An Improved Traffic Light Colour Detection and Recognition System for Autonomous Vehicles

Temilade Temitope FASINA LCU/PG/002661

**Being an MSc Thesis Submitted to the Department of Computer Science,
Faculty of Natural and Applied Sciences, Lead City University, Ibadan, Oyo State, Nigeria**

In Partial Fulfilment of the Requirements for the Award of Master of Science (MSc) Degree in Computer Science

2023

Certification

This is to certify that Temilade Temitope FASINA with matriculation number LCU/PG/001980 carried out this research work titled "An Improved Traffic Light Colour Detection and Recognition System for Autonomous Vehicles" in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan, Oyo State, for the award of Master of Science (MSc) degree in Computer Science and that this has not been previously submitted.

…………………………..	**…………………………..**
Dr Wilson Sakpere	**Date**
(Supervisor)	

…………………………..	**…………………………..**
Dr Wilson Sakpere	**Date**
(Head of Department)	

Dedication

This research work is dedicated to Almighty God for his mercies and abundant grace upon my life and family.

Acknowledgement

I would like to express my deepest gratitude to my thesis supervisor and Head of Department, Dr Wilson Sakpere, for his sincere critique, untiring guidance and motivation in making this research work a huge success. I wish to especially thank him for being patient and helpful throughout the system's development process.

My sincere thanks go to members of staff of the Computer Science Department, Lead City University Ibadan, whose individual contributions influenced my final submissions in this research work.

I want to register my special thanks and gratitude to my best friend and husband, Mr Fasina for his unending prayers, support and understanding during this thesis work. To my children and my mother, thank you so much for your support and prayers.

Finally, my thanks go to my friends who have given me full support, without which I would not have completed this thesis work.

Though the above-mentioned institutions and persons have assisted in the process of this research work, I alone stand responsible for the errors, if any, found in the work.

Abstract

This study introduces significant advancements in traffic light detection and recognition using an improved YOLOv4 algorithm. Two key optimization techniques, shallow feature enhancement and bounding box uncertainty prediction, were incorporated to address the limitations of the original YOLOv4 algorithm. The results demonstrate substantial improvements in accuracy for traffic light detection and recognition. In the experiments conducted with the LISA traffic light dataset, the AUC (Area Under the Curve) increased to 97.03% and 95.31% for the two datasets of LISA and LaRa, respectively, in traffic light detection. Additionally, the map (mean Average Precision) improved to 81.34% and 78.88% for recognition trials. Despite a slight increase in detection time, the system remained capable of real-time traffic light detection. The use of bounding box uncertainty prediction further enhanced the YOLOv4 algorithm, resulting in AUC values of 96.84% and 94.73%, as well as mAP values of 79.93% and 78.23% for the LISA and LaRa datasets in traffic light detection. Importantly, this enhancement reduced detection times to 27.59 and 33.45 milliseconds, respectively. To further improve traffic light detection and recognition systems, it is recommended that the collection of diverse and extensive datasets, accurate annotation of data, data augmentation, semantic segmentation, real-time object tracking, the utilization of deep learning models, transfer learning, proper calibration, multimodal sensor fusion, redundancy, real-time processing, machine learning anomaly detection, continuous testing, and regulatory compliance are done.

Keywords: Machine Learning, Traffic Light Recognition, Deep Learning, Autonomous Vehicle

Word Count: 225

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation	**Meaning**
AI	Artificial Intelligence
AV	Autonomous Vehicles
BOVW	Bag of Visual Words
CATboost	Category boosting
DNN	Deep Neural Networks
FFT	Fast Fourier Transform
EFB	Exclusive Feature Bundling

ELM	Extreme Learning Machines
ERP	Effective Radiated Power
HOG	Histogram of Oriented Gradients
HIS	Hue, Saturation, Intensity
GOSS	Gradient-Based Onside Sampling
ITS	Intelligent Transportation Systems
MSERs	Maximally Stable Extremal Regions
ROI	Region of Interest
STD	Scene Script Detection
SVM	Support Vector Machines
SWT	Stroke Width Transform
TTD	Traffic Script Detection
WaDe	Wave-based Detector
YOLOv4	You Only Look Once Version 4

Chapter One Introduction

1.1 Background to the study

A lot of improvements in engineering and ICT have resulted in the phase of Intelligent municipalities which has in turn resulted in the phase of the automatic cars. Autonomous cars are automobiles that can circumnavigate from one position to another with miniature or no humanoid interference. There are diverse grades of autonomy designated for automated cars[1]. This diverse grade of autonomy is designated by the grade of humanoid involvement required for automobile steering. Appraisal of the present standing is an important prerequisite for all automated automobiles. This in general connotes the automobile should be capable of detecting all things along its steering or immobile parts. This aids the automobiles in the precarious responsibilities of item evading. Detecting things in its surroundings likewise assists the automobile to spot transportation symbols which is very serious for harmless and effective movement.

Self-driving automobiles depend on diverse kinds of devices for circumnavigation[2]. These devices aid these automobiles in achieving errands like localization of the present station, article recognition, and hindrance evasion. GPS devices, radars, cameras, and laser scanners are different sensors employed in modern autonomous vehicles[3,4]. Cameras and high-performance graphics processing units (GPUs) have become more inexpensive and this has led to augmented implementation of picture processing and computer vision devices. In general, self-driving automobiles utilize cameras fixed to them to amass visual information about their surroundings[5]. The images taken by these cameras are handled through visual devices which aids the automobiles in decoding information for self-driven vehicle decision-making.

Initially in the 20th epoch, the insurrection in the automobile industry increased drivers, so much so that the existing road systems were incapable of regulating vehicular progression[7]. This provoked the building of systems to direct, standardize, and caution road users; to encourage nontoxic and effective transportation. Infrastructure devices, such as signal lights, signs, and tarmac patterns employed to converse or direct car users while plying the road are called vehicular regulated systems. They are particularly critical in intricate surroundings like connections or at-grade intersections, where a lot of data must be interconnected to car users. Equilibrium must be attained between notifying car users adequately and distracting drivers without overloading them with data. The amount of data, and the quantity of time accessible to capture it, affects a car user's capability to get data from vehicular-influenced systems.

Consequently, blunders from pressure and oversight may be triggered by great speed and overpowering volumes of data[8]. For vehicular-influenced systems to work correctly, all drivers must sternly obey; or else, risky conditions might happen. Vehicular-influenced systems are seldom intentionally snubbed by motorists. The inability to follow these transportation control systems is occasionally accidental and may be triggered by such elements as speeding through a juncture to make it in time, violent driving by following the vehicle in front, interruption delusions of or defective Movement control systems[9,10].

The difficulty of driving has been streamlined as most of the driving sub-tasks are automatic. Driving can be easy, which outcomes in distracted motorists to whom serious occasions may be perceived with additional deferral. Owing to exhaustion and psychological burden, traumatic driving where the motorist is exceedingly focused and concentrating might yield a delay in response time[11]. Even though extensive independent driving is years from now, driver aid devices can monitor the surroundings and caution or act in dangerous circumstances. Motorist maintenance aid devices must reward motorist limitations to best sustain the motorist. Failing to sense and spot transport control systems is an instance of motorists' inadequacy. Researchers review that some transport control systems are more noticeable to some motorists than others; for instance, speed perimeter symbols are virtually always perceived, but pedestrian crossing symbols are recurrently ignored[12,13].

The metropolitan setting offers a diversity of problems for all apparatuses of driver-aid devices, particularly for those that depend on computer vision. One important concern is identifying Traffic lights at junctures. 683 people died and 133,000 were hurt in accidents that involved red light running in the USA, 2022[14]. Idealistically, transport lights should be capable of connecting visually and employing infrastructure to automobiles (I2V) through radio interaction. I2V would take enormous organizational investments to introduce it on a comprehensive level, which is not occurring any moment from now[15,16]. Since junctures are some of the most intricate problems in driving conditions, making graphics recognition of transportation lights is an essential part of driver-aid devices. One such condition where driver aid devices can aid motorists is the yellow light condition, in which the motorist must select whether to stop or carry on when arriving at a juncture with a yellow transportation light.

In the realm of human reaction time and decision-making, the interval between 2.5 to 5.5 seconds is indeed crucial[17]. This timeframe is often associated with the critical period before individuals must make a decision or take action. The concept of "juncture17" appears to be a point of decision or a critical event in a driving scenario.

The central part of this interlude, which falls around the 4-second mark, is particularly significant. It represents the point at which motorists' response times tend to be the longest and where decision-making can be most challenging. This is consistent with existing research on the human factors involved in driving, where reaction times are influenced by various factors, including the complexity of the situation, visibility, and cognitive load.

Although transportation lights are schemed to be effortlessly identifiable, ecological elements such as poor or sub-optimal engagement of transportation controls can make effective recognition or detection challenging. The concerns may comprise halo instabilities and colour nature variations, for example, as a consequence of environmental circumstances or controls from other light sources, obstruction and incomplete obstruction owing to other matters or offangle opinions, predominantly connected with sustained Traffic lights, partial fashioned because of flopping or coloured, or dirty lights.

Wrong positive situations are triggered by billboards, reflections, brake lights, and pedestrian crossing lights, concerns regarding the harmonization between the camera's shutter speed and the task sequence of traffic light LED[17]. The discrepancies in traffic light lamps can be credited to dust, flaws, or a moderately sluggish task sequence of the LEDs. The task sequence is great enough so great that the flashing lights are not conspicuous to the humanoid view. Cameras using swift shutter swiftness may trigger complications leading to some frames not comprising a lit traffic light lamp. Another element that may influence the presence of lights is permeation. Owing to the shift between day and night, the camera's situations must be in sync to let the optimum quantity of light in and avert under- or over-permeation. A study presents an adaptive camera device scheme, that can change the shutter and gain backgrounds centred on the brilliance of the pixels in the upper part of the picture[18].

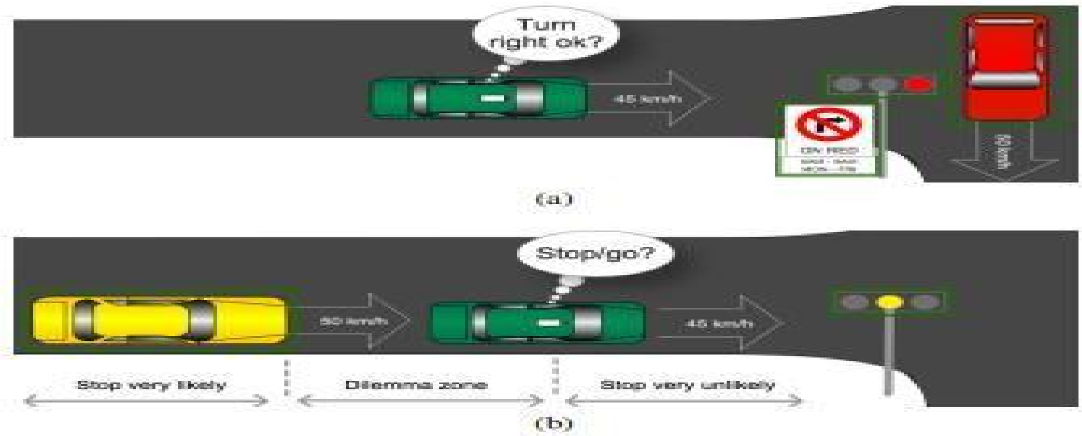


Figure 1.1: Motorist Aid device at juncture situations (a) Turn right on red support (b) Dilemma zone support[18]

1.2 Statement of the Problem

The advent of autonomous vehicles (AVs) marks a significant leap in the evolution of transportation, promising increased safety, efficiency, and accessibility. Central to the operation of AVs is their ability to interpret and react to their surroundings, a task that fundamentally includes the accurate detection and recognition of traffic lights. However, the current technology underpinning traffic light colour detection in AVs is fraught with challenges, posing risks to the safety and reliability of these vehicles. The primary issue lies in the inconsistency and inaccuracy in detecting and recognizing traffic light colours under varied environmental conditions. Factors such as lighting variations (e.g., bright sunlight, overcast skies, nighttime conditions), weather phenomena (e.g., rain, fog, snow), and differing distances and angles of approach can significantly impair the ability of AVs to accurately identify traffic light colours[10]. This challenge is compounded in complex urban environments where multiple traffic lights and varying backgrounds can lead to misinterpretation of signals. Furthermore, the current detection algorithms predominantly rely on simplistic models that do not adequately account for these variabilities, leading to a higher probability of erroneous interpretations[10]. Such inaccuracies can result in non-compliance with traffic rules, posing a threat to the safety of passengers in the AVs, as well as other road users, including drivers of non-autonomous vehicles, cyclists, and pedestrians[11]. However, how these issues can be adequately tackled and resolved remains an open issue in research[13].

1.3 Aim and Objectives of the Study

This research aims to enhance the accuracy of traffic light colour detection and recognition for autonomous vehicles. The specific objectives are to:

i. Develop a model that integrates features from both shallow and deep network layers, using the YOLOv4 algorithm as a base, to improve the accuracy of traffic light colour detection.

ii. Enhance the image processing methods to better handle variations in light conditions, distances and angles, ensuring consistent recognition of traffic light colours.

iii. Incorporate bounding box uncertainty prediction into the YOLOv4 algorithm to improve the precision of traffic light colour detection.

iv. Test and evaluate the improved system for its effectiveness and reliability under different conditions.

1.4 Significance of the Study

This research contributes significantly to the field of autonomous vehicle technology. By addressing the crucial aspect of traffic light detection and recognition, the study helps in overcoming one of the major hurdles in the development of fully autonomous driving systems. The findings and advancements from this research could lead to a deeper understanding of how autonomous vehicles interact with complex urban environments, thereby enhancing the overall safety and efficiency of these systems.

Improving traffic light detection in autonomous vehicles has direct implications for road safety. Enhanced accuracy in recognizing traffic signals reduces the risk of accidents caused by signal misinterpretation, thus ensuring safer roads for both autonomous and non-autonomous vehicles, as well as pedestrians. This aspect of the study aligns with broader societal goals of reducing traffic-related fatalities and injuries.

This study represents a quintessential example of interdisciplinary research, merging concepts from computer science, engineering, transportation studies, and cognitive sciences. The outcomes of this research can stimulate further interdisciplinary collaborations, leading to innovative solutions in various fields. As autonomous vehicles become more prevalent, there is a growing need for comprehensive policy and regulatory frameworks governing their operation. This study provides empirical data and insights that can inform policymakers and stakeholders in developing standards and regulations that ensure the safe integration of autonomous vehicles into existing transportation systems. Improving the accuracy of traffic light detection can lead to more efficient routing and reduced idling for autonomous vehicles, potentially contributing to lower fuel consumption and reduced emissions. This aspect of the study holds significance in the context of global efforts to combat climate change and promote sustainability in transportation.

1.5 Scope of the Study

This study focuses on enhancing the YOLOv4 algorithm for detecting the various colours of the traffic light and not working on the traffic light detection. It uses feature fusion enhancement strategies by merging features from various layers within the YOLOv4 architecture to improve colour detection accuracy for small targets. It incorporates the bounding box uncertainty prediction into the YOLOv4 algorithm.

1.6 Operational Definition of Terms

Algorithm: A set of rules or instructions given to an AI, computer program, or system to help in problem-solving and decision-making. In this context, it refers to the computational processes used for traffic light detection and recognition.

Bounding Box: A bounding box is a rectangular box that is wrapped as tightly as possible around a visual object (in this case, a traffic light). It is used to define the region of interest or the area where an object is located in the image.

Feature Fusion: This refers to the process of combining different types of features (like shape, texture, or colour) from multiple sources or layers within a neural network to improve the accuracy of detection or classification tasks.

Image Processing: A method in computer vision and digital image analysis where algorithms are used to perform operations on images to enhance them or extract useful information. **Neural Network**: A series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data. They are used extensively in AI applications for pattern recognition, classification, and more.

Real-time Processing: The capability of a system to process data instantly as it is received, with minimal delay, which is crucial in autonomous vehicle systems for immediate response to changing traffic conditions.

Semantic Features: In machine learning and image processing, these refer to the aspects of an image that relate to the meaning of objects and scenes in the image, such as the identity of visible objects.

Traffic Light Detection: A process in autonomous vehicle technology where the system identifies and interprets traffic lights in the vehicle's surroundings using cameras and sensors.

YOLOv4: An acronym for "You Only Look Once version 4," a high-speed, real-time object detection system used in AI and computer vision. In this study, it refers to the specific algorithm being enhanced for traffic light detection.

Endnotes

1. LC Possatti, R Guidolini, VB Cardoso, RF Berriel, TM Paixão, C Badue, AF De Souza, T Oliveira-Santos. *Traffic light recognition using deep learning and prior maps for autonomous cars*. In 2019 international joint conference on neural networks (IJCNN) 2019 Jul 14 (pp. 1-8). IEEE.

2. V Ilci, C Toth. *High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation*. Sensors. 2020 Feb 7;20(3):899.

3. M Hirabayashi, A Sujiwo, A Monrroy, S Kato, M Edahiro. *Traffic light recognition using high-definition map features.* Robotics and Autonomous Systems. 2019 Jan 1;111:62-72.

4. K Yoneda, A Kuramoto, N Suganuma, T Asaka, M Aldibaja, R Yanase. *Robust traffic light and arrow detection using digital map with spatial prior information for automated driving*. Sensors. 2020 Feb 21;20(4):1181.

5. LG Galvão, MN Huda. *Pedestrian and vehicle behaviour prediction in autonomous vehicle system—A review*. Expert Systems with Applications. 2023 Oct 12:121983.

6. E Khatab, A Onsy, A Abouelfarag. *Evaluation of 3D vulnerable objects' detection using a multi-sensors system for autonomous vehicles*. Sensors. 2022 Feb 21;22(4):1663.

7. X Wang, J Han, H Xiang, B Wang, G Wang, H Shi, L Chen, Q Wang. *A lightweight traffic lights detection and recognition method for the mobile platform*. Drones. 2023 Apr 27;7(5):293.

8. E Al-Ezaly, MH El-Bakry, A Abo-Elfetoh, S Elhishi. *An innovative traffic light recognition method using vehicular ad-hoc networks.* Scientific reports. 2023 Mar 10;13(1):4009.

9. S Pavlitska, N Lambing, AK Bangaru, JM Zöllner. *Traffic light recognition using convolutional neural networks: A survey*. arXiv preprint arXiv:2309.02158. 2023 Sep 5.

10. R Ayachi, M Afif, Y Said, AB Abdelali. *An embedded implementation of a traffic light detection system for advanced driver assistance systems*. InIndustrial Transformation 2022 May 9 (pp. 237-250). CRC Press.

11. S Wangsiripitak, K Hano, S Kuchii. *Traffic light and crosswalk detection and localization using vehicular camera.* In2022 14th International Conference on Knowledge and Smart Technology (KST) 2022 Jan 26 (pp. 108-113). IEEE.

12. D Chand, S Gupta, I Kavati. TSCTNet: *Traffic signal and countdown timer detection network for autonomous vehicles.* **International Journal of Computer Information Systems & Industrial Management Applications**. 2021 Jan 1;13.

13. R Ayachi. *Contributing to GPU and FPGA implementation of Deep Learning (DL) based intelligent vision algorithms for advanced driver assistance systems (ADAS)* (Doctoral dissertation, University of Monastir).2021

14. Y Qiang. *An optimization algorithm of intelligent traffic image recognition technology*. InProceedings of the 7th International Conference on Cyber Security and Information Engineering 2022 Sep 23 (pp. 424-428).

15. T Sahar, H Khadami, M Rauf. *Efficient detection and recognition of traffic lights for autonomous vehicles using CNN*. **Sukkur IBA Journal of Emerging Technologies**. 2022;5(2):49-56.

16. S Yan, X Liu, W Qian, Q Chen. *An end-to-end traffic light detection algorithm based on deep learning.* In 2021 international conference on security, pattern analysis, and cybernetics (SPAC) 2021 Jun 18 (pp. 370-373). IEEE.

17. AP Rangari, AR Chouthmol, C Kadadas, P Pal, SK Singh. *Deep Learning based smart traffic light system using Image Processing with YOLO v7*. In2022 4th International Conference on Circuits, Control, Communication and Computing (I4C) 2022 Dec 21 (pp. 129-132). IEEE.

18. S Bali, T Kumar, SS Tyagi. *Lightweight deep learning model for traffic light detection*. In2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS) 2022 Oct 10 (pp. 689-694). IEEE.

Chapter Two Literature Review

2.1 Conceptual Framework

2.1.1 Automobile Device

Many experiments have been carried out on the usage of visualisation detectors for the detection of transportation lighting. Although transportation light locations can be discovered on high-definition charts and automobiles must continually be conscious of their current standings. At a range of an extra 100 meters, it is significant to decide the present position of transportation luminosities to dawdle down for security. After appropriately recognising the transportation illumination condition, the required detection range can be estimated by evaluating the decelerating space from the automobile to the halt line.

Based on the study on automobile devices, the highest brake without triggering commuters any discomfort is approximately 0.1 G (= $0.098/^{2}$). For example, the decelerating space is approximately 98 m when the vehicle decelerates by 0.1 G while travelling at a pace of 50 km/h. When taking into cognisance the situation when the traffic signal is positioned at a position away from the break mark, the detection range may also upsurge. In automatic movement, it is essential to identify transportation illuminations at ranges larger than 100 meters to execute an ordinary connection technique. It is significant to deliberate the efficiency of the procedures, while taking into cognisance the interchange between the predictable consequence and the hardware requirement, to produce a practicable way of transportation illumination detection. For example, connecting a high-definition camera is a modest method to lengthen the detection distance. The handling interval might upsurge if the tenacity is amplified, nevertheless. Moreover, if the region of the image is lessened it has the potential to miss transportation illuminations.

Comprehending in what way to differentiate little pixel items is critical from the perspective of executing a Transportation Illumination recognition technique into use. However, the self-localisation element in programmed movement with high-definition charts correctly computes the automobile posture by chart-corresponding with a variety of devices or picture sensors. Spot accuracy of amid 0.1 and 0.2 m is characteristically supposed to be vital for automatic movement policymaking and pathway scheduling. Using the registered traffic light location and the current vehicle pose, a region-of-interest (ROI) position for the transportation illuminations can be estimated, assuming that the precise vehicle site on the digital map is predicted. The study part of transportation illuminations can be reduced by mining the ROI. Then, calculation costs and incorrect recognitions not unlike false-positive as well as false-negative recognitions can be reduced.

Associating traffic lights registered on a chart with traffic lights in a picture is a vital constituent of chart-based detection as well as developing detection execution. To create an approach resolution at the juncture while using high-definition charts, it is significant to identify the position of the germane transportation illuminations. Many approaches are being planned at present for detecting transportation illuminations.

Based on leading-edge studies, the detecting procedure can be elucidated below:

1. Select the area of the examination: Using the proposed chart, a region of interest (ROI) is extracted from the selected image.

2. Mine selected objects: As candidate traffic lights, circular lighting areas or rectangular objects are extracted from the search region.
3. Categorise the selected object conditions: Light colours and bolt light ways can be sensed using direct colour sieves or device learning procedures.

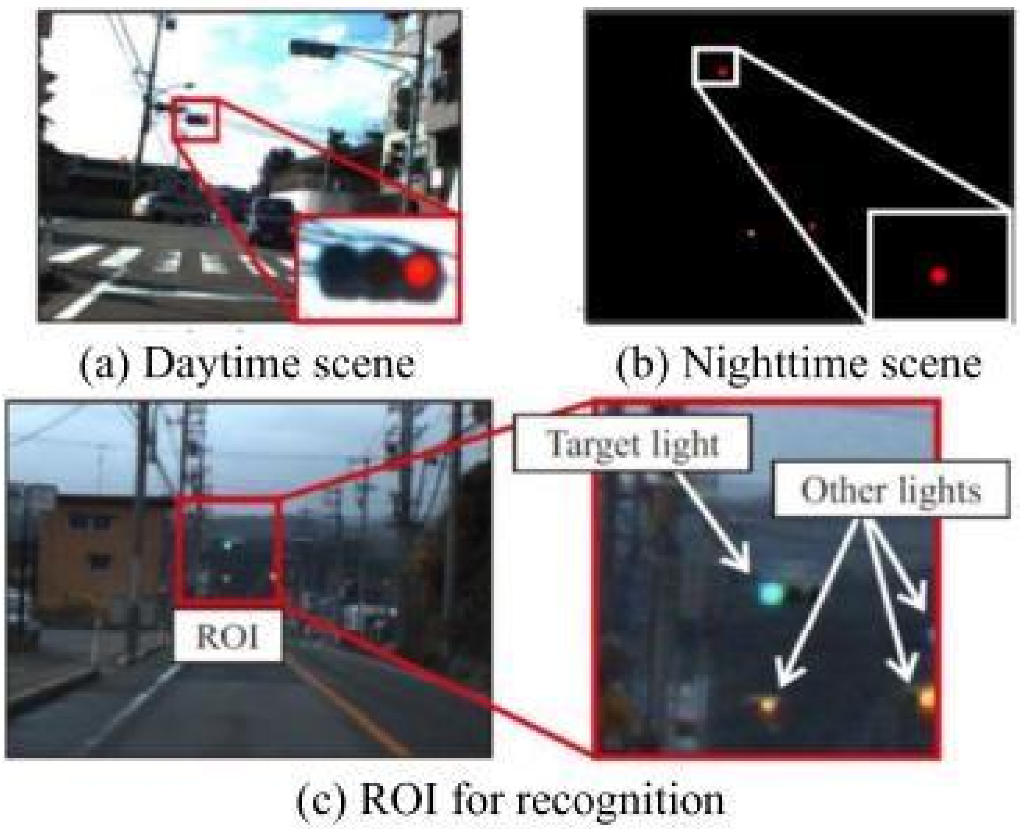


Figure 2.1 Traffic Light with Different ROI and Brightness[22]

Although many methods have been recognised, many of them comprise mining candidates centred on their exclusive colour spaces and round shapes; and categorising them as traffic illuminations or arrow illuminations. Whatever the nation, bolt and circular illuminations are the utmost general kinds of transportation lights. Because the examined area is restrained in ROI-based detection, the detection of round items is one of the effective approaches to detecting light zones. Some of the methods highlighted in studies employ spot sensors to mine prospective items by binarizing the image in addition to sectioning pixels[7]. Even slight, pixel-sized round items can be sensed by it. Then, using specific form equivalent and machine language, the detection of the whole form of the traffic lights is put into use. In addition, researchers concluded on the consequences of item tracing in steadying detection consequences[7].

