TINY ARCHITECTURE IMPLEMENTATION FOR ADAS TO PREVENT PEDESTRIAN ACCIDENT USING DEEP LEARNING

DR.K.SIVARAMAN, Kamineni Sandeep[1], Konda Balaji[2], K Joshua Sahai[3]

[1]*Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai-73*

[2] *Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai-73*

[3] *Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai-73*

---***---

Abstract - Pedestrian identification is a fundamental and critical undertaking keen video observation framework, as it gives data to semantic comprehension of the video recordings. It has an undeniable expansion to auto applications because of the potential for further developing wellbeing frameworks. In 2017, many car manufacturers offer this as an ADAS option. It is used in many cars and vehicles for automatic driver assistance systems. But when the ADAS systems detects pedestrians and fails to apply ABS on time accidents might happen. To overcome this situation our project provides an audio alert to the human passengers or driver in the vehicle so that they can apply manual brakes. In this project we will require Python libraries such as Pytorch and OpenCV. Pytorch is a broadly utilized AI library. It is popular for the YOLO (You Only Look Once) algorithm which is built for Object Detection, we are using the YOLO algorithm and customize it to detect objects on a pedestrian dataset. Python has a library pyttsx3, that is fit to switch text-over completely to audio. It extracts the label from the detected pedestrians in the video and converts the text label into speech. The algorithm detects the pedestrians when they are in a close approach to the vehicles and sends the identifies labels to the **pyttsx3** speaker engine. An audio alert is generated by the engine and alerts the passengers or drivers.

Key Words**:** YOLO, Pytorch, pyttsx3, object detection**,** python

1.INTRODUCTION

ADAS (Advanced Driver Assistance Systems) are detached and dynamic security frameworks intended to eliminate the human blunder part while working vehicles of many kinds. ADAS frameworks utilize trend setting innovations to help the driver during driving, and subsequently work on drivers' exhibition. ADAS utilizes a mix of sensor innovations to see the world around the vehicle, and afterward either give data to the driver or makes a move when important. The main component of the ADAS system is pedestrian detection. Upon detecting a pedestrian, the ADAS system is supposed to apply emergency brakes. According to the World Health Organization, each year, approximately 1.35 million people are killed as a result of traffic accidents. Between 20 million and 50 million others are affected by non-fatal injuries and many are disabled as a result. Half of road traffic fatalities belong to the category of vulnerable road users, cyclists, and pedestrians. There are also cases where the ABS failed. This algorithm generates an audio alert to alert the drivers to be ready to take manual action in case the ABS system fails, by detecting the pedestrians in the effective region using the YOLOv7 algorithm.

2. LITERATURE REVIEW

In 2018, AAA conducted tests on ADAS vehicles, specifically focused on pedestrian detection. The test resulted in the publication of devastating results by the AAA. The test brought about AAA's distribution of decimating results. AAA led AEB-P testing the previous fall on four 2019 model-year vehicles: a Chevrolet Malibu with Front Walker Slowing down, a Honda Accord with Honda Detecting Impact Stopping mechanism, a Tesla Model 3 with Programmed Crisis Slowing down and a Toyota Camry with Toyota Security Sense. The key findings are:

If an adult was encountered crossing the road in daylight by a test vehicle going 20 mph, the car avoided hitting then pedestrian only 40 percent of the time. More regrettable, in the event that the test vehicle going at 20 mph met a kid shooting into traffic from between two vehicles, the youngster got nailed 89% of the time. At 30 mph, none of the test vehicles stayed away from a crash.

Even though the Automatic Vehicle developers constantly working on improvising the Automatic Emergency Braking Systems. For the time being there has not been much progress made. So, in order to overcome the failure of braking systems, we provide an alert system for the passengers and driver to take the manual control.

3. EXISTING SYSTEM

In the existing system we observe that emergency brakes are applied when the object is detected by the vehicle and if the ABS fails the object is hit by the vehicle. The collision probability of the system when the ABS fails is noticeably higher in the existing system. Even if the system detects the pedestrian correctly, it might fail to apply the brakes on time.

