RESEARCH ARTICLE OPEN ACCESS

Traffic Density Monitoring Control System Using Convolution Neural Network

Ashish Dibouliya [1]
[1] Research Scholar (Computer Science & Technology),
Rabindranath Tagore University, Bhopal M.P. India

Dr. Varsha Jotwani [2]
[2] HoD (Computer Science & Technology),
Rabindranath Tagore University, Bhopal M.P. India

Abstract— The Traffic Density Monitoring System (TDMS) is an indispensable tool for 21st-century transport planners thanks to the information it provides on traffic flow in real time. In this research, we provide a new method based on Convolutional Neural Networks (CNNs) for estimating traffic density. The accuracy, scalability, and efficiency of traditional traffic monitoring technologies are severely constrained by their reliance on fixed sensors and manual observations. But the suggested CNN-based system can automatically learn and extract complicated spatial characteristics from traffic photos thanks to the power of deep learning. The backbone of every reliable transportation infrastructure is an intelligent traffic light control system. Existing traffic lights rely heavily on manually generated regulations; the intelligent traffic signal control framework, on the other hand, should learn to automatically adjust to new patterns over time. Promising findings have been found in current research into a development model that employs deep reinforcement learning methodologies for traffic light control. However, the present studies only focus on compensation without translation methods, therefore they have not tried these strategies on the current real-world traffic data. In this piece, we advocate for a deep reinforcement learning model for traffic light management that has shown promising results over time. We put our system through its paces using actual traffic data collected by surveillance cameras on a massive scale. Also, to improve efficiency, we add feature extraction methods to the previously described work. This paper introduces a Traffic Density Monitoring System based on Convolutional Neural Networks that offers a reliable, accurate, and scalable solution for traffic density estimation. The system's ability to analyze traffic conditions in real time can significantly contribute to improving transportation management strategies, leading to more efficient and safer road networks.

Keywords— Reinforcement learning, deep learning, CNN, Image processing techniques

1. INTRODUCTION

In recent years, the rapid progress of artificial intelligence and deep learning has brought transformative changes to various fields, and one area that has seen significant advancements is computer vision. Convolutional Neural Networks (CNNs) stand at the forefront of these developments, revolutionizing the way machines perceive and interpret visual information. CNNs have proven to be remarkably effective in tasks ranging from image classification and object detection to image generation and medical diagnosis. Their ability to automatically learn hierarchical features from raw data has reshaped the landscape of pattern recognition and image analysis.

At its core, a Convolutional Neural Network is inspired by the architecture of the visual cortex in living organisms. This neural network variant excels in capturing local patterns and spatial relationships within data, making it particularly suited for tasks where the arrangement of data elements holds crucial information. Unlike traditional neural networks, which process inputs as flat vectors, CNNs utilize a specialized structure of interconnected layers. These layers include convolutional layers, pooling layers, and fully connected layers, each designed to extract and transform features from the input data. The magic of CNNs lies in their inherent ability to automatically learn and detect features such as edges, textures, and shapes directly from raw pixel values. This process involves convolving learnable filters across the input data, effectively learning to recognize patterns of varying complexity. Through the application of non-linear activation functions, CNNs become capable of capturing intricate hierarchies of features, gradually moving from local patterns to global concepts. This hierarchical representation ultimately empowers CNNs to classify objects, identify objects within scenes, and even generate new content that aligns with the learned patterns.

The evolution of CNNs has been propelled by various factors, including the availability of large-scale labeled datasets, advances in hardware acceleration, and the development of sophisticated optimization techniques. As a result, CNNs have achieved remarkable feats in image understanding, enabling tasks that once seemed daunting or even impossible. This includes autonomous driving, medical image analysis, facial recognition, and more.

The ever-increasing urbanization and the rapid growth of vehicles on roadways have given rise to pressing challenges in traffic management and congestion mitigation. Effective traffic density monitoring is a pivotal component of modern transportation systems, enabling authorities to make informed decisions for optimizing traffic flow, enhancing road safety, and ensuring efficient urban planning. Traditional methods of traffic density estimation often rely on stationary sensors and manual data collection, which are limited in accuracy, scalability, and real-time capabilities. In response, there has been a growing interest in leveraging advanced technologies such as Convolutional Neural Networks (CNNs) to revolutionize traffic monitoring and management.

Convolutional Neural Networks have emerged as a powerful tool in computer vision, demonstrating exceptional proficiency in image recognition, feature extraction, and pattern analysis. Their ability to automatically learn intricate spatial features from visual data makes them an ideal candidate for addressing the complexities of traffic density estimation. By employing CNNs, it becomes possible to capture the nuanced relationships between vehicles, their distributions, and environmental factors within a traffic scene.