There have been lots of examples where implementation has been improved by applying deep neural networks (DNN) in current years[7,11]. A critical choice for spotting bolt illuminations is a sensor centred on device knowledge. These approaches have a 90% detection frequency and can detect traffic lights at a distance greater than 100 meters. Conversely, real-time procedure is vital for executing if processes are going to be explored in computerised automobiles. It is likewise mandatory to spot traffic illuminations in a passable handling interval by the automobile's speed to decrease the detection deferral. For example, an automobile goes 14 metres per sec. when travelling at a speed of 50 km/h. The mandatory interval must then be estimated although taking into cognisance the receptive slowing down for practical development.

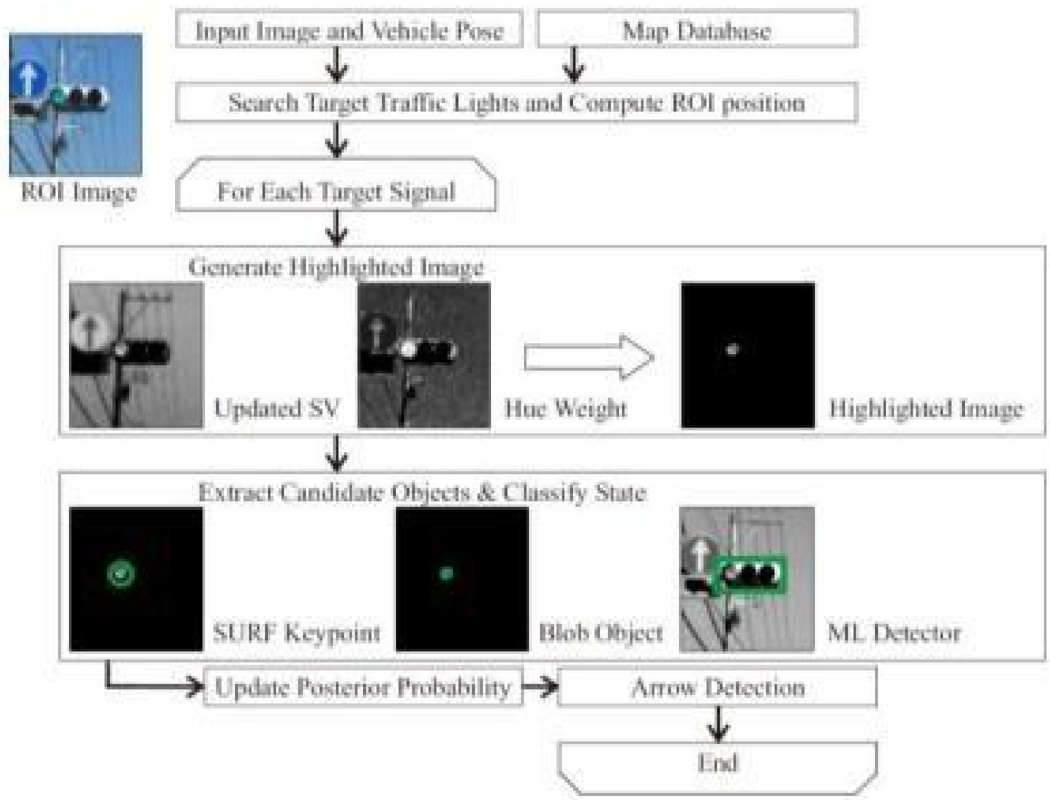


Figure 2.2 Flowchart Showing Algorithm for Traffic Light System[33]

Detecting the situation of the traffic lights in the object that resembles the high-definition chart is the objective of transportation illumination detection. The igniting position of transportation illuminations must be appropriately recognized all through the daytime and at nighttime. Traffic light locations in the camera pictures can be decided from the location of the traffic light and the automobile expending the position information of the distinct transportation lights stored on the HD map. The objective of executing a policymaking procedure for self-sufficient automobiles at junctures is to recognise the traffic light associated with the ROI.

Therefore, there will be a false-positive recognition if an additional transportation illumination is sensed in the definite ROI.

1. Explore mark traffic lights and approximate ROI in pictures expending High-definition charts.
2. Create a highlighted picture as a feature image which emphasizes the illuminations of traffic lights in the ROI picture
3. Expending three diverse kinds of mining methods, identify applications for traffic lights in the produced emphasized picture.
4. Estimate the possibility that a region comprises transportation illuminations using time-series investigation.
5. If the aim of transportation illumination offers feature data for bolt illumination, identify it.

The bulk of the prevailing detection methods classify traffic light candidates primarily by using separate features such as blob regions, circular objects, and general shapes. A strategy that combines these characteristics to find candidates offers a robust application. Once applying recognition methods that concentrate on the transportation light's illumination region, it has the potential to recognize the transportation illuminations even when their general shape is concealed, for instance at dusk or in obstruction. A forgoing vehicle, lorry and van are instances of rounding automobiles that obstruct transportation lights. In the obstructed state, it can be interesting to see the complete form of the traffic light. However, it is essential that the illumination region can be utilized to decide the transportation illumination state. Even at dusk, the situation is similar wherever the general form of a

Transportation Illumination can’t be seen.

The capacity to recognise bolt guidelines is a prerequisite for bolt recognition. When the transportation illumination on a high-definition chart has the attribute of bolt illuminations, it is contemplated to be a bolt sign. Bolt illuminations are characteristically lighted at red as well as yellow traffic lights in Japanese rush hour. An arrow recognition ROI is computed from the identified transportation illumination location after a yellow or red sign has been spotted. The right bolt sensor is cultured before expending AdaBoost for the bolt detection procedure before being smeared to the recovered ROI. The ROI image is rotated and the same detector is used to seek items to find left/straight arrows.

By perceiving the candidate objects, categorizing the illumination colour, and computing the confidence expending the existence possibility, transportation illumination detection is achieved by expanding this technique. By expending the previous data offered in the digital chart, our effort substantially develops the detection execution, predominantly for distant arrow illuminations. When there are several candidate items for a transportation light, it’s probable to weigh the candidates in the possibility of appraising the device centred on the expanse of the transportation lights. It should reduce conscriptual false-positive detections. On the other hand, by supplying the pattern of the target traffic light as prior information in arrow recognition, recognition can be improved.

The researchers' group is performing trial runs in the Tokyo waterside district of Japan as quota of the SIP scheme, as hitherto stated, and is collecting a momentous quantity of information employing onboard devices fitted on the trial cars, containing picture information for traffic light identification. A portion of information was collected throughout test drives for the SIP scheme in the Japanese city of Tokyo utilizing onboard devices fixed on trial cars, comprising graphic information for traffic light recognition.

In this assessment, the image list at the approaching of successions of intersections or junctions is well-defined as a situation, and a huge number of setting information are studied jointly to appraise the condition that is supremely comparable to the actual independent driving in the urban region. 233 connection technique situations centred on daytime, rear light, night, and raining or rainy climate information made up the assessment information. There are 42,603 total graphics information structures, 97,985 overall transportation illuminations, and 42,603 overall ROIs that require to be assessed. These comprised 8,555 transportation illuminations with red and arrow illuminations and 81,273 transportation illuminations with only lit-up transportation signs. Additionally, 8,157 transportation illuminations were obstructed by automobiles or structures, making it difficult to decide whether the ROI signs were illumined. The expanse between the automobiles and the transportation illuminations was disconnected into 10 m segments expending this information, and the mean F-value was calculated alongside the detection degree.

The detection handling was measured both in stints of the distinct transportation illuminations and in terms of every connection. This is because it is assumed that when multiple traffic signals are interconnected at a common intersection, the traffic signal status can be reliably determined by applying a majority voting method based on the statuses of various traffic lights, even when some of them may be obscured by obstacles and thus difficult to perceive. The detection degree after a camera with a telephoto lens in addition to a normal lens was evaluated. Although, the cameras currently in use have Full HD resolution to evaluate how much the recognition rate increases if a camera with a higher resolution is made available in the future. Thus, a lens with a 27° and 53° arena of vision is now being employed.

After the Transportation Illumination detection degree was evaluated in stints of definite transportation illuminations, it turned out to be evident that the degree drops for arrow illuminations that are situated alongside space away, even though those nearby could be detected precisely. This is because, with the cameras presently utilized, the amount of pixels is 10 pixels or smaller for an arrow illumination at an expanse of above, for instance, 90 meters, allowing it to decide the course of the arrow with exactitude.

However, it was recognised that integrating the outcomes of several transportation illuminations at an intersection and/or the joint use of telephoto lenses can develop the longexpanse detection degree. It was also established that practically 99.0% of the traffic lights inside 120 m were detectible, as revealed in Table 2.1.

Table 2.1: Mean Recognition Accuracy Within 120 m

Recognition accuracy	For each Traffic light	For each Intersection	For each Intersection with a telephoto lens
Green light	0.984	0.996	0.997
Red light	0.977	0.987	0.992
Arrow signal	0.908	0.960	0.982
Mean value	0.956	0.981	0.990

Even though the trial only comprised a lesser ratio of the Tokyo waterfront region in Japan, it was discovered that transportation illuminations were 99% detectable within 120 meters. Established on the consequences of these evaluations, we deliberate situations in which it turns out to be perplexing to spot transportation illuminations with current technology in this segment.

It was established that traffic lights could be detected with an accuracy of about 99% from the evaluation results using the onboard sensor data from autonomous driving currently being undertaken by the authors in the Tokyo waterfront area. On the other hand, the estimation's verdicts exposed that in circumstances like "occlusion", "nighttime," "backlight", and "conscriptual assimilation." It is puzzling to spot transportation illuminations. Because data from multiple traffic mounted at junctures can be employed, it was also established that these influences are generally transient and do not significantly affect decision-making to enter an intersection.

On the other hand, it is anticipated that detection expanding the present-day on-board cameras will become theoretically challenging in the situation where all the traffic lights can't be graphically detected because of the influences of the rear light or front illumination while stopped at the commencement of a juncture. It trails that investigation into the optimum fitting of transportation illuminations is essential assuming that on-board cameras will be employed to recognise them. However, dependable policymaking and detection are essential for independent driving devices, therefore it is anticipated to upsurge dependability by emerging laid-off schemes. Even though connecting numerous on-board cameras is one mode to create a laid-off scheme, it should be cautiously observed that this may not tackle the challenge of illumination complications as it is tentatively conceivable that the cause of the challenge will remain the same if we utilize the same principle of sensors.

Consequently, connecting a circulation illumination with interconnection technology is also desired, for example. To get prepared for the imminence of independent driving, it is vital that we not only develop onboard sensor-centred detection technology but also accelerate deliberations around how structure should be constructed from the position of easiness of detection. This has the prospective to upsurge the dependability of the independent driving device while also facilitating to upsurge the security of humanoid motorists. This research pursues to advance a traffic detection model with a dependable traffic light detection model for secured independent driving.

2.1.2 Traffic Sign Detection

Getting hold of data from several transportation symbols is critical in many applications, like Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. In general, a transportation signal detection scheme comprises two related subjects: Traffic Sign Recognition (TSR) and Traffic Sign Detection (TSD). The previous goals are to recognise the markers of identified road symbols into definite classifications/subsections, while the latter intends to correctly notice traffic signs in images. Although the subject has fascinated study attention in the computer vision domain for over two decades, there are still concerns owing to innumerable intricacies, like varied light circumstances, arresting directions and exposed fixed surroundings[17,18].

2.1.3 Traffic Sign Detection Methods

Traffic Sign Detection Methods are akin to other item recognition jobs in computer revelation. Further precisely, recognizing the object areas with vaulting packages that firmly comprise traffic symbols. Matching with general items, road symbols are commonly set up to possess firm and modest figures, and even and fascinating colours. The three-dimensional connection with other graphic items alongside the motorway is also a significant sign that could be used in the improvement of Traffic Sign Detection Methods. Consequently, depending on how road symbols are sectioned, methods can be segmented into colourcentred procedures, and geometry-centred procedures circumstances also should be considered. Several approaches have been projected in contemporary ages, comprising diverse methods established on edge arithmetical analysis, morphological filtering, Bag Of Visual Words, and AdaBoost. It should be noted that the printed procedures are habitually planned with reference to definite regulated settings[18,19,20].

LPD is typically trailed by eccentric subdivision, which halts the harvested authorized plate section into distinct types. Efficient subdivision is a different vital stage for LPR as the final stage of personality segmentation hangs on fundamentally on the worth of the classification yield. Subdivision is generally demanding in use, particularly when executing with graphics with intrinsically loud, small pixels and lowly climate situations. The utmost general exercise of personality dissection is centred on histogram analysis and thresholding. Previous information about the arrangement of the fonts on the plate is frequently employed, for instance, the spacing between two fonts. Despite the exertions done, no commonly established technique can unravel the challenges like twisted or lightened plate recognition. Furthermore, independently controlling LPD and font subdivision has the shortcoming that inappropriate recognition can unswervingly give rise to incorrect font subdivision.

Consequently, concurrent plate recognition and font subdivision is critical for a dependable LPR scheme.

In general, Traffic Script Detection (TTD) is nearly connected to Scene Script Detection (STD). As a result, the studies contained with these two subjects are studied. Despite there being many procedures that pay specific consideration to the recognition of sign-based traffic symbols, the technique unambiguously emphasises the recognition of the scriptcentred road symbols has enticed far less consideration. For the previous decades, only limited procedures have been proposed.

In a study, transportation panels are sensed by expending of Shi and Tomasi interface, Gaussian fusion models and K-means algorithm. Traffic panel candidates are segmented hinged on white and blue colours, trailed by Fast Fourier Transform (FFT) for shape classification. A TPD technique is described in a study[36], which noticed panels based on colour separation and Bag of visual words (BOVW). Maximally Stable Extremal Regions (MSERs) are used in recognizing script-based traffic signs and Mser-based script recognition and communication algorithms for independent automobiles for TPD[37,38].

Nonetheless, TPD remains problematic.

2.1.3.1 Traffic Script Detection

In general, Traffic Script Detection (TTD) is strictly linked to Scene Script Detection (STD). Consequently, the studies related to these two subjects are studied. In the face of several procedures that recompense specific consideration to the recognition of character-based road signs, the technique unambiguously emphasises the recognition of the script-based road signs has enticed much less consideration. Throughout the past ten years, only a few approaches have been projected. In a study, transportation plates are sensed by expending Shi and Tomasi features, Gaussian mixture models and K-means algorithm[39]. Transportation panel candidates are fragmented and hinged on white and blue colours, trailed with Fast Fourier Transform (FFT) for form categorization in Adaptive transportation highway sign sections script abstraction. A TPD technique is described in Script recognition and detection on transportation plates from road-level images expending graphic presence, which perceived plates built on colour separation and Bag of visual words (BOVW)[41].

MSERs are utilized in recognizing script-based traffic signs and MSER-based script recognition and communication algorithms for independent automobiles for TPD. Nevertheless, TPD still poses concerns owing to the enormous unpredictability of transportation plates. In the material realm, all plate is specifically made for demonstrating diverse data, giving rise to random discrepancies in dimension, colour, and form. Consequently, there may be a recognition breakdown of TTD that is caused by the recognition breakdown of TPD. The traditional approach to recognizing text on transportation signs needs to be of high quality. In the techniques outlined in "Script recognition and detection on transportation plates from road-level imagery using graphic analysis for script-based traffic signs" and "MSER-based script recognition and communication procedure for autonomous vehicles," transportation scripts are identified by applying Maximally Stable Extremal Regions (MSERs). Two important factors influence this approach:

(i) MSERs select regions that exhibit high stability and uniformity in their color characteristics.

(ii) Transportation scripts are typically designed with consistent and visually appealing colors to make them easily detectable by both drivers and pedestrians.

Utilizing MSERs as region identifiers is a logical and efficient choice. Inspired by the methodologies presented in "Script recognition and detection on transportation plates from road-level imagery using graphic analysis for script-based traffic signs" and "MSER-based script recognition and communication procedure for autonomous vehicles," the proposed system also incorporates the use of MSERs. On the other hand, the techniques are premeditated for spotting Spain and English terms, where the terms are generally a collection of many reliable notes. Dissimilar to Spain and English words, Chinese fonts are generally assembled of numerous varying lashes, the execution of the scheme will be impacted if only MSERs are used as area originators.

2.1.3.2 Scene Script Detection Methods

Subsequently, mutually TTD and STD goal to sense script in the wild, TTD can be meaningfully motivated by STD. Over the previous epoch, with the growth of dominant computer visualization techniques, STD has been swiftly advanced. Established on how candidate provinces are projected, methods to this challenge can be commonly segmented into two sets: interconnected element-centred techniques and sliding window-based techniques[45,46]. Linked element-based methods purposed to segment distinct character constituents established on data like colour, verge, grade and stroke thickness, and then collect the character constituents with related characteristics together to create characters. The symbolic procedures in this collection are Stroke Width Transform (SWT), MSERs., and Extremal Regions (ERs)[58].

SWT is an indigenous picture operative which calculates the thickness of the strokes connected to every resolution, and it has been useful in techniques like Detecting scripts of arbitrary orientations in natural images and script localization in natural images using stroke feature transform and script covariance descriptors A unified framework for multi-oriented script detection and recognition. In disparity to SWT, MSERs choose areas that are exceptionally firm in their appearance, and MSERs-centred techniques. have attained outstanding execution in STD. Reasonably than areas with exceptionally firm appearance, ERs are linked constituents of a picture binarised at dissimilar thresholds without solidity necessitates, outstanding outcomes also have been attained in ERs-based methods[53,55,56].

Gliding window-centred techniques execute script recognition as a general binary arrangement challenge. A diversity of device learning techniques have been used in these techniques. For instance, AdaBoosting. is used in Sensing and reading scripts in ordinary scenarios, Adaboost for script detection in natural scenes, Reading scripts in uncontrolled conditions and Script current: A combined script recognition scheme in regular scenario images for STD, and WaldBoost. is used with Histogram of Oriented Gradients (HOG) features in A synthesis procedure to spot and contain scripts in regular scenario images.

Rather than employing a boosted cascade classifier, color-based methods perform object recognition based on visual appearance data. These methods typically offer low computational costs and robust resilience against expected distortions. In one instance, a traffic sign recognition system was developed using HSI (Hue, Saturation, Intensity) color space segmentation and Support Vector Machines (SVM). This system is outlined in a paper titled "Traffic-symbol recognition and detection based on Support Vector Machines." The approach involves organizing appearance features into clusters to carry out image segmentation and shape analysis for road sign recognition.

Maximally Stable Extremal Regions (MSERs) are introduced in the context of real-time recognition and detection of highway transportation signs. This approach is followed by the utilization of Support Vector Machine (SVM) classifiers that have been trained using Histogram of Oriented Gradients (HOG) features, specifically tailored for traffic signal recognition applications. A technique that pools the data of the appearance, significance, three-dimensional and conscription connection is projected in Road symbol recognition through chart-centred standing and separation algorithms. In Towards actual-time transportation symbol recognition and classification., colour possibility model and MSERs are mutually utilized to produce traffic symbol candidates. A fusion area suggestion technique which comprises MSERs, as well as a Wave-based Detector (WaDe), is described in Traffic sign recognition through attention area extraction.

On the contrary, recognizing the symmetrical shape is another crucial aspect of road symbol recognition. Several mechanisms have been developed in this direction, including the Radial Symmetry Detector and the Triangular Detector. These techniques, such as the Triangular Symmetry Detector (TSD), make use of mutual appearance separation and robust shape analysis. They have been integrated into new road symbol recognition methods, which involve both appearance separation and robust shape analysis, as well as the recognition and placement of road symbols through convolutional neural networks (CNNs). TSD has also been closely associated with machine learning. For instance, AdaBoosting is employed in TSD for road symbol detection, using advanced AdaBoost recognition and forest-ecoc classification techniques. In the context of high-definition images, an improved CommonFinder AdaBoost (CF.AdaBoost) procedure is proposed for rapid multiclass road symbol recognition. Additionally, ACF (Aggregate Channel Features) and ICF (Integral Channel Features) sensors are applied in the recognition of U.S. traffic signs[86].

In precise and effective road symbol recognition employing categorised AdaBoost and support vector regression, a precise and effective traffic signal recognition technique is projected by discovering both AdaBoost and maintenance vector reversion for categorising sensor knowledge. Owing to the influence of pictorial learning from unusual data, deep learning has attained common attention in contemporary ages. Consequently, CNNs have likewise been used in TSD schemes. An actual-time TSD scheme that is built on CNN characters with a gliding window system is described in a real-world method for the recognition and organization of road symbols using convolutional neural networks[105]. In Road symbol recognition and detection using completely convolutional network-guided proposals, a TSD scheme is projected with two deep learning techniques, together with a Fully Convolutional Network (FCN) for road symbol application and deep CNN for image categorization.

TSR can be commonly explained as an arrangement recognition challenge, alongside numerous developed standard procedures from device memory. Among the copious models, SVM validates its noble execution, which has been used in Support-vector networks, Support vector devices for road symbols detection and Real-time detection and recognition of road traffic signs. SIFT and SURF descriptors taught in MLP attained great detection rates in reliable transportation symbol detection.

K-d tree with HOG configurations are described in road symbol categorization using k-d trees and random forests. Radial Basis Function (RBF) natural system and K-D tree are employed in two-stage road sign detection and recognition to recognize the state of traffic symbols. A symbol resemblance quantity with SimBoost and fuzzy reversion tree technique is recommended in the Vigorous category connection standard for road symbol recognition. A collaboration of categorizers hinged on the Error-Correcting Output Code (ECOC) technique is presented in road symbol detection employing evolutionary Adaboost detection and forest-ecoc classification, where the ECOC is schemed via a wood of peak tree assemblies that are entrenched in the ECOC medium. Sparse Representation Classification (SRC) recommended for facial detection is employed for TSR in parse-illustration-centered chart implanting for traffic sign recognition. The sparse representation-based graph implanting technique accomplishes great execution by merging character selection and subspace knowledge. In Traffic sign recognition via multimodal tree-structure embedded multi-task learning. multi-modal tree-structure entrenched Multi 2.3 License Plate Recognition Task Learning (MTL) is suggested to resolve TSR concern, which picks characteristics for improved detection and mutually connected characteristics for related jobs.

As for every detection challenge, characteristic illustration is a vital feature for device execution. How to construct distinctive and characteristic properties has been the dominant phase of computer visualization study. Profiting from dominant duty-explicit structures, CNN-based techniques attained high-tech execution in the struggle for the German Traffic Symbol Recognition Benchmark (GTSRB), which was done by the International Joint Conference on Neural Networks 2011. A multi-scale CNN is presented in Traffic symbol detection with multi-scale convolutional networks, dissimilar from customary CNN in that the yield of the final phase is served to a categorizer, in multi-scale CNN. The outputs of all the phases are served to a categorizer, producing exceptional execution.

A committee CNN and MLP is projected in “A committee of natural systems for road symbol classification, which is able to mechanically study task-specific invariant traits in a categorized way. Both these two techniques outdo the humanoid execution, nevertheless, the calculation cost is comparatively great. Hinge-loss CNN. additional enriched the detection performance by introducing a cost function that is widely employed in SVM.

2.1.4 License Plate Recognition

License Plate Recognition (LPR) techniques facilitate automatic recognition and detection of the documented amounts of automobiles through digital pictures. It is a main element for numerous safety and Intellectual Transportation System (ITS). Instances consist of vehicle bust-top entrée mechanism, automated road fee schemes, suspicious automobile scrutiny and tracing, and programmed recognition of terminated recordings. LPR has been reviewed for many years with a variation of techniques recommended for diverse types of authorized plates. A lot of nations have set up the technology as a major component of metropolitan transportation setups.

It is difficult to produce an all-inclusive review of the available journals on LPR. By chance, some journal papers give a vital foundation of data. There is a countless deal of unpredictability amid authorized plates amid diverse nations concerning the features, appearance, fonts employed, and arrangement. The majority of the current LPR devices are built on a general configuration, the sequential configuration of: (i) LPD, (ii) character separation, (iii) character detection. The plate recognition is contemplated as the blockage challenge in a sum of contemporary journals[130]. To construct an efficient LPR scheme, license plates not only need to be sensed appropriately but also quick enough to meet the necessities of actual-time applications.

In place of employing HOG interfaces, CNN interfaces are used in End-to-end script detection with convolutional neural networks and Deep features for script spotting, and the outcomes specify that CNN interfaces can function efficiently in gliding window systems. In future, an area originator hinged on ACF sensor and Edge Boxes is projected in interpreting script in the remote with convolutional neural networks.

Candidate area application is the preliminary phase of a script recognition scheme, its execution is of critical significance to exploit of whole scheme. The coupled element-based techniques have demonstrated vast benefits in calculation swiftness and toughness to scale, feature and revolution disparities. Nonetheless, it is challenging to notice script apparatuses, which are built of remote components, blurry or partly enclosed. To solve these challenges, Stroke Feature Transform (SFT) is projected in text localization in natural images using stroke feature transform and script covariance descriptors. by integrating colour signals of script resolutions. Area topology is used in Script indigenization in actual-sphere pictures expending proficiently trimmed comprehensive search as a picker, making sure in a protracted type of MSERs, termed MSERs++.

In the Script-attentional complication natural system for scenario script detection, ContrastEnhanced MSERs (CE-MSERs) is established upon MSERs, which senses extremely stimulating script designs. Colour-enhanced Contrasting Extremal Regions

(CERs) are described in Robust script recognition in neutral scenario images by general appearanceboosted divergent external areas and natural networks.

In contrast, gliding window-built techniques have fortes of toughness to sound and fuzziness, since durable components and categorizers are recommended. Script features with standalone components can likewise be introduced completely. The main restraint is the high calculation charge when scripts with dissimilar gauges, features and revolutions have to be sensed. Many of the prevailing techniques have centred on sensing parallel or nearparallel script outlines. There are merely scarce techniques that can tackle non-parallel script outlines. Multi-focused script recognition systems are established by combining candidates and amassing sets alongside related alignments. In future, this technique will likewise be used by the Script-consideration convolutional neural system for scenario script detection[111]. By fabricating a graph of MSERs for respective input pictures, the script outline recognition challenge is transported into a graph splitting challenge in Alignment with robust script line detection in natural images[116]. Multi-alignment STD schemes based on script candidates grouping are reported in Robust script detection in natural scene images and multi-

orientation scene script detection with adaptive clustering[112,113].

