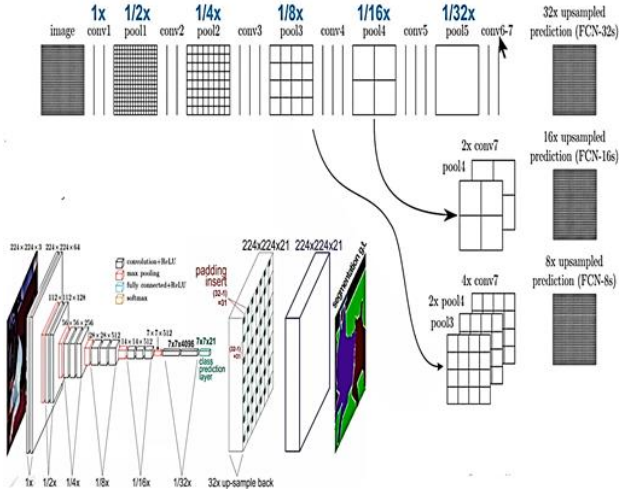


Fig.17. Different versions of FCN [12].

There are various versions in FCN, namely FCN-32s, FCN -16s and FCN-8s. The versions vary with increased accuracy in prediction of masks. As shown in fig. 17,

the FCN-16 has an additional layer which is equal to the convolution of pool 4-pixel values and two times up sampled pixel values of conv-7 layer. At the same time these two layers are replaced by a single layer as mentioned. Now consider an input with 'k' classes with a Region of Interest of a particular type of class in the k classes, having a 'm x m' pixel dimension. For each class in k, a binary mask of dimension m x m is created. Hence a loss of 'km2' is incurred. After that a particular color for a mask is blended onto the pixels from the code and the associated intensity can also be controlled from the code.

III. RESULTS AND ANALYSIS

Casse-1: Results of YOLO algorithm

For YOLOv2 algorithm to execute and detect the objects, we have employed Microsoft Visual C++ 2017 to build the *.exe* file. We have implemented the pre-trained yolo9000 weights and its configurations. Our system consists of a NVIDIA GEFORCE 940 MX enabled GPU. The results in this section depicts the performance of the YOLOv2 model on both still images and on video records. Fig. 6 and Fig. 18 portrays the detection and labelling of the objects in a single image.

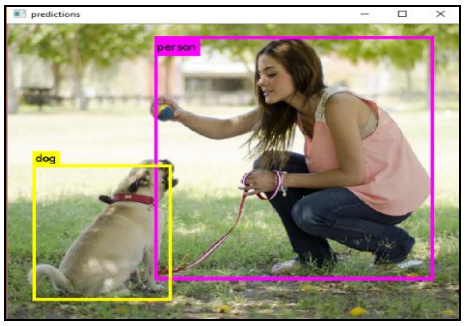


Fig 18. Detection and labeling of two objects.

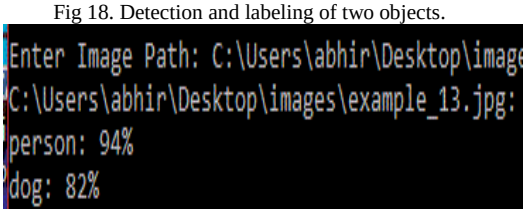


Fig 19. Confidence levels of two objects.

In Fig. 18, there are two objects and it detects them with comprehensive confidence levels, as shown in Fig. 19. The time for computing the detection algorithm for this image was close to 0.49s.

Fig 20. Detection and labelling of multiple objects.

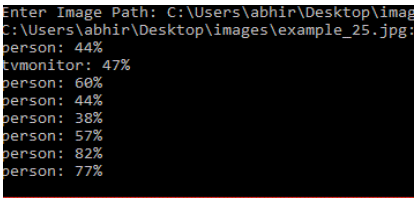


Fig 21. Confidence levels of multiple objects.

Further increase the number of objects in an image the speed of execution doesn't drop. The YOLOv2 model detects majority of the objects with a proficient confidence level. This is portrayed in Fig. 20 and Fig. 21, which has more number of objects compared to Fig. 18. The time for execution for this image was close to 0.5s.

