Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020)
IEEE Conference Record # 48766; IEEE Xplore ISBN: 978-1-7281-5371-1

Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning

Ruben J Franklin, Mohana
Department of Electronics & Telecommunication Engineering,
RV College of Engineering® Bengaluru-560059 Karnataka, India

***Abstract*—The number of new vehicles on the road is increasing rapidly, which in turn causes highly congested roads and serving as a reason to break traffic rules by violating them. This leads to a high number of road accidents. Traffic violation detection systems using computer vision are a very efficient tool to reduce traffic violations by tracking and Penalizing. The proposed system was implemented using YOLOV3 object detection for traffic violation detections such as signal jump, vehicle speed, and the number of vehicles. Further, the system is optimized in terms of accuracy. Using the Region of interest and location of the vehicle in the duration of frames, determining signal jump. This implementation obtained an accuracy of 97.67% for vehicle count detection and an accuracy of 89.24% for speed violation detection.**

Keywords – Convolutional neural network, speed violation, signal jump, traffic violation, YOLOV3.

I. INTRODUCTION

Traffic violation detection systems are being developed as the number of vehicles on the road is increasing at a large rate. This increase in number makes it difficult for trigger base traffic violation detection system to keep up with the high volume of traffic and it is not designed to detect multiple traffic violation at the same time for a given vehicle as a result for an increase in traffic rule violation due to the lack of advance detection. The traffic rule violation leads to various road accidents, traffic administration poses numerous basic difficulties in most present-day urban areas. Manual checking of vehicles is troublesome, and mistake-inclined due to feeble and problematic human memory. Consequently, a need arises for a traffic violation detection system to deal with this errand, which can identify criminal traffic offenses [5], for example, signal jump, over speeding and vehicle count. The first traffic monitoring began with human traffic cops assigned at every junction to monitor traffic, which required human resources and presence at all times. This consequently became difficult with the increase in vehicles. This gave way to trigger based traffic detection systems, which were specialized systems meant for detecting one type of violation – speed. These were costly, could only be placed at one point on the road, and were easy to avoid. A new system was required that could be operational 24x7, with least or no human resource requirement and which could identify multiple violations with high accuracy. This is how traffic violation detection using computer vision came into being. traffic violation detection using computer vision is mainly based on computer vision technologies, which are related to image processing, artificial intelligence, and deep learning. This detects objects with instances of semantic objects and class in digital images and videos [6]. The video captured from the camera is reused for various other domains and hence the operation cost is significantly reduced. Monitoring infrastructure can be scaled outside the city to rural & highways without additional cost. Since video captured from moving vehicles, the whole city is covered instead of a few signals and junctions. This also enables traffic police to capture traffic violations happening in small lanes to highways. This does not have the one-time infrastructure cost involved in other monitoring systems like Cameras. It saves time and relieves traffic police personnel to focus on other productive work. Computer vision-based detection is based on the principle of vehicle classification, environment awareness, and traffic violation detection.

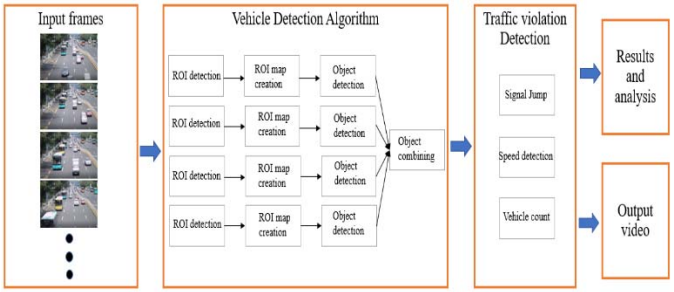

Fig.1. Block Diagram of the Traffic Violation Detection system.

Figure 1 shows the basic block diagram of the traffic violation detection system, the proposed architecture of surveillance system with intelligent detection and tracking of multiple vehicles from the surveillance input video using YOLOv3 as an object detection algorithm[12] [13]. This is done through a neural network and an object detection model which are used in the classification of the moving objects into different respective classes, thus achieving vehicle classification. Next, from the same given video footage, traffic lights, zebra crossing, different lanes, and traffic signs are classified this comes under environment awareness. Combining these two, now violations are detected based on violations are then detected these can occur on the road which are signal jump, speed detection, and vehicle count. The main objective is to detect multiple vehicle violation detections and it gives a more detailed picture of concepts and technology involved in creating a traffic violation detection system using computer vision. It also aims to throw light on some of the applications and the latest developments being made in the said field [18] [19].