In general, scripts on road plates are tilted owing to seizing angles. Moreover, the intricate arrangement of scripts on traffic panels upshot in not only multi-aligned script outlines, but as well as crossed script lines. Consequently, to tackle the challenges of multi-aligned and crisscrossed script outline recognition, enhanced techniques ought to be established.

As the ordinary features of English, there is a lot of intra-expression and inter-expression conscriptual data inside scripts. It is exceptionally unusual that English script alone comprises one letter. As a result, many current techniques presume that scripts ought to have at least 2 constituents, or else they are unswervingly observed as uproars[114,115,116]. In disparity with English script recognition, Chinese script recognition is more intricate and demanding owing to the absence of intra-character conscriptual data. In China, the majority of traffic scripts comprise only two or three features, even many individual characters. Hence, dominant categorizers ought to be used in Chinese TTD schemes, so that distinct features can be precisely perceived.

2.1.5 Convolutional Neural Networks

Whereas the profound CNNs attain a sequence of successes in several CPU vision activities, there are a great deal of studies that fixated on refining the system construction of the novel AlexNets. In overfeat, integrated detection, indigenous and recognition expending convolutional networks, multi-scale and gliding window systems are executed inside CNN to instantaneously categorise, find and perceive things in pictures. A new imagining method is projected in visualizing and comprehending Convolutional Networks for envisioning and comprehending the purpose of intermediary character coatings and the process of the whole system.

Moreover, the CNNs. projected by Visual Geometry Group (VGG) specifies the alternative significant course of system construction scheme, which is system penetration. By proposing a small core size (3×3) in all coatings, the calculated charge is considerably cheap and it turns out to be possible to increase more convolutional coatings. Consequently, the prominent VGG Nets with system deepness from 11 to 19 are produced and have remained broadly employed in loads of consequent labour. In future, the 22-coat GoogLeNets are described, as well as the system constructions are excellently enhanced and built upon Hebbian standard and multi-scale handling. Owing to the cautiously made constructions, GoogLeNets succeeded as the winner in ILSVRC 2014. Profound CNNs unsurprisingly excerpt small-level structures to great-level structures in an endways multi-layer system., and the echelons of traits can be improved by boosting the profundity of systems.

The significance of system depth has been proven by both VGG Nets and GoogLeNets, which use extra convolutional coatings and accomplish improved execution than the AlexNets[165]. Nevertheless, with the rising of system deepness, the preparation of the system turns out to be increasingly problematic. The most infamous challenges are gradient disappearing and igniting. Luckily, with the help of techniques such as regularized initialisation and standardization coating, the slope challenges have been principally tackled, and systems with very profound penetration have begun to meet. Consequently, an added grave challenge is revealed in Highway networks and Convolutional natural systems at limited interval cost. As the system depth surges, both preparation and testing accurateness get flooded initially, and then tainted. This challenge is recognised as degradation[172], which is not produced by overfitting, hence the deepness of the system. To resolve the dilapidation challenge, systems built upon profound residual learning[172] are recommended.

By recommending individuality shortcut networks, all few stacked coatings are capable of learning residual charting more reasonably than uniform charting. Officially, signifying a wanted uniform charting $H(x)$ and a residual charting $F(x) = H(x)-x$, then the novel desired charting can be characterized as H(x) = F(x)+x. The outcomes of experiments demonstrate that the optimisation of residual mapping is much more stress-free than novel uniform charting. Consequently, the deepness of systems and accurateness of CNNs are intensely amplified, as well as ResNets won the champ in ILSVRC 2015.

Lately, so that overwhelmed slope disappearing and dilapidation challenges, the small pathway from initial coatings to future coatings has been broadly employed in several CNN designs. For example, Expressway Systems as well as ResNets propose personality shortcut networks for training profound CNNs. Stochastic Depth Networks employ a new workingout scheme that haphazardly drips a subsection of coatings of ResNets and evades them with personality interaction. FractalNets improve an unconventional approach termed droplet pathway for drilling ultra-deep systems. In future, the interaction approach is additionally concentrated in densely connected convolutional networks. To safeguard the all-out data movement classified systems, every coating obtains feedback from all previous coatings as well as permits all succeeding coatings. Owing to the tightly linked arrangement, the system designs are called DenseNets. Dissimilar from ResNets, structures from previous coatings are pooled by concatenation in place of synopsis. Likening to ResNets, DenseNets has the following benefits:

(i) greater factor effectiveness
(ii) stress-free training
(iii) sturdier standardising consequence for decreasing overfitting.

Lives can be saved by having ADAS that monitor the surroundings and caution or interfere in serious circumstances. The traffic control gadgets that interconnect with drivers comprise the traffic lights, traffic signs and roadway markings. For this purpose, automotive firms are concentrating on ADAS studies to improve safety. Advancement in machine knowledge techniques has opened novel doorways for development in the automotive field, particularly for ADAS where object recognition is the crucial function. Generally, when individuals think of Artificial Intelligence (AI) nowadays, they mean machine education i.e., teaching a machine to study a wanted behaviour.

2.1.6 Artificial Neural Network (ANN)

An artificial neural network is a computing device that consists of an assemblage of linked components called neurons, also recognized as nodes. They are prearranged in the layers. There are three kinds of layers in every ANN: the input layer, hidden layers, and the output layer. If an ANN has beyond a single hidden layer, the ANN is said to be a deep artificial neural system or deep neural system. Convolutional neural network (CNN) CNN is an unusual class of deep neural network that is used for Computer Vision (CV) for scrutinizing visual images and script analysis for natural language processing (NLP). Its design is similar to that of the connectivity design of neurons in the human brain. The neurons have knowledgeable weights and biases. A simple CNN is a system of layers, and every layer of a CNN changes one volume of activations to another through differentiable activities. The main part of CNN is to reduce the images into a form that's easier to process, without losing important features which are essential for getting a decent prediction.

This is vital when designing a design to be not only decent at learning features but also scalable to huge datasets. The key benefit of CNN in comparison to its prototypes where the filters were hand-assembled is that it absorbs directly from the input information with the spontaneous generation of feature maps. One more benefit of CNNs is using 'transfer learning' that not only aids in instances of lesser datasets but also reduces the training time considerably by converging faster. Where the clue behind transfer learning is taking a model skilled on one task and applying it to another related task. The impression is that a model has already some or all of the bulks for the second task so no need to train from scratch and the model can be applied much faster. There are two ways to use transfer learning:

1. Fine-tuning a CNN

2. Using the CNN as a fixed feature extractor

CNN Fundamentals

CNN is a computer image deep-learning network that can identify and categorize image features. CNN design was inspired by the organization and activities of the visual cortex. It is crafted to look like the association between neurons in the man's brain. Image identification is a task that men have been doing from infancy. Youngsters were trained to recognize fruits and vegetables, such as bananas, apples, and watermelons. Is it conceivable to instruct computers to execute the same thing? Is it possible for people to form a device that can perceive and comprehend just like human beings? The response is yes to all of these queries.

Human beings must validate an algorithm of millions of pictures before a computer can generalize the input and make forecasts for pictures that it has never perceived before. Just as in Figure 1, Human beings can see the rose but computers can see the arithmetic information. Therefore, programming a device that can scrutinize and identify pictures is a composite job.

Figure 2.3: How Computers Visualize an Image[25]

Individuals used to increase the figures of an image of roses or, for instance, mark pictures in the databank and write a program to associate the mark picture with the databank to understand if it contained a rose or an explicit target picture, but the key restraint was that it could not identify any lone picture that was not in the databank. As a consequence, there was a great demand for a system that spontaneously spotted and recognized threedimensional properties in pictures. Figure 2.4 displays numerous CNN mechanisms. To study the developments in CNN design, it is very imperative to comprehend the several CNN mechanisms and their uses.

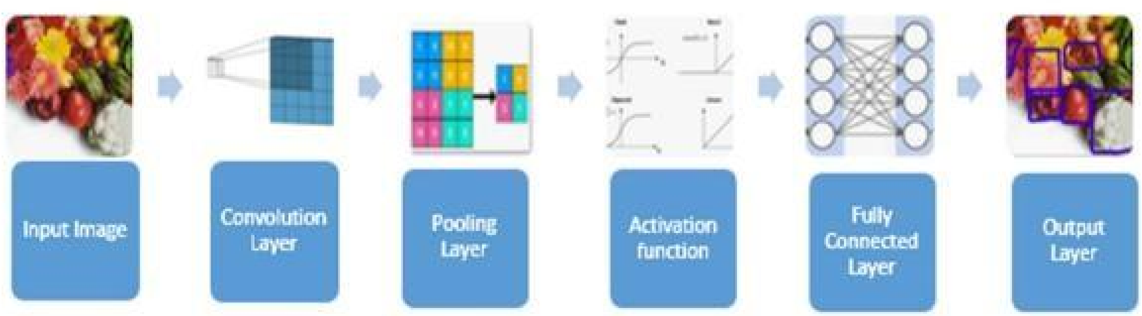


Figure 2.4 CNN Components[30]

Input Image: Pixels are the building blocks of computer pictures. They are the graphic data's binary image. A series of pixels vacillating from 0–255 are organized in a matrix-like arrangement in the digital picture. Its pixel worth stipulates each pixel's brilliance and shade. When human beings perceive a picture, their brains process a massive volume of data in the first second.

Every neuron in the human brain has its receptive arena and is associated with other neurons to cover the full visual ground. The receptive area is a small share of the visual area, where every neuron in the organic vision scheme reacts to incitements. In the same manner, every neuron in CNN examines information in only its accessible field. The CNN interfaces are automated to recognize simpler designs first, like lines and curves, before developing more composite outlines, like faces and images. Consequently, it is conceivable to say that utilizing CNN may offer vision to computers. The convolution interface is a very significant layer in the CNN design. It takes a picture as an input and employs a 3×3 or 5×5 filter, as shown in Figure 2.5.

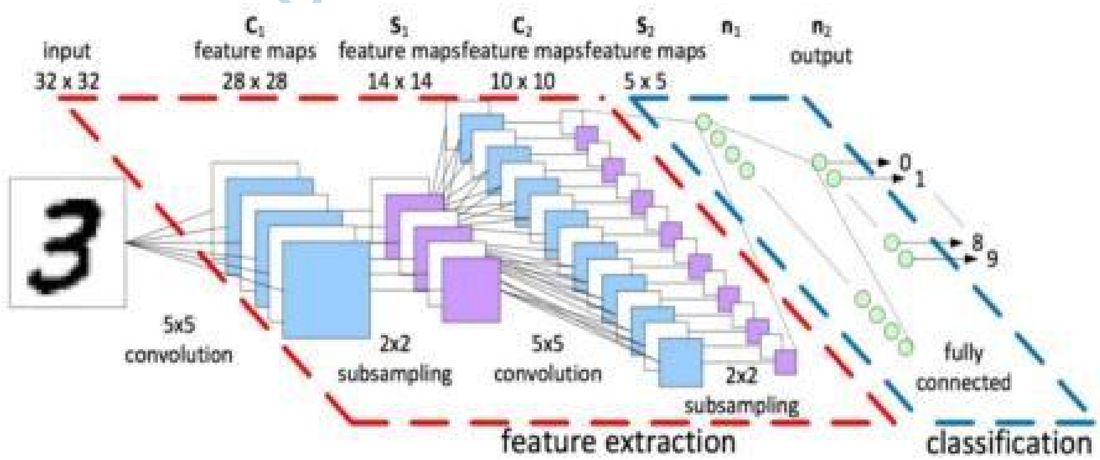


Figure 2.5 Convolution Layer[120]

In Figure 2.6, the green filter slides over the input picture, which is shown in blue, one pixel at a time, beginning at the top left. As it travels over the picture, the filter reproduces its worth with the picture's intersecting worth and then improves them all together to produce a sole significant yield for every overlap until the whole picture is reached.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Figure 2.6: Input and Filter Image[100]

The kernel has a similar penetration as the input picture when pictures have many networks, like RGB (red, green, blue). As revealed in Figure 2.6, matrix reproduction is executed between the Kn and In tons ([K1, I1], [K2, I2], [K3, I3]), and the outcomes are then pooled with the preference to harvest a dense one-depth network. Superimposing receptive areas occur for every neuron in the productivity matrix. The main ConvLayer generally captures low-level features like gradient alignment, boundaries, colour, and so on. The scheme acclimatizes to the high-level features by adding interfaces, offering us a system with an allinclusive understanding of the pictures in the dataset. Figures 2.6 and 2.7 demonstrate the stages of complexity.

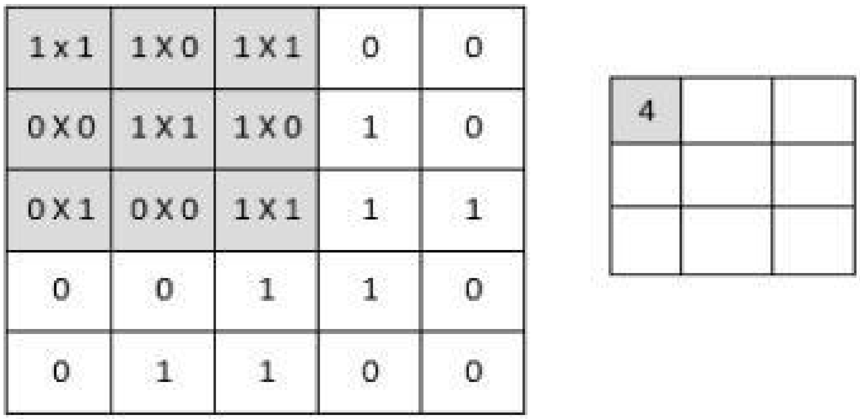


Figure 2.7. Calculation of Filter Slides Over Input Image[100]

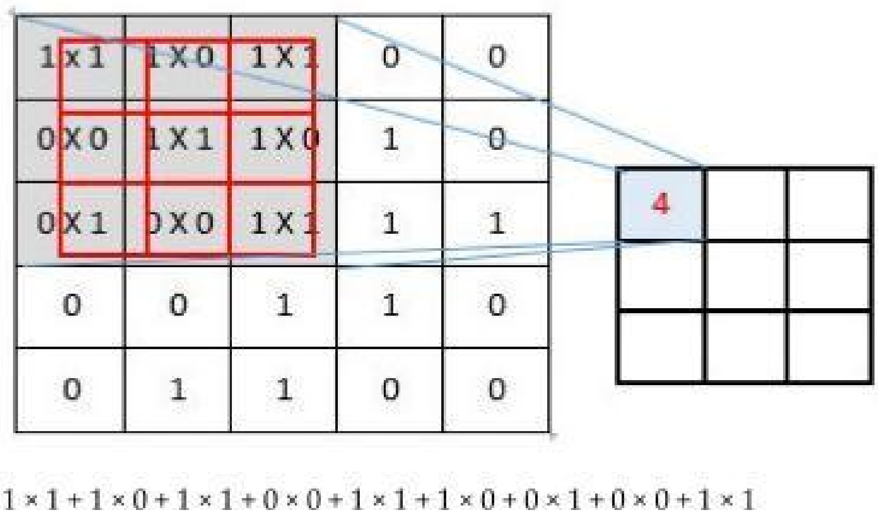


Figure 2.8: First Step of Convolution[110].

2.1.7 Feature Extraction

CNN is recognized for its capability to abstract features spontaneously. Figure 2.9 illustrates the matrix design for RGB pictures. Padding is commonly used in CNN to retain the scope of the character maps from dwindling at each interface, which is objectionable. The process yields two kinds of results:

1. A kind in which the dimensionality of the intricate characteristic is reduced in contrast to the input
2. A kind in which the dimensionality is not decreased but is either improved or sustained. Padding is employed to gratify this task.

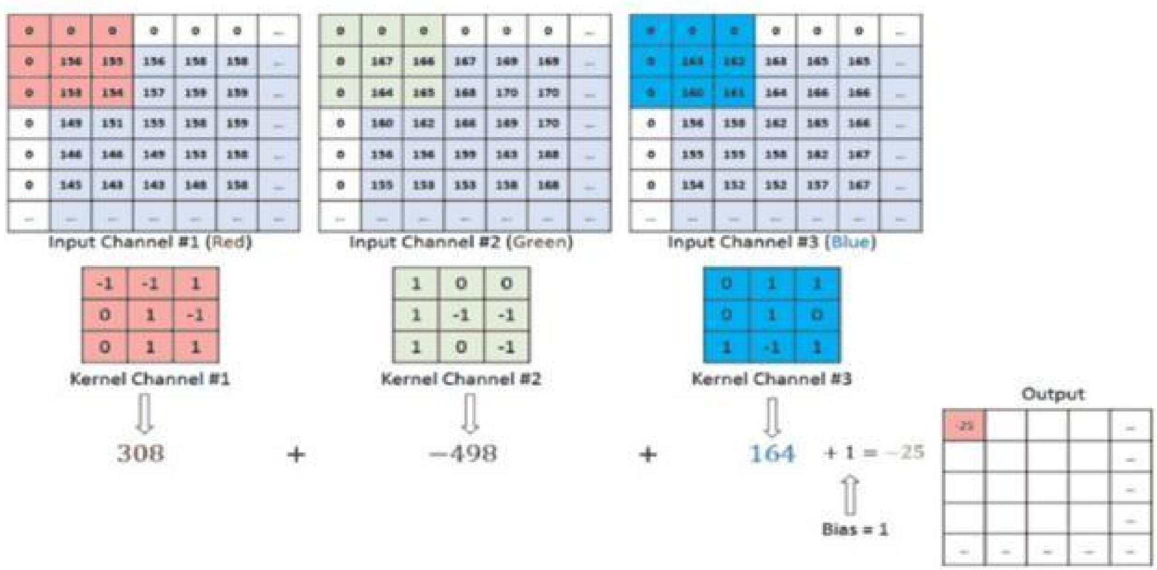


Figure 2.9. Matrix Calculation[65]

For example, when the 5×5×1 picture is reinforced into a 7×7×1 image and then applied to the 3×3×1 kernel over it, the complex matrix is observed to be of dimensions 5×5×1, as shown in Figure 2.10 It indicates that the output image has the same dimensions as the input image (same padding). If the same procedure is conducted without padding, an image with reduced dimensions can be received in the output. As a result, a 5×5×1 image will become a 3×3×1 image.

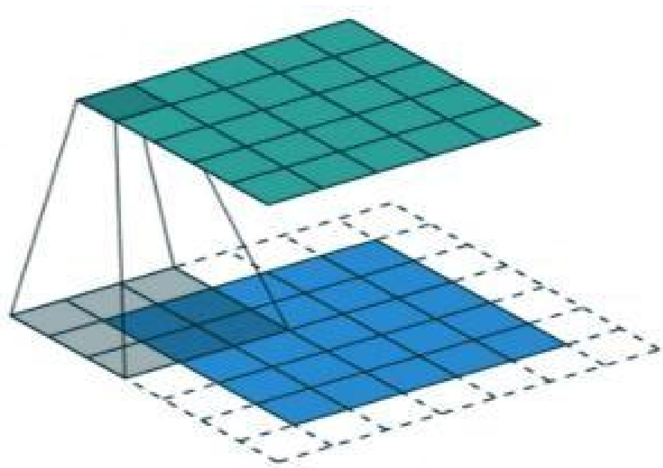


Figure 2.10: Padding[78]

The kernel goes through the width and height of the image during the progressing pass. It produces a graphic picture of the receptive area in question. It produces an activation chart, a two-dimensional illustration of the picture that displays the kernel's retort at every threedimensional location of the picture. A stride is the magnitude of the kernel when it glides. Assume the input picture is W × W × D in magnitude. If the number of kernels with a threedimensional measurement of F, stride S, and padding P is unidentified, the resultant volume can be calculated utilizing the following method:

$$Wout = \frac{W - F + 2P}{S} + 1$$

This will produce an output with size Wout × Wout × Dout.

2.1.8 Pooling Layer

After attaining the feature maps, it is essential to add a pooling (sub-sampling) layer in CNN, next to a convolution layer. The task of the pooling layer is to shrink the convolved feature's three-dimensional size. Because of the dimensionality reduction, the computer power essential to process the information is reduced. This also helps in the removal of important features that are positional and rotational invariant, which conserves the model's applied training. Pooling decreases the training time while also averting over-fitting. There are two kinds of pooling: maximum pooling and average pooling. Maximum Pooling The tensor is the input to the pooling layer. A kernel of size n×n (2×2 in the abovementioned instance) is moved through the matrix in the situation of maximum pooling, as demonstrated in Figure 2.11, and the maximum value is selected and positioned in the suitable place of the output matrix.

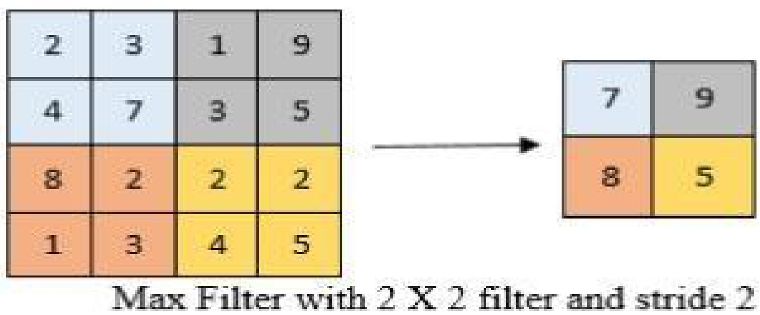
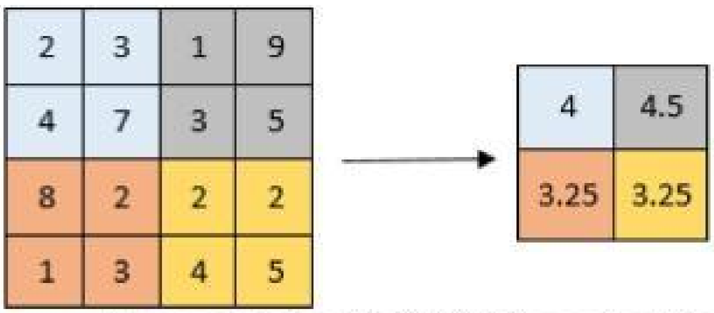


Figure 2.11: Max Filter[111]

2.1.8.1 Average Pooling

A kernel of size n×n is moved across the matrix in the average pooling, and the average of all the numbers is gotten for every point and positioned in the equivalent location of the yielded matrix. This is reiterated for every of the input tensor's networks. Consequently, we have the output tensor. It is significant to keep in mind that, while pooling decreases the picture's height and breadth, the number of channels (depth) remains the same. The pooling layer analyzes a summary statistic of the connecting outputs to substitute the system output at definite points.

Consequently, it helps in decreasing the representation's three-dimensional dimension, which decreases the volume of calculation and weights essential. The pooling procedure is executed self-sufficiently on every slice of the image. The average of the rectangle neighbourhood, the L2 norm of the rectangle neighbourhood, and a weighted average reliant on the distance from the chief pixel are all pooling roles as revealed in Figure 2.12. The most recurrent technique, on the other hand, is maximum pooling, which reports the neighbourhood's most important output.



Average Pool with 2 X 2 filter and stride 2

Figure 2.12: Average Pooling[77]

2.1.9 Nonlinearity Layer (Activation Function)

The activation function executes an important part of CNN interfaces. The output of the filter is provided to another arithmetical role termed an activation function. ReLu, which connotes for corrected linear component, is the most general activation function employed in CNN feature extraction. The key purpose behind using the activation function is to determine the result of neural systems, like yes or no. The activation function charts the result values between −1 to 1 or 0 to 1, etc. (it hangs on the activation function). The activation functions can be classified into two types.

i. Linear Activation Function

This utilizes function F(x) = CY. It takes the input and multiplies it with constant c (weight of each neuron) and yields the output signal comparative to the input. The linear function can be improved than the stage function, as it only gives the yes or no answer and not the multiple answers.

ii. Non-linear Activation Function

Functions In up-to-date neural systems, non-linear activation functions are utilized. They allow the model to construct complex mappings amid the system's inputs and outputs, which are important for learning and modelling composite information, including pictures, video, audio, and non-linear or high-dimensional information sets.

2.1.10 Fully Connected Layer

A fully linked layer is nothing more than a feed-forward neural system as revealed in Figure 2.13. Completely linked layers are found at the system's very bottom layers. A completely linked layer obtains input from the final pooling or convolutional interface's output interface, which is compressed before being conveyed as input. Flattening the output involves unrolling all values from the output that were gotten after the last pooling or convolutional layer into a vector (3D matrix).

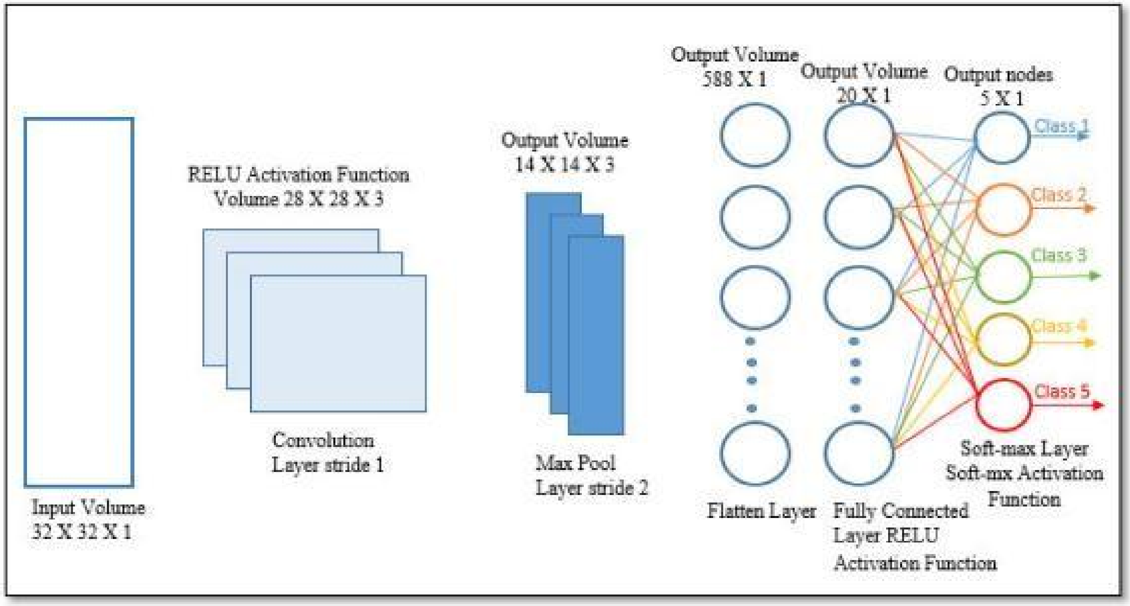


Figure 2.13: Fully Connected Layer[129]

Adding an FC layer is a simple method to study nonlinear amalgamations of high-level structures exemplified by the convolutional interface's output. In that space, the FC interface is learning a feasibly nonlinear function.