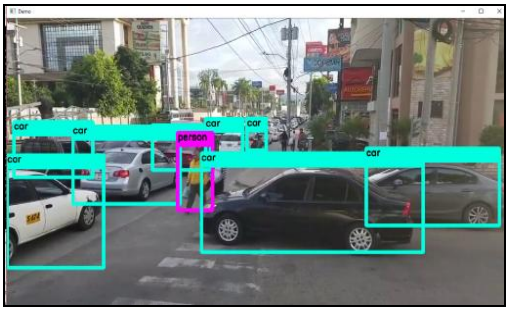


Fig 22. Detection and Labeling of single object in multiple instances.

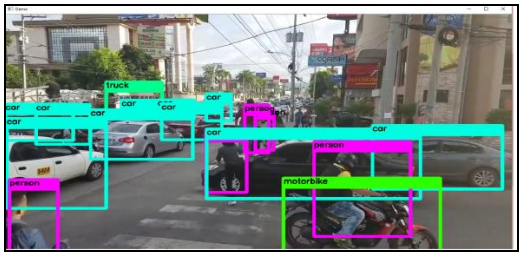


Fig 23. Detection and Labeling of multiple instances of multiple objects.

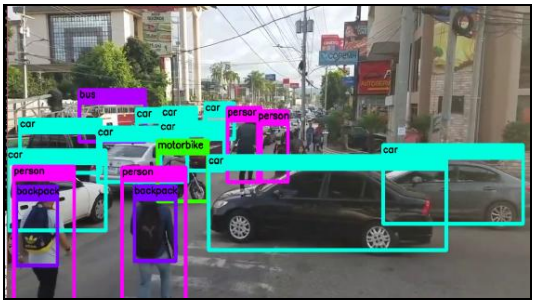


Fig 24. Real time object detection and labeling of various objects on a road.

As if it moves from images to video inputs, the scenario completely changes. The objects in a video will now continuously change its co-ordinates. YOLOv2 algorithm here is continuously detect and label the objects with a proficient confidence level. We have taken some still images from the video record that we had given as input. Fig. 22, 23 and 24 depicts the variation in the number of objects in the video. As the number of objects kept increasing, it didn't affect the detection of other neighboring objects. It gives good detection and classification performance.

Casse-2: Results of RCNN algorithm

Case 2 describes the obtained simulation results of RCNN algorithm and performance analysis.

A. Detection of single class objects in videos using Haar cascade classifier

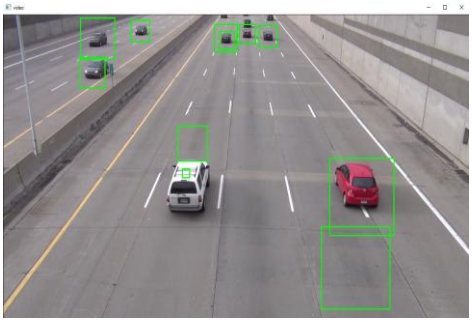


Fig 25. Erroneous detection by the classifier

Fig. 25 shows some faulty detection of cars due to plausible a smaller number of positive samples compared to negative samples. As we increase the number of positive samples, it results in fine tuning of the classifier. After that, the Adaboost algorithm can efficiently cascade off the weak classifiers. The more efficient detection is shown in Fig 26 (a) and (b).

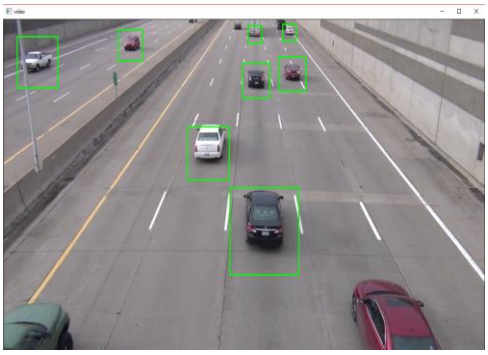


Fig 26 (a). Efficient detection by the classifier.