II. FUNDAMENTALS OF TRAFFIC VIOLATION DETECTION

This section describes the fundamental concepts used for implementation.

A. Image processing

Image processing refers to the analysis and manipulation of a digital image to enhance its quality and interpret information present in it [20].

B. YOLOv3

YOLOv3 algorithm is mainly constructed of a particular connected Convolutional Neural Network. Layers of CNN is a class of deep neural networks (DNN) to analyze digital images or videos. It is a class of multi-layer perceptron, refers to a fully connected network (FCN) in which all neurons of one layer are connected to other neurons in the next layer [1] [2]. CNN is a specialized category of neural networks used effectively to process data in a grid-like manner. Parameter sharing is a distinguishing feature of CNNs, which is of great significance in object detection. CNN's are different from fully connected forward neural networks, each of its trained weight is used, such that in overall CNN architecture every component makes use of the kernel in the input position, meaning that in each location and position, a category of all the parameters is learned instead of a different category of parameters. This property is very valuable in encapsulating the whole entire environment [3] [14] [15].

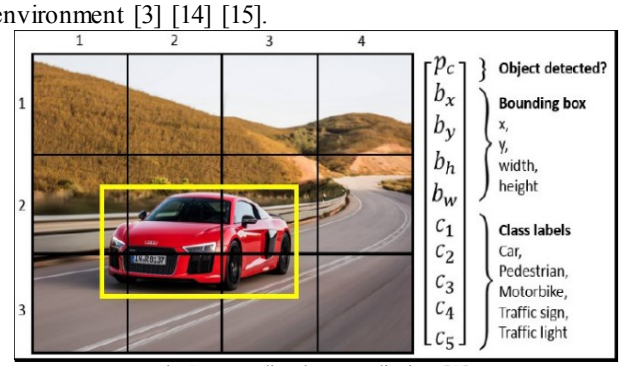

Fig 2. Bounding box prediction [9].

Figure 2 shows the bounding box prediction. CNN takes input as a digital image and output is

- Objects with boundary boxes, Coordinates of this boxes and positions concerning the frame,
- The probability score of all the object in a digital image in the respective bounding box's,
- Probabilities on every object present inside the frame in its bounding box.

The detection of the object is found in three different layers, where input dimensions of the given digital image are in height and width, those are 13x13, 26x26, and 52x52. detection takes place in three various scales [9]. The output of the digital image from the three deleted output layers have the same height and width concerning the inputs digital image, but depth will always depend on the definition [9]:

$$depth = (4 + 1 + class\ probabilities)\ x\ 3,$$

where, four denotes properties of bounding boxes such as height (*bh*), width (*bw*), x and y denote the position of the given bounding boxes (*bx*, *by*) inside a given digital image, one is the probability of the detected object in the box (*pc*) and class probabilities are the probabilities for each class (*c1*, *c2*, …). That sum is multiplied by three, this done cause each of the cells predicts 3 bounding boxes in a grid. This results in the output of 10647 bounding box predictions in every digital image.

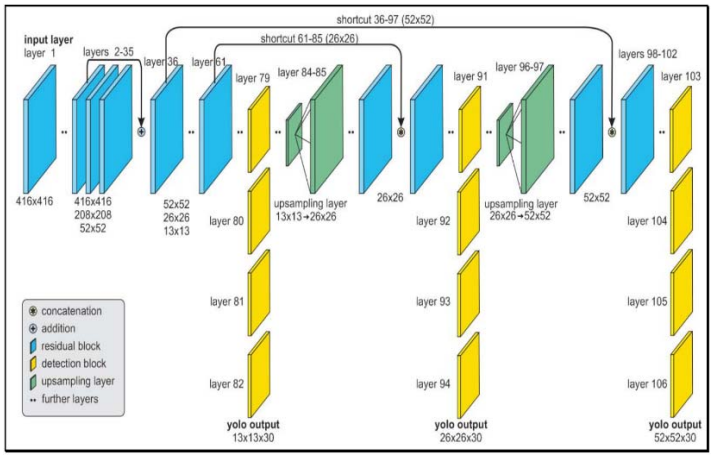

Fig 3. YOLO network [9]

The algorithm begins with a feature extraction of a single digital image in frame-by-frame format from the digital video, after which feature extracted digital image is rendered to different sizes of 416x416 image. As shown in figure 3[9], YOLO v3 CNN which has a total of a hundred and six layers. Leaving the convolutional neural network layer [8], the network can detect multiple objects simultaneously in a given single input digital image. The notion of breaking down given digital input images into n number of grids is only used in the YOLOv3 algorithm when relating to another algorithm. The given digital input frame is broken down to a *S x S* grid, where the grid gives an output prediction of three bounding boxes. Prediction scores less than 0.5 are discarded, by this method false predictions are filtered.