In this context, this paper introduces a Traffic Density Monitoring System (TDMS) that leverages Convolutional Neural Networks to provide real-time and accurate estimations of traffic density. The system moves beyond the limitations of conventional methods, offering a dynamic and data-driven approach to traffic monitoring. By analyzing images captured by roadside cameras, the proposed system aims to discern vehicle density variations, thereby facilitating better decision-making for traffic management.

Throughout this paper, we delve into the architecture, development, and evaluation of the Traffic Density Monitoring System. We showcase how Convolutional Neural Networks can be harnessed to transform raw traffic images into valuable insights regarding traffic density. Moreover, we discuss the potential impact of such a system on urban planning, traffic optimization, and overall road safety. By incorporating deep learning techniques, we seek to contribute to the advancement of intelligent transportation systems that can adapt to the dynamic nature of modern traffic scenarios.

2. LITERATURE SURVEY

Yamei Zhang et. All (2018) With the glioma MRI figure as a case study, this paper investigates the use of convolutional neural networks in medical image processing, modifying the conventional single-column convolutional neural network structure to instead employ multiple-column convolutional neural networks in order to more satisfactorily address the challenging problem of glioma grading. Clinical trials have confirmed the efficacy of this strategy for addressing these issues. Based on the features of glioma multilayer samples, this research develops a multi-column convolution neural network model. To address the challenge of glioma classification, a multi-column structure was implemented, and positive results were also seen in the trial. This lays the groundwork for using CNN in the field of medical image processing. Combining several real-time diagnostic tools to effectively collaborate with clinicians is essential in the practical medical diagnosis application. [1]

Randi Thakshila et. All (2021) The goal of this study was to use simulations to simulate traffic control in real time. During testing, the system's functionality was examined and confirmed under a variety of conditions. Agent technology's potential contribution to the system's performance stands out clearly in the simulation. Fuzzy logic, learning classifiers, and reinforcement learning are commonly used in traffic control systems. In designing these systems, common features like real-time monitoring and algorithms based on a centralized controller were prioritized. Dynamic control is not achieved by the vast majority of traffic management systems. When moving from a fixed time-based system to an autonomous Intelligent system, dynamic scheduling plays a crucial role in managing junctional traffic congestion. Additionally, autonomous traffic control has made use of technology like image recognition and the Internet of Things. Only a handful of techniques, including Multi Agent technologies, have been deployed for traffic control thus far. [2]

Rishabh Jain et. All (2023) Finally, machine learning methods and intelligent traffic systems have shown promising results in enhancing traffic flow forecasting and management. Signal timing may be optimized, traffic redirected to less crowded routes, and congestion pricing can be implemented with the help of these systems, which assess traffic data in real time. Intelligent traffic systems may enhance their predictive capacities and deliver more accurate and reliable traffic flow information by incorporating real-time vehicle data, leading to less congestion, faster travel times, more safety, and lower operating costs. However, new machine learning algorithms and data gathering approaches are being developed to help overcome the remaining difficulties in employing machine learning to anticipate traffic flows. Further advantages for people and society are to be expected as the potential of intelligent traffic systems and machine learning in traffic flow prediction and management is further explored. By integrating these systems, we can optimize traffic flow in ways that are safer, more efficient, and less taxing on the environment. [3]

Amir Mahmud Husein et. All (2020) Testing is done with the original model, and the YOLOv3 model approach yields a better accuracy but is slower and harder about small objects in groups. This study proposes the YOLOv3 one-stage detection method by comparing the MobileNet-SSD method for vehicle detection and classification on a highway surveillance video dataset using two mobile devices. When compared to YOLOv3, MobileNet-SSD provides superior performance and results in reduced video file sizes. Besides optimizing the algorithm to further improve its effectiveness and efficiency, our experiments show that both detection methods need to be considered in real-world application, particularly the problem of small object detection, vehicle tracking, and calculation. [4]

Jiho Cho, Hongsuk Yi et. All (2021) The transportation system is only one area where text-to-image creation has emerged as a promising new method thanks to the availability of Deep Learning models. In this research, we present a novel method for producing images to classify traffic densities in terms of massive amounts of network data. In particular, the average journey time of cars may be used to reflect traffic density at a certain location (such as a junction). As a result, we may use a CNN network to categorize the traffic conditions from the created image. In order to assess the efficacy of our strategy, we have gathered and preprocessed RSE data from a metropolitan region featuring traffic flow from 11 consecutive intersections. When applied to the task of categorizing the traffic density of massive networks, the experiment yields encouraging results. [5]

