TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM USING DEEP LEARNING

Naveen Varma Pandeti, Sumanth Reddy Busireddy, Siri Sushma Penumatsa
Department of Computer Science
Central Michigan University, USA
Email:pande3n@cmich.edu, busir1s@cmich.edu, penum1s@cmich.edu

I. Abstract

The most important challenge to sustainable mobility is persistent congestions of differing strength and duration in the dense transport networks. The standard Adaptive Traffic Signal Control cannot properly address this kind of congestion. Deep learning-based mechanisms have proved their significance to anticipate in adjective outcomes to improve the decision making on the predictions of traffic length. The deep learning models have long been used in many application domains which needed the identification and prioritization of adverse factors for a simplifying human life. Several methods are being popularly used to handle real time problems occurring from traffic congestion. This study demonstrates the capability of DL models to overcome the traffic congestion by simply allowing the vehicles through a signal depending on the length of vehicles. Our proposed method integrates a numeral of approach,intended to advance the cooperativeness of the explore operation. In this work, we develop the application to regulate the traffic by releasing better signal at desired time intervals. we Have also implemented the application to detect the number of vehicles in the images from the user and gives vehicles counts. To detect the vehicles count here we are using the YOLO pretrained weights.

II. Introduction

A. motivation

Efficient traffic prediction is crucial for the optimization of intelligent transportation systems. By accurately forecasting traffic patterns, congestion hotspots can be identified in advance, enabling proactive measures to alleviate traffic congestion and improve overall traffic flow. [1] [2] [3] Deep learning techniques, such as You Only Look Once (YOLO), have demonstrated remarkable success in object detection tasks. In this study, we leverage the power of YOLO to develop a robust traffic prediction model, aiming to enhance the efficiency and effectiveness of intelligent transportation systems.

B. Problem Statement

The existing traffic prediction Studies for intelligent transportation systems often suffer from limited accuracy and insufficient real-time capabilities. Additionally, the integration of external factors such as weather conditions and events is often overlooked. This study aims to address these limitations by proposing a deep learning-based approach, specifically utilizing You Only Look Once (YOLO) for traffic prediction. The goal is to develop a more accurate and robust model that can handle real-time traffic data and incorporate external factors for enhanced intelligent transportation system performance.

C. The Objective of The Project

The objective of this project is to develop a deep learning-based traffic prediction model using You Only Look Once (YOLO) for intelligent transportation systems. The primary goals include improving the accuracy of traffic congestion prediction, enabling real-time prediction capabilities, and integrating external factors such as weather conditions and events for more comprehensive and reliable predictions. The project aims to enhance the efficiency and effectiveness of intelligent transportation systems, leading to improved traffic management and optimized resource allocation.

D. Scope

[4]The scope of this project encompasses the development and implementation of a deep learning-based traffic prediction system using You Only Look Once (YOLO) for intelligent transportation systems. The project includes the collection and preprocessing of historical traffic data, integration of real-time traffic data sources, training and fine-tuning the YOLO model for accurate traffic prediction, and incorporating external factors such as weather conditions and events. The scope also covers the evaluation of the model's performance and its potential integration into existing intelligent transportation systems.

E. Project Introduction

Over the last decade by solving many very complex and sophisticated real-world problems. [5] [6] The application areas included almost all the real-world domains such as healthcare, autonomous vehicle (AV), business applications, and image processing. DL algorithms' learning is typically based on trial-and-error method quite opposite of conventional algorithms, which follows the programming instructions based on decision statements like if-else. One of the most significant areas of DL is simplifying human problems, in many application areas including medical domain, governments every sector is showing their interest to introduce AI to their systems. Various models have wide applicability in working with the conditions of real time. There are lots of studies performed for regulating traffic using deep learning techniques such as image segmentation, object detection etc.., In particular, the study is focused on live traffic regulating near a traffic signal and study is also focused on the decreasing the waiting time depending on vehicle counts and early response. These systems can be very helpful in decision making to handle the present scenario to guide early interventions to manage these traffic regulations very effectively. This study aims to provide an better system which can be able to release the traffic depending on the count of vehicles. This project targets to develop web application in order to handle the "Traffic Congestion". Python-Flask is used as front end which is used to craft the user interface. MySQL is used as back end and used to craft the database and save the particulars. Anybody with a little computer knowledge can approach and deal with the software with ease; hence it can be termed user friendly.