2.2 Methodological Review

There are 3 sub-classifications into which computer vision challenges can be categorized. They are recognition, organization, and tracking. Correspondingly, researchers offer an analysis of transportation symbol detection. Whereas the tracking phase offers threedimensional and temporal data between settings, the recognition and organization phases function chronologically in each setting. Discovering the Transportation Light candidate is a subject of the recognition challenge. The organization is then carried out based on the interfaces taken out from the identified candidates. Tracking Transportation

Lights (TLs) through a sequence of frames requires the usage of data as regards the location of Transportation Light condition or state. Any TLR scheme that addresses the abovementioned challenges can be divided into four phases, specifically: detection, feature extraction, classification, and tracking.

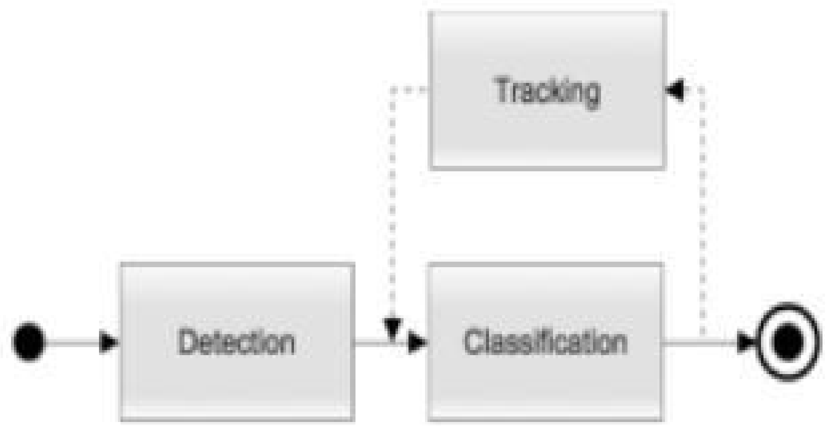


Figure 2.14 Breakdown of a Computer Vision-Based Traffic Light Recognition System

The support scheme should be capable of differentiating between diverse light conditions when manifold TLs are concurrently noticeable to the motorist, and decide which Transportation Light is pertinent to the motorist; a duty that can be perplexing even to a humanoid. An intricate transportation scenario condition in Figure 2.14, where three imminent junctures are all observable simultaneously.

Of the junctures, one comprises turn tracks that have their individual TLs. Deciding whether a Transportation Light applies to the ego-automobile signifies the main concern for TLR devices, comparatively to DAS. The significance of Transportation Light is carefully associated with its location compared to the ego-automobile. The most advanced technique for determining this concern is understood, where a presumption is centred on the juncture width and the forecast coordination of the TLS[10]. An author suggests a different and less vibrant method, where the roads are chronicled ahead and applicable TLs are physically interpreted offline[7]. In the annotated regions, interfaces are extracted; then the scheme can identify the applicable TLs on that definite road. The data amassed through the TLR device for DAS must be connected to the motorist, if possible, in a manner that is not invasive and complements the motorist's intellectual burden as little as possible. When a motorist is not concentrating, a security device or scheme can be triggered, or it may even be used to decide whether the motorist has sensed particular items or if the motorist hasn't, he should be made conscious of it. Consequently, it is likely to use the combination of information from looking-in and looking-out devices. To predict motorists' actions, the authors pool an extensive number of looking-in events, like head position evaluations, and hand and foot tracking; with looking-out actions, like automobile factors, track and highway geometry scrutiny, and neighbouring automobile routes. The attention of this research does not spread to the DAS performance element.

All through recognition, TLR devices characteristically hunt for a variation of Transportation Light apparatuses. In some circumstances, the element's physical connection can be employed. The procedures of recognition can either be knowledge-based or prototypical-based. A candid and commonly used colour prototypical for model-based recognition entails colour thickness thresholds for every Transportation light colour.

Sensors centred on this type of model are predominantly susceptible to over-correct explicit exercise information collections. Of the quite a few sensors offered by researchers[7], the finest executing sensor is a Gaussian pixel classifier manufactured from thousands of physically annotated images. Uncertain clustering is presented to produce exceptional symbols of a specified colour[11,12]. In disparity to fixed clustering, also recognized as solid clustering, information points in vague clustering can belong to one or more clusters. The connection level to clusters can differ, detonating that an information point can have a greater probability or likelihood of belonging to one cluster or another.

General to all sensors that depend on colour transmissions is their sensitivity to colour conflicts, which can be produced by a variation of effects. A general recognition technique is limelight recognition using the white top hat procedure, which is vigorous to conflicts in colour. This technique is centred on greyscale imageries, as can be perceived in research[8,9]. The V network from the HSV colour space is exploited in a multi-feature combinationbased transportation light detection algorithm for smart automobiles with a similar result. Shape prototypes are applied as a sieving step following the colour-based subdivision or as an alternate to the overriding colour prototypes. According to the author, the edge map of a Laplacian edge recognition sieve is exposed to the Hough transformation[23]. A variety of the adapted spherical Hough transformation which outdoes the standard Hough transformation is the involvement of Traffic Light Recognition with Color and Edge Data. It exploits for

full circles.

By means of also skimming for other circles around lively lights, magnifies this impression. Before probing for estimated ellipses in the canny edge pixels around candidate BLOBs, the authors apply a Laplacian sieve that excerpts a vivid border while disregarding halo effects[6]. Researchers engaged the usage of smart radial symmetry to find circles, and the local maximum and minimum were engaged to decide the practice limits of the assumed circle[18]. Authors engaged morphological procedures and thresholding to discover article limits; the margins are then topologically evaluated and Transportation Light candidate rectangles are recognized.

Learning-Based: A cascade classifier centred on Haar topographies was an initial effort at learning-based recognition[7,8]. On the other hand, in contrast, their Gaussian colour categorizer outdone it. Three documents that engaged some more efficient learning-based sensors have just been in recent times printed. For TLS recognition, authors pool the incidence priors from a probabilistic prior chart with a recognition score centred on SVM cataloguing of Histogram of Oriented Gradients (HoG) structures. Configuration detection employs an ML classifier to perceive Transportation Light pixels by extracting structures from colour, pattern, pixel position, and standardized voting histograms.

In a study, the novel RGB structures were changed, and then from them were removed as total blocks of pixels in 10 networks. The removed structures are categorized using a deepness learning tree as perceived in research[33]. In another research, the learning tree deepness is raised to 4 and the scale space is expanded to develop detector performance. The necessity for loads of information and calculation is a general prerequisite for learningbased sensors. Owing to the momentous volume of information employed throughout the training procedure, they have resilient toughness to variability and a low predisposition to overfit[34].

Auxiliary Detection: According to a study, when approaching pre-annotated intersections, GPS location is utilized to trigger the recognition device[23]. This stage was later developed by adding maps comprising much more specific data on Transportation Light locations[7,10,24,28,31]. Correct GPS measurement and manually interpreted Transportation Light placement were engaged to produce these maps. Hue and saturation histograms of each Transportation Light were stored throughout the early Transportation Light mapping[35]. According to a study, the probable conditions of the distinctive TLs were also interpreted to reduce incorrect positives. Visual TLs recognition can be enhanced upon by discharge on the aforementioned maps. The maps upsurge accurateness in Transportation Light candidates, making it easier not to allow incorrect candidates. Significantly, great dependence on maps can give rise to grave missed recognitions, in circumstances such as while highways are being built.

A general attribute of the organization is colour. The usage of colour thicknesses from sectional areas can be perceived in research. The attributes in research are established on HSV histograms. With the colour attribute, figure and construction are regularly engaged in Transportation Light attributes. Part ratio, measurement, and region are only a few instances of the numerous features that form figure data. The comparative position of TLs' elements constitutes physical data. Figure and physical data are general structures because

Transportation Light lamps may regularly be effortlessly differentiated from the conscriptual setting. Colour pattern equivalent is utilizedin Robust and real-time

transportation light detection in intricate city surroundings.

According to a study, the Support Vector Machine (SVM) classifier engages a blend of BLOB width, height, centre synchronisation, region, magnitude, the totality of pixels, brilliance time, and geometric time attributes. In a research study focused on analyzing image regions containing multiple transportation traffic lights, more advanced attribute descriptors were explored, building upon the edge information represented by Histogram of Oriented Gradients (HoG). Researchers opted for 2D Gabor Wavelet features instead of the Haar attributes, leading to more sophisticated attribute representations. The study employs a three-dimensional scripture blueprint to categorize TLs. Precisely, they calculate a Local Binary Pattern (LBP) histogram for the Transportation Light as well as for five areas of equivalent magnitude in every colour network, before constructing a characteristic vector out of the concatenated LBP histograms. Devices that depend on colour, characteristics, or physical qualities will face problems in a diversity of real-world circumstances. Development in toughness is executed by engaging many characteristic types comprising several kinds of data.

A mixture of scores from configuration, figure, colour, and geolocation data is engaged by researchers to categorize Transportation Light candidates and decide if a Transportation Light should happen there[42]. Researchers simply evaluate the condition as the champion of a popular count on the number of pixels classified by experientially proven thresholds.

Founded on a controversial technique that selects the optimum light from Transportation Light location and magnitude decides a Transportation Light condition for the full fragmented structure[45].

Using the colour circulation, the authors split Transportation Light candidates perpendicularly into three classes[58]. Researchers focus on classifying the arrow kind of their Transportation Light candidates, and they accomplish this by categorizing Gabor picture interfaces according to their nearby neighbours after decreasing their exhausting 2D selfgoverning element analysis[57]. HoG structures from the Transportation Light vessel and SVM are employed by authors to categorize the TLs[56]. A neural system is used in researches[39,40] to determine the state of the revealed TLs. A pattern corresponding by regularized cross-correlation is employed in Strong and real-time traffic light detection in compound city surroundings, applying adaptive pattern matching in researches[50,52,53]. For classification centred on HSV histograms, researchers use SVM[49].

Cascading classifiers based on Haar structures are employed in research disparities in the learning-based AdaBoost cascade Haar feature classifier with their anticipated adaptive pattern matching method[58,59]. Their model-based approach considerably outclassed their learning-based method, so it was discovered. The research employs SVM to categorize LBP structure vectors to determine a TL's condition from the arrangement of its threedimensional scripture. The value of the structures is necessary for fruitful arrangement. The major studies employ a classifier to examine the recovered structures and liken the outcomes with an array of smart Traffic Light conditions to decide which match is the best. The research that follow classify TLs employing heuristics. A proportion of the variables perceived in a real-world performance can influence arrangement centred on, for example, heuristically derived thresholds.

All the device learning-based procedures culture a model with training information examples, which necessitate huge bulks of information with very great discrepancy to be very strong. Trailing is regularly employed to administer sluggards produced by stuff like obstruction, and to reduce fabricated positives to decrease sound. Tables I and II specify that nearly half of the procedures deliberately engage some kind of tracking. By observing at aforementioned structures, temporal tracing can notice whether a candidate has formerly been revealed in a similar part and whether it portions the similar features as a specific candidate in the contemporary structure. This is a forthright approach that has been engaged, researchers employed a temporal choice procedure that bases the concluding organization on temporal constancy to decrease fabricated recognitions by a third[37,38,45].

Comparable procedures are employed in Semantic dissection-based transportation light recognition during the day and night, where a Transportation Light must be perceived in 3 successive structures before it is permitted. It is obvious from their outcomes that accumulating this type of tracing amplified general accuracy by 12.16% at the cost of general recollection by 6.27%. The research employs manifold goal temporal trailing and forecasts TL's position established on the self-worth automobile's speed[49]. In addition, as TLs are ready to disappear from the area of sight, they alter the top hat kernel's size, permeation, and strengthened thresholds, when TLs are just around to vanish from the region of sight. This makes detection probable over a broader expanse variety or interval. The identified condition from the classifier is added into a variation of HMMs, one for every prospective kind of Transportation Light and one for non-TL objects, before it gets to the last phase decision[49].

The final projected phase is then selected as the model that best narrowly ties the perceived order of conditions. HMM is also engaged in a study, but only for a sole Transportation Light type[61]. A study experiments with a Kalman filter and an element sieve for trailing the comparative drive between the automobile and TLs and evaluates the distance to TLs by reverse standpoint mapping[36]. The steadiness of the setting and colour of the TLs is then engaged to sieve them. An Interrelating Multiple Model sieve is engaged in tracing both the posture and rank of a transportation light through an interrelating multiple model sieve to trace the condition and site of a definite TL[51].

To trace the condition and the spot in time, the forecast in the model engages Kalman sieves. A Markov chain with weighted possibilities is utilized to establish the condition and was initially called in a study to find the present condition established on subsequent conditions. Transportation Light mapping localization, and condition recognition for self-driving automobiles regulate the localization discrepancy between the projected and actual Transportation Light area employing preceding charts and a histogram sieve. The connection tracing technique engaged in studies adopts that a noticed TL's condition won't adjust significantly over the progression of a series of structures[39,40,42]. For example, if a cord of red conditions is discovered, it is possible that the condition in the subsequent structure will also be red, causing an unevenly related incidence. Utilize CAMSHIFT tracing of candidates centred on presence across structures[58]. The major objectives of tracing are to tackle with sole unsuccessful recognitions conveyed by, for instance, obstruction, and to sieve out sound.

Most of the reviewed journals use tracing in a rudimentary temporal dependability check; nevertheless, a few go beyond and integrate previous possibilities. Connection tracing and opinion tracing, in general, are the two chief kinds of tracing that are engaged. Since connection tracing regularly hangs on similar structures as the sensor, it is difficult to supplement the sensor if it flops. Consequently, point tracing can make employment of past data, which offers a superior base for improving the sensor.

The studied research has estimated TLR scheme execution in a diversity of ways, making it hard to associate competing methods. Moreover, the appraisal conditions engaged in some studies are not clearly specified. The bulk of the time, an exceptional native assortment of structures are utilized for estimation. The bulk of these indigenous datasets are minor and have lesser disparities. Exactitude, memory, and accuracy are the three parameters employed most often to gauge device execution. Consequently, where possible, consequences from the studied TLR schemes can be encapsulated by these parameters. In a study, the parameters of memory, exactitude and accurateness are all well-defined. In equations (1), (2), and (3), the descriptions are specified. The abbreviations for correct positives, fabricated positives, made-up negatives and factual negatives are TP, FP, FN, and

TN.

$$Prc = \frac{\quad}{+} \qquad (1)$$

The nearer the Accuracy is to 1, the sturdier the indicator that all the detected Transportation Light conditions have been appropriately detected.

$$r = \frac{\quad}{+F} \qquad (2)$$

The nearer the Memory is to 1, the sturdier the signal that all the Transportation Light conditions were efficaciously recognised by the device, in a specified video arrangement.

$$r = \frac{+F}{+F+F+} \qquad (3)$$

If the accuracy is close to one, the device senses all TLs without creating any wrong positives. Factual negatives are characteristically comprised in the Accuracy calculation as revealed by the third calculation, but they are seldom applied in TLR scheme calculation. These execution parameters may intermittently go by other labels, such as detection degree n as an alternative to exactitude, or recognition degree in place of recollection. A Transportation Light is categorized as a TP if it has been recognized once across the whole successions of structures where it appears, as can be seen in research. As this offers a more precise representation of a specified device's performance, we recommend appraising FPs and TPs structure by structure.

2.2.1 Architectural Evolution of Deep CNNs

Figure 2.16 designates the several design classifications of CNN variations. This section elucidates all classifications in element.

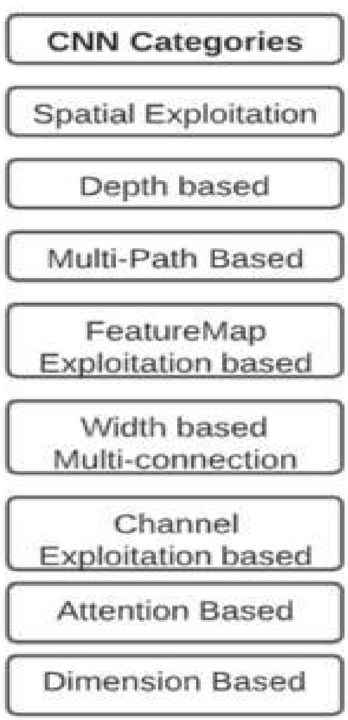


Figure 2.15: CNN Variants Categories[150].

2.2.2 Spatial Exploitation-Based CNNs

There are several considerations in CNN, which include biases, weights, the number of layers, neurons, activation function, stride, filter size, learning degree, and so on. Diverse connection stages can be examined using diverse filter magnitudes because convolutional processes consider the vicinity (locality) of input pixels. Diverse filter dimensions comprise various degrees of granularity; naturally, fine-grained data is mined by filters in smaller sizes, although coarse-grained data is mined by filters in big sizes. Consequently, in the early 2000s, three-dimensional filters were utilized by researchers to upsurge performance. It was perceived that there is a connection between a three-dimensional filter and system learning. Countless studies carried out during this time epoch specified that, by regulating filters, CNN could execute more efficiently on rough and fine-grained details.

2.2.3 CNN Based on Depth

The chief impression behind deep CNN design is that, with the assistance of extra mappings (nonlinear) and more unconventional feature pyramids, the system can estimate the objective task efficiently. The system's penetration has been a significant factor in controlled teaching. Deep systems symbolize precise task courses more efficiently than superficial schemes. In 2001, a theorem was recommended termed "universal approximation". It explains that one concealed layer may estimate any role. On the other hand, this occurs at the expense of an exponentially massive amount of neurons as well as a computationally unworkable consequence. It was postulated in 2011 that deeper networks can sustain the system's theatrical influence at a smaller cost. Deep systems are computationally more effective for difficult operations, according to a researcher, who established this empirically in 2013[193]. VGG and Inception executed best in the ILSVRC2014 competition, strengthening the idea that depth is a significant factor in regulating the learning capacity of systems.

2.2.3 CNNs with Multiple Paths Deep

CNNs are frequently good at complex tasks. Occasionally they may suffer from performance deprivation, eruption problems, or gradient vanishing, which are created by increasing the depth rather than overfitting. The disappearing gradient concern gives rise to a bigger test blunder and training blunder. The model of cross-layer connectivity or multipath was projected for deep training systems. Shortcut networks or many paths can have a connection from one layer to another systematically by eluding some in-between levels, permitting the modified flow of data between the layers. The system is divided into segments utilizing cross-layer connectivity. These paths resolve the disappearing gradient challenge by spreading the gradient to lesser layers.

2.2.4 Feature-Map Exploitation Based CNNs

CNN became dominant for MV jobs owing to its capability to discharge categorized learning and automatic feature extraction. The execution of organization, subdivision, and recognition components is deeply controlled by feature selection. CNN selects structures animatedly by regulating the weights associated with a kernel, also identified as a mask. Additionally, diverse feature removal phases are executed, permitting several kinds of features (known as feature maps or channels in CNN). Nevertheless, some of the feature maps have little or no role in object discrimination. Excessive feature sets may offer a sound effect, giving rise to the over-fitting of the system. This infers that in addition to system designing, the collection of feature maps can perform a crucial function in amassing system generalization.

2.2.5 Multi-Connection

Depending on the Width During CNN developments, the stress was predominantly on leveraging the probability of the deepness and the productivity of links in system regularization from 2012 to 2015. It was revealed that the system width is correspondingly essential. This infers that, in addition to deepness, width is a vital element in evolving learning viewpoints. It is revealed that neural systems with ReLU activation tasks must be extensive enough to maintain a general estimation property while also up surging in depth.

One important subject with deep neural network designs is that several layers may fail to acquire appreciated features. Even though stacking countless layers (raising depth) may learn different feature illustrations, it does not always improve the NN's learning capacity. Moreover, any deep system cannot illogically estimate a class of constant works on a compact set if the system's maximum width is not superior to the input measurement. Therefore, the study emphasis changed from deep and narrow designs to wide and thin designs to address this subject.

2.2.6 Exploitation-Based Feature-Map (ChannelFMap) CNNs

Because of its capability to execute categorized learning and spontaneous feature removal, CNN has attracted much attention in computer vision challenges. The execution of organization, subdivision, and recognition components is deeply determined by feature selection. CNN chooses features enthusiastically by regulating the burdens connected with a kernel, also identified as a mask.

Additionally, several feature removal stages are used in CNN to excavate many kinds of features. On the other hand, some feature maps have little or no importance in object discernment. Substantial feature arrays may deliver a sound effect, triggering the system to overfit. This infers that, in addition to system designing, the choice of feature maps can play an indispensable part in amassing system overview. Feature maps and network languages are often employed interchangeably in the study.

2.2.7 CNNs that are Based on Attention

Various stages of concept play an indispensable part in influencing the NN's discernment power. Diverse pyramids of concepts centred on qualities pertinent to image localization and detection play a vital role in knowledge. This influence is recognized as attention in the human visual scheme. Humanoids can perceive any scene by incorporating half-finished glimpses of it and concentrating on conscript-relevant features. This method concentrates on definite areas and understands several clarifications of objects at a definite location, therefore developing visual configuration capture. RNN and LSTM integrate an additional or fewer similar analyses.

RNN and LSTM systems employ responsiveness elements as advanced features, and the novel analysts are biased based on their repetition in previous rounds. The theory of responsiveness in the convolutional neural network is employed by countless researchers to develop illustrations and overwhelm computational restrictions. This idea of responsiveness also adds to CNN becoming smart enough to differentiate objects even in hectic backgrounds and compound situations.

2.2.8. Dimension-Based CNN

The typical convolutions layer encrypts both channel-wise and three-dimensional data concurrently, however, it is computationally exorbitant. The efficacy of regular convolutions was improved by the introduction of independent (or depth-wise separable) convolutions., which encrypt three-dimensional and channel-wise data distinctly using point-wise and depth-wise complications, correspondingly. This factorization is far more effective, nonetheless, it places a substantial computational weight on point-wise complications, making them a computational blockage.

Road and traffic signs must be appropriately set up in the crucial locations and an inventory of them is idyllically required to help guarantee suitable modernizing and conservation. An automatic way of noticing and recognizing traffic symbols can have an important influence on this objective by providing a swift technique for spotting, categorizing and logging symbols. This technique aids in improving the inventory correctly and unfailingly.

Once this is completed, the recognition of damaged or concealed symbols becomes easier for the humanoid operator. Road and traffic sign detection is the arena of learning that can be employed to help the improvement of an inventory scheme (for which real-time detection is not mandatory) or help the advancement of an in-car advisory scheme (when real-time detection is essential). Both road symbol inventory and road sign detection are concerned with traffic symbols, face similarity problems and use automatic discovery and detection. A road and traffic symbol recognition scheme could in standard be advanced as part of an

Intelligent Transport System (ITS) that uninterruptedly monitors the driver, the automobile, and the road in order, for instance, to notify the driver in time about imminent judgment points concerning steering and possibly dangerous traffic circumstances.

2.2.9 Traffic Sign Detection System and Road Accidents

Road accidents are the main reason for the death of young persons between 15 and 29 years old. Among 20 to 50 million individuals are incapacitated every year, while 1.3 million die owing to road accidents, of which 91% occur in low and average-income nations. Due to various motives which comprise driver training and behaviour, law enforcement, and lack of suitable road set-up. On the other hand, technology can also play a significant part in driverhelp schemes that contribute to the watchfulness of the driver and improved driving behaviours. Most road accidents happen in city areas, particularly at road connections and crossroads. Statistics for road accidents indicate that a substantial amount occurred at road connections. For example, 22% in the USA, 58.7% in Japan in 1995, 13.75% in Ecuador in 2015 and 9.22% in Chile in 2014. Consequently, the significance of building systems for road connection recognition, which contrasts other features like pedestrian recognition, lanetracking and driver sleepiness or interruption has not gotten enough consideration[218,219].

Previous studies have emphasised pavement segmentation to recognize intersections by analysing the continuousness and curvature of the road restrictions. On the other hand, obstructions in city environs make the investigation of limits hard. Consequently, the application of an advanced driving assistance system (ADAS). necessitates a component proficient in identifying road symbols in general, and explicitly those established at junctures.

Road sign recognition using detectible-spectrum cameras may take diverse methods. Certain devices implement character categorization methods that involve utilizing a sliding-window technique to compute features across diverse overlapping regions. These extracted features are then input into a previously trained classifier for further classification or processing.. The disadvantage of this approach is that various sites, as well as balances, must be verified using classifiers that may require computationally challenging teaching stages.

More modern approaches frame a two-step method, wherein applicant or application areas are calculated mainly by certain "class-agnostic" subdivision procedures, for instance digging out collections of pixels that combine some distinctiveness without inevitably recognising whether they rightly fit into the same group of items. In a subsequent phase, some organization or judgement procedure is utilized to finish the recognition determining if some groups of articles required are there or not. The systems recommended in research can be seen among current methods for traffic symbol recognition using district scheme approaches together with classifiers. The most modern methods of subdivision and organization engaged in traffic symbol recognition are discussed below.

