**COLLABORATIVE PROJECT WITH INTEL**

**PROJECT TITLE : ROAD OBJECT DETECTION WITH DEEP LEARNING**

**TEAM NAME :** **TEAM\_QUBIT**

**Team Mentor : Dr T V RAJINI KANTH, Professor & Head,**

**Department of CSE-AI&ML**

[**rajinitv@gmail.com**](mailto:rajinitv@gmail.com)**, Ph. No: 9849414375**

**Team Members : V. Sri Varun**  Roll no: 20311A0414 ECE Team lead

**Y. Dharan Datta**  Roll no: 20311A04W6 ECE Team member

**T. Praneeth Reddy** Roll no: 20311A0535 CSE Team member

**Institute Name** **:** **Sreenidhi Institute of Science and Technology**

**Yamnampet, Ghatkesar**

**Hyderabad - 501301**

**Date of submission :** **12 -07- 2023**

**ABSTRACT:** Road safety is a major concern today. To overcome that realistic environmental issues it is required to develop suitable tools or GUI etc. Deep Automated driving and vehicle safety systems need object detection. It is important that object detection be accurate overall and robust to weather and environmental conditions and run in real-time. As a consequence of this approach, they require image processing algorithms to inspect the contents of images. This article compares the accuracy of a few major image processing algorithms: Deep Neural Networks (DNN), Region-based Fully Convolutional Networks (R-FCN), Mask Region-based Convolutional Neural Networks (Mask R-CNN), Single Shot Multi-Box Detector (SSD), RetinaNet, and You Only Look Once (YOLO).

Here we have taken care of Helmet-No Helmet detection and license plate photo capturing

**Key Words: Road Safety, Helmet-No Helmet, DNN**

**INTRODUCTION**

The automobile industries have developed rapidly since the first demonstration in the 1980s and the vehicle navigation and intelligence system have improved. However, the increase in road vehicles raises traffic congestion, road safety, pollution, etc. Autonomous driving is a challenging task; a small error in the system can lead to fatal accidents. Visual data play an essential role in enabling advanced driver-assistance systems in autonomous vehicles. The low cost and wide availability of vision-based sensors offer great potential to detect road incidents. Additionally, emerging autonomous vehicles use various sensors and deep learning methods to detect and classify four classes (such as a vehicle, pedestrian, traffic sign, and traffic light) to improve safety by monitoring the current road environment.

**MOTIVATION**

The motivation for studying road object detection is to understand the existing Object Detection Models, analyze the respective deep learning-based algorithm's performance for object detection under practical constraints. Some of the challenges that road object detection faces include:

* Come up with solutions to road traffic rule violations handing when there was no human monitoring
* The trade-off between speed and accuracy, as autonomous vehicles require real-time detection with high precision13.
* To understand and Hand-on experience of Object Detection algorithms implementation in Python.
* To contribute to society with new inventions to overcome practical problems which can be overcome with Object detection
* Collision avoidance systems have significantly decreased the injury and death rate in accidents. With the development of both software algorithms and hardware components, it is getting more precise for cameras to detect multiple objects in a real-time scenario using DNN.

The Intel Industrial Training initiative Unnati Program helps the students in getting the flavor of the Industrial View of the work planning, interaction, and guidance of the Intel Team and friendly competition with other college students.

**DATA SOURCES**

Studied and analyzed the Indian Driving Dataset IDD Lite for understanding the object detection procedure [IDD (iiit.ac.in)](https://idd.insaan.iiit.ac.in/) and Google Images.

We have taken a few images with Mobile Camera on Hyderabad Road which has various objects like cars, trucks, helmets and no-helmet detections, and number plate capturing.

Developed code to take the input as images and object detection labels and create frames too with identified license plates of bikes without helmets.

**PYTHON LIBRARIES USED IN THE PROGRAMS**

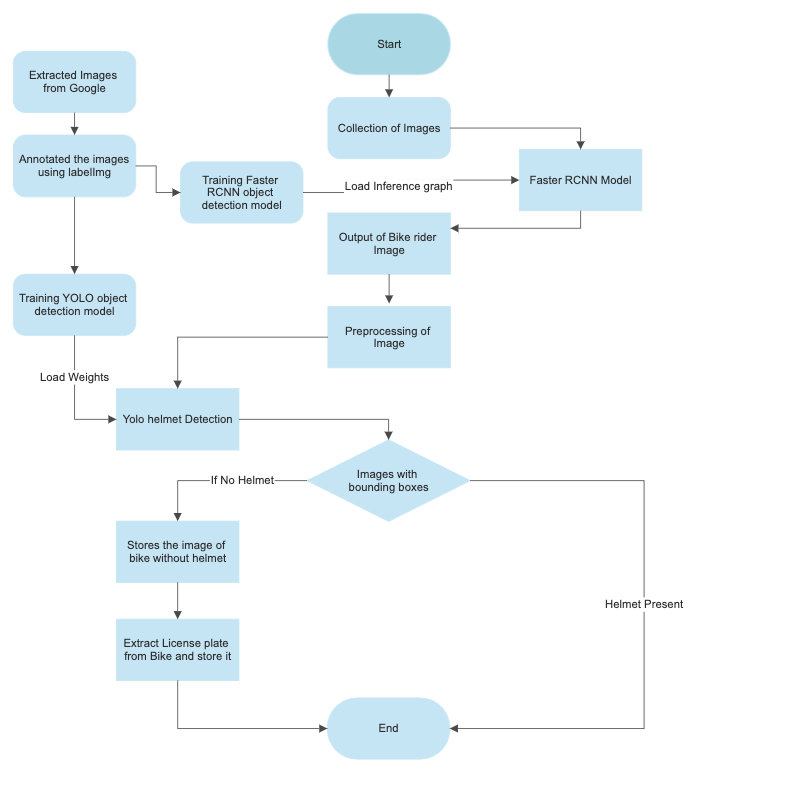
**CV2 or CV:** OPEN CV Python library of Python bindings designed to solve computer vision problems. **OpenCV** is a huge open-source library for computer vision, machine learning, and image processing.