III. LITERATURE REVIEW

A. Case Study 1 :

[7] Advanced vehicle guidance systems use real-time traffic information to route traffic and to avoid congestion. Unfortunately, these systems can only react upon the presence of traffic jams and not to prevent the creation of unnecessary congestion. Anticipatory vehicle routing is promising in that respect, because this approach allows directing vehicle routing by accounting for traffic forecast information. This paper presents a decentralized approach for anticipatory vehicle routing that is particularly useful in large-scale dynamic environments. The approach is based on delegate multiagent systems, i.e., an environment-centric coordination mechanism that is, in part, inspired by ant behavior. Antlike agents explore the environment on behalf of vehicles and detect a congestion forecast, allowing vehicles to reroute. The approach is explained in depth and is evaluated by comparison with three alternative routing strategies. The experiments are done in simulation of a real-world traffic environment. The experiments indicate a considerable performance gain compared with the most advanced strategy under test, i.e., a traffic-message-channel-based routing strategy.

B. Case Study 2:

[8] Mobile Ad Hoc Network (MANET) has the ability to self-configure and establish a mobile wireless mesh that can be used in extreme conditions, such as in areas affected by disasters. One of the routings in MANET is AODV routing. AODV is one of the reactive routing needed to send data. However, in the implementation of disaster conditions, AODV has weaknesses that are vulnerable to extreme environmental conditions. In this study, communication will be modeled that leads to disruption due to disaster. MANET AODV-DTN is used to improve network performance. With this system, the Probability Delivery Ratio (PDR) parameter value can be increased as evidenced by the variable modification of the number of nodes to be 0.431%, reducing the average delay by 63.525%, and producing the energy consumption increased by 0.170%. Simulation with the variable modification of speed obtained by PDR 0.482%, reducing the average delay by 78.710% and energy consumption increased by 0.167%. Modification of buffer size variables obtained 0.729% PDR results, reducing the average delay of 71.603% and energy consumption increased by 0.161%. From these data, MANET AODV-DTN is better than MANET AODV.

C. Case Study 3 :

[9] The rapid uptake of mobile devices and the rising popularity of mobile applications and services pose unprecedented demands on mobile and wireless networking infrastructure. Upcoming 5G systems are evolving to support exploding mobile traffic volumes, real-time extraction of fine-grained analytics, and agile management of network resources, so as to maximize user experience. Fulfilling these tasks is challenging, as mobile environments are increasingly complex, heterogeneous, and evolving. One potential solution is to resort to advanced machine learning techniques, in order to help manage the rise in data volumes and algorithm-driven applications. The recent success of deep learning underpins new and powerful tools that tackle problems in this space. In this paper we bridge the gap between deep learning and mobile and wireless networking research, by presenting a comprehensive survey of the crossovers between the two areas. We first briefly introduce essential background and state-of-theart in deep learning techniques with potential applications to networking. We then discuss several techniques and platforms that facilitate the efficient deployment of deep learning onto mobile systems. Subsequently, we provide an encyclopedic review of mobile and wireless networking research based on deep learning, which we categorize by different domains. Drawing from our experience, we discuss how to tailor deep learning to mobile environments. We complete this survey by pinpointing current challenges and open future directions for research.

D. Case Study 4 :

[10] Travel time is a fundamental measure in transportation. Accurate travel-time prediction also is crucial to the development of intelligent transportation systems and advanced traveler information systems. We apply support vector regression (SVR) for travel-time prediction and compare its results to other baseline travel-time prediction methods using real highway traffic data. Since support vector machines have greater generalization ability and guarantee global minima for given training data, it is believed that SVR will perform well for time series analysis. Compared to other baseline predictors, our results show that the SVR predictor can significantly reduce both relative mean errors and root-mean-squared errors of predicted travel times. We demonstrate the feasibility of applying SVR in travel-time prediction and prove that SVR is applicable and performs well for traffic data analysis.

