International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

An Improved Pedestrian Detection Algorithm using Integration of Resnet and Yolo V2

Geethapriya. S, P. Kumar

Abstract: ***Pedestrian detection is one of the important tasks in object detection technology. The pedestrian detection algorithm has been used in applications like intelligent video surveillance, traffic analysis, and autonomous driving. In recent years, many pedestrian detection algorithms have been proposed but the key drawback is the accuracy and speed, which can be improved my integrating efficient algorithms. The proposed model improves the pedestrian detection algorithm by integrating two efficient algorithms together. The model is developed using the joint version of ResNet and YOLO v2, which preforms feature extraction and classification respectively. By using this model the efficiency of the system is increased by improving the accuracy rate so that it can be used with real-time applications. The model has been compared with existing models like SSD, Faster R-CNN and Mask R-CNN. Comparing with these models, the proposed model provides mAP value higher than these existing models with less loss function when tested on the INRIA dataset.***

Keywords: mAP - Mean Average Precision, R-CNN – Region-based Convolutional Neural Network, ResNet – Residual Neural Network, SSD - Single Shot Detector, YOLO - You Only Look Once.

I. INTRODUCTION

Object detection is the computer vision and image processing technology that detects and defines the objects. Object detections combine image classification and object localization tasks for detecting the objects from an image which can also detect multiple objects from an image. Image classification is used for predicting the class of the object in an image. Object localization is to locate one or more objects present in the image and to locate them using a bounding box. This process can be used for detection objects for autonomous cars, which contain classes like cars, trucks, pedestrians, etc. By using this in autonomous cars it helps out with minimizing the specialized sensors in the car. Pedestrian detection is not only used for autonomous cars but also it is an important object in artificial intelligence because it is the key source for the machine to interact with. Pedestrian detection is used in many real-time applications for interaction, security purpose and much more. Some of the pedestrian detection applications are autonomous driving, traffic analysis, intelligent surveillance, industrial automation, etc.

Manuscript received on April 02, 2020.
Revised Manuscript received on April 20, 2020.
Manuscript published on May 30, 2020.
* Correspondence Author
S. Geethapriya*, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India. E-mail: geethapriya.s.2018.mecse@rajalakshmi.edu.in
Dr. P. Kumar, Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India. E-mail: kumar@rajalakshmi.edu.in

The evolution of object detection is so far improved from the traditional methods, which minimizes the number of evaluation steps in the current methodologies and also more efficient. In the traditional method, a fixed sliding window is used to slide from left to right and top to bottom in the image to localize the object in the image at different locations. After this process, an image pyramid is used to detect objects at varying scales. After the completion of these two processes, the Region of Interest (ROI) is extracted and fed into the convolutional neural network. For each evaluation of these steps, if the classification probability is higher than the threshold value then a bounding box is used to locate and label the object in the image. Finally, a non-max suppression is applied to the bounding boxes to finalize the final predicted or detected object in the image. The second method for object detection is by using a pre-trained network. The pre-trained network is a network that is already trained using a set of data sets for some other problem statement. By using this pre-trained network can make use of it for training the model, by using the same dataset or by using own dataset. This method is so effective because it is not necessary to start from scratch for building a model. The only thing have to do is to customize the model according to the problem statement. Also, the classification part in the network can be changed by removing the final fully connected layer and change it to the classification algorithms that needed. Some of the pre-trained networks are ResNet, VGG16, MobileNet and BaseNet. The classification part can be altered to R-CNN, YOLO, etc. One of the advantages of using deep learning algorithm is neural networks are capable of extracting features itself without any additional process for it. In this proposed system, the pre-trained ResNet model is used as a feature extractor and the model is trained using the INRIA dataset. The classification part in the network is removed and replaced with the YOLO v2 network. The reason for YOLO v2 is it is the only network in the YOLO family to detect even the smaller objects from any corner of the image. So, by using these two networks together the accuracy of detection can be increased and the model can be used for real-time applications.