From the perspective of road symbol recognition, blob generation and colour examination are the key procedures engaged to section districts of attention. Exceptional labourers have been positioned to create the colour-built subdivision healthy to great disparities in light as well as climate situations. RGB was changed into grayscale pictures using the red as well as the blue mechanisms and scientifically acquired thresholds to produce ROIs[229]. Researchers employed 3 colour places derivative from the RGB, the main to highpoint traffic symbols by a prevalence of blue and red colours, the subsequent unit is for symbols with penetrating red and the third one for the lively blue[230]. Researchers have constructed the Gaussian space (EEλEλλ), where items subjugated by the green-red and blue-yellow colours are emphasized. The predesigned areas are one by one changed to standardized values $C\lambda = E\lambda/E$ and $C\lambda\lambda = E\lambda\lambda/E$, which are fed to a k-means collecting. to produce the ROIs. Researchers performed two RGB-based chromatic filters for ROIS production, one for symbols that have a red colour frequency, and an additional sieve for red-yellow prevalence; in equal conditions, thresholds are distinguished in relation to mean and differences. Researchers have employed the L*a*b* space to sense symbols wherein the blue, green, yellow and red colours are controlled.

Established on the k-means bunching algorithm, the writers construct a classifier that engages the a* and b* mechanisms. Researchers employ the H and S mechanisms of the HSV space to train a categorizer and device the colour subdivision that produces ROIs[236]. Further lately, researchers have employed the H element of the HSI space, in which the traffic symbols are emphasized to construct a grayscale picture where a set of ROIs is produced[237]. Researchers employed the HSV sieve times to produce an array of ROIs. In conclusion, researchers used three dissimilar item application approaches (Selective Search, Edge Boxes and BING) and convolutional neural systems for organisation, attaining an exactitude of 88% on average.

Recognition

This phase characteristically uses character classifiers and consequently necessitates a character depicter and an organization procedure. One of the furthermost common character depicters is the histogram of oriented gradients (HOG), which makes available data around items' figures. Contemporary researches in traffic symbol recognition use the HOG descriptor[127]. Researchers use the PHOG depicter, a variation of the HOG descriptor[128]. Other depicters are founded on the distinct Fourier transform, the Hough transform, the SURF technique, the rates of the neighbouring resolutions in an ROI, or predetermined outline depicters for rudimentary figures (spherical, trilateral, or quadrilateral)[130].

Regarding categorizers, the majority of the novel researchers in road symbol recognition engage SVM (support vector machine) categorizers; see for instance[128,129,130]. Another general organization method depends on non-natural neural networks (NN). For instance, contemporary research by researchers associates an NN- categorizers with ELM (Extreme Learning Machine), and Researchers depend on Multilayer Perceptron (MLP)[130]. The modest k-NN (k-nearest neighbours) algorithm is engaged in the road symbol recognition technique suggested in robust road symbol detection with trait abstraction and k-NN cataloguing methods[128].

In recent times, Deep Learning methods have been employed for instantaneous recognition as well as the detection of road symbols. Convolutional Neural Networks (CNN) is also engaged in countless of the major modern researches which recommend new designs for programmed symbol recognition[243]. Other stratagems, like the one engaged by researchers depend on an amalgamation of DBM (Deep Boltzmann Machine) and CCA (Canonical Correlation Analysis) for trait abstraction as well as organization. Researchers have also investigated RBNN (Radial Basis Neural Networks) for classification[130].

The core road symbol databanks match the following nations: Germany, United Kingdom, Spain, Japan, China and Malaysia. Every nation has its guidelines and principles regarding road symbols, divided into supervisory, prevention and data classifications. In general, they don't trail the Vienna Convention-Complaint for road symbols. Therefore nation-precise databanks are essential for the advancement of road symbol recognition schemes. On the other hand, there is an absence of databanks with road symbols in Nigeria.

2.3 Review of Related Works

In a work on detecting traffic light colour using a machine learning approach. Using HSV colour representation, our approach is to extract features based on an area of X×X pixels. The traffic light colour model is then created by applying a learning algorithm on a set of examples of features representing pixels of traffic and non-traffic light colours. The learned model is then used to classify whether an area of pixels contains traffic light colour or not. Evaluation of this approach reveals that it significantly improves the detection performance over the one based on the value-range colour segmentation technique[98].

In a work that integrates a new robust machine learning-based solution by combining a Convolutional Neural Network (CNN) with computer vision techniques to achieve a realtime traffic light detector. The proposed detection and recognition algorithm is capable of recognizing traffic lights on low-power small-form platforms, which are lightweight, portable, and can be mounted on Automatic Vehicles (AVs) in daylight scenarios. The Laboratory for Intelligent and Safe Automobiles (LISA) open-source dataset is utilized with augmentation methods to increase the accuracy of the solution. The proposed approach achieves 93.42% accuracy at a speed of 30.01 Frames Per Second (FPS) on an NVIDIA

Jetson Xavier platform without using hardware accelerators such as Field-Programmable Gate Array (FPGA). This solution is expected to promote the quicker adoption and wider deployment of AVs by increasing the chances of avoiding crashes and ultimately saving lives[126].

2.4 Chapter Summary and Gap in Literature

The literatures connected to research are studied in this chapter. The literature related to research is studied, an outline of computer vision is presented, and the related methods of computer vision are introduced, including image acquisition, data preprocessing, feature extraction, detection/segmentation, detection and decision-making. Depending on how road symbols are subdivided, TSD methods can be separated into appearance-built techniques, geometry-built techniques and machine learning-built techniques. For TSR, machine learning techniques such as SVM, RF and CNN have been extensively used, and outstanding outcomes have been accomplished.

Many of the current LPR devices are considered for certain regulated surroundings, or else, the execution may be principally abridged. To strategise a vigorous and active LPR scheme, it should be capable of being employed in an extensive variety of atmospheres, and senseauthorized plates with great exactitude. Over the last ten years, with the growth of authoritative computer vision techniques, script recognition has swiftly advanced.

Nevertheless, there exist complications like Chinese script recognition and multi-oriented and crisscrossed script outline recognition. The advancement of CNNs has been studied. As the swelling of the deepness of systems, CNNs facial training challenges like slope disappearance, slope ignition and dilapidation. With the assistance of naturalization as well as learning techniques, the challenges have been finely reduced. On the other hand, they are not completely resolved, novel techniques are projected to solve these challenges.

Endnotes

1. Z Chen; X. Huang. *Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking.* IEEE Intell. Transport Syst. Mag., vol. 8, pp. 28-42, 2019.

2. G Siogkas; E. Skodras; E. Dermates. *Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information* in International Conference on Computer Vision Theory and Applications, 2019.

3. J Levinson; J. Askeland; J. Dolson; S. Thrun. *Traffic light mapping, localization, and state detection for autonomous vehicles* in 2020 IEEE International Conference on Robotics and Automation, 2020.

4. M Omachi and S. Omachi. *Traffic light detection with color and edge information* in 2019 2nd IEEE International Conference on Computer Science and Information Technology, 2019.

5. V.John, K Yoneda, B Qi, Z Liu, S Mita. T*raffic light recognition in varying illumination using deep learning and saliency map* in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2019.

6. R de Charette, F Nashashibi. Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates in 2019 IEEE Intelligent Vehicles Symposium, 2019.

7. G Trehard; E Pollard; B Bradai; F Nashashibi. *Tracking both pose and status of a traffic light via an Interacting Multiple Model filter* in 17th International Conference on Information Fusion (FUSION), 2020

8. M Hirabayashi, A Sujiwo, A Monrroy A, Kato S, M Edahiro. *Traffic light recognition using high-definition map features*. Robotics and Autonomous Systems. 2019 Jan 1;111:62-72.

9. K Behrendt; L. Novak; R. Botros, "*A deep learning approach to traffic lights: Detection, tracking, and classification,*" 2017.

10. A Pon, O Adrienko, A Harakeh, SL Waslander. *A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection*," in 2018 15th Conference on Computer and Robot Vision (CRV), 2018.

11. CN Anagnostopoulos, IE Anagnostopoulos, V Loumos, E Kayafas. *A license platerecognition algorithm for intelligent transportation system applications*. IEEE Transactions on Intelligent Transportation Systems. 2018 Sep 6;7(3):377-92.

12. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," IEEE Transactions on Intelligent Transportation Systems, vol. 9, pp. 377–391, Sept 2008.

13. C. N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," IEEE Intelligent Transportation Systems Magazine, vol. 6, pp. 59–67, Spring 2019.

14. CN Anagnostopoulos CN. *License plate recognition: A brief tutorial. IEEE Intelligent Transportation Systems Magazine.* 2014 Jan 20;6(1):59-67.

15. S Du, M Ibrahim, M Shehata, and W Badawy. *Automatic license plate recognition (ALPR): A state-of-the-art review,*" IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, pp. 311–325, Feb 2013

16. N.F Yaacob, N Rusli, S.N Bohari. *A review analysis of accident factor on road accident cases using Haddon Matrix approach*. In Proceedings of the Second International Conference on the Future of ASEAN (ICoFA) 2017–Volume 2: Science and Technology 2018 (pp. 55-65). Springer Singapore.

17. E.A Yihun. *The causal Attribution of road Traffic Accident among victims and drivers in case of Gondar town, North West Ethiopia*. **IJASSH**. 2019 Dec 12.

18. J.A Andeta. *Road-traffic accident prediction model : Predicting the Number of Casualties [Internet] [Dissertation]. 2021*. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-20146

19. B. Kumeda, F Zhang, F Zhou, S Hussain, A Almasri, M Assef. *Classification of road traffic accident data using machine learning algorithms*. In2019 IEEE 11th international conference on communication software and networks (ICCSN) 2019 Jun 12 (pp. 682-687). IEEE.

20. P.A Nandurge, N.V Dharwadkar. *Analyzing road accident data using machine learning paradigms*. In2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2017 Feb 10 (pp. 604-610). IEEE

21. Z.E Abou Elassad, H Mousannif, H Al Moatassime. *A real-time crash prediction fusion framework: An imbalance-aware strategy for collision avoidance systems*. Transportation research part C: emerging technologies. 2020 Sep 1;118:102708.

22. H Park, A Haghani. *Real-time prediction of secondary incident occurrences using vehicle probe data.* Transportation Research Part C: Emerging Technologies. 2016 Sep 1;70:69-85.

23. M Zahid, Y Chen, S Khan, A Jamal, M Ijaz, T Ahmed. *Predicting risky and aggressive driving behavior among taxi drivers: do spatio-temporal attributes matter?* **International journal of environmental research and public health**. 2020 Jun;17(11):3937.

24. D.H Kim DH, Ramjan LM, Mak KK. Prediction of vehicle crashes by drivers' characteristics and past traffic violations in Korea using a zero-inflated negative binomial model. Traffic injury prevention. 2016 Jan 2;17(1):86-90.

25. Wang C, Liu L, Xu C, Lv W. Predicting future driving risk of crash-involved drivers based on a systematic machine learning framework. International journal of environmental research and public health. 2019 Feb;16(3):334.

26. Onyemaechi NO, Ofoma UR. The public health threat of road traffic accidents in Nigeria: A call to action. Annals of medical and health sciences research. 2016;6(4):199-204.

27. Apeagee BB, Haaor SA. *A Logistic Regression Model of Road Traffic Fatalities in Benue State: Imp*lication to Public Health. Nigerian Annals of Pure and Applied Sciences. 2020 Nov 15:3(3a):46-52.

28. https://www.statista.com/statistics/1296025/total-number-of-road-casualties-in-nigeria/

29. T.O Helly. *Appraisal of Road Accidents in Akungba Akoko, Ondo State, Nigeria.*

30. S.K Chinnamgari. *R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5*. Packt Publishing Ltd; 2019 Jan 14.

31. H Hihn, D.A Braun. *Specialization in Hierarchical Learning Systems: A Unified Information-theoretic Approach for Supervised, Unsupervised and Reinforcement Learning. Neural Processing Letters*. 2020 Dec;52(3):2319-52.

32. J.A Sidey-Gibbons, C.J Sidey-Gibbons. *Machine learning in medicine: a practical introduction*. BMC medical research methodology. 2019 Dec;19:1-8.

33. I.H Sarker. *Machine learning: Algorithms, real-world applications and research direction*s. SN computer science. 2021 May;2(3):160.

34. Y Bengio, A Lodi, A Prouvost. *Machine learning for combinatorial optimization: a methodological tour d'horizon*. **European Journal of Operational Research**. 2021 Apr 16;290(2):405-21.

35. T Bokaba, W Doorsamy, B.S Paul. *Comparative study of machine learning classifiers for modelling road traffic accidents*. Applied Sciences. 2022 Jan;12(2):828.

36. H Saigo, S Nowozin, T Kadowaki, T Kudo, K Tsuda. *gBoost: a mathematical programming approach to graph classification and regression*. Machine Learning. 2009 Apr;75:69-89.

37. A.V Konstantinov, L.V Utkin. *Interpretable machine learning with an ensemble of gradient boosting machines*. Knowledge-Based Systems. 2021 Jun 21;222:106993.

38. M Awad, & R Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers* (p. 268). Springer nature, 2015.

39. Y Guo, TvHastie, R Tibshirani. Regularized linear discriminant analysis and its application in microarrays. Biostatistics. 2007 Jan 1;8(1):86-100.

40. S Ghosh, A Dasgupta, A Swetapadma. *A study on support vector machine based linear and non-linear pattern classification*. In2019 International Conference on Intelligent Sustainable Systems (ICISS) 2019 Feb 21 (pp. 24-28). IEEE

41. I Ahmad, M Basheri, M.J Iqbal, A Rahim. *Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection*. IEEE access. 2018 May 30;6:33789-95.

42. X Zhai, M Chen, W Lu. *Fuel ratio optimization of blast furnace based on data mining. ISIJ International*. 2020 Nov 15;60(11):2471-6.

43. M Sabzekar, S.M Hasheminejad. *Robust regression using support vector regressions. Chaos, Solitons & Fractals*. 2021 Mar 1;144:110738.

44. P Tsirikoglou, S Abraham, F Contino, C Lacor, G Ghorbaniasl. *A hyperparameters selection technique for support vector regression models. Applied Soft Computing*. 2017 Dec 1;61:139-48.

45. D.A Pisner, D.M *Schnyer. Support vector machine. InMachine learning 2020* Jan 1 (pp. 101-121). Academic Press.

46. M.S Adnan, S Zaidi, P Bhargava. *A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. Construction and building materials.* 2020 Jul 10;248:118475.

47. M Wang, S Jia, E Chen, S Yang, P Liu, Z Qi. *A derived least square fast learning network model. Applied Intelligence*. 2020 Dec;50:4176-94.

48. S Mangalathu, S.H Hwang, E Choi, J.S Jeon. *Rapid seismic damage evaluation of bridge portfolios using machine learning techniques.* Engineering Structures. 2019 Dec 15;201:109785.

49. S Afraei, K Shahriar, S.H Madani. *Developing intelligent classification models for rock burst prediction after recognizing significant predictor variables, Section 2: Designing classifiers. Tunnelling and Underground Space Technology*. 2019 Feb 1;84:522-37.

50. L Ali, S.U Khan, N.A Golilarz, I Yakubu, I Qasim, A Noor, R Nour. *A feature-driven decision support system for heart failure prediction based on statistical model and Gaussian naive bayes*. Computational and Mathematical Methods in Medicine. 2019 Nov;2019.

51. D Bassi, H Singh. *Optimizing hyperparameters for improvement in software vulnerability prediction models.* InAdvances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2022 2022 Jul 28 (pp. 533-544). Singapore: Springer Nature Singapore.

52. V Mishra, S.M Agarwal, N Puri. *Comprehensive and comparative analysis of neural network*. **International Journal of Computer Application**. 2018;2(8):126-37

53. J.M Johnson, T.M Khoshgoftaar. *Survey on deep learning with class imbalance*. **Journal of Big Data**. 2019 Dec;6(1):1-54.

54. A Darwish, D Ezzat, AE Hassanien. *An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis*. Swarm and evolutionary computation. 2020 Feb 1;52:100616.

55. Y Ren, L Zhang, P.N Suganthan. *Ensemble classification and regression-recent developments, applications and future directions.* **IEEE Computational intelligence magazine**. 2016 Jan 12;11(1):41-53.

56. A Mosavi, F Sajedi Hosseini, B Choubin, M Goodarzi, A.A Dineva, E Rafiei Sardooi. *Ensemble boosting and bagging based machine learning models for groundwater potential prediction*. Water Resources Management. 2021 Jan;35:23-37.

57. N.A Mashudi, N Ahmad, N.M Noor. *Classification of adult autistic spectrum disorder using machine learning approach*. **IAES International Journal of Artificial Intelligence.** 2021 Sep 1;10(3):743.

58. A Taherkhani, G Cosma, T.M McGinnity. *AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning*. Neurocomputing. 2020 Sep 3;404:351-66.

59. E Suganya, CRajan. *An adaboost-modified classifier using particle swarm optimization and stochastic diffusion search in wireless IoT networks*. Wireless Networks. 2021 May;27:2287-99.

60. R Sibindi, R.W Mwangi, A.G Waititu. *A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices*. Engineering Reports. 2022:e12599.

61. Yang X, Wang Y, Byrne R, Schneider G, Yang S. Concepts of artificial intelligence for computer-assisted drug discovery. Chemical reviews. 2019 Jul 11;119(18):10520-94..

62. C Zhang, X Lei, L Liu. *Predicting metabolite–disease associations based on LightGBM model*. Frontiers in Genetics. 2021 Apr 13;12:660275.

63. O.I Abiodun, A Jantan, K.V Dada, N.A Mohamed, H Arshad. *State-of-theart in artificial neural network applications: A survey*. Heliyon 4 e00938. DOI: 10.1016/j.heliyon.2018. e00938.

64. W Qi, H Su, C Yang, G Ferrigno, E De Momi, A Aliverti. *A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone*. Sensors. 2019 Aug 29;19(17):3731.

65. A Golab, E Gooya, A Falou, M Cabon. *A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP)*. Decision Science Letters. 2022;11(4):407-18..

66. A.B Shaik, S Srinivasan. *A brief survey on random forest ensembles in classification model*. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2 2019 (pp. 253-260). Springer Singapore.

67. I Reis, D Baron, S Shahaf. *Probabilistic random forest: A machine learning algorithm for noisy data sets*. **The Astronomical Journal.** 2018 Dec 20;157(1):16.

68. B.O Yigin, O Algin, G Saygili. *Comparison of morphometric parameters in prediction of hydrocephalus using random forests*. Computers in Biology and Medicine. 2020 Jan 1;116:103547.

69. N.M Abdulkareem, A.M Abdulazeez. *Machine learning classification based on Radom Forest Algorithm: A review*. **International Journal of Science and Business**. 2021;5(2):128-42

70. D Denisko, M.M Hoffman. *Classification and interaction in random forests. Proceedings of the National Academy of Sciences*. 2018 Feb 20;115(8):1690-2.

71. L.V Utkin, M.S Kovalev, F.P Coolen. *Imprecise weighted extensions of random forests for classification and regression*. Applied Soft Computing. 2020 Jul 1;92:106324.

72. L Demidova, M Ivkina. *Defining the Ranges Boundaries of the Optimal Parameters Values for the Random Forest Classifier.* In 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA) 2019 Nov 20 (pp. 518-522). IEEE

73. M.A Sulaiman. *Evaluating data mining classification methods performance in Internet of things applications*. **Journal of Soft Computing and Data Mining**. 2020 Dec 6;1(2):11-25.

74. M.L Kolhe, S Tiwari, M.C Trivedi & K.K Mishra (Eds.). *Advances in Data and Information Sciences: Proceedings of ICDIS 2019* (Vol. 94). Springer Singapore. https://doi.org/10.1007/978- 981-15-0694-9

75. K Gajowniczek, I Grzegorczyk, T Ząbkowski, C Bajaj. *Weighted random forests to improve arrhythmia classification*. Electronics. 2020 Jan 3;9(1):99.

76. Y Sun, Y Li, Q Zeng & Y Bian. *Application Research of Text Classification Based on Random Forest Algorithm.* 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 370–374. https://doi.org/10.1109/AEMCSE50948.2020.00086

77. S Koley, A.K Sadhu, P Mitra, B Chakraborty, C Chakraborty. *Delineation and diagnosis of brain tumors from post contrast T1-weighted MR images using rough granular computing and random forest*. Applied Soft Computing. 2016 Apr 1;41:45365.

78. B Charbuty, A Abdulazeez. *Classification based on decision tree algorithm for machine learning*. **Journal of Applied Science and Technology Trends**. 2021 Mar 24;2(01):20-8.

79. Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "*Predicting diabetes mellitus with machine learning techniques*," Frontiers in genetics, vol. 9, p. 515, 2018

80. T. A. Assegie and P. S. Nair, "*Handwritten digits recognition with decision tree classification: a machine learning approach*," **International Journal of Electrical and Computer Engineering**, vol. 9, no. 5, p. 4446, 2019

81. F. De Felice, Crocetti D, Parisi M, Maiuri V, Moscarelli E, Caiazzo R, Bulzonetti N, Musio D, Tombolini V."Decision tree algorithm in locally advanced rectal cancer: an example of over-interpretation and misuse of a machine learning approach," **Journal of Cancer Research and Clinical Oncology**, vol. 146, no. 3, pp. 761–765, 2020

82. I. H. Sarker, A. Colman, J. Han, A. I. Khan, Y. B. Abushark, and K. Salah, "*Behavdt: a behavioral decision tree learning to build usercentric context-aware predictive model*," Mobile Networks and Applications, vol. 25, no. 3, pp. 1151–1161, 2020

83. X. Hu, C. Rudin, and M. Seltzer, "Optimal sparse decision trees," in Advances in Neural Information Processing Systems, 2019, pp. 7267– 7275

84. S. Patil and U. Kulkarni, "*Accuracy Prediction for Distributed Decision Tree using Machine Learning approach*," in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Apr. 2019, pp. 1365–1371, DOI: 10.1109/ICOEI.2019.8862580

85. D. Hussain, M. A. Al-Antari, M. A. Al-Masni, S.-M. Han, and T.-S. Kim, "*Femur segmentation in DXA imaging using a machine learning decision tree,*" **Journal of Xray Science and Technology**, vol. 26, no. 5, pp. 727–746, 2018

86. N. Linty, A. Farasin, A. Favenza, and F. Dovis, "*Detection of GNSS Ionospheric Scintillations Based on Machine Learning Decision Tree*," IEEE Transactions on Aerospace and Electronic Systems, vol. 55, no. 1, pp. 303–317, Feb. 2019, DOI: 10.1109/TAES.2018.2850385.

87. W. Kuang, Y. Chan, S. Tsang, and W. Siu, "*Machine Learning-Based Fast Intra Mode Decision for HEVC Screen Content Coding via Decision Trees,*" IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 5, pp. 1481–1496, May 2020, DOI: 10.1109/TCSVT.2019.2903547.

88. I. Ramadhan, P. Sukarno, and M. A. Nugroho, "*Comparative Analysis of K-Nearest Neighbor and Decision Tree in Detecting Distributed Denial of Service,*" in 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, Jun. 2020, pp. 1–4, DOI: 10.1109/ICoICT49345.2020.9166380.

89. V. M. E. Batitis, M. J. G. Caballes, A. A. Ciudad, M. D. Diaz, R. D. Flores, and E. R. E. Tolentin, "*Image Classification of Abnormal Red Blood Cells Using Decision Tree Algorithm,*" in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Mar. 2020, pp. 498–504, DOI: 10.1109/ICCMC48092.2020.ICCMC-00093.

90. Y. Zhang, J. Liu, Z. Zhang, and J. Huang, "*Prediction of Daily Smoking Behavior Based on Decision Tree Machine Learning Algorithm*," in 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Jul. 2019, pp. 330–333, DOI: 10.1109/ICEIEC.2019.8784698

91. S. Nandhini and J. M. K.S, "*Performance Evaluation of Machine Learning Algorithms for Email Spam Detection*," in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Feb. 2020, pp. 1–4, DOI: 10.1109/ic-ETITE47903.2020.312.

92. A. I. Taloba and S. S. I. Ismail, "*An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection*," in 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Dec. 2019, pp. 99–104, DOI: 10.1109/ICICIS46948.2019.901475

93. M. O. Arowolo, M. Adebiyi, A. Adebiyi, and O. Okesola, "*PCA Model For RNA-Seq Malaria Vector Data Classification Using KNN And Decision Tree Algorithm,*" in 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Mar. 2020, pp. 1–8, DOI: 10.1109/ICMCECS47690.2020.240881

94. S. Pathan, P. Kumar, R. Pai, and S. V. Bhandary, "*Automated detection of optic disc contours in fundus images using decision tree classifier*," Biocybernetics and Biomedical Engineering, vol. 40, no. 1, pp. 52–64, 2020

95. A. A. Nagra, Han F, Ling QH, Abubaker M, Ahmad F, Mehta S, Apasiba AT."*Hybrid self-inertia weight adaptive particle swarm optimisation with local search using C4. 5 decision tree classifier for feature selection problems,*" Connection Science, vol. 32, no. 1, pp. 16–36, 2020.

96. A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, *"A novel hierarchical intrusion detection system based on decision tree and rules-based models*,"

in 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2019, pp. 228–233.

97. M. Li, H. Xu, and Y. Deng, "*Evidential decision tree based on belief entropy," Entropy*, vol. 21, no. 9, p. 897, 2019

98. P. Sathiyanarayanan, S. Pavithra, M. S. SARANYA, and M. Makeswari, "*Identification of Breast Cancer Using The Decision Tree Algorithm*," in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1–6.

99. C Iwendi, A.K Bashir, A Peshkar, R Sujatha, J.M Chatterjee, S Pasupuleti, R Mishra, S Pillai, O Jo. *COVID-19 patient health prediction using boosted random forest algorithm*. Frontiers in public health. 2020 Jul 3; 8:357.

100.F Zhang, X Yang. *Improving land cover classification in an urbanized coastal area by random forests: The role of variable selection*. Remote Sensing of Environment. 2020 Dec 15; 251:112105.

101.J.F Saenz-Cogollo, M Agelli. *Investigating feature selection and random forests for inter-patient heartbeat classification*. Algorithms. 2020 Mar 25;13(4):75.

102.R Shiroyama, M Wang, C Yoshimura. *Effect of sample size on habitat suitability estimation using random forests: a case of bluegill, Lepomis macrochirus*. **InAnnales de Limnologie-International Journal of Limnology** 2020 (Vol. 56, p. 13). EDP Sciences.

103.D Wang, AX Zhu. *Soil mapping based on the integration of the similarity-based approach and random forests*. Land. 2020 May 29;9(6):174.

104.D Yeap, M.M McCartney, M.Y Rajapakse, A.G Fung, N.J Kenyon, C.E Davis. *Peak detection and random forests classification software for gas chromatography/differential mobility spectrometry (GC/DMS) data.* Chemometrics and Intelligent Laboratory Systems. 2020 Aug 15; 203:104085.