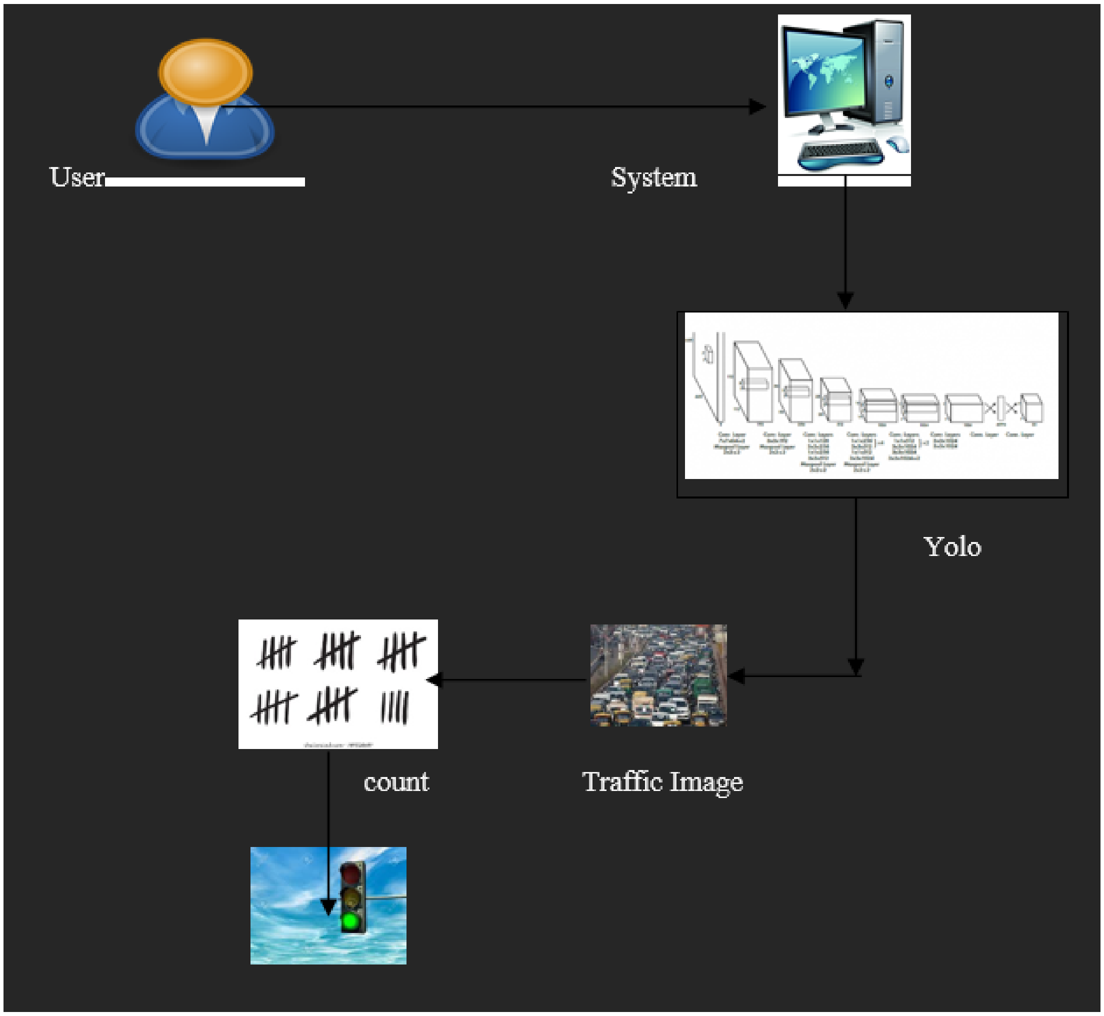

Fig. 1: System Architecture.

IV. System Analysis

To fully understand just how powerful this traffic congestion problem is, we need to look at it from the very beginning which is traditional way of controlling traffic. Traditional way basically uses, a person should make traffic observations to estimate or provide the clearance to the lane which consists of high count of vehicles. Traditional method later updated by using remotes-controlled system to give the right signal to the lanes. And these practices fail because the persons may not be available at every time at the traffic center. In previos Studies [11] [12] [13] [14] we have studied that there are some disadvantages like :

- Low efficiency.
- Time consuming.
- High complexities.
- Resources consuming

So, We proposed this application that can be considered a useful system since it helps to reduce the limitations obtained traditional methods. By providing support through the a DL based analysis, it can be able to generate best results for attributes without any overlap. The system is developed in a Flask based Python environment. The models involved in this application are YOLO that helps in improving **High efficiency, Time Saving, Low complexities.**

V. HARDWARE AND SOFTWARE REQUIREMENTS

A. *Hardware Configuration:*

The system specifications outline the key components for an efficient computing setup. Featuring an I3/Intel Processor, 4GB minimum RAM for smooth multitasking, and a spacious 128GB Hard Disk, this configuration is suitable for general tasks. The detailed specifications provide insights into the system's capabilities and suitability for various computing needs.

Component	**Specification**	**Additional Information**
Processor	I3/Intel Processor	Suitable for general tasks
RAM	4GB (minimum)	Allows for smooth multitasking
Hard Disk	128GB	Adequate storage capacity

TABLE I: Detailed System Specifications

B. Software Configuration

The development environment is built on Windows 11, utilizing Python 3.6+ for server-side scripting. PyCharm serves as the integrated development environment (IDE), fostering a seamless coding experience. Key libraries, including Pandas, Numpy, Sklearn, and Yolo, enhance functionality, while Flask provides a robust framework for web development.

Component	**Specification**
Operating System	Windows 8+
Server-side Script	Python 3.6+
IDE	PyCharm
Libraries Used	Pandas, Numpy, Sklearn, Yolo
Framework	Flask

TABLE II: Development Environment Specifications

VI. Methodology

[15] [16] The main aim of the proposed methodology is to develop and implement an intelligent traffic prediction system using deep learning techniques, specifically leveraging the You Only Look Once (YOLO) model. The primary objectives include improving the accuracy of traffic congestion prediction, enabling real-time prediction capabilities, events for enhanced performance within intelligent transportation systems. The methodology also seeks to evaluate the system's potential for integration into existing traffic management frameworks and to contribute to the optimization of resource allocation in traffic flow.