A. Disadvantages of Proposed System

1) The vehicle may hit the pedestrian if the ABS fails.
2) According to AAA tests the vehicle hitting the pedestrian has a probability of 89%.

4. PROPOSED SYSTEM

When we compare with the current features present in an ADAS system the proposed system has a voice alert system which alerts the passengers when a pedestrian comes in the path of the vehicle. And the driver would be ready to take the manual action if needed. The voice alert helps when the driver is drowsy or un-attentive towards the path, in case of an ABS failure.

A. Advantages of Proposed System

1) The vehicle alerts the driver or passenger when a pedestrian is in a close range of approach.
2) The audio alerts the driver to be ready to take the manual action if required.

5. MODULES

The modules to run the project:

1) Getting the feed from monocular camera**:** The system acquires the real time data from the monocular camera mounter to the vehicle.

2) Processing the video: the footage is processed to be understandable by the detection algorithm.

3) Running detection algorithm on the acquired footage: The detection algorithm detects the objects in the footage and gives the number of pedestrians in the video.

4) Creating a voice alert: The algorithm extracts the text from the detected label and creates an audio alert.

5) Alerting the driver**:** The driver is alerted by playing the alert.

6. SYSTEM REQUIREMENT

A Software Requirements specification (SRS) – a requirements specification for a software system- is a complete description of behavior of a system to be developed. It contains a collection of cases that describe all the interactions that users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are those that place constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints

A. Fundamental Requirements

The Functional Prerequisites The operations and activities that a system must be able to perform are specified in the specification. Functional requirements should include specific screen functions, work-flow outlines, and any other business or compliance requirements that the system must meet. It also depends on the type of software, the number of expected users, and the system on which the software is used.

1) The monocular camera recording the footage.
2) The system checks the real time integrity of the footage.
3) The system performs detection algorithm on the footage.

B. Non-Fundamental Requirements

A non-functional requirement in systems engineering is one that specifies criteria that can be used to judge the operation of a system rather than specific behaviors. They differ from functional requirements, which define specific behavior or functions. Non-functional requirements can be regarded as system performance quality attributes.

1) Efficiency Requirement: When the pedestrian detection algorithm is implemented, the count of detected pedestrians is converted into an audio to alert the driver or passenger.

2) Reliability Requirement: The system should accurately perform the detection of pedestrian when they are in a close range of approach when emergency brake fails.

3) Usability Requirement: The system is designed as an alert system for the drivers or passengers when a pedestrian comes into a close range of collision.

7. DATASET USED FOR TRAINING

An extract of the INRIA persons dataset is used to train the model with YOLOv7 tiny architecture. The train dataset consisted of 644 images with annotation files. The test dataset consists of 218 images to test with the respective annotations.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Fig -1: Sample images used for training

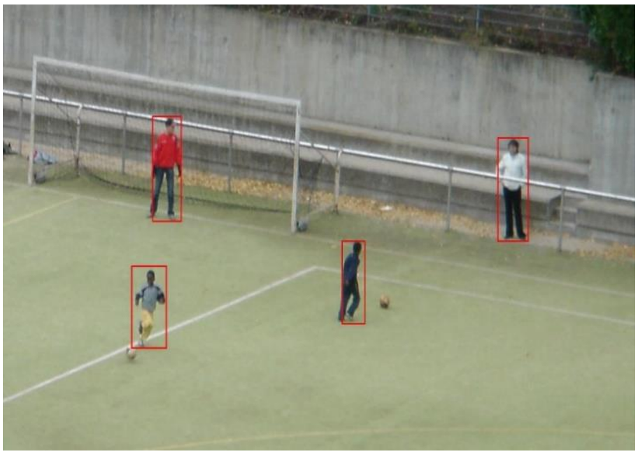

Fig -2: Labeled images used for training

8. Proposed Yolo Architecture For Pedestrian Detection

YOLO is an abbreviation for "You Only Look Once," and it refers to a popular family of real-time object detection algorithms. The first YOLO object detector was released in 2016. Joseph Redmon, Ali Farhadi, and Santosh Davila created it. When it was first introduced, this architecture outperformed other object detectors in terms of speed, and it quickly established itself as the industry standard for real-time computer vision applications.