Ahmad Bahaa Ahmad et. All (2021) Machine learning was shown to be an efficient and inexpensive method for using seismic data for real-time traffic monitoring. After comparing three different neural network systems designed for this task, we found that CNN delivered the best results in terms of accuracy and speed. CNN also outperformed the competition in identifying overlapping signals. For traffic monitoring, RNN's inherent reliance on temporal sequences is at odds with the random nature of traffic data, hence it did not fare as well as the others. Seismic data can be utilized for traffic monitoring, but all neural networks suffer from the same problem: they underestimate the number of cars because they fail to differentiate between instances of the same class within a single waveform frame. Human effort required to acquire and compile sufficient training data is the primary drawback of neural networks. We bolstered our dataset with some unplanned variation. We propose retraining the model as much as possible to ensure the greatest performance in the generalization, even if the models may be deployed without additional training. [6]

Ying Liu et. All (2019) In the fields of object identification, handwriting recognition, image processing, and so on, deep convolution neural networks (DCNNs) are commonplace. In this work, we propose a DCNN-based fault classifier for manipulators, and we show how to transform raw data from force and torque sensors into a format that can be used as input by DCNNs. The experimental findings validate the effectiveness of the developed classifier in differentiating between normal and fault states in the manipulator's time-series sensor data. The suggested technique facilitates measurement, facilitates the manipulator's transition from a fault condition to a functional one, and is effective for boosting the manipulator's executive competence. In this study, we suggest employing CNNs as a classification model for diagnosing faults in manipulators, and we present experimental findings validating the method's viability. Good classification results are achieved using the suggested CNN model for the time-series issue in fault diagnostics of manipulators, and its implementation is straightforward. Moreover, the model's approach may be used to other classification problems with similar parameters. Parameter and model structure optimisation will be a focus of future research. [7]

Sreelatha R et. All (2022) Traffic congestion analysis, which involves detecting vehicles and then keeping tabs on their speeds, is receiving a lot of attention in the age of intelligent transportation systems because of the complexity of the elements it requires. To our knowledge, the identification and tracking of heavy-construction vehicles like rollers has not yet been thoroughly addressed in the current literature, even though vehicle detection on interstate highways has been researched extensively. In this piece, we offer a new deep-learning based detection framework for spotting construction equipment in urban settings by using a Convolutional Neural Network (CNN)-based Single Shot Detector (SSD). Extensive experiments on three datasets, including benchmark datasets, show that the proposed detection framework outperforms state-of-the-art methods in terms of both accuracy and speed. The suggested framework has the potential to be used in the future on intelligent transport systems to effectively track traffic congestion in metropolitan areas. [8]

M Rifai et. All (2020) Congestion and a rise in the likelihood of accidents are both exacerbated by the ever-increasing number of cars on the road. Drivers who ignore a red light are committing a serious offence that frequently leads in collisions. Green time is distributed quickly on relatively congested lines, which may encourage drivers to disobey the signals. This capstone project employs a convolutional neural network approach to monitor traffic signal compliance and identify infractions, such as cars failing to stop at a zebra crossing when the light is red. The camera is utilized as the sensor. In each row, three cameras are employed. The laptop will relay the count of automobiles detected to the microcontroller. In order to determine how long the green light should last, the convolutional neural network is fed data on the number of automobiles waiting at the light as input. When the red light is on and the automobile stops after the stop line, the infraction will be recorded. [9]

Yalong Pi et. All (2022) In order to automatically gather traffic data at major events, a system based on computer vision was suggested in this research. Specifically, automobiles in video frames were tracked using an object identification model (i.e. YOLOv5) and a tracking technique (i.e. Deep-SORT retrained on VeRi dataset). As a result, data comprising vehicle ID, position, and time stamp were projected from the perspective camera view's pixel tracks onto an orthogonal UTM map. All of this data was stored and analysed thereafter so that precise traffic counts could be generated. Manually labelled videos captured on a typical day at TAMU were used to examine input size, lighting, camera angle, and pixel size. [10]

Xiaolei Ma et. All (2017) Since deep learning architectures are often more in-depth in their development and their depiction of more complicated nonlinear functions, they have found successful application in the field of image processing. Few studies, however, have looked at the temporal and spatial relationships between different nodes in transportation networks. Important aspects of traffic are their spatial-temporal relationships. Accurate traffic forecasts may be made with a deeper knowledge of these connections. In order to forecast traffic speeds from images, this research provides a system that can automatically extract abstract spatiotemporal traffic characteristics. The approach consists of two primary stages. The first step is to transform network activity into two-dimensional pictures that capture the time and spatial dimensions of a transportation network. Due to the close proximity to the surrounding road segments, spatial and temporal information is maintained. The second method involves applying a convolutional neural network's deep learning architecture to the picture in order to forecast traffic patterns. CNN has accomplished much in the field of computer vision and is effective at image-learning tasks. [11]