Types of Algorithms used :

A. CNN (Convolutional Neural Network)

[17] In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analysing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels that shift over input features and provide translation equivariant responses. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme. The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

It is widely used in many Applications Such as : Image recognition, Video analysis, Natural language processing, Anomaly Detection, Drug discovery, Health risk assessment and biomarkers of aging discovery etc..

B. YOLO

[18]Yolo is a part of object detection, Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e., the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found.

It is widely used in computer vision tasks such as image annotation, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person.

VII. Implementation

A. Required libraries

[19] In the implementation of the Intelligent Traffic Congestion (ITC) system, a structured approach is followed to maximize the potential of deep learning, specifically the You Only Look Once (YOLO) model for efficient traffic prediction. The first step is to install all of the required libraries and packages, which include Python 3.6+, Pandas, Numpy, Sklearn, and the YOLO framework. Collectively, these libraries provide a stable environment for constructing and training the deep learning model.

B. Required Software Packages

Miniconda3 manages Python dependencies for seamless integration of CNN and YOLO models in Traffic Prediction for Intelligent Transportation Systems. MySQL facilitates efficient data storage, while PyCharm and VSCode streamline code development and collaboration, enhancing the overall effectiveness of deep learning-based traffic prediction projects.

Software	Description
Miniconda3-latest-Windows-x86_64	Miniconda is a lightweight distribution of Python that includes a minimal set of utilities, package manager (conda), and dependencies. It facilitates the management of Python environments and package dependencies for various projects.
mysql-installer-community-8.0.32.0	MySQL Installer for the Community Edition simplifies the installation and configuration of MySQL databases on Windows. It includes MySQL Server, MySQL Workbench, and other essential components, providing an intuitive interface for database management.
pycharm-community-2022.2.3	PyCharm is an integrated development environment (IDE) for Python. The Community Edition is free and open-source, offering features like code completion, syntax highlighting, and support for web development frameworks.
VSCodeUserSetup-x64-1.72.2	Visual Studio Code (VSCode) is a lightweight, open-source code editor. Known for its extensibility, multi-language support, and rich feature set, it is widely used for web development and offers built-in Git integration and debugging support.

TABLE III: Software Information

C. *Training the Model*

The second stage includes picture loading, which is essential for training a model based on real-world traffic conditions.The system processes a wide range of pictures to ensure that the model is trained on an extensive dataset that reflects a diverse set of traffic scenarios. In the third step, the photographs are preprocessed to enhance the training model. Deep learning's power, such as using the You Only Look Once (YOLO) model for effective traffic prediction.This involves tasks such as normalization, resizing, and other image augmentation techniques to optimize the model's learning process.

D. *Working of Algorithms*

Moving to the core algorithms,The fourth phase includes the design and development of Convolutional Neural Network (CNN) with a strong focus on YOLO. CNN, a deep neural network class, is excellent at analyzing visual imagery, making it perfect for image-based applications such as traffic prediction. YOLO, in addition, stands out for its accuracy in object identification, which is critical for recognizing and forecasting traffic patterns. Finally, the best-trained model is deployed to provide real-time clearance signals for effective traffic management, resulting in an intelligent and responsive traffic prediction system for both X-Junction And Y- Junction.

Algorithm 1 Flask Application - Part 1

Import Libraries:
Import Flask, render_template, session, redirect, url_for, request, os, cv2, matplotlib
Import cvlib for object detection
Initialize Flask App:
Create Flask app instance
Global Variables:
Declare global variables for vehicle counts and output filenames
Define Routes:
Define routes for home, image uploads, and result views
Object Detection Function:
Define a function for vehicle detection using YOLO
Main Execution:
Run Flask app in debug mode

The Python code provided is a Flask application for traffic analysis that employs YOLO (You Only Look Once) object recognition. It counts vehicles in submitted images using the cvlib library. The program provides paths for uploading pictures, analyzing them using YOLO, and showing the results. It has the features for handling picture uploads, object detection, and analyzing vehicle counts at intersections. The code stores vehicle counts and filenames in a global variable. It also includes basic error handling. The program runs as a Flask server and provides a basic UI for evaluating traffic conditions and calculating vehicle counts at various intersections.

Algorithm 2 Flask Application - Part 2

Handling Image Uploads:
Handle image uploads, YOLO processing, and result display
Save processed images in 'static/outputs' directory
Junction Analysis Routes:
Implement routes for analyzing vehicle counts at junctions (xjunction, yjunction)
Upload and Display:
Upload images, process using YOLO, and display results
Redirect to subsequent upload pages on successful processing
Error Handling:
Implement basic error handling with try-except blocks

VIII. TESTING AND RESULTS

The provided code implements a Flask web application with a key feature: a "Follow Link" button on the homepage. When users visit the application and click on this button, they are seamlessly redirected to a developed application focused on Traffic Prediction for Intelligent Transportation System using deep Learning. This functionality enhances user experience, allowing easy navigation to the specific application dedicated to traffic prediction, thereby showcasing the seamless integration and accessibility of intelligent transportation solutions.