YOLOv7 is a real-time object detector with a single stage. In July of 22nd, it was introduced to the YOLO family. The YOLOv7 paper claims that it is the fastest and most accurate real-time object detector to date. YOLOv7 raised the bar significantly by improving its performance.

The YOLO architecture is based on FCNN (Fully Connected Neural Network). However, Transformer-inspired versions have recently joined the YOLO family. In a separate post, we will go over transformer-based detectors. For the time being, let's concentrate on FCNN (Fully Convolutional Neural Network)-based YOLO object detectors.

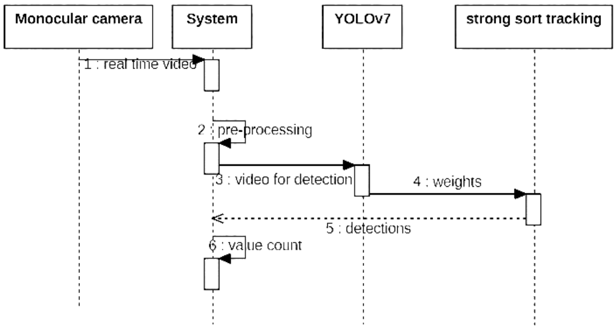

Fig -3: Yolov7 Working

The Fig 4 diagram provides a way to model the workflow of a development process.

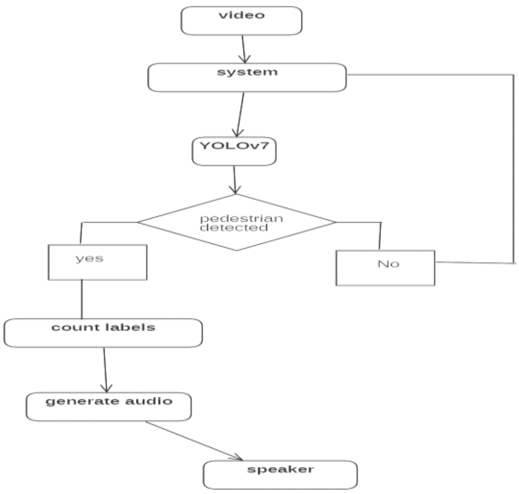

Fig -4: Yolov7 Working

8. EXPERIMENT AND RESULTS

The INRIA person dataset was created by collecting high-resolution images from all websites in order to detect people on the road. The dataset consists of 814 images, of which 644 have annotation files for training and 218 are for testing. During the training, the initial learning rate is 9.02. After extensive testing and careful consideration of training time and GPU memory, this study sets the number of epochs and batch size to 60 and 8, respectively, to ensure that training is completed on schedule. Since many of the modules of the upgraded network architecture are the same as the layers in the original YOLOv7 module, this research conducts ablation experiments to enhance YOLOv7 using the pre-trained model supplied on the official website. The precision of the experiment is not considerably impacted by this strategy, which also saves time and money. The PyTorch 1.10.0 environment, an Intel (R) Xeon (R) Platinum 8255C Processor operating at 2.50 GHz, and the training and reasoning of all models were employed in this experiment on the Nvidia GPU. Fig 5 illustrates example of detection on image chosen from the INRIA person dataset. After detection is complete, the camera footage of the recognized pedestrians is given audio. Using the recognized video, an audio alert asking how many pedestrians are detected. Fig 6 illustrates the detection of the pedestrian and plays an audio.

Fig -5: Predicted Image

Fig -6: Detection of the pedestrian and plays an audio

9. CONCLUSION

In this paper we use YOLOv7 algorithm to predict the pedestrians in a vehicles path and alert the driver with a audio message. The algorithm detected the pedestrians with an average confidence of 0.95. The metrics of the algorithm is shown in Fig 8. The proposed work in different weather and light conditions the pedestrians fine tune the model giving the input which pedestrian in different weather and light conditions using the same process in the research paper.