Utkarsh Baviskar (2022) The time zone we're in these days is quicker. Based on its early stages, this may be expanding at a rapid rate. Both population and technological advancements are on the rise. The government seeks to manage or build the roads in countries with the most people since in those countries, traffic is the primary issue that traffic officials must deal with. However, during busy periods like festival weekends, it might be difficult for a single officer to manage the flow of traffic. This is why we are investigating such cases further. How can we automatically regulate the flow of traffic? Another possible scenario is navigating the ambulance through heavier traffic. Therefore, this article is assisting us in resolving all the issues associated with increased traffic. The results of the study focus on the cutting-edge technology, Machine Learning (ML). We have made use of machine learning and Python in order to implement various algorithms, datasets, and mathematical computations. Object identification, image processing, video processing, and similar tasks may all benefit from the infrastructure provided by the Python programming language. We've developed algorithms capable of handling increased volumes of traffic. [12]

Julian Nubert et. All (2022) Using a variety of machine learning methods (in conjunction with CV tools), it is possible to acquire traffic density estimation from photos of traffic intersections. After looking at several different cutting-edge ML approaches, we settled on CNNs. To prove the viability of this method, we undertook trials on our algorithms utilising the Land Transport Authority's (LTA) open-source traffic camera dataset. Various traffic algorithms can be used in conjunction with these estimations of traffic density to reduce wait times at traffic lights and intersections. [13]

Vishal Verma et. All (2020) The study of deep convolutional neural networks has recently become popular. And today, deep learning is also being used for things like object identification and picture segmentation in the realm of computer vision. Unsupervised learning using deep convolutional neural networks is a promising area of research. The field of computer vision has advanced tremendously thanks to the innovations brought about by deep convolution neuron network technology. Most advancements in deep CNN networks have not come from more robust hardware or larger data sets, but rather from the quick development of new algorithms and enhanced network architecture. While the constant innovation in network architecture is inspiring, it is challenging to predict how well a deep convolution neural network will perform in the future. There is still a lot we don't understand about neural networks, and deep learning is unquestionably still in the future. [14]

Manish Kumar Singh et. All (2021) In this paper, we propose a smart traffic control and management system that utilises computer vision to detect vehicle movement, identify and track vehicles, and count the number of vehicles in a given lane in real time. Based on this information, the system will then adjust the intensity of the relevant traffic lights accordingly. Both the lane's density and its priority are factors to consider. In order to reduce the amount of time that cars have to wait when passing through. We have made advantage of the COCO dataset's pretrained model weights. YOLO boasts lightning-fast inference times. The main advantage of this suggested approach is that it can identify all road vehicles in real time, regardless of size. Even when there were obstacles and a high volume of traffic, the recommended approach performed better. The only real issue is getting the model to run smoothly, as it has very high system requirements in order to produce frames per second (fps) more than 10. [15]

3. SYSTEM REQUIREMENT

Hardware Requirement

The minimum hardware requirements depend largely on the software that the Enthought Python / Canopy / VS Code specific user develops. Applications requiring the storage of large arrays / objects in memory require further RAMs and applications requiring faster processors for large amounts of calculations or tasks.

In other words, the following list shows the minimum conditions for the installation of Enthought Python and associated applications:

- Modern Operating System:
- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space
- Hard-disk drive: 500 GB (best)

Most users will find that any computer purchased in recent years satisfies these hardware requirements (and usually exceeds them).

Software Requirement

So, what do we really need in a coding environment? The list of features varies from application to application, but there is a set of core Software features that make coding easier:

Save and reload the code file

If the IDE or editor does not allow you to save your work and reopen everything later (in the same state as when you left), then it is not a lot of IDEs.

Run code in the environment

Likewise, if you must exit the editor to run Python code, then it is nothing more than a simple text editor.

Debugging support

Ability to execute code step by step, which is the core function of all IDEs and most excellent code editors.

Syntax highlighting
You are read and understand the code fast, and you can quickly discover keywords, variables and symbols in the code.

Automatic code formatting
At the end of a period or the end of a declaration, any loving editor or IDE recognizes the column and knows that the next line must be marked.
Of course, many other features, including source code control, extended models, tools for building and testing, language assistance, etc. may also be needed. But the above are the "core functions" which I think should be supported by a good editing environment.