In order to enhance the accuracy and robustness of our Traffic Prediction application for Intelligent Transportation Systems, We encourage users to submit a minimum of four images for X-junction scenarios and a minimum of three images for Y-junction scenarios . This diverse dataset allows our deep learning models, based on Convolutional Neural Networks (CNN) and YOLO (You Only Look Once) algorithm, to better comprehend and adapt to varying traffic conditions.By providing a richer set of visual data, users

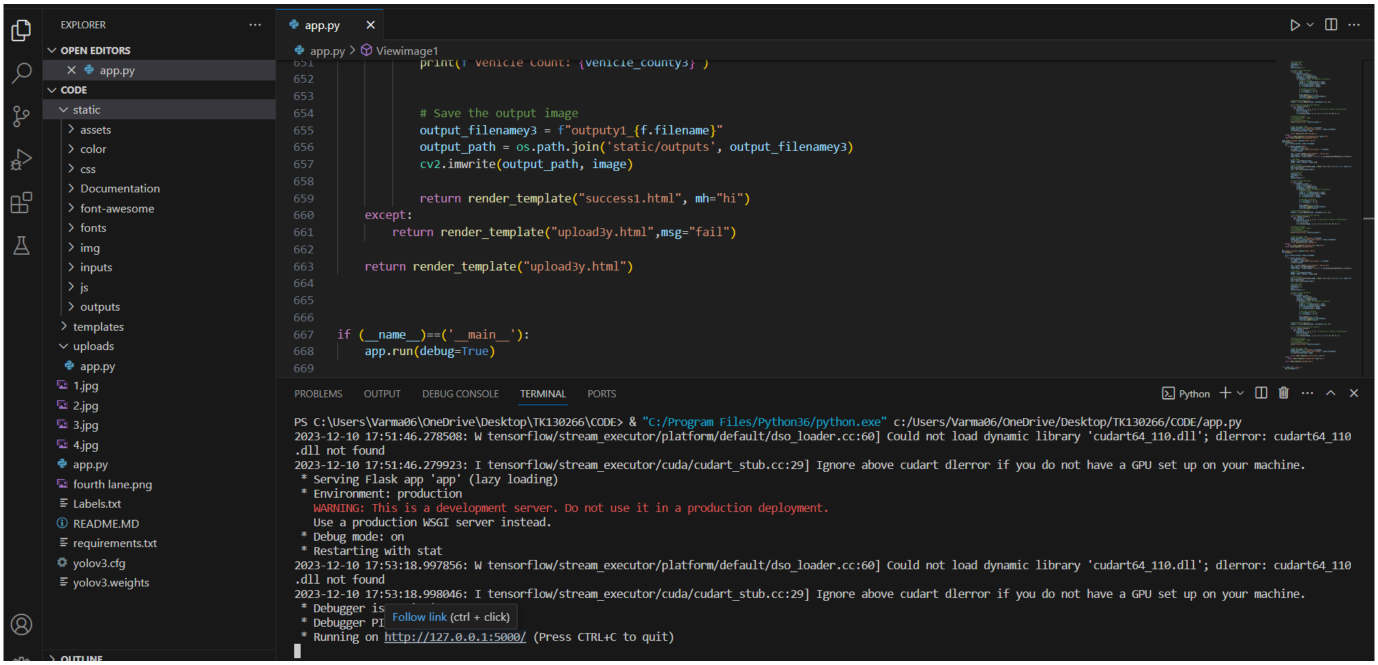

Fig. 2: Execution Result

Fig. 3: Traffic Control

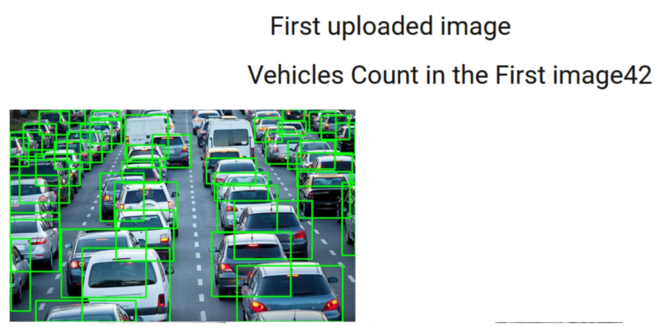

Fig. 4: Image 1 Results

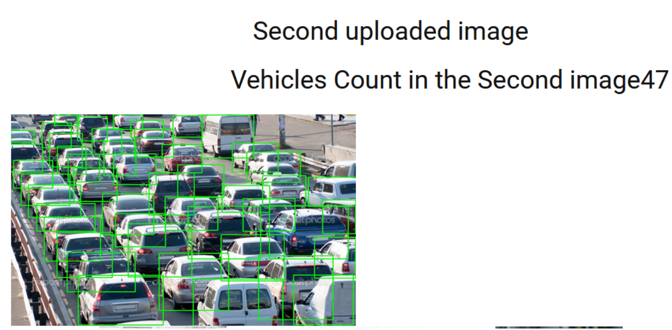

Fig. 5: Image 2 Results

contribute to the training and optimization of our models, ultimately improving the application's capability to predict traffic dynamics effectively.Images that are submitted serve as significant inputs for the algorithms, allowing the system to generate more accurate forecasts and improve overall traffic control methods.

[20] In the field of intelligent traffic control, our project takes a comprehensive approach to enhance traffic clearing at intersections. The software frequently change signal timings with the intent of selecting lanes based on vehicle count. It evaluates real-time traffic conditions by utilizing powerful algorithms such as Convolutional Neural Networks (CNN) and YOLO. Lanes with a higher vehicle count receive higher priority, with longer green signals for faster clearing. This adaptive technique assures effective traffic management by reducing congestion and optimizing total vehicle flow. This technology contributes to a more responsive and efficient urban transportation network by intelligently responding to the dynamic nature of traffic, improving the travel experience for everyone.