II. LITERATURE SURVEY

Different models in deep learning where proposed and developed for object detection in recent years. Existing models have various algorithms for both feature extraction and classification. Some model does both the process using single algorithm which is the advantage of using Deep Learning. The existing algorithms for object detection are discussed below.

The various methods that are used for extracting features are Haar-like [16], HOG [7], [10], [16], Gaussian models which are mat lab features for extraction of features. The CNN methods are LDCF [15], ZFnet [12]. And after the growth of deep learning Fast R-CNN [11] and YOLO [8], [11] were widely used. HOG is the mostly used method for feature extraction. HOG provides an accurate result for pedestrian detection. And along with HOG, CSS [10] and Haar-like [16] are used for extracting features. The other methods for extracting features are, Selective Search [13] which includes the feature of exhaustive search and segmentation. It groups similar regions using shape, color, and texture. Transfer learning [9] is used for the system for feature extraction and classification. In this, the model trained for another problem is used for solving this problem statement by using the knowledge from it. Some of the pre-trained models for training the network are ImageNet, ResNet, etc. When building the system with deep learning the same methods are mostly used for both feature extraction and classification. Fast R-CNN [11] and YOLO [2], [11] are of those type, used for both feature extraction and classification. The methods for Classifications are SVM [7], [10], [16], AdaBoost [16], and K-means clustering. These are the most common methods used in a certain period. The idea of SVM classifier algorithm is simple; it creates a hyper plane which separated two classifiers. AdaBoost or Adaptive boosting uses the ensemble learning technique to classification. This combines multiple week classifiers to make it as a strong classifier.

III. PROPOSED METHOD

The model contains a neural network with Res Net as a feature extractor and YOLO v2 for classification. The choice of these two algorithms is to improve the efficiency of the object detection system by increasing the accuracy through detecting even smaller objects accurately. A neural network is created using the ResNet network which does the feature extraction process. ResNet is used for its better learning and accuracy with deeper networks. The degradation problem occurs in deeper networks which saturates the accuracy of the model by repeating the same process again and again at higher levels. To overcome this problem the ResNet model is used, which avoids a step which is processed more than twice, so that accuracy may not saturates even with deeper networks. And for classification, the fully connected network is altered into YOLO v2 network. YOLO v2 is a better network among neural networks to detect smaller objects accurately. It is better than YOLO v1 in accuracy and better than YOLOv3 in speed. The model is trained using the INRIA dataset with both positive and negative images. The image given into the model will be divided into N x N grid cells and will be assigned for extracting features in it. Then the extracted feature will be used in the classification process for detecting the object in the given image. Once the object is identified it will be marked using a bounding box. The accuracy of detection of an object is found using the difference between the actual bounding boxes versus predicted bounding box which is known as Intersection over Union. Mean Average Precision or Average Precision is a metric to calculate the accuracy of the object detectors. Precision measures how accurate the prediction was. The model was trained and tested using the INRIA dataset with both positive and negative images. The prediction with IoU greater than or equal to 0.75 is considered as the best prediction and it is detected using the bounding boxes. The further explanation of the network has been described in section IV.