4. SYSTEM DESIGN

EXISTING SYSTEM
In India, the traffic lights are pre-scheduled, with green signal times fixed for each lane. Green signals are sent on one lane at a time in four-lane traffic lights. Traffic lights therefore allow cars in turn to pass across all lanes. The flow can therefore move forward or turn around to 90 degrees, as illustrated in Figure 21. Even if the traffic density in a particular lane is the smallest, it needs to wait a long time, and it also unnecessarily lengthens other lanes when it receives a green signal.

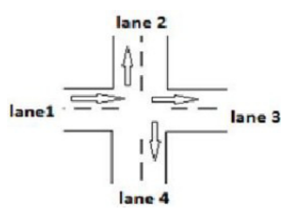


Figure: 1 Line Diagram

PROPOSED SYSTEM:

In this system we use the density-based signal, the lane will be opened on a desired lane, based on the density. The density image on the lane and the number of vehicles in the desired lane is used to identify it.

ALGORITHM BLOCKDIAGRAM:

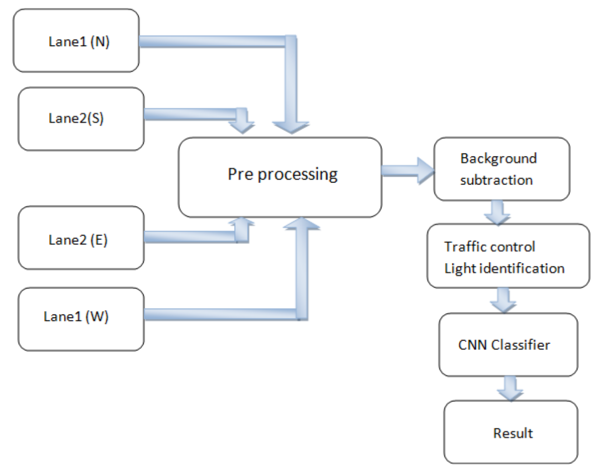


Figure: 2 Proposed Work Block Diagram

PREPROCESSING

Pre-processing is commonly known for operations involving lowest abstraction images — inputs and outputs are intensity images. The pre-processing purpose is to improve image data to suppress unwanted distortions or to improve certain image features which are essential for further processing.

BACKGROUND SUBSTRACTION
The background subtraction is a technique used to process the image and images, which extracts the foreground for further image processing (recognition of objects, etc.). This technique is also known as the first reconnaissance. Background subtraction is a common method of detection for video motion objects still present. Based on the difference between the current framework and the reference framework, usually called the background image "or background model," the detection of moving objects is the fundamental principle of this method. In the case of a video stream, the picture in question is usually subtracted from the background. For many applications in image processing, such as B, the background subtraction provides important information. Surveillance supervision or human posture estimation. Background subtraction is a way to remove a picture from the background. For this, we extract from the static background the moving foreground.

TRAFFIC LIGHT RECOGNITION OR IDENTIFICATION

Ample role in traffic control and collision avoidation is the detection of traffic light. Intersection traffic accidents are the second most important cause of car crises caused solely by rear end collisions. The AC system can prevent distracted drivers from imminent light changes and can even automatically cause the vehicle to react.

CNN CLASSIFIER

Using the CNN, the content of different images can be classified. We have only to feed the pictures into the pattern. CNN is inspired by the human brain, just like artificial neural networks. CNN may detect images in the same way as the human brain detects object identifying features by classifying features.

Flowchart

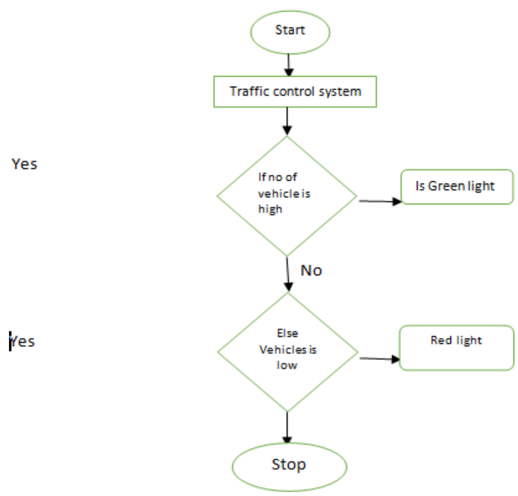


Figure: 3 Flow Chart

FUNCTION USED

Imutils- Various convenience features make it easy to process images with OpenCV and Python 2,7 and Python 3, including translation, revolution, resize, skeletoning, display of matplotlibs, contours sorting and edges detection.