105.A Azhar, N.M Ariff, M.A Bakar, A Roslan. *Classification of driver injury severity for accidents involving heavy vehicles with decision tree and random forest*. Sustainability. 2022 Mar 30;14(7):4101.

106.A da Cruz Figueira, C.S Pitombo, A.P Larocca. *Identification of rules induced through decision tree algorithm for detection of traffic accidents with victims: A study case from Brazi*l. Case studies on transport policy. 2017 Jun 1;5(2):200-7.

107.A Saracoglu, H Ozen. *Estimation of traffic incident duration: A comparative study of decision tree models*. **Arabian Journal for Science and Engineering**. 2020 Oct;45(10):8099-110.

108.G Shiran, R Imaninasab, R Khayamim. *Crash severity analysis of highways based on multinomial logistic regression model, decision tree techniques, and artificial neural network: a Modeling comparison*. Sustainability. 2021 May 18;13(10):5670.

109.M Yan, Y Shen. *Traffic accident severity prediction based on random forest.* Sustainability. 2022 Feb 2;14(3):1729.

110.K Assi, S.M Rahman, U Mansoor, N Ratrout. *Predicting crash injury severity with machine learning algorithm synergized with clustering technique: A promising protocol.* **International journal of environmental research and public healt**h. 2020 Aug;17(15):5497.

111.P Abdullah, T Sipos. *Drivers' Behavior and Traffic Accident Analysis Using Decision Tree Method*. Sustainability. 2022 Sep 9;14(18):11339.

112.S Alagarsamy, P Nagaraj, B Srikanth, C.V Krishna, G Bharath, S.S Kalyan. *A Novel Machine Learning Technique for Predicting Road Accidents*. In2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS) 2023 Feb 2 (pp. 1547-1551). IEEE.

113.A Kwok-Fai Lui, Y.H Chan, K.H Lo, W.T Cheng, H.T Cheung. *Predictive Screening of Accident Black Spots based on Deep Neural Models of Road Networks and Facilities: A Case Study based on a District in Hong Kong.* In2021 5th International Conference on Computer Science and Artificial Intelligence 2021 Dec 4 (pp. 422-428).

114.Z Ma, G Mei, S Cuomo. *An analytic framework using deep learning for prediction of traffic accident injury severity based on contributing factors*. Accident Analysis & Prevention. 2021 Sep 1; 160:106322.

115.J.S Kong, K.H Lee, O.H Kim, H.Y Lee, C.Y Kang, D Choi, S.C Kim, H Jeong, D.R Kang, T.E Sung. *Machine learning-based injury severity prediction of level 1 trauma center enrolled patients associated with car-to-car crashes in Korea.* Computers in biology and medicine. 2023 Feb 1; 153:106393.

116.S Malik, H El Sayed, M.A Khan, M.J Khan. *Road Accident Severity Prediction—A Comparative Analysis of Machine Learning Algorithms.* In2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT) 2021 Dec 12 (pp. 69-74). IEEE.

117.S Elyassami, Y Hamid, T Habuza. *Road crashes analysis and prediction using gradient boosted and random forest trees*. In2020 6th IEEE Congress on Information Science and Technology (CiSt) 2021 Jun 5 (pp. 520-525). IEEE.

118.U Fareed, U Khadam, M.M Iqbal, M.J Iqbal. *Road Accidents Investigation and Forecasting Using Data Mining Techniques.* **KIET Journal of Computing and Information Sciences**. 2023;6(1):28-49.

119.B Kumeda, F Zhang, F Zhou, S Hussain, A Almasri, M Assefa. *Classification of road traffic accident data using machine learning algorithms*. In2019 IEEE 11th international conference on communication software and networks (ICCSN) 2019 Jun 12 (pp. 682-687). IEEE.

120.T Bokaba, W Doorsamy, B.S Paul. *Comparative study of machine learning classifiers for modelling road traffic accidents*. Applied Sciences. 2022 Jan;12(2):828.

121.D Dias, J.S Silva, A Bernardino. *The prediction of road-accident risk through data mining: A case study from Setubal, Portugal*. InInformatics 2023 Jan 30 (Vol. 10, No. 1, p. 17). MDPI.

122.M Gnjatović, I Košanin, N Maček, D Joksimović. *Clustering of Road Traffic Accidents as a Gestalt Problem*. Applied Sciences. 2022 Apr 29;12(9):4543.

123.X Zhang, S Qi, A Zheng, Y Luo, S Hao. *Data-Driven Analysis of Fatal Urban Traffic Accident Characteristics and Safety Enhancement Research*. Sustainability. 2023 Feb 10;15(4):3259.

124.Y Zhang, Y Sung. *Hybrid Traffic Accident Classification Models*. Mathematics. 2023 Feb 19;11(4):1050.

125.H.M Alnami, I Mahgoub, H Al–Najada. *Highway accident severity prediction for optimal resource allocation of emergency vehicles and personnel*. In2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) 2021 Jan 27 (pp. 1231-1238). IEEE.

126.M Manzoor, M Umer, S Sadiq, A Ishaq, S Ullah, H.A Madni, C Bisogni. *RFCNN: Traffic accident severity prediction based on decision level fusion of machine and deep learning model.* IEEE Access. 2021 Sep 14; 9:128359-71.

127.J.A Sowdagur, B.T Rozbully-Sowdagur, G Suddul. *An Artificial Neural Network Approach for Road Accident Severity Prediction*. In2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC) 2022 May 25 (pp. 267-270). IEEE.

128.P Chirag, M Supreetha. *Road accident prediction and classification using machine learning*. In2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) 2022 Oct 16 (pp. 1-8). IEEE.

129. JL Binangkit, DH Widyantoro. *Increasing accuracy of traffic light color detection and recognition using machine learning*. In 2016 10th international conference on telecommunication systems services and applications (TSSA) 2016 Oct 6 (pp. 1-5). IEEE.

130.G Golcarenarenji, I Martinez-Alpiste, Q Wang, JM Alcaraz-Calero. *Robust real-time traffic light detector on small-form platform for autonomous vehicles*. **Journal of Intelligent Transportation Systems**. 2023 Apr 24:1-1.

Chapter Three

Methodology

3.1 Research Approach

For this project, the object detection method is used to find instances of traffic light signs in each picture frame of the videos being analysed. Object detection algorithms frequently use deep learning or machine learning to generate insightful results. Humans are capable of quickly identifying and locating objects of interest when viewing photos or videos. Object detection is used to automate the replication of this intelligence. The Yolo4 image detector Object of the Deep Learning Toolbox is used in this project for our image classification task. The process involves the following steps.

1. Preparing the Dataset that would be used for training and evaluation of the image detector
2. Training the Yolo4 detector object.
3. Evaluation of Detector object.

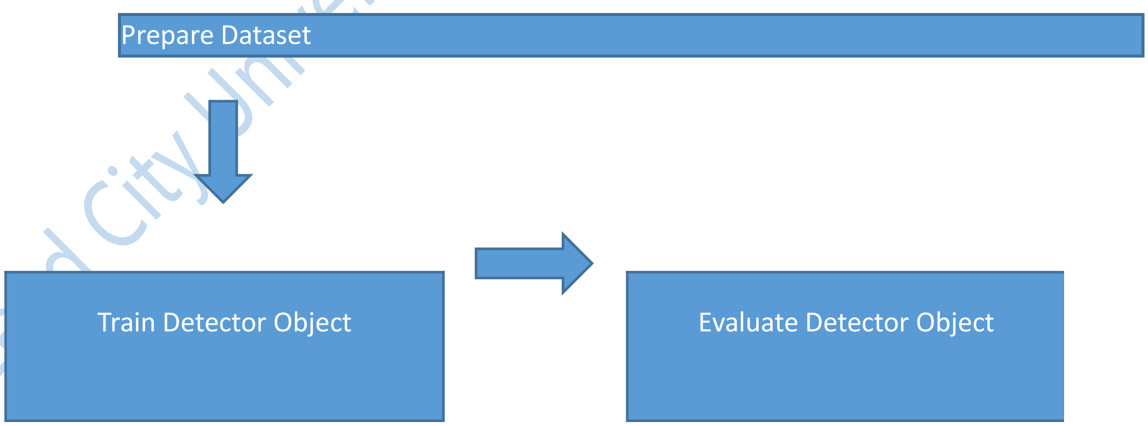


Figure 3.1: Flow Chart for Creating a Detector Model
Source: Researcher's Design, 2023

3.2 YOLOv4 Algorithm Network Structure

The feature extraction network CSPdarknet-53 is used by the YOLOv4 algorithm. The receptive field increases while the dimension of the feature map shrinks as the network gets deeper. While this is happening, the semantic features become more and more clear while the features eventually become abstract. However, as the location data gets fuzzier, it becomes impossible to precisely detect small targets. In the conscript of this, a shallow feature augmentation strategy is suggested, which combines shallow features with high-level semantic features to increase the YOLOv4 algorithm's ability to precisely place and recognize small targets.

To guarantee the viability of the shallow feature enhancement method and produce the best results without breaking its network structure, the original feature extraction network of the YOLOv4 algorithm, CSPDarknet-53, uses a CSP strategy. This study proposes two feature fusion enhancement strategies: merging features from the 11th and 127th layers while merging features from the 23rd and 117th layers after processing, and the second which entails merging the features from the 23rd layer and 127th layer and fusing the features from the 54th layer and layer 117 after processing. When putting the techniques into practice, two factors need to be taken into account. First, to integrate the 127th layer features with the 11th layer shallow features, the 127th layer features must be upsampled four times. Second, the produced features have dimensions as high as 304x304x255, making it impossible to recognize signal lights in real time even with extensive additional calculations.

To improve the shallow features, the following techniques are used: (1) high-level semantic features from the 117th layer are up-sampled; (2) shallow features from the 54th layer are spliced; (3) convolution is performed, and (4) high-level semantic features from the 127th layer are upsampled and shallow features from the 23rd layer are spliced. The first-scale feature is obtained following the convolution step. The first-scale feature is obtained following the convolution operation. The dimensions of the first-scale feature are 152×152×255, those of the second-scale feature are 76×76×255, and those of the third-scale feature are the same at

$19\times19\times255$. The detection accuracy of the majority of minor targets is enhanced while maintaining excellent detection accuracy for specific larger traffic signals. Because the enhanced algorithm preserves the CSPDarknet-53 network topology, it guarantees that some significant traffic lights continue to have excellent detection accuracy. At the same time, it combines superficial features with clearer and more specific location data with the deep features to increase the detection accuracy of traffic signals for the majority of small targets by taking into consideration the benefits of shallow features and deep features.

Figure 3.2 depicts the network architecture of the improved YOLOv4 algorithm. First, the input traffic light data needs to be processed and the image input required by the YOLOv4 algorithm is a multiple of 32. Low detection accuracy results from setting the input size too low. According to experimental verification, increasing the input size does not significantly improve detection accuracy while doing so marginally increases detection time since the calculation may be too large. As a result, as input for the entire network, the traffic light data are scaled from the size of $1280\times960\times3$ to the size of $608\times608\times3$ in three channels.

Second, the feature uses the 53-CSPDarket network feature extraction of input data, for the convolution operation, input data of sizes 3×3 and 1×1 are alternatively employed. The aforementioned technique is used to carry out shallow feature fusion based on the original YOLOv4 algorithm to avoid the issue of the dimensions of the feature graph decreasing with the deepening of the convolution depth, with the features gradually becoming abstracted and insensitive to small target detection. To increase the detection and recognition accuracy of traffic lights by the YOLOv4 algorithm, the three-scale characteristic information is created and the detection and recognition of traffic lights are completed.

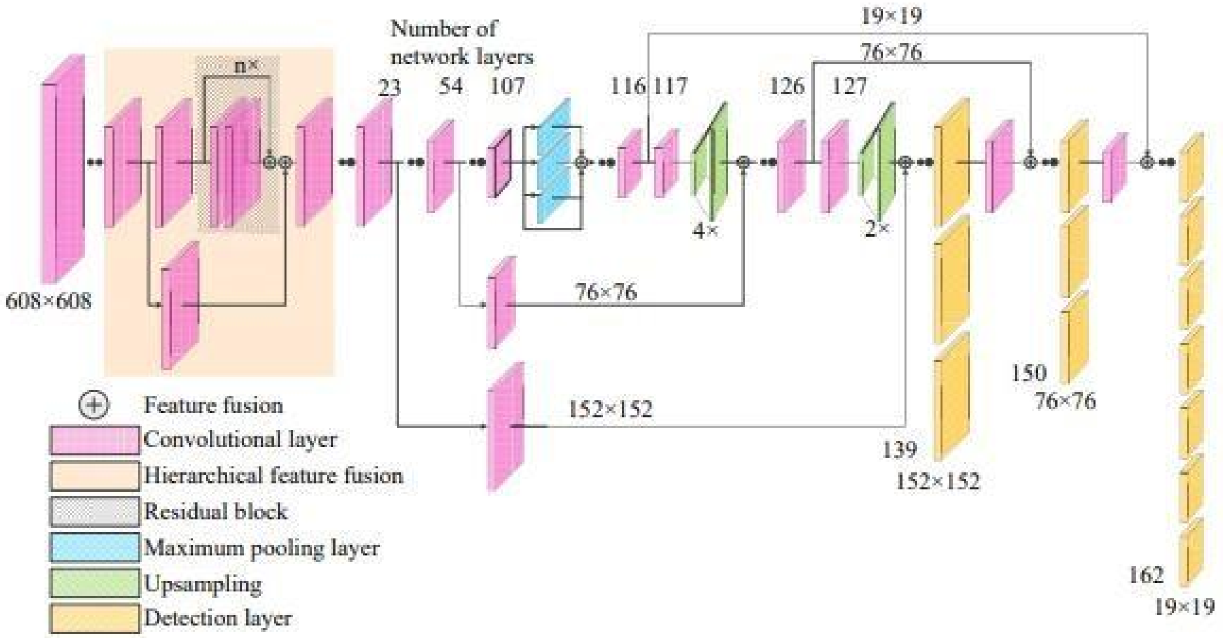


Figure 3.2: The Network Structure of the Improved YOLO4 Algorithm[100].

3.3 Prediction of Bounding Box

The precision of the bounding box is not processed during the original YOLOv4 algorithm's bounding box prediction; just the coordinate information is predicted. Consequently, it is impossible to determine from the results of research, whether the predicted bounding box coordinates were accurate. A bounding box uncertainty prediction is added to the YOLOv4 algorithm to forecast the uncertainty of each coordinate information and enhance the precision of the predicted bounding box, thereby strengthening the YOLOv4 algorithm's capacity to detect traffic lights.

Bounding box regression is used in the original YOLOv4 technique to extract the coordinate information (and) of the bounding box's centre point and the size information (and h) of the bounding box. The dependability of the bounding box cannot be reflected by these parameters; they can only offer the position and size of the bounding box. In this study, the calculation of the prediction frame's uncertainty is combined with the computation of the confidence level, and it is modelled using a single Gaussian model of , , , and $_h$. Equation is used to express the Gaussian model (4).

$$(\mid) \quad = (; \ (,\Sigma x)( ) \qquad (4)$$

Here, ( ) is the mean function and Σ(x) is the variance function.

The positional information of the anticipated bounding box is modelled as the mean and variance to forecast the bounding box's uncertainty. The bounding box's outputs are , Σ, , Σ, , Σ, $_h$, Σ_h . The Gaussian parameters of , , , and $_h$ are preprocessed using the Sigmoid function by Equations (5) – (7), due to the detection layer structure in the network.

$$(\hat{\ }_y)\mu t$$

$$(\hat{\ }=), = \quad = \sigma(\hat{\Sigma} t_y)$$

$$\Sigma = \Sigma, \Sigma(\hat{\ }\ ) \quad = \sigma(\hat{\Sigma} t_h)$$

$$\Sigma = \Sigma, \Sigma_h(\hat{\ }\ )$$

$$( ) = \frac{1}{1+\exp(-x)} \qquad (6)$$

The predicted coordinate of the bounding box is the average value of each coordinate in the detection layer, and each variance is the uncertainty of the corresponding coordinate. The sigmoid function is used to process and – which indicates the centre coordinates of the bounding box in the grid – as values that range from 0 to 1, as seen in equation (5). In YOLOv4, and $_h$ are represented by and $_h$. The Sigmoid function is not used for processing since their scale changes can exceed the grid size of the centre point of the bounding box. The Sigmoid function converts each coordinate's variance in Equation (6) to a value between 0 and 1. In the Gaussian distribution, the variance determines how much the distribution will change, i.e., the greater the variance is the greater the change in the distribution. Since each variance represents the uncertainty of its corresponding coordinate, the closer the processed variance's distance is from 0 the lesser uncertainty is present and how reliable the predicted bounding box is; a distance close to 1 determines how much uncertainty is present and how unreliable the predicted bounding box is.

Equation 7 is used to calculate how the bounding box confidence calculation method changes during prediction:

$$onr = \Pr(r) \times {I^{P}}_{Pn}{}^{h} \times (1 - rPeu_{P}) \tag{7}$$

where $rPeu_P$ is the average uncertainty for each coordinate information.

If a bounding box of the same size is used to detect a small target as opposed to a large target, the variance will be higher since the small target takes up fewer pixels, leading to a low confidence in the bounding box, and the bounding box can be easily abandoned. Because the variance of the bounding box with a smaller size is smaller in small target detection and the mean value derived for the uncertainty will be smaller, the predicted bounding box has a better degree of confidence.

3.4 Performance Analysis of YOLOv4 Algorithm

The pictures in the VOC2007 dataset have varying pixel sizes, but they are typically 500×375 pixels (horizontal image) or 375×500 pixels (vertical image), with no more than a 100-pixel[1,2] difference between the width and height. Small targets from the VOC2007 data set was selected, whose width and height are less than one-tenth of the original image (that is, a target that occupies 50×37 pixels or 37×50 pixels, and the width and height deviation does not exceed 10 pixels) for experimental verification to assess of the improved algorithm for the problem of small target detection. There are 1164 tags overall for the eight different sorts of small targets, which include people, dogs, sheep, bottles, birds, boats, aeroplanes, and birds.

The YOLOv4 algorithms known as YOLOv4-v1 and YOLOv4-v2 merely boost the shallow feature enhancement mechanism, and the accuracy of Gaussian model calculation coordinates, respectively. Improved YOLOv4 refers to the YOLOv4 algorithm that has been improved utilizing two newer techniques. YOLOv4-v1, YOLOv4-v2, and improved-YOLOv4 were experimented on the VOC2007 small target data set.

The improved YOLOv4 algorithm provides a much higher accuracy of detection for small targets. The detection accuracy of the improved YOLOv4 algorithm is 8.08% greater than the mean average precision (mAP) value of the original YOLOv4 algorithm by applying two algorithms simultaneously. The improved YOLOv4 algorithm has also greatly increased its overall accuracy and recall rate. Even though the improved YOLOv4 algorithm's detection speed employing the shallow feature fusion method has slowed down, it can still detect the target in real-time.

Endnotes

1. S. Akshayaa and S. Nithin, "*Comparative study of pedestrian detection techniques for driver assistance system*," in Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021.

2. Y. He, C. Zhu, J. Wang, M. Savvides and X. Zhang, "*bounding box regression with uncertainty for accurate object detection*," in IEEE Conference on Computer Vision and Pattern Recognition, 2019.

Chapter Four

Results and Discussion

4.1 Experiment and Analysis

Experiments were conducted using the LISA traffic light dataset of the University of California, San Diego's Intelligent and Safe Automobile Laboratory to validate the performance of the improved YOLOv4 algorithm for traffic light detection. The Bumblebee XB3 camera was positioned in the centre of the car's roof during sample collection for this data set to picture the traffic signal lights. The massive amount of data gathered encompassed settings including bright illumination, target coverage, and night. These scenes increased the traffic signal light recognition challenge and allowed us to test the algorithm's robustness better. The dataset was labelled completely, as shown in Table 4.1, where LISA-dayTrain is the training set, and LISAdaySeq1 is the test set.

Table 4.1 Overview of the LISA Traffic Light Dataset

Sequence Name	Number of Images	Number of Tags	Image Size
LISA-dayTrain	14,025	40,764	1280 × 960
LISA-daySeq2	6894	11,144	1280 × 960
LISA-daySeq1	4060	10,308	1280 x 960

Source: Research Design, 2023

4.2 Algorithm and Anchor Parameter

The anchor mechanism is extended by the YOLOv4 algorithm. An anchor is a set of a priori candidate frames with a predefined aspect ratio that are used to limit the range of the projected object. During the network's training phase, the size of the predicted bounding box is continuously changed using the pre-set anchor parameters, and the ideal predicted bounding box is gradually determined. As a result, when training traffic signal data, it is critical to define appropriate network parameters based on the traffic signal data's properties. K-means++ clustering algorithm was used to cluster traffic light data, compute the similarity of input samples, and acquire the anchor parameters best suited for traffic light data to filter the anchor parameters best suited for traffic lights.

To cluster the traffic light data, we chose k distinct clusters and utilized the K-means++ algorithm. As demonstrated in Figure 4.1, the average intersection over union (AVG IOU) of the actual and predicted boxes changed with k.

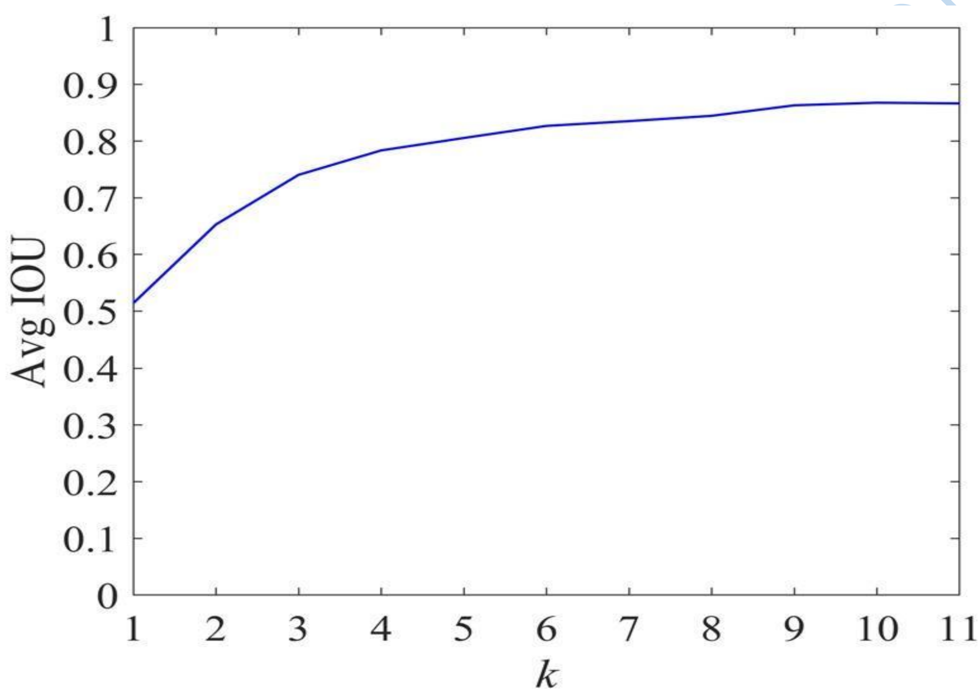


Figure 4.1: Average Intersection Over Union.
Source: Research Design, 2023

Figure 4.1 shows that as the number of clusters k rose, the average intersection over union tended to become flatter. The bigger the value of k, the smaller the gap between the real box and the predicted box, which helped reduce the training error. As a result, we selected to use the a priori aspect ratio for k = 9 as the anchor parameter of the improved YOLOv4 algorithm in this study; the a priori aspect ratios for other k values are provided in Table 4.2.

Table 4.2: A Priori Aspect Ratio Corresponding to Different k Values

k = 6	k = 7	k = 68	k = 9	k = 10	k =11
(6,13)	(7,14)	(6,12)	(6,13)	(6,13)	(6,13)
(8,16)	(10,21)	(7,17)	(8,16)	(8,16)	(8,16)
(10,24)	(12,28)	(9,17)	(10,24)	(10,24)	(10,23)
(13,28)	(16,34)	(10,24)	(13,27)	(13,27)	(13,24)
(18,40)	(18,44)	(13,28)	(16,34)	(15,34)	(13,30)
(26,53)	(26,53)	(16,37)	(18,44)	(20,39)	(17,32)
-	(27,58)	(21,44)	(25,41)	(18,47)	(17,41)
-	-	(27,58)	(23,55)	(25,42)	(25,41)
-	-	-	(29,59)	(23,54)	(22,52)
-	-	-	-	(29,59)	(28,55)
-	-	-	-	-	(30,70)

Source: Research Design, 2023

4.3 Model Training Analysis

To construct the anchor parameters for the traffic light data set and replace the original parameters used in the operation, all experiments used the K-means++ clustering algorithm. With a learning rate of 0.01 as the starting point, the maximum number of iterations was set at

50,000. The cosine function attenuation approach attenuated the learning rate to prevent the gradient explosion from occurring during the training phase when the learning rate was too high.

Figure 4.2 depicts the cosine function's attenuation curve.

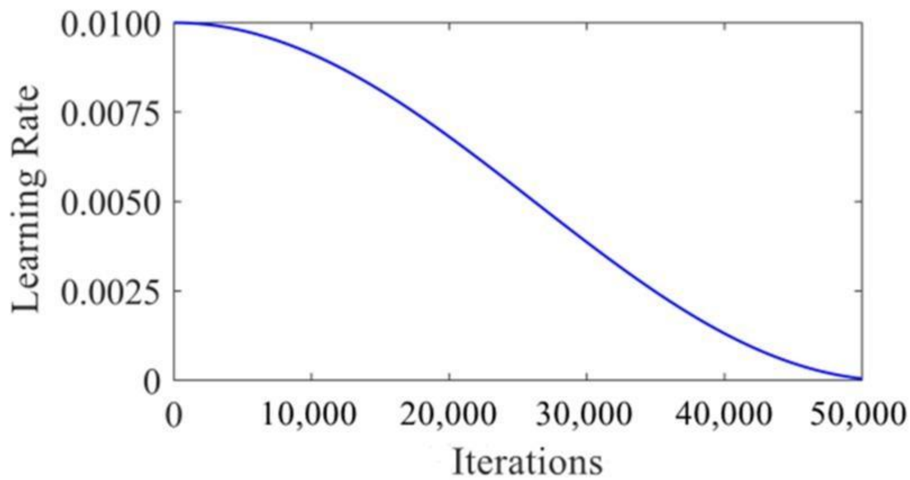


Figure 4.2: Attenuation Curve of the Cosine Function
Source: Research Design, 2023

All network training parameters were captured during the training procedure. Figure 4.3 depicts the average loss curve as it varies with the number of repetitions. The statistic shows that the average loss during the initial training period was relatively high. The average loss reduced continuously as the number of iterations rose until it ultimately stabilized, producing an optimal training effect.