A. SYSTEM DESIGN

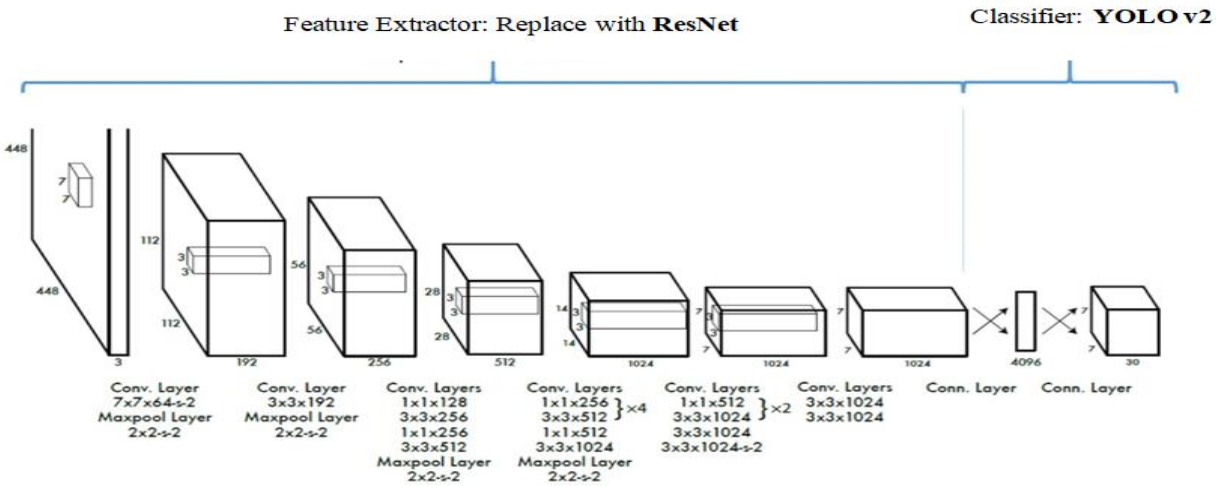

Figure 3.1: CNN architecture of YOLO

The overall architecture of the YOLO network is shown in figure 3.1. The network until fully connected layer is altered using the ResNet network, so that feature extraction is done using ResNet and YOLO v2 performs the classification process. First, a convolutional neural network with ResNet as a base is created. The CNN contains multiple convolutional plus ReLU layer and pooling layers alternatively. These layers are the building blocks of the convolutional neural network. The convolutional layer applies a filter to the input images and results in a map of activation. After every result of the convolutional layer, the map of activation or the feature map indicates the strength and locations of the detected feature in the input image and predicts the class to which it belongs.

ReLU in the convolutional layer is used to convert the non-linear properties to linear. Most of the real-world data is non-linear, so to deal with real-world data ReLU layer is used to convert it to linear for further process.

Another building block of CNN is the pooling layer, which progressively reduces the spatial size of the input to reduce the computation and parameter of the network, maintains the most important information. The last layer in the convolutional neural network is fully connected layers, this layer is sub-divided into three layers namely fully connected input layer, first fully connected layer and fully connected output layer. In a fully connected input layer, the output of the previous layer is given as input, which flattens and converts them into a single vector for making them as an input to the next layer. This single vector values are given as input to the first fully connected layer, which takes inputs and applies weights for the feature analysis to predict the labels. Finally, the fully connected output layer gives the final probability for each label.

B. MODULE DESCRIPTION:

The whole procedure for developing the pedestrian detection system is divided into four modules. The modules are:

- Creating a dataset
- Building a neural network
- ROI and Feature Extraction
- Classification and Detection

Though the pre-trained ResNet model is being used for developing the neural network, the dataset for training the model has been taken from INRIA pedestrian dataset and trained the pre-trained model for customizing it to the proposed system. The INRIA pedestrian dataset contains 15560 positive images and 6744 negative images for training and testing the model. All the images are pre-processed and set for training the network. The result of the system completely resides on how the network has been trained and with which kind of inputs. So the dataset should be good enough to train the network and should train the network with all possible ways for facing a new image for detection so that problems like overfitting will not occur while testing the model. The next module in developing the system is developing the neural network. For building a neural network the weights and parameters for the network have been set. In this, the weights and parameters of the ResNet model have been used for building the model. The pre-processed datasets are given has inputs to train the model. Then the sigmoid is the activation function used here for building the network. After the completion of single forward propagation the network starts learning and in the end it results with a loss function. To minimize the loss function, the back propagation method can be used to go back to the previous layers and can change the weights accordingly so that the loss function can be minimized. This can be repeated until the network is trained up to the expected level. The next step is extracting the features from the images. One of the advantages of using deep learning techniques is, the base network or neural network itself does the feature extraction process and doesn't need any additional process for it. The process that occurs inside the network for extracting features is the region of interest method, which is done before extracting features. By using this method the region where the object resides in the image can be taken separately for the extraction process so that the space needed to be processed can be minimized as compared to processing the whole image. So that the time required for the extraction process will be minimized. The final module in developing the system is the classification and detection process. The extracted features of the images are assigned for the classification process. The classification process is carried over by YOLO v2. The confidence score for the images with the extracted feature is calculated using the following eight values, which is further explained in section 4.