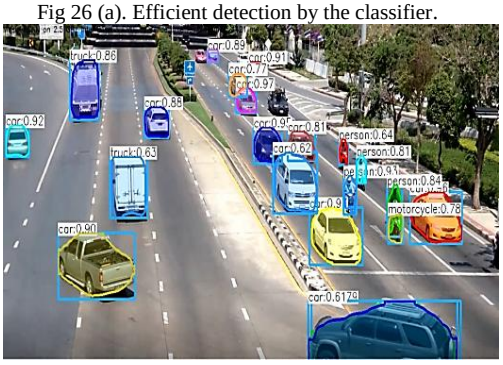


Fig 26 (b). Efficient detection by the classifier.

B. Detection and Classification of Multi Class Objects using Mask R-CNN

Detection in static input

The Mask R-CNN is the state of art detection and classification algorithm. It can efficiently detect multiple classes of objects for the given input. The approach used, it is known as instance segmentation.

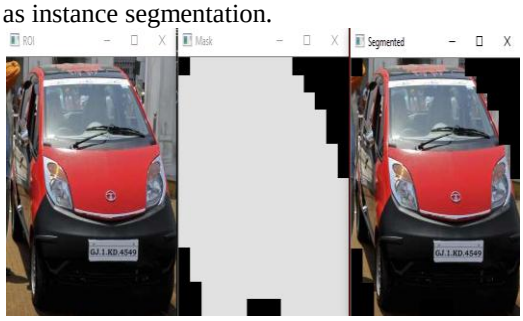


Fig 27. (a) Formation of RoI (b) Mask Formation (c) Segmentation.

As shown in Fig. 27(a), the first step is the formation of Region of Interest (RoI) to form bounding box around the object, next step as shown in Fig. 27(b), formation of mask and at the end network dealiantes the concerned object as shown in Fig. 27(c).

Fig 28. Blending the mask with colour.

After the mask generation is done for each object, confidence levels are obtained from FCN layer of network and masks are blended with distinct colours. As shown in Fig. 28, class labels along with their confidence levels are annoted onto the particular class of object.

Detection of objects in videos

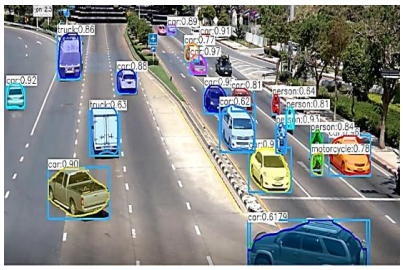


Fig 29. Detection and mask formation of vehicles.

The process of classification works on same lines in case of video records. Apparently in videos the frames are read at a continuous rate. While training such algorithms with regards to multiple class objects is how efficiently can an algorithm detect the objects in close vicinity to each other. As shown in Fig. 29, it consists of vehicles which are not very close to each other. The speed at which the frames are read is quite slow because the process is single handedly working on the CPU with configuration of 2.5 GHz Intel core i5 CPU. The inference time for reading each frame was within a range of 4000ms to 5500ms. The frames per sec obtained at the output are around 0.22.

C. Performance Analysis

```
===== TRAINING 10-stage =====
<BEGIN
POS count : consumed   1900 : 1909
NEG count : acceptanceRatio    900 : 0.00118292
Precalculation time: 18.116
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3| 0.999474|     0.75|
+----+---------+---------+
|   4|        1| 0.768889|
+----+---------+---------+
|   5| 0.999474| 0.748889|
+----+---------+---------+
|   6|        1| 0.735556|
+----+---------+---------+
|   7| 0.999474| 0.627778|
+----+---------+---------+
|   8| 0.999474| 0.624444|
+----+---------+---------+
|   9| 0.999474| 0.398889|
+----+---------+---------+
END>
```

Fig 30. Training of Haar classifier.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Fig.30 shows the Training of Haar classifier. N is the no. of used features, H R is the hit rate based on threshold stage, FA is false alarm based on threshold stage. Fig. 31 shows display of output frames.