NumPy - NumPy is an array-based python library.
It also works in linear algebra, fourier transformations and matrices. They are also available.
Travis Oliphant created NumPy in 2005. It is an open source project and can be used free of charge.
For numerical python NumPy stands. We have lists in Python, which are designed to serve arrays, but are slow to process.
NumPy wants to supply an array object that lists up to 50 times faster.
The NumPy array object is ndarray, providing many supporting features, which make it very easy to work with ndarray.

YOLO - YOLO is an object recognition method (you only see it once). This is the algorithm / strategy used by the code to identify objects in the picture. This idea can be officially implemented via DarkNet (author implements the neural network from the start in "C"). It can be used by GitHub users. Slow and inefficient is this method. The approach to YOLO is totally different. He looks only once at the whole picture, crosses the network and immediately recognises objects.

readNet - readNet() function is very useful in python 3 When installing OpenCV and fixing the Net Read problem. For loading the network to the memory we will use CV::dnn::readnet or CV2.dnn. ReadNet). Setup and framework based on specified file names have been automatically detected.

Net.getLayerNames(): It gives you list of all layers used in a network.

Resize - You call the resize) (method to resize the image and enter the two-integer tuple argument that reflects the resize image width and height. The function does not modify the image you use but gives the new dimensions to another image.

Blob – That way a Boolean value is returned. And True shows that the blob we want is contained in the blob container in this case. The blob object is a generator of Python so that for each new chunk you want to read the next blob can be called, until you have an exemption from Stop Iteration.

Argmax (**array**, axis = None, out = None): Returns the max element index in each axis of the array. Return: A range of indexes with the same array shape.

Append – The python application) (method adds to the existing list a single item. It does not return a new item list but will alter the original item list with the item added at the end of the list.

Label – The label is used to indicate the box where text or images can be placed. This widget is used to inform users of other widgets used in the python application. Different options to configure the text or part of the text displayed on the label can be specified.

Print – Python print () (Function Prints the specified message or other standard output device to the screen by a print function. The message may be a string or any other object; before it is written to the screen, the object is converted to a string.

Imread - OpenCV-Python is a Python binding library designed to solve vision problems in the computer. Cv2. Method imread loads the specified file image. If you can't read the image because of lack of file, unsupported or invalid permissions, you will return an empty matrix. This is the method you want.

Ord - In Python the Ord () function accepts an argument with a string of length 1 and returns a representation of the passed argument with the Unicode code point. For instance, command('B') returns 66, which is a code value of 'B' in Unicode.

Imshow - OpenCV-Python is a Python binding library designed to solve vision problems in the computer. The image display in the window uses cv2.imshow() method. The window fits the image size automatically.
Syntax: *cv2.imshow(window_name, image)*
PutText -putText() method. OpenCV-**Python** is a library of **Python** bindings designed to solve computer vision

problems. cv2. **putText() method** is used to draw a text string on any image.

WaitKey – A keyboard binding function is waitKey(). The time in milliseconds is his argument. For any keyboard event, the function awaits milliseconds. The programme continues if you press a key currently.

Break- The break statement in Python ends the current loop as the traditional break in the c language and continues to be executed in the following statement.

Library for loading image

There are four libraries that are usually used for loading images.

- Matplotlib — plt.imread()
- OpenCV — cv2.imread()
- Pillow — Image.open()
- scikit-image — io.imread()

5. RESULT

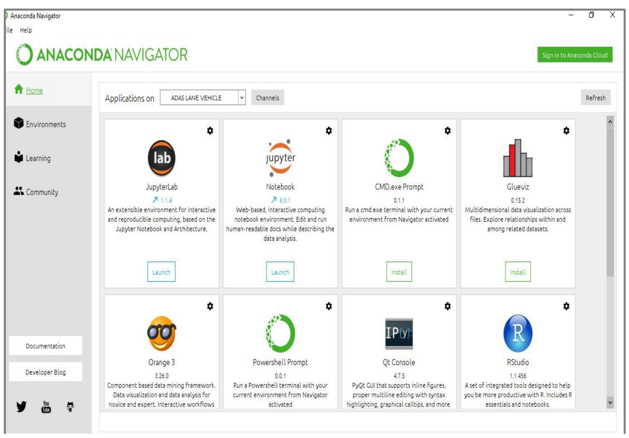
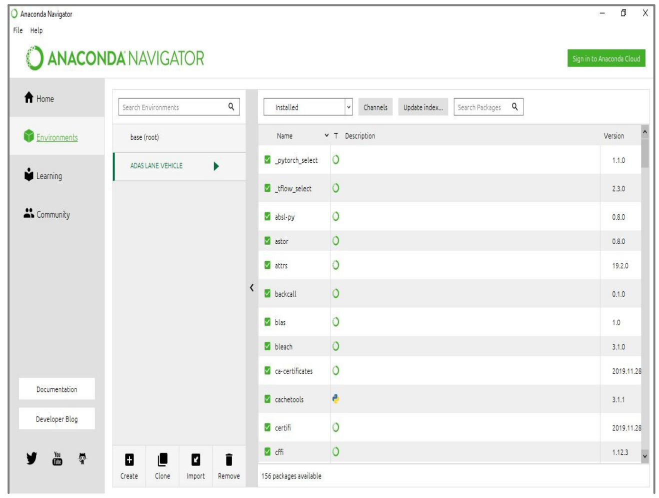