IV. EXPERIMENT AND RESULT

The experiment has been done using the INRIA person dataset as the training samples. 80 % of the dataset is used for training the model and rest 20 % is assigned for testing the model. For the proposed model, a pre-trained ResNet model is taken and trained the model using the INRIA dataset. The model has been modified using the YOLO v2 network. The structure of the model is shown in table 1.

Table 1: Structure of YOLO v2 network

LAYERS	FILTERS	SIZE/STRD(DIL)	INPUT	OUPUT
0 CONV	32	3 x 3/ 1	416 x 416 x 3	416 x 416 x 32 0.299 BF
1 MAX			416 x 416 x 32	208 x 208 x 32 0.006 BF
2 CONV	64	3 x 3/ 1	208 x 208 x 32	208 x 208 x 64 1.595 BF
3 MAX			208 x 208 x 64	104 x 104 x 64 0.003 BF
4 CONV	128	3 x 3/ 1	104 x 104 x 64	104 x 104 x 128 1.595 BF
5 CONV	64		104 x 104 x 128	104 x 104 x 64 0.177 BF
6 CONV	128	3 x 3/ 1	104 x 104 x 64	104 x 104 x 128 1.595 BF
7 MAX			104 x 104 x 128	52 x 52 x 128 0.001 BF
8 CONV	256	3 x 3/ 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
9 CONV	128		52 x 52 x 256	52 x 52 x 128 0.177 BF
10 CONV	256	3 x 3/ 1	52 x 52 x 128	52 x 52 x 256 1.595 BF
11 MAX			52 x 52 x 256	26 x 26 x 256 0.001 BF
12 CONV	512	3 x 3/ 1	26 x 26 x 256	26 x 26 x 512 1.595 BF

13 CONV	256		26 x 26 x 512	26 x 26 x 256 0.177 BF
14 CONV	512	3 x 3/ 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
15 CONV	256		26 x 26 x 512	26 x 26 x 256 0.177 BF
16 CONV	512	3 x 3/ 1	26 x 26 x 256	26 x 26 x 512 1.595 BF
17 MAX			26 x 26 x 512	13 x 13 x 512 0.000 BF
18 CONV	1024	3 x 3/ 1	13 x 13 x 512	13 x 13 x1024 1.595 BF
19 CONV	512		13 x 13 x 1024	13 x 13 x 512 0.177 BF
20 CONV	1024	3 x 3/ 1	13 x 13 x 512	13 x 13 x1024 1.595 BF
21 CONV	512		13 x 13 x 1024	13 x 13 x 512 0.177 BF
22 CONV	1024	3 x 3/ 1	13 x 13 x 512	13 x 13 x1024 1.595 BF
23 CONV	1024	3 x 3/ 1	13 x 13 x 1024	13 x 13 x1024 3.190 BF
24 CONV	1024	3 x 3/ 1	13 x 13 x 1024	13 x 13 x1024 3.190 BF
25 ROUTE	16			
26 REORG_OLD		/2	26 x 26 x 512	13 x 13 x2048
27 ROUTE	26 24			
28 CONV	1024	3 x 3/ 1	13 x 13 x3072	13 x 13 x1024 9.569 BF
29 CONV	35	1 x 1/ 1	13 x 13 x1024	13 x 13 x 35 0.012 BF
30 DETECTION				

The above structure is the flow in which the images are to be processed in the network and finally it is going to be detected. The network is trained using a huge volume to pedestrian data so that the network can detect any pedestrian data in the future for application purposes. The image given into the model will be divided into N x N grid cells and will be assigned for extracting features in it. Then the extracted feature will be used in the classification process for detecting the object in the given image. The classification process is carried over by YOLO v2. The confidence score for the images with the extracted feature is calculated using the following eight values,

$$Y = pc, bx, by, bh, bw, c1, c2, c3.$$

pc – Represents whether an object is present in the frame or not. If present pc=1 else 0.
bx, by, bh, bw – are the bounding boxes of the objects (if present).
c1, c2, c3 – are the classes. If the object is a car then c1 and c3 will be 0 and c2 will be 1.