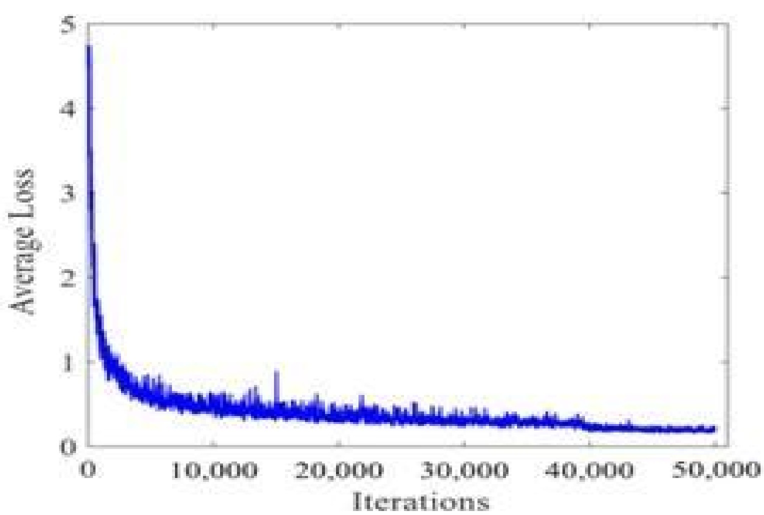


Figure 4.3: Relation Curve of Average Loss and the Number of Iterations Source: Research Design, 2023

Figure 4.4 depicts the average intersection over the union curve with the number of iterations. The figure shows that the average intersection over union was relatively tiny at the start of the training. The average intersection over union rapidly grew as the number of iterations increased. After 45,000 iterations, the average intersection over union might be kept at around 0.9 to provide the appropriate training impact.

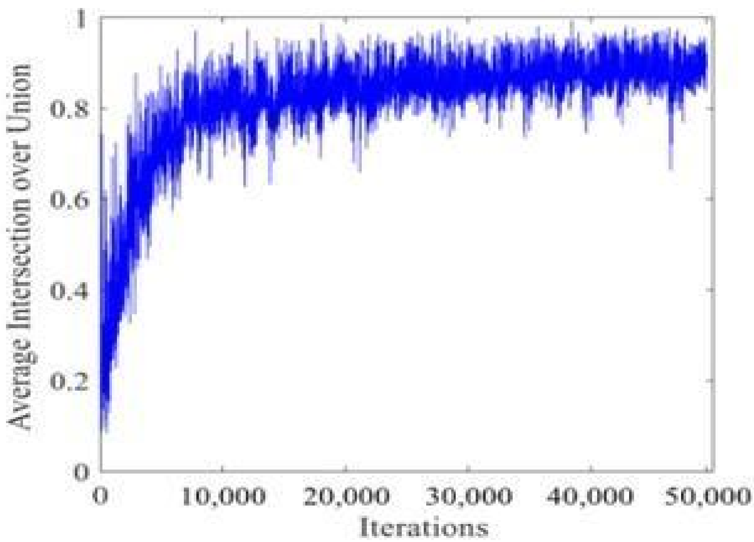


Figure 4.4 Relation Curve of the Average Intersection Over Union and the Number of Iterations
Source: Research Design, 2023

4.4 Analysis of Traffic Lights Detection Performance

The challenge with traffic light detection was that the traffic light only occupied a tiny number of effective pixels. In natural sceneries, the background of traffic lights is complicated and changing. The detecting effect of traffic lights is affected by intense light, evening, and weather conditions. To validate the efficacy of the algorithm proposed in this paper for small target traffic light detection against a complex background, the AUC value specified in the Vision for Intelligent Vehicles and Applications (VIVA) Challenge Competition was used as the evaluation index for this section of the experiment. The intersection over a union of valid positive samples should be greater than 0.5 in the calculation.

We detected traffic lights following the VIVA Challenge Competition's Traffic Light category standards. The LISA traffic light dataset ran six experiments using the Faster R-CNN, YOLOv3, YOLOv4, YOLOV4-V1, YOLOV4-V2, and improved YOLOv4 algorithms. Following thorough training, the detection results of several traffic signal lights in various scenarios were compared, as shown in Figure 4.5. Figure 4.5 depicts the detection effect maps and local area magnified images of traffic lights in the two circumstances of intense lighting and evening. The figure shows that the YOLOv3 algorithm missed targets and caused erroneous detections of traffic lights in bright light and at night, whereas the Faster R-CNN, YOLOv4, YOLOv4-v1, YOLOv4v2, and improved YOLOv4 algorithms avoided missed and false detections.

The improved YOLOv4 algorithm was able to reduce missed drastically, and erroneous detection of traffic lights by analysing 4060 traffic light data points in the LISA-daySeq1 test set, and detection accuracy was significantly improved. Table 4.3 displays the performance indicators for traffic light detection.

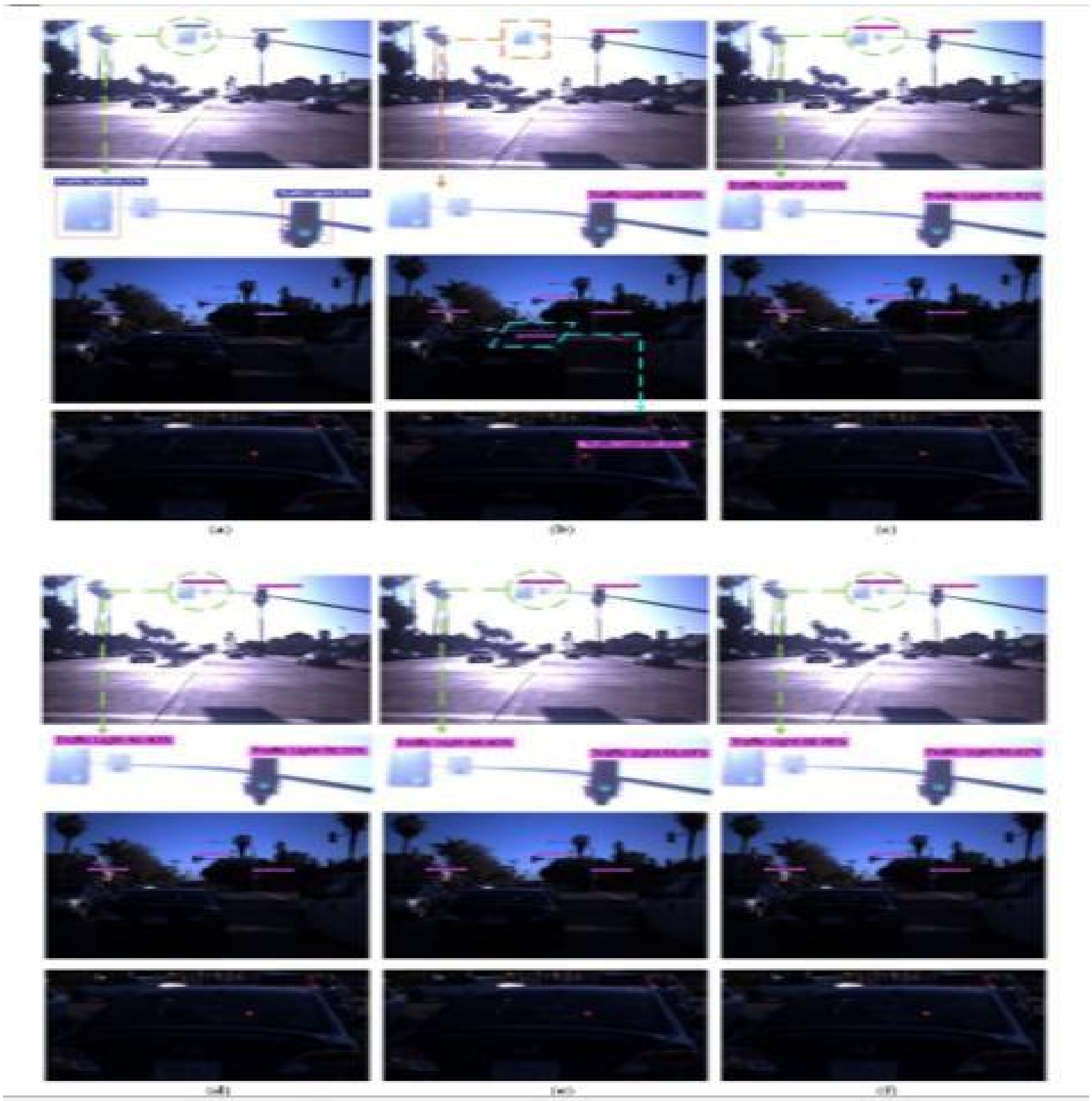


Figure 4.5: Comparison of Detection Results Using Multiple Algorithms for Traffic Lights in Different Scenarios (a) Faster R-CNN. (b) YOLOv3. (c) YOLOv4. (d) YOLOv4-v1. (e) YOLOv4-v2. (f) Improved YOLOv4.
Source: Research Design, 2023

The results demonstrate that, while the Faster R-CNN algorithm had a high AUC value, the detection time was longer, implying that it could not detect traffic lights in real-time. When compared to the original YOLOv4-v1 technique, the AUC value, precision, and recall rate of the

YOLOv4-v2 algorithm were greatly improved. The AUC value, precision rate, and recall rate of the improved YOLOv4 algorithm were much more significant than those of the original YOLOv3 algorithm, showing that the two improved approaches suggested in this study were suitable for traffic light detection. Because the shallow feature enhancement technique was utilized in the YOLOv4-v2 and improved YOLOv4 algorithms, which increased the number of network calculations, the detection performance was slower than that of the original YOLOv4 algorithm. However, the improved YOLOv4 algorithm's average detection time for a single image was only 33.74 ms. Compared to the Faster R-CNN algorithm, which took 101.48 ms, the improved YOLOv4 algorithm exhibited a better detection accuracy while still attaining real-time detection of traffic lights. The yellow rectangle is the missing area, the blue parallelogram is the wrong spot, and the green circle is the proper spot.

Table 4.3: Traffic Light Detection Performance Index in the LISA Dataset

Algorithm	AUC/%	Precision/%	Recall/%	Detection Speed/ms
ACF	40.17	-	-	-
YOLOv2	90.49	-	-	-
Faster R-CNN	97.01	98.25	95.93	101.48
YOLOv3	92.32	93.03	92.97	24.38
YOLOv4	96.58	96.86	95.62	28.33
YOLOv4-v1	96.84	97.41	96.13	27.59
YOLOv4-v2	97.03	97.96	96.17	33.99
Improved YOLOv4	97.58	98.74	96.81	33.74

4.5 Analysis of Traffic Lights Recognition Performance

In this section, the traffic light recognition experiment classified green, red, and yellow traffic lights as Go, Stop, and Warning to test the effectiveness and robustness of the improved YOLOv4 algorithm for traffic light recognition, and the experimental evaluation index was mAP, which is commonly used in target detection algorithms. Six tests were performed using the Faster R-CNN, YOLOv3, YOLOv4, YOLOv4-v1, YOLOv4-v2, and improved YOLOv4 algorithms.

As shown in Figure 4.6, the detection results of numerous traffic signal lights in different situations were compared, with the yellow rectangle representing the missing location, the blue parallelogram representing the wrong spot, and the green circle representing the correct site. Figure 4.6 depicts the detection effect maps and local area magnified images of traffic lights in the two circumstances of intense illumination and evening. Under bright lights, there were two instances of green traffic lights in the illustration. The YOLOv3 algorithm missed detection, while the YOLOv4, YOLOv4-v1, and YOLOv4-v2 methods all had one false detection and one valid detection case. All of the green traffic lights in the figure were accurately identified by the Faster R-CNN and improved YOLOv4 algorithms, and there were three examples of red traffic lights in the figure in the evening. Only one conspicuous red traffic light was accurately detected by the YOLOv3 algorithm, whereas two red traffic lights were correctly identified by the YOLOv4, YOLOv4-v1, and YOLOv4-v2 algorithms. Using the Faster R-CNN and improved YOLOv4 algorithms, all red traffic signal lights in the figure were correctly spotted. After testing on 4060 photos in the test set, the improved YOLOv4 algorithm effectively reduced the rate of missing and erroneous traffic light detections, and traffic light recognition accuracy was significantly improved.

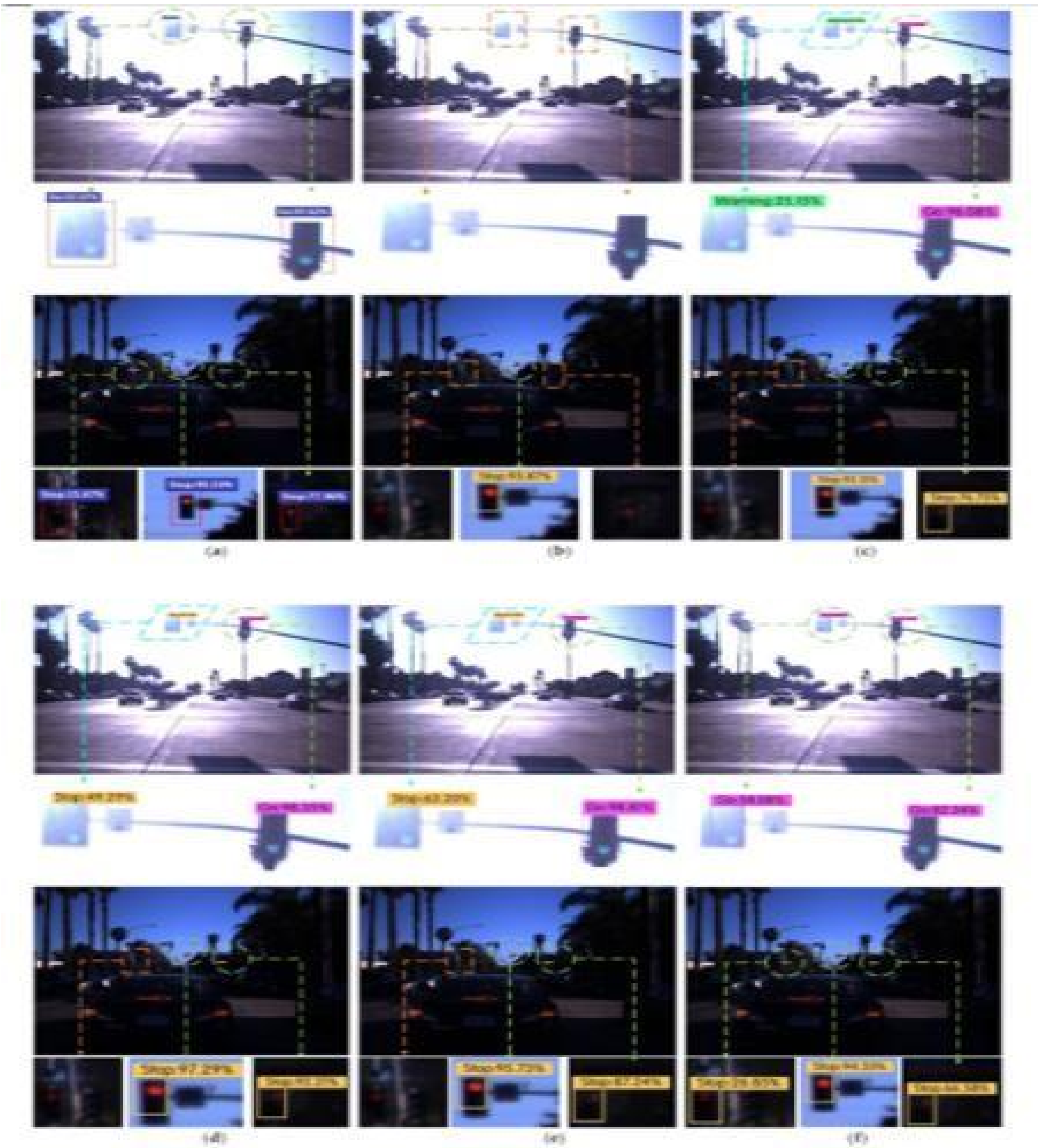


Figure 4.6: Comparison of Recognition Results Using Multiple Algorithms for Traffic Lights In Different Scenarios. (a) Faster R-CNN. (b) YOLOv3. (c) YOLOv4. (d) YOLOv4-v1. (e) YOLOv4-v2. (f) Improved YOLOv4
Source: Research Design, 2023

Table 5 displays the performance metrics for traffic light identification. The map values of the YOLOv4-v1 and YOLOv4-v2 algorithms were significantly improved in the identification experiments when compared to the original YOLOv4 algorithm, demonstrating that the proposed two improved methods of shallow feature enhancement and boundary box uncertainty prediction could effectively improve the identification accuracy of the YOLOv4 algorithm for traffic lights.

Table 4.4: Traffic Light Recognition Performance Index in the LISA Dataset

Algorithm	AP/% Go	Stop	Warning	mAP/%	Precision/%	Recall/%	Detection Speed/ms
Faster R-CNN	73.29	91.63	78.97	81.29	82.18	83.79	101.48
YOLOv3	63.28	85.02	74.71	74.33	75.17	80.30	24.38
YOLOv4	71.46	89.97	76.43	79.29	80.17	81.99	28.33
YOLOv4-v1	73.02	90.23	76.54	79.93	81.42	82.97	27.59
YOLOv4-v2	73.93	91.86	78.25	81.34	82.07	83.58	33.99
Improved YOLOv4	76.67	91.26	78.53	82.15	83.59	84.85	33.74

Source: Research Design, 2023

Chapter Five

Conclusion

5.1 Summary of Results

This project has demonstrated the advantages of using the improved YOLOv4 algorithm for traffic light detection and recognition. This solution included incorporating a shallow feature augmentation mechanism and a bounding box uncertainty prediction mechanism. The improved YOLOv4 approach effectively addressed the problem that the YOLOv4 algorithm was not sensitive to small objects, and traffic light detection and recognition accuracy was much improved. The experimental analysis was carried out using the LISA traffic light data set yielding the following results.

To optimize the YOLOv4 method, a shallow feature enhancement mechanism was used. The detection and recognition of traffic lights have been significantly enhanced. In the traffic light detection experiments, the AUC is increased to 97.03% and 95.31% for the two data sets of LISA and LaRa, respectively, while the mAP is increased to 81.34% and 78.88% for the recognition trials. Due to enhanced network calculations, the detection time was increased to 33.99 ms and 39.63 ms, respectively. The results indicate that the method could greatly improve traffic light detection and recognition accuracy. Although the amount of network calculation required was somewhat increased, the system could still achieve real-time detection and recognition of traffic lights.

A bounding box uncertainty prediction technique was used to optimize the YOLOv4 algorithm, which significantly increased its accuracy in detecting and recognizing traffic lights. The detection and recognition of traffic lights have been significantly enhanced. In the traffic light detection studies, the AUC for the two data sets of LISA and LaRa was increased to 96.84% and 94.73%, respectively, and the mAP was increased to 79.93% and 78.23%, respectively. The detection times were lowered to 27.59 and 33.45 milliseconds, respectively. The results demonstrate that, compared to the upgraded YOLOv4 algorithm, this method had a minimal difference in detection and recognition time but significantly increased the YOLOv4 algorithm's accuracy in traffic light detection and recognition.

5.2 Recommendations

The improved YOLOv4 algorithm used two optimization methods: shallow feature enhancement and bounding box uncertainty prediction. In the traffic light detection studies, the AUC for the two LISA and LaRa data sets increased by 1% and 1.19%, respectively, compared to the original YOLOv4 algorithm. In the recognition experiments, the mAP was enhanced by 2.86% and 2.56%, respectively, compared to the original YOLOv4 algorithm. The improved robustness of the YOLOv4 algorithm was demonstrated by a significant decrease in missed and erroneous detection cases under difficult traffic signal light backgrounds such as bright lighting, target blocking, and evening. Furthermore, when the modified YOLOv4 algorithm was evaluated on data sets acquired by other cameras, the AUC and mAP improved, proving the technique's scalability. Although the calculation cost grew as the number of network calculations increased, the additional detection time was only at the ms level, allowing for real-time detection of traffic lights. This demonstrates that the strategy presented in this work is a viable option for usage in real-world circumstances.

Due to the complexity and changeability of the background in traffic light detection settings, avoiding missed and incorrect detections of traffic lights remains difficult. Given the effectiveness of the improved YOLOv4 algorithm in reducing the number of missed and false detection cases, the target tracking should be focused on the identified traffic light to predict the movement trajectory and status of the traffic light relative to the vehicle, thus improving the improved YOLOv4 algorithm's reliability for traffic light detection and recognition.

Diverse and Extensive Dataset: Collect a diverse and extensive dataset of traffic light images from various locations, lighting conditions, and weather scenarios. This dataset should include images of traffic lights at different angles, distances, and orientations. Annotated Data: Ensure that the dataset is accurately annotated with information about the location of the traffic lights and their current color states. Data Augmentation: Apply data augmentation techniques to artificially increase the size and diversity of your dataset. This can include variations in brightness, contrast, and weather conditions. Semantic Segmentation: Implement semantic segmentation to precisely locate the traffic lights in the images. This can provide valuable information for colour recognition. Real-time Object Tracking: Use object tracking algorithms to track the detected traffic lights across multiple frames, confirming their colour consistency. Deep Learning Models: Train deep learning models, such as convolutional neural networks (CNNs), specifically for traffic light detection and recognition tasks. Transfer Learning: Leverage pretrained CNN models and fine-tune them for your traffic light recognition task. This can save time and resources. Calibration: Ensure that the cameras and sensors are properly calibrated to account for any distortion or misalignment that can affect colour recognition accuracy. Multimodal Sensor Fusion: Combine data from different sensors, such as cameras, LiDAR, and radar, to enhance traffic light detection and recognition. This can provide complementary information. Redundancy: Implement redundancy in your sensor systems to ensure that the failure of one sensor does not compromise the overall system's accuracy. Real-time Processing: Use dedicated hardware accelerators, like GPUs or TPUs, to perform real-time image processing and recognition tasks, crucial for autonomous vehicles. Machine Learning Anomaly Detection: Develop algorithms to detect and handle anomalies, such as malfunctioning or obscured traffic lights, to ensure the vehicle's safety. Continuous Testing: Continuously test the traffic light recognition system in various real-world conditions to identify areas for improvement. Regulatory Compliance: Ensure that your system complies with all relevant regulations for autonomous vehicle operation and traffic light recognition in the regions where your vehicles will be deployed.

5.3 Contribution to Knowledge

This research presents several significant contributions to knowledge in both the fields of autonomous vehicle technology and traffic management systems. They include:

i. Development of more sophisticated algorithms that are capable of accurately identifying traffic light colours under various environmental conditions, such as different lighting, weather, and obstructed views.

ii. Increase in the accuracy of traffic light recognition and the overall safety of autonomous vehicles. This is crucial for public trust and the broader acceptance of autonomous vehicles on roads.

iii. Provision of valuable data and insights that can inform international standards and regulations.

5.4 Suggestions for Further Studies

Multi-Sensor Fusion: Investigate more advanced techniques for combining data from various sensors, such as cameras, LiDAR, radar, and GPS, to enhance traffic light detection accuracy. Research could focus on optimizing the integration of information from different sensors to improve overall system reliability. Real-Time Reinforcement Learning: Explore the application of reinforcement learning techniques for real-time decision-making regarding traffic light recognition. Reinforcement learning can adapt to changing traffic conditions and traffic light patterns. Environmental Variability: Study the impact of environmental variability, such as extreme weather conditions (heavy rain, snow, fog), on traffic light recognition accuracy.

Develop methods to make the recognition system robust to such challenging conditions. Generalization to New Locations: Investigate methods to improve the system's ability to generalize to new geographic locations with different traffic light designs and configurations.

Transfer learning and domain adaptation techniques can be valuable here. Edge Computing and Onboard Processing: Research ways to optimize onboard processing for traffic light recognition, which is essential for real-time decision-making in autonomous vehicles. This includes exploring hardware acceleration and efficient algorithms.

Human-Centric Interaction: Examine how autonomous vehicles can effectively communicate their understanding of traffic light states and intentions to passengers and other road users. This can include the use of natural language interfaces or graphical displays.

Data Privacy and Security: Explore methods for securing the data used for traffic light recognition, as well as ensuring the privacy of individuals captured in the images. This is especially relevant as autonomous vehicles generate and collect large amounts of data. **Regulatory and Ethical Considerations:** Investigate the legal and ethical implications of traffic light recognition in autonomous vehicles, including issues related to liability, accountability, and compliance with traffic regulations. Benchmarking and Evaluation Metrics: Develop standardized benchmark datasets and evaluation metrics to assess the accuracy and performance of traffic light recognition systems. This would facilitate fair comparisons between different methods.

Vehicle-to-Infrastructure (V2I) Communication: Study the potential benefits of V2I communication systems, which could provide real-time traffic light information to autonomous vehicles. Investigate how these systems can be integrated with onboard traffic light recognition for improved accuracy.

Machine Learning Explainability: Enhance the interpretability and explainability of machine learning models used in traffic light recognition to make it easier to understand why a particular decision was made. Continuous Learning and Adaptation: Research approaches that enable autonomous vehicles to continuously learn and adapt their traffic light recognition capabilities based on real-world experience and feedback from other vehicles.

Bibliography

Conference Proceedings

Ahmim A., Maglaras L., Ferrag M. A., Derdour M., & Janicke H., *"A Novel Hierarchical Intrusion Detection System Based on Decision Tree and Rules-Based Models*," In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2019, pp. 228–233.