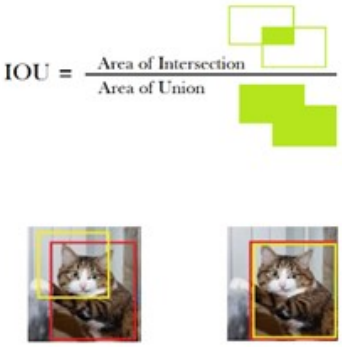

Fig 4. Filtering multiple detections

Figure 4 shows the perdition of multiple bounding boxes NOS (Non-Max Suppression). NOS excrete data by comparing NOS pc bounding box with each bounding boxes, intersect with one another [10].

III. DESIGN AND IMPLEMENTATION OF TRAFFIC VIOLATION DETECTION

In computer vision, the proposal of YOLOv3 satisfies the research and application requirements of many fields and provides novel ideas to the development of traffic violation detection system. Nowadays, the image and vision field receive the most attention from YOLOv3. YOLOv3 is

suitable for traffic violation detection due to its fast object detection and processing speed. After the initial development of the YOLOv3 theory, several research works were published to extend the idea as well as its applicability in various domains [4][16][17]. Design starts by analyzing the input key parameters, from the video sequence, which is read in the frame by frame separately. Each digital video is feed into as a frame by frame where the input can be divided into two different parts those are background part and foreground part. The background part, which contains the environment, is mostly stable and remains unchanged throughout the video and the foreground part, is where all the moving objects are detected.

The output of this is feature representation which is the most important information related to traffic violation detection. By using YOLOv3, three features are obtained and those are class of the object, bounded box, and semantic segmentation as shown in figure 5. Which is later used to determine the type of traffic violation [25].

Fig 5. Four blocks of YOLOv3 architecture [8]

A. Feature Extractor

The feature extractor YOLOV3 uses is called Darknet-53. It is familiar with the previous darknet versions from YOLO V1, where there are only 19 layers. But that was like a few years ago, and the image classification network has progressed a lot from merely deep stacks of layers. ResNet brought the idea of skip connections to help the activations to propagate through deeper layers without gradient diminishing. Darknet-53 borrows this idea and successfully extends the network from 19 to 53 layers.

B. Detector

Once the three features vectors are obtained, it canbe fed into the detector. Through this config file, multiple 1x1 and 3x3 Conv layers are used before a final 1x1 Conv layer to form the final output. For medium and small scale, it also concatenates features from the previous scale. By doing so, small scale detection can also benefit from the result of large-scale detection [7].

C. Bounding Box, Class Predictions

The YOLOv3 network predicts four co-ordinates for each bounding box. YOLOv3 gives an objectiveness score for each bounding box-using concept called logistic regression. In which one means a complete overlap of bounding box with truth object. Some of the assumptions and constraints made during implementation are:

- The observed object is a visible and direct line of sight.
- The video is captured from a single stationary source
- The captured video should contain RGB frame structure

D. Dataset Specification

Dataset used: MSCOCO

COCO is large-scale object detection, segmentation, and captioning dataset. COCO has several features: Object segmentation, Recognition in context, superpixel stuff segmentation, 330K images (>200K labeled), 1.5 million object instances, 80 object categories, 91 stuff categories, 5 captions per image, 250,000 people with key points. Figure 5 describes the specifications of dataset.


```
[ { "segmentation":
[[204.01,306.23,...206.53,307.95]],
"num_keypoints": 15,
"area": 5463.6864,
"iscrowd": 0,
"keypoints": [229,256,2,...,223,369,2],
"image_id": 289343,
"bbox": [204.01,235.08,60.84,177.36],
"category_id": 1, "id": 201376 } ]
```