These values will be again repeated if there occur multiple objects in a single frame. The images with the presence of objects are taken and the objects are detected using the bounding boxes. IOU and non-max suppressions and used to detect the object accurately. That is, once the object is identified it will be marked using a bounding box. The accuracy of detection of an object is found using the difference between the actual bounding boxes versus predicted bounding box which is known as Intersection over Union.

$$\text{Intersection over Union} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Mean Average Precision or Average Precision is a metric to calculate the accuracy of the object detectors. Precision measures how accurate the prediction was.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}}$$

In this approach, the joint version of ResNet and YOLO v2 has been come out with an accuracy of 87.7% which is the highest accuracy rate as compared with some existing models. The model is compared against three different models for object detection. Some existing models like the Single Shot Detector, Faster R-CNN, and Mask R-CNN. Among these three models, Mask R-CNN has the highest accuracy rate and minimum loss. But compared with the proposed model, it has the highest accuracy rate and less loss function. The mAP and loss value of the models are given in table 2.

Table 2: Model Comparison

Models	mAP	Loss
Single Shot Detector	62.5%	1.92
Faster R-CNN	78.8%	0.159
Mask RCNN	84.9%	0.099
Proposed Approach	87.7%	0.043

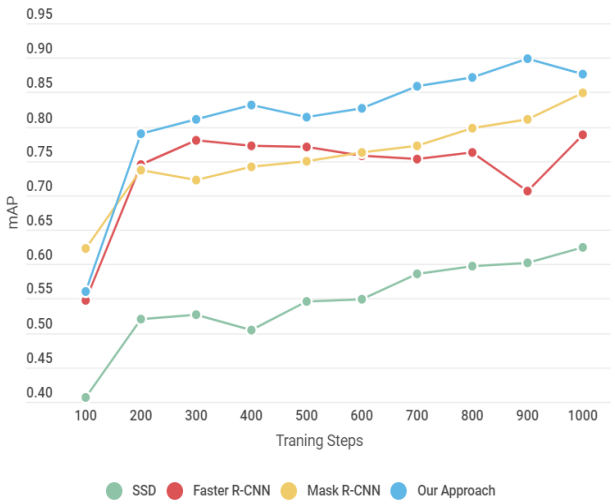

Figure 4.1: mAP Value Comparison

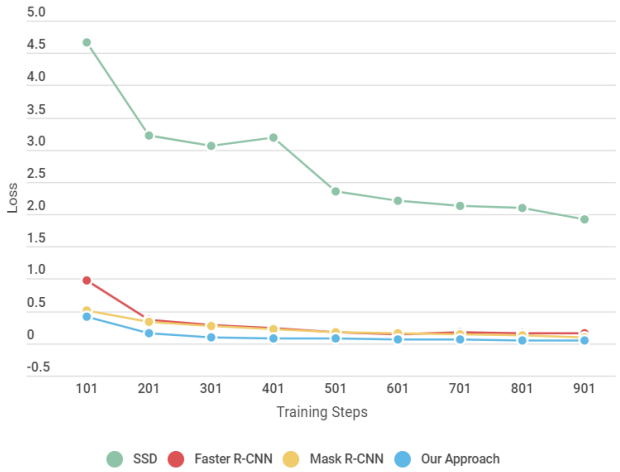

Figure 4.2: Loss Value Comparison

The comparisons of these models are plotted on a chart for both mAP and Loss values, shown in Figures 4.1 and 4.2 respectively, which shows the mAP and loss value of the networks from the 1st to the 1000th training steps. Difference can be seen between those existing models and the proposed approach, that this model has the highest accuracy rate with less loss function. Also, the output of the experiment is shown in figure 4.3.A and 4.3.B. The output image of the experiment shows the detection of pedestrians using the bounding boxes with the confidence score in it; from this the accuracy of the model can be measured. In figure 4.3.B, can see even a small part of the image is detected (marked with red color bounding box inside the car) and marked as pedestrian with the confidence score of 0.915, which is an accurate detection. From this it is shown that by using this algorithm we can detect smaller objects from any given frames. So this algorithm can be used in real-time applications.