Alagarsamy S, Nagaraj P, Srikanth B, Krishna C.V, Bharath G & Kalyan S.S. *A Novel Machine Learning Technique for Predicting Road Accidents*. **I**n 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2023, (pp. 1547-1551). IEEE.

Alnami H.M, Mahgoub I & Al–Najada H. *Highway Accident Severity Prediction for Optimal Resource Allocation of Emergency Vehicles and Personnel*. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) IEEE, 2021 (pp. 1231-1238).

Arowolo M. O., Adebiyi M., Adebiyi A., & Okesola O., "*PCA Model for RNA-Seq Malaria Vector Data Classification using KNN and Decision Tree Algorithm,*" In 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Mar. 2020, pp. 1–8, DOI: 10.1109/ICMCECS47690.2020.240881

Batitis V. M. E., Caballes M. J. G., A. A. Ciudad, M. D. Diaz, R. D. Flores, & E. R. E. Tolentin, "*Image Classification of Abnormal Red Blood Cells using Decision Tree Algorithm,*" In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 498–504, DOI: 10.1109/ICCMC48092.2020.ICCMC-00093.

Chirag P, Supreetha M. *Road Accident Prediction and Classification using Machine Learning*. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) IEEE**,** 2022, (pp. 1-8).

Demidova L & Ivkina M. *Defining the Ranges Boundaries of the Optimal Parameters Values for the Random Forest Classifier.* In 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA) IEEE, 2019, (pp. 518-522).

Denisko D & Hoffman M.M. *Classification and Interaction in Random Forests.* Proceedings of the National Academy of Sciences**,** 2018, 115(8):1690-2.

Elassad Z.E Abou, Mousannif H & Moatassime H Al. *A Real-Time Crash Prediction Fusion Framework: An Imbalance-Aware Strategy for Collision Avoidance Systems*. Transportation Research Part C: Emerging Technologies, 2020, 118:102708.

Elyassami S, Hamid Y & Habuza T. *Road Crashes Analysis and Prediction using Gradient Boosted and Random Forest Trees*. In 2020 6th IEEE Congress on Information Science and Technology (CiSt) IEEE, 2021, (pp. 520-525).

Finogeev A, Deev M & Kolesnikoff I. *Proactive Big Data Analysis for Traffic Accident Prediction*. In 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), IEEE, 2020, pp. 1-9.

Géron A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.

Kim D.H DH, Ramjan LM & Mak KK. *Prediction of Vehicle Crashes by Drivers' Characteristics and Past Traffic Violations In Korea Using A Zero-Inflated Negative Binomial Model.* Traffic Injury Prevention 17(1), 2016, 86-90.

Kumeda B, Zhang F, Zhou F, Hussain S, Almasri A & Assefa M. *Classification of Road Traffic Accident Data using Machine Learning Algorithms*. In 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN) IEEE, 2019, (pp. 682-687).

Kwok-Fai Lui A, Chan Y.H, Lo K.H, Cheng W.T & Cheung H.T. *Predictive Screening of Accident Black Spots based on Deep Neural Models of Road Networks and Facilities: A Case Study based on a District in Hong Kong.* In 2021 5th International Conference on Computer Science and Artificial Intelligence, 2021, (pp. 422-428).

Lubbe N, Jeppsson H, Ranjbar A, Fredriksson J, Bärgman J, & Östling M. *Predicted Road Traffic Fatalities in Germany: The Potential and Limitations of Vehicle Safety Technologies from Passive Safety to Highly Automated Driving*. In Proceedings of IRCOBI conference. Athena, Greece, 2018.

Malik S, Sayed H El, Khan M.A & Khan M.J. *Road Accident Severity Prediction—A Comparative Analysis of Machine Learning Algorithms.* In 2021 IEEE Global Conference on Artificial Intelligence and Internet of Things IEEE. (GCAIoT), 2021, (pp. 69-74).

Nandhini S. & J. M. K.S, "*Performance Evaluation of Machine Learning Algorithms for Email Spam Detection*," In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1–4, DOI: 10.1109/icETITE47903.2020.312.

Nandurge P.A & Dharwadkar N.V. *Analyzing Road Accident Data using Machine Learning Paradigms*. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) IEEE, 2017, (pp. 604-610).

Park H & Haghani A. *Real-Time Prediction of Secondary Incident Occurrences using Vehicle Probe Data.* Transportation Research Part C: Emerging Technologies, 2016, 70:69-85.

Pathan S., Kumar P., Pai R., & Bhandary S. V., "*Automated Detection of Optic Disc Contours in Fundus Images using Decision Tree Classifier*," Biocybernetics and Biomedical Engineering, vol. 40, no. 1, 2020, pp. 52–64

Patil S. & Kulkarni U., "*Accuracy Prediction for Distributed Decision Tree using Machine Learning approach*," In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 1365–1371. DOI: 10.1109/ICOEI.2019.8862580

Ramadhan I., Sukarno P., & Nugroho M. A., "*Comparative Analysis of K-Neares Neighbor and Decision Tree in Detecting Distributed Denial of Service,*" In 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020, pp. 1–4, DOI: 10.1109/ICoICT49345.2020.9166380.

Reiss F, Cutler B & Eichenberger Z. *Natural Language Processing with Pandas Dataframes*. In Proc. Of The 20th Python In Science Conf. (Scipy 2021), 2021, pp. 49-58.

Sathiyanarayanan P., Pavithra S., Saranya M. S., & Makeswari M., "*Identification of Breast Cancer using the Decision Tree Algorithm*," In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1–6.

Shaik A.B & Srinivasan S. *A Brief Survey on Random Forest Ensembles in Classification Model*. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Springer Singapore, Volume 2, 2019, (pp. 253-260).

Shen Z, Shehzad A, Chen S, Sun H & Liu J. *Machine Learning Based Approach on Food Recognition and Nutrition Estimation*. Procedia Computer Science, 2020, 174:448-53.

Sowdagur J.A, Rozbully-Sowdagur B.T, Suddul G. *An Artificial Neural Network Approach for Road Accident Severity Prediction*. In 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC) IEEE, 2022, (pp. 267-270).

Sun Y, Li Y, Zeng Q & Bian Y. *Application Research of Text Classification Based on Random Forest Algorithm.* 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 2020, 370–374. https://doi.org/10.1109/AEMCSE50948.2020.00086

Taloba A. I. & Ismail S. I., "*An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection*," In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), 2019, pp. 99–104. DOI: 10.1109/ICICIS46948.2019.9014756

Xia X L, Nan B. & Xu C. *Real-time traffic Accident Severity Prediction using Data Mining Technologies*. In 2017 International Conference on Network and Information Systems for Computers (ICNISC), 2017, pp. 242-245. IEEE.

Yaacob N.F, Rusli N & Bohari S.N. *A Review Analysis of Accident Factor on Road Accident Cases using Haddon Matrix Approach*. In Proceedings of the Second International

Conference on the Future of ASEAN (ICoFA), Springer Singapore, 2017–Volume 2: Science and Technology, 2018, (pp. 55-65).

Zhang Y., Liu J., Zhang Z., & Huang J., "*Prediction of Daily Smoking Behavior Based on Decision Tree Machine Learning Algorithm*," In 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 330–333, DOI: 10.1109/ICEIEC.2019.8784698

Dissertations

Abiodun O.I, Jantan A, Dada K.V, Mohamed N.A & Arshad H. *State-of-the-art in Artificial Neural Network Applications: A survey*. Heliyon, 2018, 4:11. DOI: 10.1016/j.heliyon.2018.e00938.

Adnan M.S, Zaidi S & Bhargava P. *A Novel Support Vector Regression (SVR) Model for the Prediction of Splice Strength of the Unconfined Beam Specimens. Construction and Building Materials*, 2020, 248:118475.

Afraei S, Shahriar K & Madani S.H. *Developing Intelligent Classification Models for Rock Burst Prediction after Recognizing Significant Predictor Variables, Section 2: Designing classifiers. Tunnelling and Underground Space Technology*, 2019, 84:522-37.

Andeta J.A. *Road-Traffic Accident Prediction Model: Predicting the Number of Casualties*. Master's Degree Thesis in Informatics, ECTS Spring Term, 2021. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-20146

Eneh, O.C., Okosun, A., Egbenta, I.R., Obi, N.I., Oloto, M.C., Ubani, O., Eneh, C.A. & Eneonwo C.I. *A comparative analysis of road and vehicle qualities as factors of road traffic carnage in Nigeria*. BMC Public Health, 2023, 23(1):2173. DOI: 10.1186/s12889023-17089-2

Helly T. O. *Appraisal of Road Accidents in Akungba Akoko, Ondo State, Nigeria*.

Herda G & McNabb R. *Python for Smarter Cities: Comparison of Python Libraries for Static and Interactive Visualisations of Large Vector Data*. arXiv preprint, 2022, arXiv:2202.13105.

Hihn H & Braun D.A. *Specialization in Hierarchical Learning Systems: A Unified Informationtheoretic Approach for Supervised, Unsupervised and Reinforcement Learning. Neural Processing Letters*. 2020, 52(3):2319-52.

Klein L.A. "S*ensor and Data Fusion for Intelligent Transportation Systems." Society of PhotoOptical Instrumentation Engineers*, 2019.

Li M., Xu H., & Deng Y., "*Evidential Decision Tree Based on Belief Entropy," Entropy*, 2019, 21(9):897.

Oyetubo A.O, Afolabi O.J & Ohida M.E. *Analysis of Road Traffic Safety in Minna Niger State, Nigeria. Logistics, Supply Chain, Sustainability and Global Challenges*. 2018, 9(1):23-38.

Rabbani M.B, Musarat M.A, Alaloul W.S, Maqsoom A, Bukhari H, & Rafiq W. *Road Traffic Accident Data Analysis and its Visualization*. Civil engineering and architecture. 2021, 9(5):1603-14.

Sabzekar M & Hasheminejad S.M. *Robust Regression Using Support Vector Regressions. Chaos, Solitons & Fractals*. 2021, 144:110738.

Saigo H, Nowozin S, Kadowaki T, Kudo T & Tsuda K. *gBoost: A Mathematical Programming Approach To Graph Classification And Regression*. Machine Learning. 2009. 75:69-89.

Salaudeen A.G. *Risk Factors and Safety Measures for Road Traffic Crashes Among Inter-City Commercial Drivers in Kwara State, Nigeria*, Doctoral Dissertation, University Of Ilorin. 2018

Uzondu C, Jamson S & Lai F. *Exploratory study involving observation of traffic behaviour and conflicts in Nigeria using the Traffic Conflict Technique*. Safety Science. 2018. 110(Part A):273-284. DOI: 10.1016/j.ssci.2018.08.029.

Wang M, Jia S, Chen E, Yang S, Liu P & Qi Z. *A Derived Least Square Fast Learning Network Model. Applied Intelligence*. 2020. 50:4176-94.

Zhang Y & Sung Y. *Hybrid Traffic Accident Classification Models*. Mathematics. 2023. 11(4):1050.

Journals

Abdulkareem N.M & Abdulazeez A.M. *Machine Learning Classification Based on Radom Forest Algorithm: A review*. **International Journal of Science and Business**. 2021. 5(2):128-42.

Abdullah P & Sipos T. *Drivers' Behavior and Traffic Accident Analysis using Decision Tree Method*. **Sustainability**. 2022. 14(18):11339.

Ali L, Khan S.U, Golilarz N.A, Yakubu I, Qasim I, Noor A & Nour R. *A Feature-Driven Decision Support System for Heart Failure Prediction Based on Statistical Model and Gaussian Naive Bayes*. **Computational and Mathematical Methods in Medicine**. 2019.

Alzubi J, Nayyar A & Kumar A. *Machine Learning from Theory to Algorithms: An overview*. **In Journal of Physics: Conference Series** Vol. 1142, **IOP Publishing**. 2018. 012012

Apeagee B B & Haaor S A. *A Logistic Regression Model of Road Traffic Fatalities in Benue State: Imp*lication *to Public Health*. **Nigerian Annals of Pure and Applied Sciences.** 2020. 3(3a):46-52.

Assegie T. A. & Nair P. S., "*Handwritten Digits Recognition with Decision Tree Classification: A Machine Learning Approach*," **International Journal of Electrical and Computer Engineering**, vol. 9, no. 5, 2019, 4446.

Assi K, Rahman S.M, Mansoor U & Ratrout N. *Predicting Crash Injury Severity with Machine Learning Algorithm Synergized with Clustering Technique: A Promising Protocol.* **International journal of environmental research and public health**. 15, 2020, 5497.

Audu A.A, Iyiola O.F, Popoola A.A, Adeleye B.M, Medayese S, Mosima C, & Blamah N. *The Application of Geographic Information System as an Intelligent System Towards Emergency Responses in Road Traffic Accident in Ibadan*. **Journal of Transport and Supply Chain Management**. 2021, 15:17

Awad M, & Khanna R. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers,* **Springer Nature**, 2015, 268.

Azhar A, Ariff N.M, Bakar M.A & Roslan A. *Classification of Driver Injury Severity for Accidents Involving Heavy Vehicles with Decision Tree and Random Forest*. **Sustainability**. 14(7). 2022, 4101.

Bengio Y, Lodi A & Prouvost A. *Machine Learning for Combinatorial Optimization: A Methodological Tour d'horizon*. **European Journal of Operational Research**. 290(2), 2021, 405-21.

Bokaba T, Doorsamy W & Paul B.S. *Comparative Study of Machine Learning Classifiers for Modelling Road Traffic Accidents*. **Applied Sciences.** 2022. 12(2):828.

Wang C, Liu L, Xu C & Lv W. *Predicting Future Driving Risk of Crash-Involved Drivers Based on a Systematic Machine Learning Framework*. **International Journal of Environmental Research And Public Health.** 2019. 16(3):334.

Charbuty B & Abdulazeez A. *Classification Based on Decision Tree Algorithm for Machine Learning*. **Journal of Applied Science and Technology Trends**. 2021. 2(01):20-8.

Chinnamgari S. K. *R Machine Learning Projects: Implement Supervised, Unsupervised, and Reinforcement Learning Techniques using R 3.5*. Packt Publishing Ltd. 2019.

Darwish A, Ezzat D & Hassanien A E. *An Optimized Model Based on Convolutional Neural Networks and Orthogonal Learning Particle Swarm Optimization Algorithm for Plant Diseases Diagnosis*. Swarm and evolutionary computation**.** 2020. 52:100616.

De Felice F., Crocetti D., Parisi M., Maiuri V., Moscarelli E., Caiazzo R., Bulzonetti N., Musio D. & Tombolini V. "*Decision Tree Algorithm in Locally Advanced Rectal Cancer: An Example of Over-Interpretation and Misuse of a Machine Learning Approach,*" **Journal of Cancer Research and Clinical Oncology**, vol. 146, no. 3, 2020. 761–765.

Dias D, Silva J.S & Bernardino A. *The Prediction of Road-Accident Risk through Data Mining: A Case Study from Setubal, Portugal*. In Informatics MDPI Vol. 10, No. 1, 2023. 17.

Fareed U, Khadam U, Iqbal M.M & Iqbal M.J. *Road Accidents Investigation and Forecasting using Data Mining Techniques.* **KIET Journal of Computing and Information Sciences**. 6(1), 2023:28-49.

Figueira da Cruz A, Pitombo C.S & Larocca A.P. *Identification of Rules Induced through Decision Tree Algorithm for Detection of Traffic Accidents with Victims: A Study Case from Brazi*l. Case Studies on Transport Policy. 2017. 5(2):200-7.

Fleischhauer V, Feldheiser A. & Zaunseder S. *Beat-to-Beat Blood Pressure Estimation by Photoplethysmography and Its Interpretation*. Sensors. 2022. 22(18):7037.

Gajowniczek K, Grzegorczyk I, Ząbkowski T, C Bajaj. *Weighted Random Forests to Improve Arrhythmia Classification*. Electronics**.** 2020. 9(1):99.

Gnjatović M, Košanin I, Maček N & Joksimović D. *Clustering of Road Traffic Accidents as a Gestalt Problem*. Applied Sciences**.** 2022. 12(9):4543.

Golab A, Gooya E, Falou A & Cabon M. *A Multilayer Feed-Forward Neural Network (MLFNN) For the Resource-Constrained Project Scheduling Problem (RCPSP)*. Decision Science Letters. 2022. 11(4):407-18.

Guo Y., Hastie T. & Tibshirani R. *Regularized Linear Discriminant Analysis And Its Application In Microarrays*. Biostatistics. 2007. 8(1):86-100. DOI: 10.1093/biostatistics/kxj035.

Hagenauer J & Helbich M. *A Comparative Study of Machine Learning Classifiers for Modeling Travel Mode Choice*. Expert Systems with Applications. 2017. 78:273-82.

Heinsfeld A.S, Franco A.R, Craddock R.C, Buchweitz A, & Meneguzzi F. *Identification of Autism Spectrum Disorder Using Deep Learning and the ABIDE Dataset*. NeuroImage: Clinical. 2018. 17:16-23.

Hu X., Rudin C., & Seltzer M., "*Optimal Sparse Decision Trees,*" in Advances in Neural Information Processing Systems, 2019, 7267– 7275.

Hussain D., Al-Antari M. A., Al-Masni M. A., S.-M. Han & T.-S. Kim, "*Femur Segmentation in DXA Imaging using a Machine Learning Decision Tree,*" **Journal of X-ray Science and Technology**, vol. 26, no. 5, 2018, pp. 727–746.

Iwendi C, Bashir A.K, Peshkar A, Sujatha R, Chatterjee J.M, Pasupuleti S, Mishra R, Pillai S & Jo O. *COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm*. Frontiers in Public Health. 2020. 8:357.

Jamali A.A, Ferdousi R, Razzaghi S, Li J, Safdari R & Ebrahimie E. *DrugMiner: Comparative Analysis of Machine Learning Algorithms for Prediction of Potential Druggable Proteins*. Drug Discovery Today. 21(5):718-24, 2016.

Johnson J.M & Khoshgoftaar T.M. *Survey on Deep Learning With Class Imbalance*. **Journal of Big Data**; 6(1). 2019: 1-54.

Khattak A, Almujibah H, Elamary A & Matara C.M. *Interpretable Dynamic Ensemble Selection Approach for the Prediction of Road Traffic Injury Severity: A Case Study of Pakistan's National Highway N-5*. Sustainability. 2022. 14(19):12340.

Koley S, Sadhu A.K, Mitra P, Chakraborty B & Chakraborty C. *Delineation and Diagnosis of Brain Tumors from Post Contrast T1-weighted MR Images using Rough Granular Computing and Random Forest*. Applied Soft Computing. 2016. 41:453-65.

Kong J.S, Lee K.H, Kim O.H, Lee H.Y, Kang C.Y, Choi D, Kim S.C, Jeong H, Kang D.R & Sung T.E. *Machine Learning-Based Injury Severity Prediction of Level 1 Trauma Center Enrolled Patients Associated with Car-To-Car Crashes in Korea.* Computers in Biology and Medicine. 2023. 153:106393.

Konstantinov A.V & Utkin L.V. *Interpretable Machine Learning with an Ensemble of Gradient Boosting Machine*s. Knowledge-Based Systems. 2021. 222:106993.

Ma Z, Mei G & Cuomo S. *An Analytic Framework using Deep Learning for Prediction of Traffic Accident Injury Severity Based on Contributing Factors*. Accident Analysis & Prevention. 2021. 160:106322.

Mangalathu S, Hwang S.H, Choi E & Jeon J.S. *Rapid Seismic Damage Evaluation of Bridge Portfolios using Machine Learning Techniques.* Engineering Structures. 2019. 201:109785.

Mashudi N.A, Ahmad N & Noor N.M. *Classification of Adult Autistic Spectrum Disorder using Machine Learning Approach*. IAES International Journal of Artificial Intelligence. 2021. 10(3):743.

Mishra V, Agarwal S.M & N Puri. *Comprehensive and Comparative Analysis of Neural Network*. International Journal of Computer Application. 2018. 2(8):126-37.

Mosavi A, Sajedi Hosseini F, Choubin B, Goodarzi M, Dineva A.A & Sardooi E R. *Ensemble Boosting And Bagging Based Machine Learning Models For Groundwater Potential Prediction*. Water Resources Management. 2021. 35:23-37.

Nagra A. A., Han F, Ling Q H, Abubaker M, Ahmad F, Mehta S, Apasiba A T. "*Hybrid SelfInertia Weight Adaptive Particle Swarm Optimisation with Local Search Using C4. 5 Decision Tree Classifier for Feature Selection Problems,*" Connection Science, vol. 32, no. 1, 2020, pp. 16–36.

Onyemaechi N O & Ofoma U R. *The Public Health Threat of Road Traffic Accidents in Nigeria: A Call to Action.* Annals of Medical and Health Sciences Research. 6(4) 2016; 199-204.

Paczkowski W.R. *Data Visualization: The Basics. Business Analytics*: Data Science for Business Problems. 2021:85-126.

Qi W, Su H, Yang C, Ferrigno G, De Momi E & Aliverti A. *A Fast and Robust Deep Convolutional Neural Networks for Complex Human Activity Recognition using Smartphone*. Sensors. 2019. 19(17):3731.

Reis I, Baron D & Shahaf S. *Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets*. The Astronomical Journal. 2018. 157(1):16.

Ren Y, L Zhang & Suganthan P.N. *Ensemble Classification and Regression-Recent Developments, Applications and Future Directions.* IEEE Computational Intelligence Magazine 11(1). Jan 12 2016; 41-53.

Rygielski C, Wang J.C & Yen D.C. *Data Mining Techniques for Customer Relationship Management*. Technology in society. Nov 1 2022; 24(4):483-502.

Tangirala S. *Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*. **International Journal of Advanced Computer Science and Applications**. 11(2):612-9. 2020

Saenz-Cogollo J.F & Agelli M. *Investigating Feature Selection and Random Forests for InterPatient Heartbeat Classification*. Algorithms. Mar 25 2020; 13(4):75.

Sameen M.I & Pradhan B. *Severity Prediction of Traffic Accidents with Recurrent Neural Networks*. Applied Sciences. Jun 8 2017; 7(6):476.

Saracoglu A & Ozen H. *Estimation of Traffic Incident Duration: A Comparative Study of Decision Tree Models*. **Arabian Journal for Science and Engineering**. 2020. 45(10):8099-110.

Sarker I. H., Colman A., Han J., Khan A. I., Abushark Y. B., & Salah K., "*Behavdt: A Behavioral Decision Tree Learning To Build User-centric Context-Aware Predictive Model*," Mobile Networks and Applications, vol. 25, no. 3, 2020, pp. 1151–1161

Sarker I.H. *Machine Learning: Algorithms, Real-World Applications and Research Directions*. SN computer science**.** (3):160, May 2 2021.

Shiran G, Imaninasab R & Khayamim R. *Crash Severity Analysis of Highways Based on Multinomial Logistic Regression Model, Decision Tree Techniques, and Artificial Neural Network: A Modeling Comparison*. Sustainability. 2021. 13(10):5670.

Shiroyama R, Wang M & Yoshimura C. *Effect of Sample Size on Habitat Suitability Estimation using Random Forests: A Case of Bluegill, Lepomis Macrochirus*. **In Annales de Limnologie - International Journal of Limnology** Vol. 56, p. 13. EDP Sciences, 2020.

Sibindi R, Mwangi R.W & Waititu A.G. *A Boosting Ensemble Learning Based Hybrid Light Gradient Boosting Machine and Extreme Gradient Boosting Model for Predicting House Prices*. Engineering Reports. 2022:e12599.

Sidey-Gibbons J.A & Sidey-Gibbons C.J. *Machine Learning in Medicine: A Practical Introduction*. BMC MedicalR Methodology. 19 Dec 2019: 1-8.

Suganya E & Rajan C. *An Adaboost-Modified Classifier Using Particle Swarm Optimization and Stochastic Diffusion Search in Wireless Iot Networks*. Wireless Networks. 2021. 27:228799.

Sulaiman M.A. *Evaluating data mining classification methods performance in Internet of things applications*. **Journal of Soft Computing and Data Mining**. Dec 6 2020;1(2):11-25.

Taherkhani A, Cosma G & McGinnity T.M. *AdaBoost-CNN: An Adaptive Boosting Algorithm for Convolutional Neural Networks to Classify Multi-Class Imbalanced Datasets using Transfer Learning*. Neurocomputing. Sep 3 2020; 404:351-66.

Tsirikoglou P, Abraham S, Contino F, Lacor C & Ghorbaniasl G. *A Hyperparameters Selection Technique for Support Vector Regression Models.* Applied Soft Computing. 2017. 61:13948.

Utkin L.V, Kovalev M.S & Coolen F.P. *Imprecise Weighted Extensions of Random Forests for Classification and Regression*. Applied Soft Computing. 2020. 92:106324.

Varshney K.R. *Trustworthy Machine Learning and Artificial Intelligence*. *XRDS: Crossroads*, The ACM Magazine for Students. 25(3):26-9, Apr 10, 2019.

Wang D & Zhu A X. *Soil Mapping Based on the Integration of the Similarity-Based Approach and Random Forests*. Land. 2020. 9(6):174.

Wang Y, Wang D, Geng N, Wang Y, Yin Y & Jin Y. *Stacking-Based Ensemble Learning of Decision Trees for Interpretable Prostate Cancer Detection*. Applied Soft Computing. 2019. 77:188-204.

Wen H, Zhang X & Zeng Q. *Predicting Future Driving Risk of Crash-Involved Drivers Based on a Systematic Machine Learning Framework*. **International Journal of Environmental Research and Public Health**. 16(3), 2019;334-52.

Yan M & Shen Y. *Traffic Accident Severity Prediction Based on Random Forest.* Sustainability. 2022. 14(3):1729.

Yang X, Wang Y, Byrne R, Schneider G & Yang S. *Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery*. Chemical reviews. 2019. 119(18):10520-94.

Yeap D, McCartney M.M, Rajapakse M.Y, Fung A.G, Kenyon N.J & Davis C.E. *Peak Detection and Random Forests Classification Software for Gas Chromatography/Differential Mobility Spectrometry (GC/DMS) Data.* Chemometrics and Intelligent Laboratory Systems. Aug 15 2020. 203:104085.

Yigin B.O, Algin O & Saygili G. *Comparison of Morphometric Parameters in Prediction of Hydrocephalus using Random Forests*. Computers in Biology and Medicine. Jan 1 2020. 116:103547.

Zahid M, