Journal of Science, Computing and Engineering Research (JSCER)
Volume-6, Issue- 3, March 2023.
DOI: *https://doi.org/10.46379/jscer.2023.060322*

Vehicle Collision Detection and Alert System Using Yolov3 Algorithm

Gayathri G S, Vimala Devi A, Priya B, Kalaiarasi C

Department of CSE, PERI Institute of Technology, Chennai, India.

Article Information

Received : 09 Jan 2023
Revised : 22 Feb 2023
Accepted : 03 Mar 2023
Published : 26 Mar 2023

Corresponding Author:
Gayathri G S

Abstract— According to worldwide statistics, traffic accidents are the cause of a high percentage of violent deaths. Due to this and the wide use of video surveillance and intelligent traffic systems, an automated traffic accident detection approach becomes desirable for computer vision researchers. Over the past years, automatic traffic accident detection (ATAD) based on video has become one of the most promising applications in intelligent transportation and is playing a more and more important role in ensuring travel safety. Input is a video obtained via surveillance systems. Output results are acquired instantly in real-time and we would be notified if there's a chance of collision or not. Our system is based on YOLO, Neural networks, and Deep Learning of object detection along with computer vision technology and several methods and algorithms. The existing systems are simple and effective but are only able to analyze the rear portion of the vehicle. Hence, not very effective and have low accuracy. We propose an automated, real-time system for the beforehand detection of vehicle collisions during high traffic and intimate the concerned people using the application. Our approach will work on still images, recorded- videos, and real- time live videos and will detect, classify, track and compute moving object velocity and direction using a convolution neural network. Using YOLO, it will be able to detect the front as well as the rearview of the vehicle and alert us beforehand. The Advantages of the proposed system are Secured, Interpretability, High accuracy, Lightweight model & fast processing. Moreover, this system can be used in the cases of Self-driving cars.

Keywords: ***Traffic Systems, YOLO, neural networks***

Copyright © 2023 Gayathri G S. This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

Citation: Gayathri G S, Vimala Devi A, Priya B, Kalaiarasi C. "Vehicle Collision Detection and Alert System Using Yolov3 Algorithm", Journal of Science, Computing and Engineering Research, 6(3), 112-117, 2023.

I. INTRODUCTION

Vehicles are an important way of transportation all over the world. There are many cases of road accidents every day in the world. A traffic collision, also called a motor vehicle collision, car accident, or car crash occurs when a vehicle collides with another vehicle, pedestrian, animal, road debris, or other stationary obstruction, such as a tree, pole, or building. Traffic collisions often result in injury, disability, death, and property damage as well as financial costs to both society and the individuals involved.

Road transport is the most dangerous situation people deal with on a daily basis, but casualty figures from such incidents attract less media attention than other, less frequent types of tragedy. A number of factors contribute to the risk of collisions, including vehicle design, speed of operation, road design, weather, road environment, driving skills, impairment due to alcohol or drugs, and behavior, notably aggressive driving, distracted driving, speeding, and street racing.

In 2013, 54 million people worldwide sustained injuries from traffic collisions.[2] This resulted in 1.4 million deaths in 2013, up from 1.1 million deaths in 1990.[3] About 68,000 of these occurred in children less than five years old.[3] Almost all high-income countries have decreasing death rates, while the majority of low- income countries have increasing death rates due to traffic collisions. Middle-income countries have the highest rate with 20 deaths per 100,000 inhabitants, accounting for 80% of all road fatalities with 52% of all vehicles. While the death rate in Africa is the highest (24.1 per 100,000 inhabitants), the lowest rate is to be found in Europe (10.3 per 100,000 inhabitants).

Road incidents are caused by a large number of human factors such as failing to act according to weather conditions, road design, signage, speed limits, lighting conditions, pavement markings, and roadway obstacles. A 1985 study by K. Rumar, using British and American crash reports as data, suggested 57% of crashes were due solely to driver factors, 27% to combined roadway and driver factors, 6% to combined vehicle and driver factors, 3% solely to roadway factors, 3% to combined roadway, driver, and vehicle factors, 2% solely to vehicle factors, and 1% to combined roadway and vehicle factors.

Reducing the severity of injury in crashes is more important than reducing incidence and ranking incidence by broad categories of causes is misleading regarding severe injury reduction. Vehicle and road modifications are generally more effective than behavioral change efforts with the exception of certain laws such as required use of seat belts, motorcycle helmets, and graduated licensing of teenagers.

Human factors in vehicle collisions include anything related to drivers and other road users that may contribute to a collision. Examples include driver behavior, visual and auditory acuity, decision-making ability, and reaction speed. A 1985 report based on British and American crash data found driver error, intoxication, and other human factors contribute wholly or partly to about 93% of crashes. A 2019 report from the U.S. The National Highway Traffic Safety Administration found that leading contributing factors for fatal crashes included driving too fast for conditions or in excess of the speed limit, operating under the influence, failure to yield right of way, failure to keep within the proper lane, operating a vehicle in a careless manner, and distracted driving.

Drivers distracted by mobile devices had nearly four times greater risk of crashing their cars than those who were not. Research from the Virginia Tech Transportation Institute has found that drivers who are texting while driving are 23 times more likely to be involved in a crash as non-texting drivers. Dialing a phone is the most dangerous distraction, increasing a drivers' chance of crashing by 12 times, followed by reading or writing, which increases the risk by ten times. To avoid such instances, our suggested or proposed system uses OpenCV and Python to detect vehicle collisions in real-time to avoid accidents and life-threatening situations.

Traditional vehicle detection algorithms were mainly based on artificial feature extraction. Artificial feature extraction algorithm suffers either from high computational costs, low accuracy, or insufficient robustness. The current methods are complex and the robustness is insufficient as some morphological operations, image processing, and other pre-processing must be performed for vehicle images. Although prediction results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact such as the high speed of the vehicle.

The existing systems are also not robust in predicting vehicle collision, many of the current techniques are supervised algorithms, while automobile technology is slowly moving to unsupervised and computer vision learning to manage real-world data without manual human labels. The discussed limitations have been overcome to enhance the performance of prediction of vehicle collision successfully in the presented application. The existing models cause unsatisfactory results & inaccurate data which may lead to a higher chance of road accidents and damage to human life and other assets.

We propose an automated, real-time system for the beforehand detection of vehicle collisions during high traffic and intimate the concerned people using the application. Our approach will work on still images, recorded- videos, and real-time live videos and will detect, classify, track and compute moving object velocity and direction using a convolution neural network. Using YOLO, it will be able to detect the front as well as the rearview of the vehicle and alert us beforehand. The working of yolo is quite simple as yolo is based on regression.

Unlike CNN which selects interesting parts in an image, yolo on the other hand predicts the class and bounding boxes for the whole image in one run of the algorithm. The overall accuracy of the proposed scheme has been evaluated with the traditional state-of-the-art models and the results from our proposed application show a higher accuracy rate.

When performing classification in the trained model by applying sample test data of vehicle collision detection, the proposed application gives accurate results. The proposed model works well against train data and test data further this model will provide better results for real-time data. The application is beneficial for conditions where data has to be processed in a short time and results are required instantly. It is a fast and quick method. This model is highly Cost Efficient. Thus, this model helps to build trust and develops a sense of security among people. Anaconda is an open-source package manager for Python and R.

It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda.

There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open- source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

II. IMPLEMENTATION

This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.

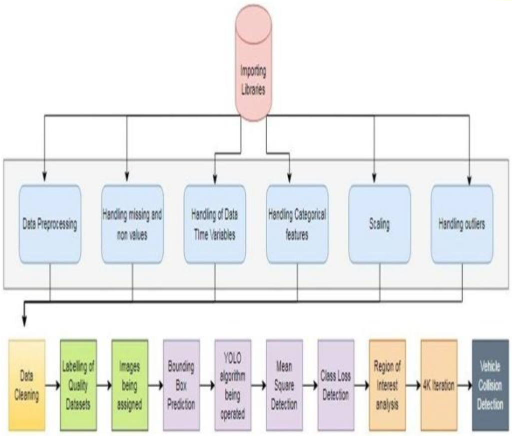
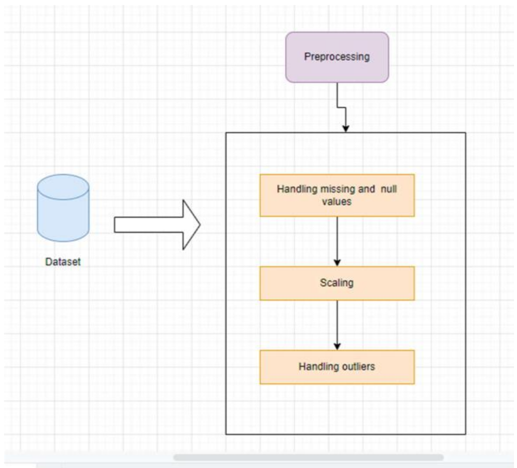


Fig 4.1 – Architecture Diagram

The process begins by setting up the environment and compiling the data. Data collection is done via datasets. The



data is collected by importing the necessary libraries from source websites. After the data is collected, the quality datasets are labeled accordingly. Preprocessing of data is performed where missing and null values are dropped in this process. It followed by another process where images are being assigned.

Yolo algorithm achieves its result by applying a neural network on an image. The image is divided in an SxS grid and comes up with a bounding box. This algorithm has 24 convolutional layers which in turn has two fully connected layers. The reduction in feature space is done by Alternating 1x1 convolutional layers from preceding layers. The object identification problem is considered to be a regression problem with the objective of spatially bounding box separation along with the probability of associated classes in the bounding boxes. A single neural network can predict the bounding boxes and class probabilities directly from the input images in just one evaluation which can be optimized end-to-end. A multi-task algorithm is proposed to extract from each frame all needed information while maintaining the principles of real-time, YOLO is object detection tool that divides the input image into an (S×S) grid.

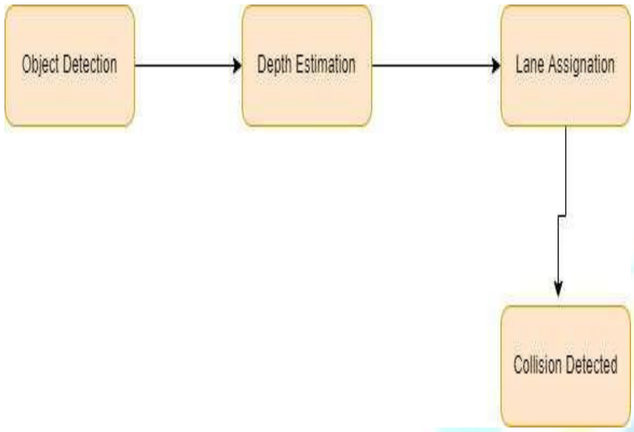
Each grid cell predicts only one object. For example, the yellow grid cell below tries to predict the "person" object whose center (the blue dot) falls inside the grid cell. Each grid cell predicts a fixed number of boundary boxes. In this example, the yellow grid cell makes two boundary box predictions (blue boxes) to locate where the object is.

There are several versions of YOLO, but version 3 was chosen according to its speed and accuracy Also, YOLO is combined with several algorithms, such as depth estimation and computer vision, in a single algorithm to achieve the desired real-time collision warning system. This algorithm uses in autonomous cars; that is why the latency is a very critical parameter that should keep minimum as possible. So, the YOLO is used, since the lower the number of classes are detected by YOLO, the less the processing will be, the higher the performance is.

First, depth estimation algorithm uses the boundary boxes (width and height) from YOLO and get the distance using multiple regression techniques the depth estimation equation is modified using trial and error algorithm that tries different values for one parameter (t (from 0 to 1 in the equation (this parameter comes out from regression algorithm and does not have any physical meaning) then the algorithm compares the output estimated distance at each value of (t (with the real distance from real dataset.

After applying depth estimation, two algorithms developed to complete the task; lane assignation and tracking algorithm. For the lane assignation algorithm, geometric principles were used to assign cars into lanes depending on each vehicle's bounding box002E.

The horizon line is dependent on the camera position in the presented work; the horizon line represented 55% of the height. As shown in the above figure the image is divided virtually with two significant lines into four quarters; the upper half is unused as it represents the sky or the far vehicles, while the down half represents the road (in the presented work), which is divided into three regions. Lane assignation is a very critical task, as it will determine which vehicle to give interest and which not, So, the performance of this task must be optimal. The problem was which point of the bounding box will be used to represent the entire box (the car)? Lane assigning algorithm is divided into two layers/phases; layer 1, is to determine at what quarter the vehicle is moving. In this layer, the center of the bounding box uses as a reference point.

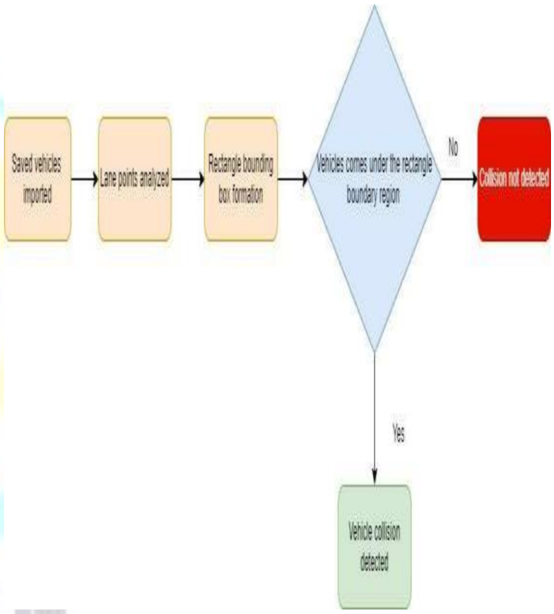


The primary role of this layer is to determine if the vehicle is above the horizon (it does not represent any interest right now), regardless of the interest or if the vehicle is below the horizon so it is in the exciting area and initially determined if it is in the right side or the left side For layer 2, two virtual lanes are created, which represent the width of the vehicle; this layer is more critical than the first layer. Thus, the results are expected to be precisely accurate, especially in the busy roads. In this layer, two different reference points are used depending on the output of the first layer, If the vehicle was detected at the right half, the bottom-left point of the bounding box is used as the reference point to the vehicle. This reference point substituted in the right virtual lane equation to determine whether the vehicle is in the emergency lane or not; if not, automatically, the car will be assigned to the right lane. While if the car was detected at the left part, the bottom-right point of the bounding box is the reference point to the car.

This reference point is used to determine whether the car is in the emergency lane or not; if not, automatically, the car will be assigned to the left lane. The above figure shows how the input image converted after object detection into a geometric problem to assign vehicles to their actual lanes The assigning technique illustration Finally, the tracking algorithm is responsible for tracking the same vehicles through all the frames, so the change in its distance could be evaluated then calculating its relative speed. The tracking algorithm is a complex problem, especially if the road is crowded with vehicles since it is a must to compare all vehicles in the current frame with all vehicles in the previous frame, then a suitable threshold was estimated.

If the new position differs by a maximum of 5% from the last position, so it is the same vehicle. Hence tracking the same vehicle through the frames is achieved but it’s heavy work and increases the delay of the system. So, here is the second importance of the lane assigning algorithm, hence after the vehicles are divided into three categories, the tracking algorithm is applied on the vehicles in the emergency lane only.

So, the algorithm will be more simple and faster, also the need to be aware of the relative speed of the cars in front of our vehicle.. After the first detection, the positions of the vehicles in the emergency lane are saved, then start comparing at every frame the new position of the vehicles in the emergency lane with the last saved positions. If the difference of vehicle’s positions is within the threshold (difference < 3%) so it considered as the same vehicle, hence the tracking is achieved.



The final step of our proposed system is to detect vehicle collisions. Saved vehicles are being imported in this module. Lane points are being analyzed and a function for the region of interest is being written. To process input video, vehicle detection, lane analysis, bounding box parameters are being analyzed. If a car X is being driven on a highway a rectangle boundary in front and back of it would be formed. If in case any vehicle comes under that rectangle boundary region, we would be notified that there is a chance for collision. This system can be used in the cases of Self-driving cars where it would analyze the collision possibility automatically and drive accordingly.