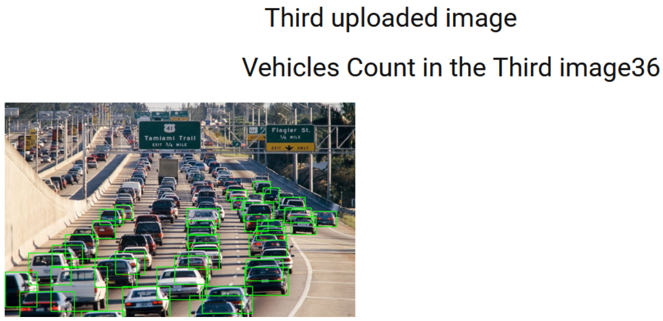

Fig. 6: Image 3 Results

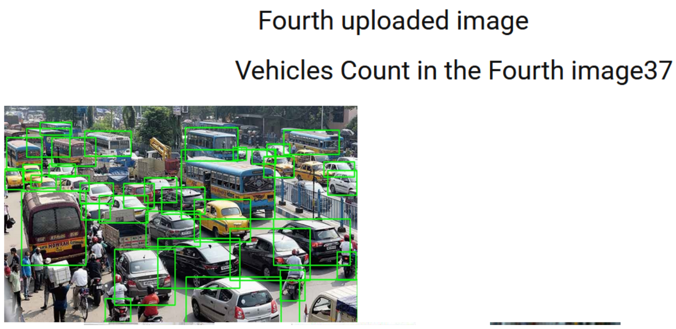

Fig. 7: Image 4 Results

Specifically, when deciding which lane to clear first, the system dynamically analyzes real-time vehicle counts. Notably, in the second lane, we observed a peak vehicle count of 47. Recognizing the importance of efficiency, the system prioritizes the lane with the highest vehicle count, aiming to swiftly clear congestion and optimize traffic flow.

IX. Conclusion And FutureWork

In conclusion, our application has effectively implemented a manual traffic signal control system, developed within a user-friendly environment using Flask and Python programming. The system's functionality involves collecting images from users to intelligently clear signals for lanes with the highest count of vehicles. By leveraging technologies like Flask and Python, we've created an accessible and responsive solution for traffic management, aiming to enhance efficiency and optimize signal control based on real-time vehicle counts.The future of machine learning and deep learning correlates with with our objective of optimizing traffic control in the context of our traffic prediction system. Ongoing developments and an increasing demand for ML knowledge indicates a promising future for improving our system's capabilities and contributing to the larger field of intelligent transportation systems.