Figure 4.3.A: Experiment Output 1

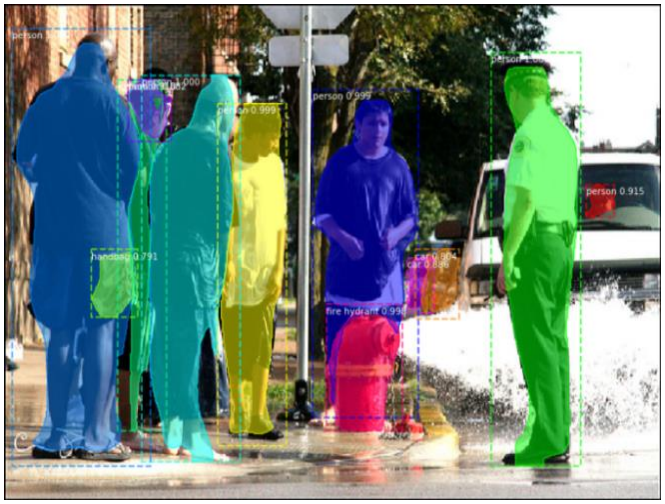

Figure 4.3.B: Experiment Output 2

V. CONCLUSION

Thus, the pedestrian detection algorithm is developed using the integration of Res Net and YOLO v2 networks. The choice of these two networks is to increase the accuracy rate and to minimize the loss function of the system to implement it in real-time applications. By experimenting with these two integrated networks the algorithm has come out with an accuracy of 87.7 percent with a loss of 0.043. The model has been compared with existing models like SSD, Faster RCNN and Mask RCNN which shows that this algorithm has the highest accuracy rate with less loss function. This algorithm can also be improvised by integrating with any other efficient algorithms. Also, it can be integrated with hardware like CCTV cameras for intelligent video surveillances in real-time applications.

REFERENCES

1. Ahmed, Z., Iniyavan, R., & P, M. M. (2019), “Enhanced Vulnerable Pedestrian Detection using Deep Learning”; International Conference on Communication and Signal Processing (ICCSP), pp: 0971-0974.
2. Ash, R., Ofri, D., Brokman, J., Friedman, I., & Moshe, Y. (2018), “Real-time Pedestrian Traffic Light Detection”; IEEE International Conference on the Science of Electrical Engineering in Israel.
3. Brunetti, A., Buongiorno, D., Trotta, G. F., & Bevilacqua, V. (2018), “Computer vision and deep learning techniques for pedestrian detection and tracking: A survey”; Neurocomputing, 300, pp: 17–33.
4. Chen, E., Tang, X., & Fu, B. (2018), “A Modified Pedestrian Retrieval Method Based on Faster R-CNN with Integration of Pedestrian Detection and Re-Identification”; International Conference on Audio, Language and Image Processing, pp: 63-66.
5. Guanqing Li, Zhiyong Song , Qiang Fu (2018), “A New Method of Image Detection for Small Datasets under the Framework of YOLO Network”; IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, pp: 1031-1035.
6. Lan, W., Dang, J., Wang, Y., & Wang, S. (2018), “Pedestrian Detection Based on YOLO Network Model”; IEEE International Conference on Mechatronics and Automation (ICMA), pp: 1547-1551.
7. Rahul Pathak,P. Sivraj, (2018), “Selection of Algorithms for Pedestrian Detection During Day and Night”; Computational Vision and Bio Inspired Computing, pp 120-133.
8. Zaatouri, K., & Ezzedine, T. (2018), “A Self-Adaptive Traffic Light Control System Based on YOLO”; International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), pp: 16-19.