Fig 6. Dataset Specifications

E. Implementation details

Fig.7. Input video

```
if (x["type"] == "convolutional"):
    #Get the info about the layer
    activation = x["activation"]
    try:
        batch_normalize = int(x["batch_normalize"])
        bias = False
    except:
        batch_normalize = 0
        bias = True

    filters= int(x["filters"])
    padding = int(x["pad"])
    kernel_size = int(x["size"])
    stride = int(x["stride"])

    if padding:
        pad = (kernel_size - 1) // 2
    else:
        pad = 0

    #Add the convolutional layer
    conv = nn.Conv2d(prev_filters, filters, kernel_size, stride, pad, bias = t
    module.add_module("conv_{0}".format(index), conv)

    #Add the Batch Norm Layer
    if batch_normalize:
        bn = nn.BatchNorm2d(filters)
        module.add_module("batch_norm_{0}".format(index), bn)

    #Check the activation.
    #It is either Linear or a Leaky ReLU for YOLO
    if activation == "leaky":
        activn = nn.LeakyReLU(0.1, inplace = True)
        module.add_module("leaky_{0}".format(index), activn)

#If it's an upsampling layer
#We use Bilinear2dUpsampling
elif (x["type"] == "upsample"):
    stride = int(x["stride"])
    upsample = nn.Upsample(scale_factor = 2, mode = "bilinear")
    module.add_module("upsample_{}".format(index), upsample)
```

Fig.8. Convolution layers

```
(9): Sequential(
  (conv_9): Conv2d (128, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (batch_norm_9): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (leaky_9): LeakyReLU(0.1, inplace)
)
(10): Sequential(
  (conv_10): Conv2d (64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bi
  (batch_norm_10): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True)
  (leaky_10): LeakyReLU(0.1, inplace)
)
(11): Sequential(
  (shortcut_11): EmptyLayer(
  )
)
```

Fig 9. Output Parameters

Figure 7 shows the frame of the input video. Figures 8 and 9 describe the details of convolution layers and output parameters.

IV. SIMULATION RESULTS AND ANALYSIS

The proposed algorithm, which was, trained on the MSCOCO dataset and the tested-on input video [11] of 1920×1080 px 30 FPS. The results were analyzed. The calculated Intersection over Union (IOU) for the detected bounding boxes between defining true positives and truth bounding boxes, is set as the value of IOU as 0.4.

A. Signal Jump and other Traffic violation detection

Fig.10. Detection of a traffic violation input frame

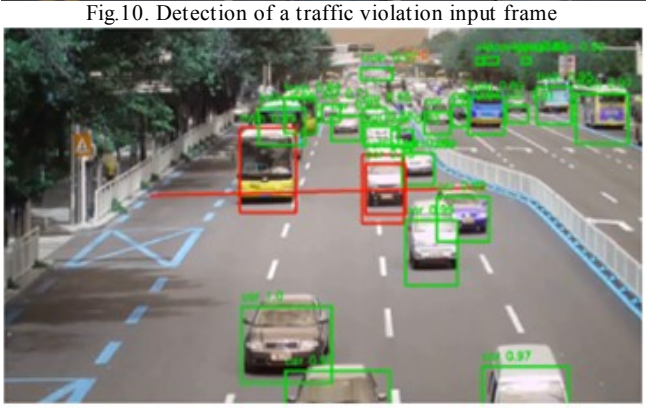

Fig 11. Detection of a traffic violation output frame: bounding box and violation detection (signal jump)

```
car: 85.616534948349%
line: (260, 350) (1021, 335)
Box: (724, 188) (789, 239)
```

Fig.12. Generated output data for one object (car).

[{"iscrowd": 0, "image_id": 1, "category_id": 1, "id": 1, "bbox": [95.5, -0.5, 201.0, 333.0], "area": 33317.25 }, {"iscrowd": 0, "image_id": 2, "category_id": 3, "id": 3, "bbox": [25.5, 42.5, 116.0, 261.0], "area": 22243.5 }, {"iscrowd": 0, "image_id": 2, "category_id": 2, "id": 4, "bbox": [100.5, 238.5, 324.0, 256.0], "area": 47241.0 }, {"iscrowd": 0, "image_id": 2, "category_id": 4, "id": 5, "bbox": [161.5, -0.5, 210.0, 264.0], "area": 14593.25 }]

Fig.13. Output Parameters of entering the frame

Figure 10 shows the input frame and Figure 11 describes the detection of a traffic violation in the image of a signal jump. The exact location of the same in the frame and boundary box with green color, upon detection of traffic violation the boundary box turns red. Detection of vehicle crossing during a red light in the image and locate the exact location. It generated the regions where there might be an object based on the input digital image. Second, it predicts the class of the object, the bounding box. The red line indicates the signal line where the vehicle must stop during the red light. Figure 12 shows the generated data concerning the region of interest and boundary boxes. Output parameters of the entire frame described in figure 13.

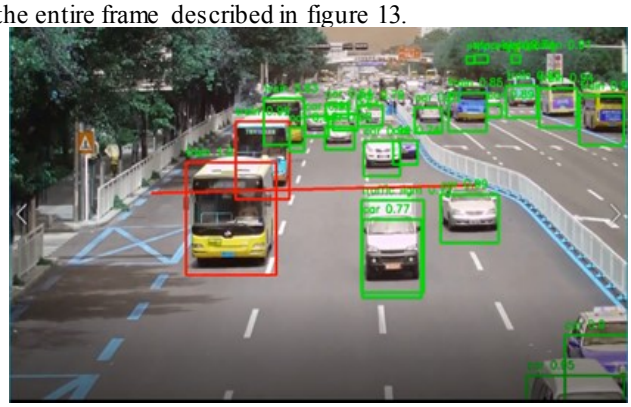

Fig.14. Failure to the detection of a traffic violation

Figure 14 shows that the car is not at the ROI, which is the traffic signal line, hence the car id not detected as violating the traffic rules. Since this model is capable of detecting vehicles in every frame it is having been enhanced to detect the speed of the vehicle in a video. The accuracy of detecting the speed of the vehicle was 5.4% times slower than the actual speed. The corresponding frame of the video is shown in Fig 9, the shown frame detects the vehicles and also the respective speed of the vehicle. This is done using the pixel calculation, area of the boundary box, and the anchor points.

B. Speed violation and lane change detection

Fig.15. Speed detection of an output frame

Since this model is capable of detecting vehicles in every frame it is having been enhanced to detect the speed of the vehicle in a video. The corresponding frame of the video is shown in figure 15. In the frame, it detects the vehicles and the respective speed of the vehicle. This is done using the pixel calculation, area of the boundary box, and the anchor points. The speed is calculated using all the three principles, the rate of change of the object is the video frame by frame gives us the speed but it is not the true speed. This because the object in the video doesn't move linearly hence the area of the boundary frame by frame gives us the actual distance traveled by the vehicle in the video. The anchor points add more perception to the parameters. With this parameter, the speed of the vehicle is determined.

C. Performance Analysis

TABLE I VEHICLE DETECTION ANALYSIS

Input Video	Frames per second	Actual Number Of Vehicles	Detected Vehicles	Accuracy %
Video 1	22	9	9	100
Video 2	23	43	41	95.34

TABLE II DETECTION TIME ANALYSIS

Input Video	Actual Number of Vehicles	Violation Detection time
Video 1	9	2.65 sec
Video 2	43	3.43 sec

Analysis of vehicle detection and its time tabulated as shown in table I and II.

V. APPLICATIONS AND RESEARCH CHALLENGES

A. Applications

- Signal jump violation detection
- Lane jump violation detection
- Speeding violation detection
- Parking violation detection
- Missing Car detection
- AI-based Traffic control
- Autonomous Traffic Management system[26]

B. Research Challenges

Lack of Data

A major requirement for the model to be trained or to be tested is the presence of data. Unavailability of labeled data in "class imbalance" increases the time required to implement a well- defined working anomaly detector. Increasing in WSN and IOT without doughty generates large amounts of data, which makes it even more difficult to label them, and differentiates between "normal" and "abnormal" data set. Due to all the facts stated above, it comes as a challenge to develop new anomaly detection algorithms.

Continuous learning and Training

As environments and rules may change over time, continuous learning and training are required. For example, change in traffic rules at midnight, at which point some rules do not apply.

Veracity

Computational power is required in both hardware and software, together with a large amount of computing time, to identify objects, labels, and their respective location in an image or video frame. Under partial concealment, resolution of the interested detection must improve localization accuracy on small objects. A centralized anomaly detection system is not realistic and feasible for real-time analysis of data.

VI. CONCLUSIONS AND FUTURE SCOPE

A. Conclusion

Detections of traffic violation in the video surveillance is challenging as the number of vehicles on the road and traffic rules are depended on the different area of the road and timings. This paper proposes that the YOLOv3 algorithm is suitable for traffic violation detection. Results show that the detection of multiple traffic violations from a single input source is archivable. The system has an accuracy of 97.67% for vehicle count detection and an accuracy of 89.24% to detect the vehicle speed. The detection time is lower for high dense traffic flow. Thus, the system operation speed is dependent on the density of traffic.

B. Future scope

With the increasing growth in traffic density all over the world, it possesses a great challenge to traffic management. Emphasis should be that large area is covered and the high volume of traffic monitoring and detection from a single input source using parallel computation. Further enhancements are required to reduce computational time at high traffic volume roads. Further, it can be implemented for larger datasets by training using GPUs and high-end FPGA kits [21] [22] [23] [24].

REFERENCES

[1] X. He et al., "A Driving Warning Method based on YOLOV3 and Neural Network", IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Zhengzhou, China, 2019.
[2] H. Qu et al., "A Pedestrian Detection Method Based on YOLOv3 Model and Image Enhanced by Retinex," 11th International Congress on Image and Signal Processing, Bio Medical Engineering and Informatics (CISP-BMEI), Beijing, China, 2018.
[3] J. Won et al., "An Improved YOLOv3-based Neural Network for De-identification Technology," 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Korea, 2019.
[4] P. Tumas et al., "Automated Image Annotation based on YOLOv3", IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, 2018.
[5] Krishna et al., "Automated traffic monitoring system using computer vision" 2016.

[6] P. Manocha et al., "Korean Traffic Sign Detection Using Deep Learning", International SoC Design Conference (ISOCC), Korea, 2018.
[7] Y. Lee et al., "Fast Detection of Objects Using a YOLOv3 Network for a Vending Machine," IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Taiwan, 2019.
[8] J. Wang et al., "Real-Time Non-Motor Vehicle Violation Detection in Traffic Scenes", IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan, 2019.
[9] A. Corovic et al., "The Real-Time Detection of Traffic Participants Using YOLO Algorithm," 26th Telecommunications Forum (TELFOR), Belgrade, 2018.
[10] N. Krittayanawach et al., "Robust Compression Technique for YOLOv3 on Real-Time Vehicle Detection", 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, 2019.
[11] github.com/anmspro/Traffic Signal Violation-Detection-System.
[12] Mohana et.al., "Simulation of Object Detection Algorithms for Video Survillance Applications", *2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2018.*
[13] Yojan Chitkara et. al.,"Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model" *International Conference on Computing Methodologies and Communication, 2019.*
[14] Mohana et.al., "Performance Evaluation of Background Modeling Methods for Object Detection and Tracking," *International Conference on Inventive Systems and Control, 2020.*
[15] N. Jain et.al., "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2019.
[16] Ruben J Franklin et.al., "Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks," *International Conference on Inventive Systems and Control, 2020.*
[17] Abhiraj Biswas et. al., "Classification of Objects in Video Records using Neural Network Framework," *International conference on Smart Systems and Inventive Technology, 2018.*
[18] Pallavi Raj et. al.,"Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications" *IEEE International Conference on Intelligent Sustainable Systems, 2019.*
[19] Harsh Jain et. al., "Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications" *International conference on Electronics and Sustainable Communication Systems, 2020.*
[20] R. K. Meghana et al., "Inspection, Identification and Repair Monitoring of Cracked Concrete structure –An application of Image processing", *International Conference on Communication and Electronics Systems (ICCES)*, 2018.
[21] 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 (RTEICT)*, 2016.
[22] 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 (RTEICT)*, 2017.
[23] 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 (ICIMIA), 2017.
[24] D. Akash et al.,"Interfacing of flash memory and DDR3 RAM memory with Kintex 7 FPGA board", *International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*,2017.
[25] Vijayakumar, T. (2019). COMPARATIVE STUDY OF CAPSULE NEURAL NETWORK IN VARIOUS APPLICATIONS. Journal of Artificial Intelligence, 1(01), 19-27.
[26] Koresh, M. H. J. D. (2019). COMPUTER VISION BASED TRAFFIC SIGN SENSING FOR SMART TRANSPORT. Journal of Innovative Image Processing (JIIP), 1(01), 11-19