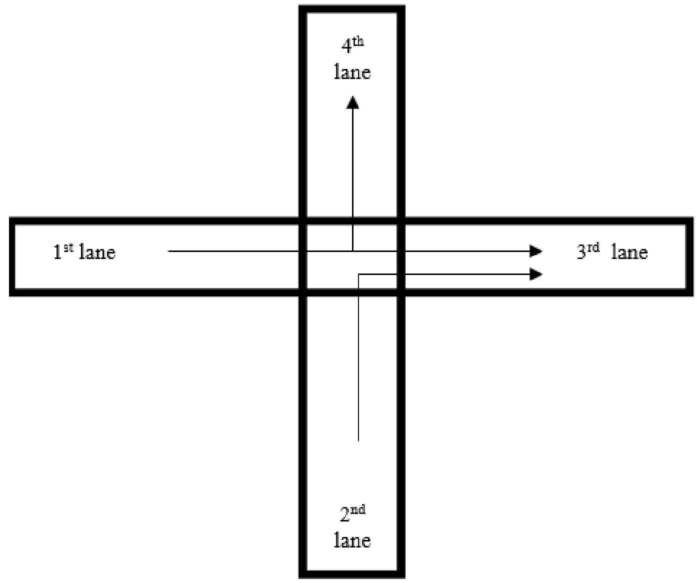

Fig. 8: X-Junction

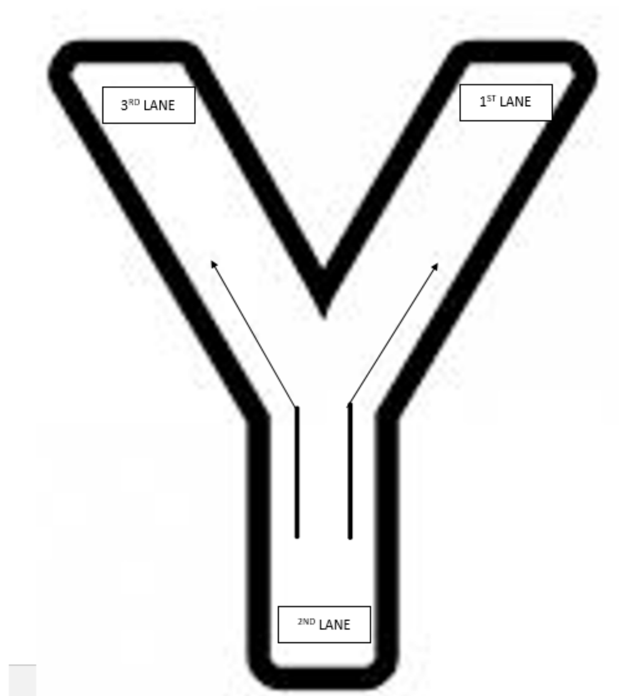

Fig. 9: Y-Junction

References

[1] M. Kumar and R. Bhatt, "An approach towards real-time object detector using open cv," in *2022 11th International Conference on System Modeling Advancement in Research Trends (SMART)*, Moradabad, India, 2022, pp. 981–985.

[2] M. Mahendru and S. K. Dubey, "Real time object detection with audio feedback using yolo vs. yolo$_v3,'' Noida, India, 28-29 January 2021$.

[3] M. S. S. Reddy, P. R. Khatravath, N. K. Surineni, and K. R. Mulinti, "Object detection and action recognition using computer vision," in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 14-16 June 2023, pp. 874–879.

[4] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *2017 International Conference on Advances in Computing, Communications and Informatics*