TABLE I. ACCURACY FOR MULTIPLE FRAMES IN HAAR CASCADE CLASSIFIER

Frames	TP	FP	TN	Total Vehicles	Accuracy
Frame 1	4	0	4	8	50%
Frame 2	6	0	2	8	75%
Frame 3	6	0	3	9	67%
Frame 4	4	0	4	8	50%
Frame 5	4	0	5	9	45%
Frame 6	5	0	3	8	62.5%
Frame 7	5	0	2	7	71%
Frame 8	4	1	4	8	50%
Frame 9	4	0	3	7	57%
Frame 10	5	0	4	9	56%

Mean True Positives- 4.7
Mean False Positives- 0.1
Mean True Negatives- 3.4
Mean Total Vehicles- 8.5
Mean Accuracy- 58.35%
Range of Accuracy- 45%-75%

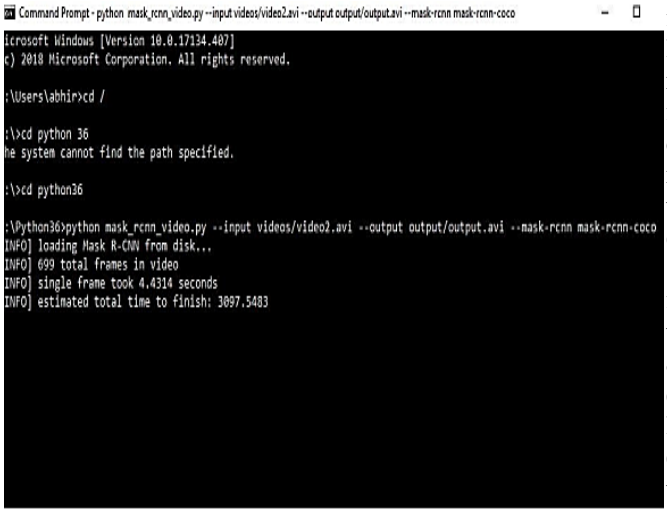


Fig.31. Displaying the output Frame per second values.

Table 1 shows the accuracy of detection in multiple frame using Haar cascade classifier. Obtained results shows that the YOLO and RCNN algorithm gives better performance.

IV. RESEARCH CHALLENGES

Video Surveillance has created a booming impact in the world of automated surveillance. This work is a base work analysis on the state of art algorithms which can be used for a multivariate application [17] [18] [19]. In this paper considered only traffic surveillance applications like accident detection [16]. Sensing modalities such as Multi-modal detection has seen some developments in the recent years where depth of thermal cameras is tested. Work is going on in the domain to increase the sensitivity of modal to detect slightest of infrared radiation given out by objects. A very practical application of image and video classification is the development of Augmented Reality [16] [21] [29]. The transposition of digital information on top of what we find on the planet is not anymore, a cutting-edge dream. Work has already begun in this domain and a number of games had been developed[32][33]. Developers are now working to power up the augmented reality by incorporating neural network training and use it in crowd behavior monitoring and augmented reality advertisement. The most important scope of classifying and detecting of variety of objects is in the powering of self-driven cars. To empower self-governing driving, man-made brainpower is being instructed to perceive different objects on streets. These include pathways, moving vehicles, etc [25]. Still work is going on to eliminate human intervention and increase seamless performance.

V. CONCLUSIONS

Classification and detection of objects have been the state-of-art approach for many areas in computer vision. In the domain of video surveillance classification of objects have been a major breakthrough. In this paper, specifically analyzed the working and performance of two state-of-art algorithms, which can be used for the purpose of video surveillance. In this paper, first introduced YOLOv2 model and YOLO9000, real-time detection systems for detecting and classifying objects in video records. YOLOv2 is agile and efficient in detecting and classifying the objects. The speed and accuracy were achieved with the aid of GPU functionalities and Anchor Box technique respectively. Furthermore, YOLOv2 can detect object movement in video records with a proficient accuracy. YOLO9000 is a real-time framework, which is able to optimize detection and classification and bridge the gap between them. YOLOv2 model and YOLO 9000 detection system collectively are able to detect and classify objects varying from multiple instances of single objects to multiple instances of multiple objects.

There was great deal of variation obtained when it switched from single class detection to multi class detection in the terms of system performance trade-offs. The Haar classifier is able to detect fast-moving vehicles quite smoothly but it can detect only single class objects. On the other hand, Mask R-CNN is able to outperform cascade classifier in terms of detecting multiple objects of different class, but at the same time runs at a slower rate with an inference time of 4.5 to 5.5 s per frame. The frames per second obtained is observed and it depend on the input size. It varied between 0.2 fps to 0.5 fps during the processing of the algorithm. The dataset used in Mask R-CNN is COCO dataset, which consists of 90 types of classes.

REFERENCES

1. Navjot Kaur et.al., "Object classification Techniques using Machine Learning Model", *International Journal of Computer Trends and Technology (IJCTT)* – Vol. 18, Dec 2014.
2. Pawan Kumar Mishra et.al.,"A Study on Classification for Static and Moving Object in Video Surveillance System", *International Journal of Image, Graphics and Signal Processing*, 2016.
3. H. V. Ravish Aradhya et.al., "Real time objects detection and positioning in multiple regions using single fixed camera view for video surveillance applications", *International Conference on Electrical Electronics Signals Communication and Optimization (EESCO)*, 2015, pp.1-6.
4. S. K. Mankani et.al., "Real-time implementation of object detection and tracking on DSP for video surveillance applications", *International Conference on Recent Trends in Electronics, Information & Communication Technology*, 2016, pp. 1965-1969.
5. Y. Li et.al.," Eye-gaze tracking system by haar cascade classifier" 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, 2016, pp. 564-567.