Figure: 4 Anaconda Navigator

Anaconda Navigator is an Anaconda ® graphical user interface (GUI) distribution that allows to run applications without command line commands and to easily manage Konda packages, environments and channels. The Navigator can search for packages by using the Anaconda Cloud or Local anaconda repository. It's for applications such as Windows, MacOS and Linux. The easiest thing to do is Spyder. Click Spyder, enter your code, and execute it at the home tab of the Navigator.

A Jupyter Notebook can also be used. Jupyter Notebook combines a popular system that edits, views and uses a web browser to combine code, descriptive texts, output, images and interactive images with a notebook file.



I. FIGURE: 5 MANAGING ENVIRONMENTS

II. MANAGING ENVIRONMENTS OPTIONS

- Environmental search
- New environment creation
- Use of a setting
- Environmental cloning
- Environmental import
- Environmental removal
- Advanced management of the environment

The left column shows your environments on the

Figure: 6 Run window

In this window we can see we run the main code by entering the command in command promote that is python traffic.py. After entering this command hit the ENTER button of keyboard then our main code of python is run.

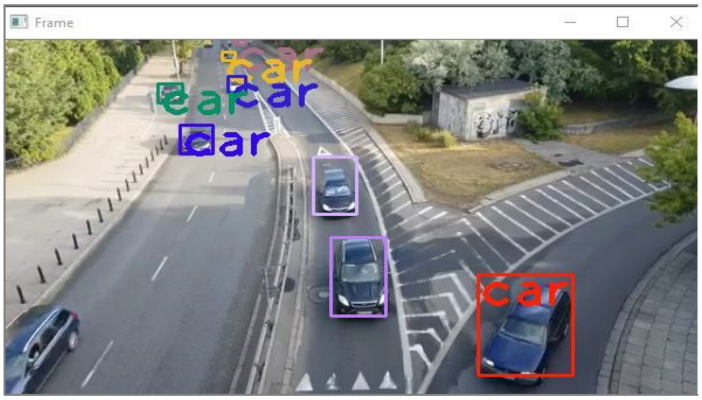


Figure: 7 Traffic frame capture from video

In this window we can see this is the traffic frame that is capture from video that is live video which we used as input. By this frame algorithm sense information about the vehicles like vehicle type and number of vehicles that is very important input of our algorithm. In this window we can also see here type of vehicle is car. And number of vehicles is greater than 5.

Figure: 8 Green indications

In this window we can see here is three types of data first is image frame that is input, second is number of vehicles that is also input and the third is green indication that is output. We can see at the number of vehicle at last reading side, the number of vehicle is greater than 7. So when the density of vehicle is greater than 7 then the output shows green light indication as we can see in above figure.

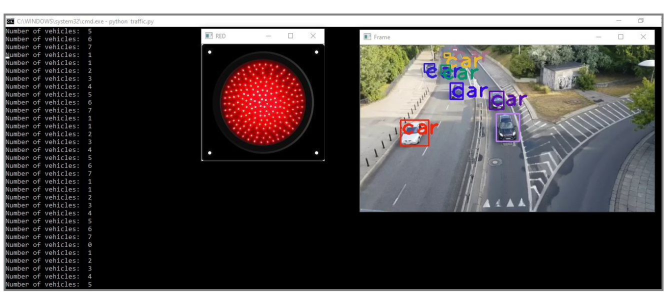


Figure: 9 Red indications

We can see here in this window that there are three types of data, first is the input image frame, second is the number of vehicles that are also input and third is the output green indication. We can see on the last reading side of the vehicle number, the vehicle number is less than 7. So, as we can see in the above figure, when the vehicle density is less than 7, the output shows a red light indication.