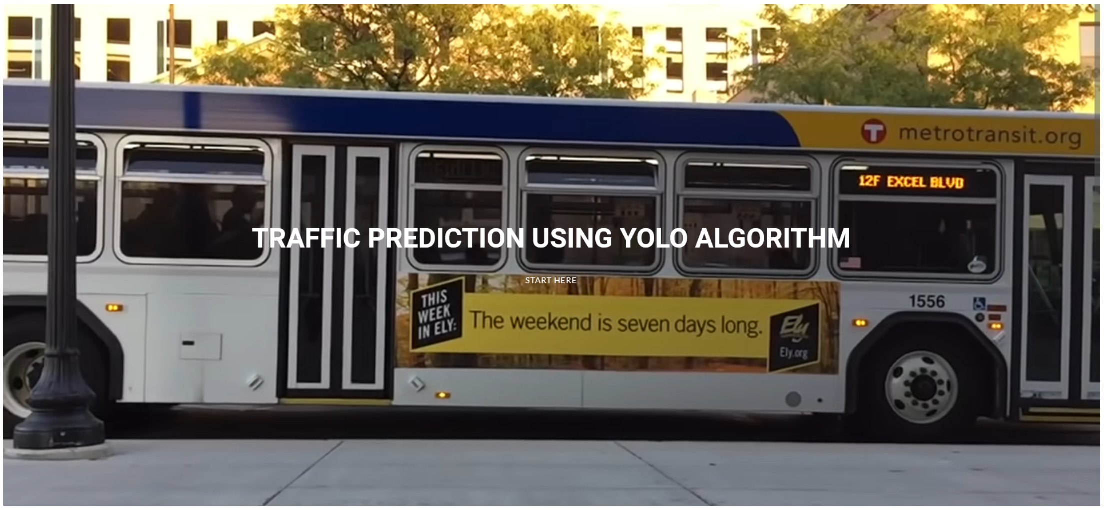

Fig. 10: Home page

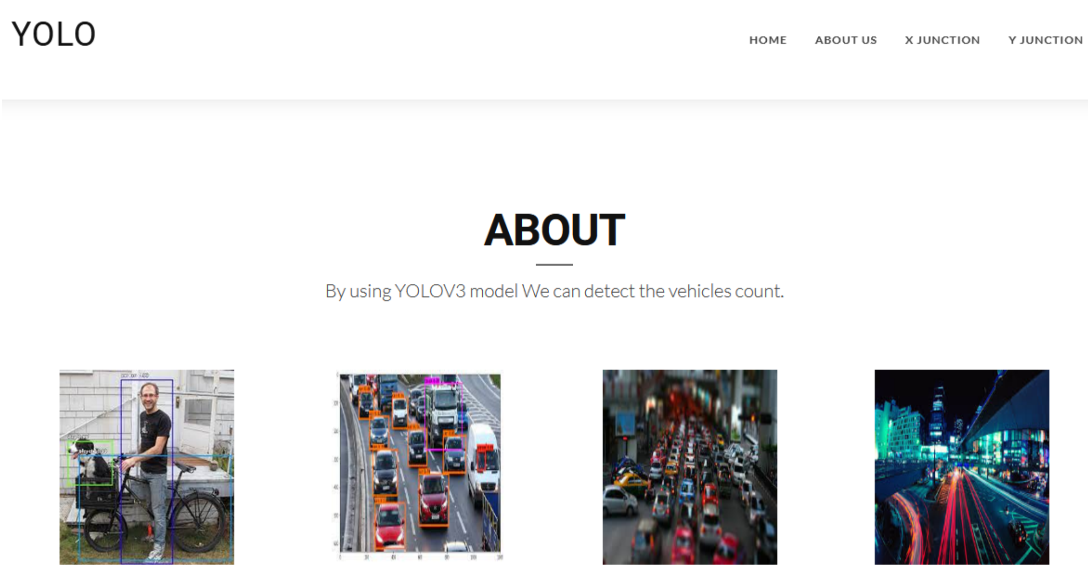

Fig. 11: Start page

(ICACCI), Udupi, India, 13-16 September 2017, pp. 2353–2358.

[5] X. Zou and E. Chung, "Network-wide lane-level traffic flow prediction via clustering and deep learning with limited data," in *2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Nice, France, 14-16 June 2023, pp. 1–6.

[6] Y. Zhong, H. Liu, and M. Lv, "Intelligent algorithm based traffic flow automatic control method at traffic intersection," in *2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT)*, Changzhou, China, 09-11 December 2022, pp. 1–10.

[7] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 364–373, 14 March 2011.

[8] M. Mahrishi and S. Morwal, "Index point detection and semantic indexing of videos -a comparative review," 04 2020.

[9] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, 13 March 2019.

[10] C.-H. Wu, J.-M. Ho, and D. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 06 December 2004.

[11] S. Makridakis, E. Spiliotis, V. Assimakopoulos, A.-A. Semenoglou, G. Mulder, and K. Nikolopoulos, "Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward," *Journal of the Operational Research Society*, vol. 74, pp. 1–20, 09 2022.

[12] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, *Machine Learning Strategies for Time Series Forecasting*, 01 2013, vol. 138.

[13] F. Harrell, K. Lee, D. Matchar, and T. Reichert, "Harrell jr fe, lee kl, matchar db, reichert taregression models for prognostic prediction: advantages, problems, and suggested solutions. cancer treat rep 69: 1071-1077," *Cancer treatment reports*, vol. 69, pp. 1071–77, 11 1985.

[14] M. Colak, C. Colak, H. Kocatürk, S. Sagiroglu, and I. Barutçu, "Predicting coronary artery disease using different artificial neural network models," *Anadolu kardiyoloji dergisi : AKD = the Anatolian journal of cardiology*, vol. 8, pp. 249–54, 08 2008.

[15] K. M and K. P. R, "Comparative analysis of yolov3-320 and yolov3-tiny for the optimised real-time object detection system," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, London, United Kingdom, 27-29 April 2022, pp. 495–500.

[16] S. T. Blue and M. Brindha, "Edge detection based boundary box construction

algorithm for improving the precision of object detection in yolov3," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 06-08 July 2019, pp. 1–5.

[17] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 21-23 August 2017, pp. 1–6.

[18] N. G. Markov, I. V. Zoev, and E. A. Mytsko, "Fpga hardware implementation of the yolo subclass convolutional neural network model in computer vision systems," in *2022 International Siberian Conference on Control and Communications (SIBCON)*, Tomsk, Russian Federation, 17-19 November 2022, pp. 1–4.

[19] H. Wan, "Deep learning:neural network, optimizing method and libraries review," in *2019 International Conference on Robots Intelligent System (ICRIS)*, Haikou, China, 15-16 June 2019, pp. 497–500.

[20] X. Zhang, "Traffic sign detection based on yolo v3," in *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, Shenyang, China, 29-31 January 2023, pp. 1044–1048.