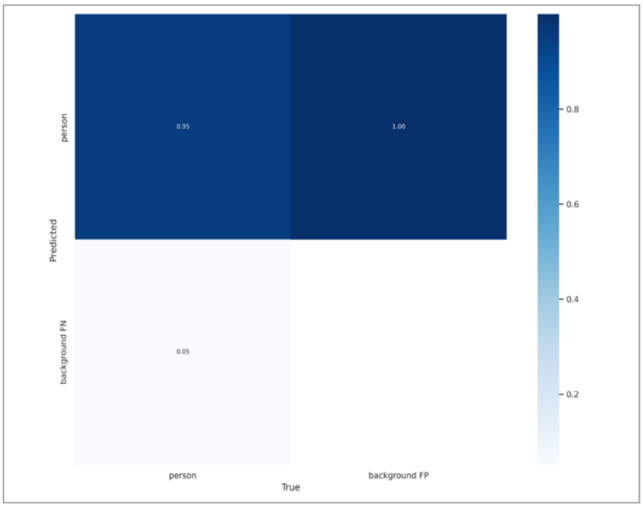

Fig -7: Metrics

ACKNOWLEDGEMENT

YOLOv7 outperforms all known object detectors in terms of both speed and accuracy in the 5 FPS to 160 FPS range, and has the highest accuracy (56.8% AP) among all known real-time object detectors with 30 FPS or higher on GPU V100. In terms of speed and accuracy, the YOLOv7-E6 object detector (56 FPS V100, 55.9% AP) outperforms both the transformer-based detector SWIN-L Cascade-Mask R-CNN (9.2 FPS A100, 53.9% AP) and the convolutional-based detector ConvNeXt-XL Cascade-Mask R-CNN (8.6 FPS A100, 55.2% Furthermore, we only train YOLOv7 on the MS COCO dataset.

https://doi.org/10.48550/arXiv.2207.02696

REFERENCES

1. Ce Li, Yachao Zhang and YanyunQu, “Object Detection Based on Deep Learning of Small Samples,” International Conference, pp.1-6, March 2018.
2. Cong Tang, YunsongFeng, Xing Yang, Chao Zheng, and Yuanpu Zhou, "Deep Learning-Based Object Detection," International Conference, pp.1-6, 2017.
3. "Deep Neural Networks for Object Detection," by Christian Szegedy, Alexander Toshev, and DumitruErhan, IEEE, pp.1-9, 2007.
4. Xiaogang Wang, “Deep Learning in Object Recognition, Detection, and Segmentation,” IEEE, pp.1-40, Apr,2014.
5. Shuai Zhang, Chong Wang and Shing-Chow Chan, “New Object Detection, Tracking, and Recognition Approaches for Video Surveillance Over Camera Network,” IEEE SENSORS JOURNAL, vol. 15, no.69, pp. 1-13, May 2015.
6. Malay Shah and Prof. RupalKapdi, "Object Detection Using Deep Neural Networks," IEEE International Conference, 2017.
7. Xiao Ma, Ke Zhou, and JiangfengZheng, "Photorealistic Face Age Progression/Regression Using a Single Generative Adversarial Network," Elsevier B.V., Neurocomputing, pp.1-16, July 2019.
8. Xindong Wu, Zhong-Qiu Zhao, PengZheng, Shou-taoXu, and Zhong-Qiu Zhao, "Object Detection with Deep Learning: A Review," IEEE, pp.1-21, 2019.
9. Sandeep Kumar, AmanBalyan and ManviChawla, “Object Detection and Recognition in Images,” IJEDR, pp.1-6, 2017.
10. Adami Fatima Zohra, SalmiKamilia, Abbas Faycal and SaadiSouad, “Detection And Classification Of Vehicles Using Deep Learning,” International Journal of Computer Science trends and technology(IJCST), vol. 6, pp.1-7,2018.