**Numpy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

**OS:** The **OS module in Python** provides functions for interacting with the operating system.

**Tensor flow:** Keras is a neural network Application Programming Interface (API) for Python developed by Google that is tightly integrated with Tensor Flow, which is used to build machine learning models. Keras' models offer a simple, user-friendly way to define a neural network, which will then be built for you by Tensor Flow.

**ARCHITECTURE**

****

**RESULTS**

**Fig 1. Input Image**

****

**Fig 2. The output of the RCNN Model i.e the bike image and without the Helmet By the YOLO Model**

****

**Fig 3. License plate photo capture by YOLO Model**

****

**CONCLUSION**

Analysed IDD Lite dataset, and developed a program to detect Helmet and Helmet Objects on a Road Object images. Learned how to work on large-scale projects, and use pre-trained models to implement the solution. Successfully implemented YOLOV3 algorithm in DNN to detect Road Objects in Python.

**FUTURE SCOPE**

It can be developed further to detect more types of Road Traffic Rule Violations to improve the efficiency of Traffic Control Systems and also immediate notification to bike users for not wearing helmet.

**REFERENCES**

**[1].**[**https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FNetwork-architecture-of-YOLOv3\_fig1\_339763978&psig=AOvVaw3bu6AXRJ9jcbj2ouae9jY2&ust=1688623769396000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCPiagvfz9v8CFQAAAAAdAAAAABAI**](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FNetwork-architecture-of-YOLOv3_fig1_339763978&psig=AOvVaw3bu6AXRJ9jcbj2ouae9jY2&ust=1688623769396000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCPiagvfz9v8CFQAAAAAdAAAAABAI)

[Electronics | Free Full-Text | Road Object Detection: A Comparative Study of Deep Learning-Based Algorithms (mdpi.com)](https://www.mdpi.com/2079-9292/10/16/1932)

[2]. [OpenCV DNN Module and Deep Learning (A Definitive guide) (learnopencv.com)](https://learnopencv.com/deep-learning-with-opencvs-dnn-module-a-definitive-guide/)

[3]. [A Gentle Introduction to Object Recognition With Deep Learning - MachineLearningMastery.com](https://machinelearningmastery.com/object-recognition-with-deep-learning/)

[4]. [Official English Documentation for ImageAI! — ImageAI 3.0.2 documentation](https://imageai.readthedocs.io/en/latest/)

[5]. [NumPy](https://numpy.org/)

[6]. [Microsoft Word - Traffic Violation Proctoring System Document.docx (anits.edu.in)](https://ece.anits.edu.in/Project%20Reports%202020-21/Sec-C/C-13.pdf)

[7]. [The Definitive Guide to Object Detection (augmentedstartups.com)](https://www.augmentedstartups.com/blog/the-definitive-guide-to-object-detection-applications-models-and-future-trends)

[8]. Mahaur, B., Singh, N. & Mishra, K.K. Road object detection: a comparative study of deep learning-based algorithms. Multimed Tools Appl 81, 14247–14282 (2022). <https://doi.org/10.1007/s11042-022-12447-5>.

[9]. Song, H., Liang, H., Li, H. et al. Vision-based vehicle detection and counting system using deep learning in highway scenes. Eur. Transp. Res. Rev. 11, 51 (2019). <https://doi.org/10.1186/s12544-019-0390-4>

[10]. Emin Güney, Cuneyt Bayilmis, An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms, January 2022IEEE Access PP(99):1-1, DOI:[10.1109/ACCESS.2022.3198954](http://dx.doi.org/10.1109/ACCESS.2022.3198954).

<https://www.diva-portal.org/smash/get/diva2:1249005/FULLTEXT01.pdf>

**SOURCE CODE AND CONFIGURATION FILES**

1. **Source code for the project is attacted to the Google Colab Link.** [**Colab Link**](https://colab.research.google.com/drive/1qCC6mbaKS7rYHspZB7qVoiboB1WzL1S6?authuser=2#scrollTo=eLqOML_ZdwVs)
2. **The models data and files are attacted to Google Drive Link.** [**Drive Link**](https://drive.google.com/drive/folders/1iqh_2374fjyBnMxlEGtc2NPEFtnaTJvS?usp=drive_link)

**(Please make sure that the entire is cloned to google drive to implementation.)**