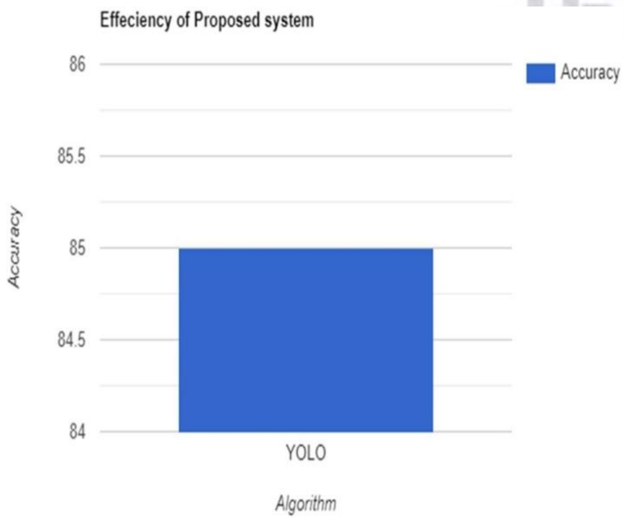
The train-test split is a technique for evaluating the performance of a machine learning algorithm. The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

System testing is the testing of complete and fully integrated software product. Usually software is only one element of a larger computer based system. Ultimately, Software is interfaced with other software/hardware systems. System testing is actually a series of different tests whose sole purpose is to exercise the full computer based system. The objective is to estimate the performance of the machine learning model on new data: data not used to train the model. This is how we expect to use the model in practice. We have chosen a split percentage of 80% : 20 % that meets our project's objectives with considerations

III. RESULTS

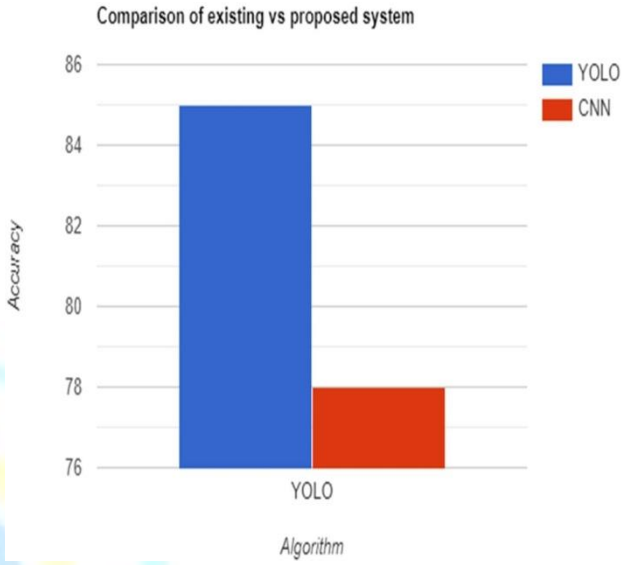
The efficiency of proposed system is plotted in the graph with respect to the accuracy of the trained model. The accuracy is 85 % for the video input and this accuracy is measured for YOLO model.

White box testing is a testing which includes the error in the coding section. It includes the error occurred during compilation and also during the development of the project.



This provides the final assurance that the software meets all functional, behavioral and performance requirements. This software is completely assembled as package. Validation succeeds when the software function in which the user expects.

The comparison of proposed system and existing system is plotted in the graph with respect to the accuracy of the existing algorithm CNN and the proposed algorithm YOLO. The accuracy is 85 % for YOLO Model and the accuracy is 78% for the CNN Model which is lower than the proposed system.



IV. CONCLUSIONS AND FUTURE WORK

The proposed system is faster than other object detection methods and predicts the object better than other object detection algorithms. The Advantages of the proposed system are Secured, Interpretability, High accuracy, Lightweight model & fast processing. Moreover, this system can be used in the cases of Self-driving cars. Where it would analyze the collision possibility automatically and drive accordingly It could be used in self-driving cars, traffic surveillance systems, traffic management, and automated driving applications.

REFERENCES

[1]. Yeong-Kang Lai, Chu-Ying Ho, Yu-Hau Huang, Chuan-Wei Huang, Yi-Xian Kuo, Yu-Chieh Chung, "Intelligent Vehicle Collision-Avoidance System with Deep Learning", 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018

[2]. Jae Gyeong Choi, Chan Woo Kong, Gyeongho Kim, Sunghoon Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras", Expert Systems with Applications, Volume 183, 30 November 2021.

[3]. P. Nirmala, T. Manimegalai, J. R. Arunkumar, S. Vimala, G. Vinoth Rajkumar, Raja Raju, "A Mechanism for Detecting the Intruder in the Network through a Stacking Dilated CNN Model", Wireless Communications and Mobile Computing, vol. 2022, Article ID 1955009, 13 pages, 2022. https://doi.org/10.1155/2022/1955009.

[4]. Aloukik Aditya, Liudu Zhou, HrishikaVachhani, Dhivya Chandrasekaran, Vijay Mago, " Collision Detection: An Improved Deep Learning Approach Using SENet and ResNext", 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021

[5]. Wan-Jung Chang, Liang-Bi Chen, Ke-Yu Su, "DeepCrash: A Deep Learning- Based Internet of Vehicles System for Head-On and Single-Vehicle Accident Detection With Emergency Notification", IEEE Access, Vol 7, P-148163 – 148175, 2019

[6]. Manu S. Pillai, Gopal Chaudhary, Manju Khari, Rubén González Crespo, "Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning", Soft Computing volume 25, pages11929–11940, 2021.

[7]. J. R. Arunkumar, S. Velmurugan, B. Chinnaiah, G. Charulatha, M. Ramkumar Prabhu et al., "Logistic regression with elliptical curve cryptography to establish secure iot," Computer Systems Science and Engineering, vol. 45, no.3, pp. 2635–2645, 2023.

[8]. G. Madhumitha, R. Senthilnathan, K. Muzammil Ayaz, J. Vignesh, Korada Madhu, " Estimation of Collision Priority on Traffic Videos using Deep Learning", 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), 2020

[9]. Yeong-Kang Lai, Yu-Hau Huang, Thomas Schumann, "Intelligent Vehicle Collision Warning System Based on a Deep Learning Approach", 2018 IEEE International Conference on Consumer Electronics-Taiwan ICCE-TW, 2018.

[10]. Prathima Chilukuri , J.R. Arun Kumar , R. Anusuya , M. Ramkumar Prabhu. "Auto Encoders and Decoders Techniques of Convolutional Neural Network Approach for Image Denoising In Deep Learning" Journal of Pharmaceutical Negative Results, 13(4), 1036–1040. https://doi.org/10.47750/pnr.2022.13.04.142 ,November 4, 2022.

[11]. R. Yugha, V. Vinodhini, J. R. Arunkumar, K. Varalakshmi, G. Karthikeyan and G. Ramkumar, "An Automated Glaucoma Detection from Fundus Images based on Deep Learning Network," 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Dharan, Nepal, 2022, pp. 757-763, doi: 10.1109/I-SMAC55078.2022.9987254.

[12]. M.Ramkumar Prabhu, A.Rajalingam, J.R.Arunkumar, R.Anusuya, "Microstrip Patch Antenna Using Combined Slots For Bandwidth Enhancement And Size" Journal of Engineering Sciences (JES),Vol 11, Issue 1, Jan / 2020, ISSN NO: 0377-9254.

[13]. HomaTaghipour, Amir BahadorParsa, Rishabh Singh Chauhan, Sybil Derrible, Abolfazl (Kouros) Mohammadian, "A novel deep ensemble-based approach to detect crashes using sequential traffic data", IATSS Research, 9 November 2021.

[14]. M.Ramkumar Prabhu, J.R.Arunkumar, A.Rajalingam, R.Anusuya "A Modified Square Patch Antenna with Rhombus slot for High bandwidth" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-9, July 2019.

[15]. Karishma Pawar, Vahida Attar, "Deep learning-based detection and localization of road accidents from traffic surveillance videos", ICT Express, 15 November 2021

[16]. Yan Miao, Fu Liu, Tao Hou, Lu Liu, Yun Liu, "A Nighttime Vehicle Detection Method Based on YOLO v3", 2020 Chinese Automation Congress (CAC), 2020