9. Ghosh, S., Amon, P., Hutter, A., & Kaup, A. (2017), "Reliable pedestrian detection using a deep neural network trained on pedestrian counts"; IEEE International Conference on Image Processing.
10. Hongmeng Song, Wenmin Wang (2017), "Collaborative Deep Networks for Pedestrian Detection"; IEEE Third International Conference on Multimedia Big Data.
11. Naghavi, S. H., Avaznia, C., & Talebi, H. (2017), "Integrated real-time object detection for self-driving vehicles"; 10th Iranian Conference on Machine Vision and Image Processing.
12. Zhang, H., Du, Y., Ning, S., Zhang, Y., Yang, S., & Du, C. (2017), "Pedestrian Detection Method Based on Faster R-CNN. 2017 13th International Conference on Computational Intelligence and Security.
13. Hailong Li, Zhendong Wu, & Jianwu Zhang. (2016), "Pedestrian detection based on deep learning model"; 9th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics.
14. Peng, Q., Luo, W., Hong, G., Feng, M., Xia, Y., Yu, Li, M. (2016), "Pedestrian Detection for Transformer Substation Based on Gaussian Mixture Model and YOLO"; 8th International Conference on Intelligent Human-Machine Systems and Cybernetics.
15. Tomè, D., Monti, F., Baroffio, L., Bondi, L., Tagliasacchi, M., & Tubaro, S. (2016), "Deep Convolutional Neural Networks for pedestrian detection, Signal Processing: Image Communication, pp: 482–489.
16. Wei, Y., Tian, Q., & Guo, T. (2013), "An Improved Pedestrian Detection Algorithm Integrating Haar-Like Features and HOG Descriptors"; Advances in Mechanical Engineering.
17. A. Krizhevsky, I. Sutskever, G. E. Hinton (2012), "Imagenet classification with deep convolutional neural networks"; Advances in neural information processing systems.
18. David Gero´ nimo, Antonio M. Lo´ pez, Angel D. Sappa (2010), "Survey of Pedestrian Detection for Advanced Driver Assistance Systems", IEEE Transaction on pattern analysis and machine intelligence, VOL. 32, NO. 7, pp: 1239-1258.
19. Hong, Z., Zhang, L., & Wang, P. (2018). "Pedestrian Detection Based on YOLO-D Network"; IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp: 802-806.
20. Xie, C., Li, P., & Sun, Y. (2019)," Pedestrian Detection and Location Algorithm Based on Deep Learning*". International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), pp: 582-585.
21. Nguyen, K., Fookes, C., & Sridharan, S. (2015), "Improving Deep Convolutional neural networks with unsupervised feature learning", IEEE International Conference on Image Processing (ICIP), pp: 2270-2274.
22. Chen, X., Guo, R., Luo, W., & Fu, C. (2018). Visual Crowd Counting with Improved Inception-ResNet-A Module. 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp: 112-119.

AUTHORS PROFILE

S. Geethapriya, currently pursuing her final year M.E Computer Science and Engineering at Rajalakshmi Engineering College, Chennai. She has published few papers in the reputed journals. She is currently working in the research area of Deep Learning. **Mail id: geethapriya.s.2018.mecse@rajalakshmi.edu.in**

Dr. P. Kumar, Professor, Computer Science and Engineering has been with Rajalakshmi Engineering College, Chennai since June 1998. He has 23 years of teaching and published more than 30 papers in the referred National and International Journals. He has guided many UG and PG projects and one of his projects have won Best Project in IBM TGMC. He also published a book on Computer Programming. **Mail id: kumar@rajalakshmi.edu.in**