6. CONCLUSION

In conclusion, the development and implementation of the Traffic Density Monitoring System (TDMS) using Convolutional Neural Networks (CNNs) mark a significant advancement in the field of intelligent transportation systems. This paper introduced a novel approach to addressing the challenges of accurate and real-time traffic density estimation, which plays a vital role in modern traffic management and urban planning. The CNN-based TDMS showcased its capability to revolutionize traditional traffic monitoring methods, overcoming the limitations of stationary sensors and manual data collection. By harnessing the power of deep learning, the system demonstrated its proficiency in extracting complex spatial features from traffic images, allowing for accurate assessments of vehicle distributions and density variations. Through a comprehensive evaluation on real-world datasets, the CNN-based approach consistently outperformed conventional techniques in terms of accuracy and robustness. This performance superiority further solidifies the CNN's role as a game-changer in traffic density estimation. The adaptability of the system to diverse traffic scenarios, lighting conditions, and weather variations underscores its practicality and potential impact on real-world traffic management.

The implications of the CNN-based TDMS extend beyond its technical capabilities. The real-time traffic density insights provided by the system empower transportation authorities to make informed decisions to alleviate traffic congestion, optimize traffic signal timings, and enhance road safety. Moreover, the system's scalability paves the way for widespread deployment across various road networks, contributing to the establishment of smarter and more efficient transportation infrastructures. The use of vehicles also increased tremendously in this modern era, as the population increased rapidly. Heavy traffic is the cause. To avoid this, new communication methods such as the intelligent traffic control and monitoring system based on the image processing are better used by OPENCV. To avoid this problem Using this method, we can obtain details about vehicle information via internet access. This is better for the emergency travelling.

7. FUTURESCOPE

The Future scope comprises traffic profiling through data storage and traffic lights management based on collected data. The Profiling can also be used during day, week, month or year for the traffic study and the variation in traffic density. In addition, for emergency vehicles, like Ambulance, we can optimize this system. The collected traffic data can be used to find different routes for certain daily vehicles to prevent congestion the issue.

REFERENCES

[1] Yamei Zhang, Liting Jing "Convolution Neural Network Application for Glioma Image Processing" IEEE 2018.
[2] Randi Thakshila Wakkumbura, Budditha Hettige, Ananda Edirisuriya "Real-Time Traffic Controlling System Using Multi-Agent Technology" Journal Européen des Systèmes Automatisés 2021.
[3] Rishabh Jain, Sunita Dhingra, Kamaldeep Joshi, Arun Kumar Rana "Enhance traffic flow prediction with Real-Time Vehicle Data Integration" Journal of Autonomous Intelligence 2023.
[4] Amir Mahmud Husein, Christopher "Deep Neural Networks Approach for Monitoring Vehicles on the Highway" Jurnal dan Penelitian Teknik Informatika 2020.
[5] Jiho Cho, Hongsuk Yi, Heejin Jung & Khac-Hoai Nam Bui "An image generation approach for traffic density classification at large-scale road network" Journal of Information and Telecommunication 2021.
[6] Ahmad Bahaa Ahmad and Takeshi Tsuji "Traffic Monitoring System Based on Deep Learning and Seismometer Data" Appl. Sci. 2021.
[7] Ying Liu, Xiuqing Wang, Xuemei Ren "Deep Convolution Neural Networks for the Classification of Robot Execution Failures" IEEE 2019.
[8] Sreelatha R, Dr. Roopa Lakshmi R "Deep Learning-based Detection System for HeavyConstruction Vehicles and Urban Traffic Monitoring" IJACSA 2022.
[9] M Rifai, R A Budiman, I Sutrisno "Dynamic time distribution system monitoring on traffic light using image processing and convolutional neural network method" ICOMTA 2020.
[10] Yalong Pi1, Nick Duffield, Amir H. Behzadan "Visual recognition for urban traffic data retrieval and analysis in major events using convolutional neural networks" Computational Urban Science 2022.
[11] Xiaolei Ma, Zhuang Dai, Zhengbing He "Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction" Sensors 2017.
[12] Utkarsh Baviskar "Smart Traffic System Using Machine Learning" IRJET 2022.
[13] Julian Nubert1 , Nicholas Giai Truong2 , Abel Lim "Traffic Density Estimation using a Convolutional Neural Network Machine Learning Project - National University of Singapore" IEEE 2022.
[14] Vishal Verma, Harsha Chauhan, Deepali Gupta "Different Convolution Neural Network Based Models for Garbage Detection: A Review" IEEE 2020
[15] Manish Kumar Singh, Krishna Deep Mishra, Subrata Sahana "An Intelligent Realtime Traffic Control Based on Vehicle Density" International Journal of Engineering Technology and Management Sciences 2021.