6. S. Sajjanar et.al.,"Implementation of real time moving object detection and tracking on FPGA for video surveillance applications", *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics*,2016, pp. 289-295.
7. Mohana et.al., "Elegant and efficient algorithms for real time object detection, counting and classification for video surveillance applications from single fixed camera" *International Conference on Circuits, Controls, Communications and Computing* (I4C),2016, pp. 1-7.
8. Ekrem Başer et.al., "Detection and classification of vehicles in traffic by using haar cascade classifier", *58th ISERD International Conference, Prague, Czech Republic*, 23rd-24th December 2016.
9. K. He et.a., "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988.
10. S. Zhang et.al.,"New Object Detection, Tracking, and Recognition Approaches for Video Surveillance over Camera Network," *in IEEE Sensors Journal*, vol. 15, no. 5, pp. 2679-2691.
11. Shu Wang et.al.,"Vehicle Type Classification via Adaptive Feature Clustering for Traffic Surveillance Video", 2012, *Broadband Wireless Communication and Sensor Network Technology*, China.
12. Arsalan Mousavian et.al.,"3D Bounding Box Estimation Using Deep Learning and Geometry", 2017 *IEEE Conference on on Computer Vision and Pattern Recognition (CVPR).*
13. V. P. Korakoppa et.al.,"Implementation of highly efficient sorting algorithm for median filtering using FPGA Spartan 6", *International Conference on Innovative Mechanisms for Industry Applications*, 2017, pp. 253-257.
14. V. P. Korakoppa et.al.,"An area efficient FPGA implementation of moving object detection and face detection using adaptive threshold method,"*International Conference on Recent Trends in Electronics, Information & Communication Technology*, pp. 1606-1611.
15. Priyadarshini N K et.al.,"Analysis of moving objects in videos" GSJ: Volume 6, Issue 1, January 2018.
16. Shu Wang et.al., "Vehicle Type Classification via Adaptive Feature Clustering for Traffic Surveillance Video", 2012, Broadband Wireless Communication and Sensor Network Technology, China.
17. Ipek Baris et.al.,"Classification and tracking of traffic scene objects with hybrid camera systems", 2017 IEEE *20th International Conference on Intelligent Transportation Systems (ITSC)*.
18. B. Karasulu et.al., "Moving Object Detection and Tracking in Videos", Performance Evaluation Software, Springer Briefs 7 in Computer Science,pp.7-30, 2013.
19. M R, Sunitha et.al.,"A Survey on Moving Object Detection and Tracking Techniques", *International Journal Of Engineering And Computer Science*, 2016.
20. Ren S et.al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.", 2017 *IEEE Transactions on Pattern Analysis & Machine Intelligence.*
21. Krishna C V et.al., "A review of Artificial Intelligence methods for data science and data analytics: Applications and Research Challenges" *International Conference on I-SMAC (IoT in social, mobile, Analytics and cloud)*, 2018, pp. 599-601.
22. Mohana et.al., "Simulation of object detection Algorithms for video Survillance applications", *International Conference On I-SMAC (IoT in social, mobile, Analytics and cloud),(I-SMAC- 2018)*, pp. 664-668.
23. Manjunath Jogin et.al., "Feature extraction using Convolution Neural Networks (CNN) and Deep Learning" *International Conference On Recent Trends In Electronics Information Communication Technology*, 2018, pp.2319-2323.
24. B. T. Nalla et.al., "Image Dehazing for Object Recognition using Faster RCNN," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 01-07.
25. Mahapatra et.al., "Simulation Based Algorithm for Tracking Multi-Moving Object Using Gaussian Filtering with Lucas-Kanade Approach", *Procedia Computer Science*, pp.790-795.
26. T. Xu et.al., "An improved TLD target tracking algorithm," 2016 IEEE *International Conference on Information and Automation (ICIA), Ningbo*, 2016, pp. 2051-2055.
27. G. Chandan et.al., "Real Time Object Detection and Tracking Using Deep Learning and OpenCV", *International Conference on Inventive Research in Computing Applications (ICIRCA), 2018*, pp. 1305-1308.
28. Nehashree M R et.al., "Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications" *International Conference on Intelligent Sustainable Systems (ICISS-2019).*
29. Meghana R K et.al., "Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model" *International Conference on Computing Methodologies and Communication (ICCMC 2019).*
30. Joseph Redmon et.al., "YOLO9000 Better, Faster, Stronger", 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*
31. Shunji Funasaka et.al., "Single Kernel Soft Synchronization Technique for Task Arrays on CUDA-enabled GPUs, with Applications", 2017 *Fifth International Symposium on Computing and Networking (CANDAR).*
32. Apoorva Raghunandan et.al., "Object Detection Algorithms for video surveillance applications", *International conference on communication and signal processing (ICCSP)*,2018, pp. 0570-0575.
33. Akshay Mangawati et.al., "Object Tracking Algorithms for video surveillance applications", *International conference on communication and signal processing (ICCSP)*, 2018, pp. 0676-0680.

AUTHORS PROFILE

Mohana Born on January 11, 1985 in Karnataka, India, obtained his BE degree in Telecommunication from RV College of Engineering in 2008, M. Tech degree in computer science & Engineering from RV College of Engineering in 2012. He is currently working at RV College of Engineering, Bangalore 560059, as assistant professor in Electronics and Telecommunication Engineering Department from past 10 years. He is a life member in Indian Society for Technical Education (ISTE). His research interests are in the areas of signal and image processing, Deep Learning and Artificial Intelligence.

H V Ravish Aradhya born on March, 27th, 1969 in Karnataka, India, obtained his BE degree in Electronics from RV College of Engineering in 1991, ME degree in Electronics from University Visvesvaraya college of Engineering, Bangalore, in 1995, and Ph D degree from Visvesvaraya Technological University, Belagavi, India in 2014.
He is currently serving RV College of Engineering, Bangalore 560 059, as professor in Electronics & Communication Engineering Department the past 23 years. With a vast 26 years of teaching experience, he has thirteen text books written/adapted/reviewed for leading publishers like McGraw-Hill and Pearson Education, guided many undergraduate and post graduate projects. To his credit; he has 68 International & National journal/conference papers presented. He is a life member in Indian Society for Technical Education (ISTE) and Institution of Electronics and Telecommunication Engineers (IETE). His research interests are in the areas of VLSI design, embedded systems, Microprocessor and Microcontroller applications, and Computer networking.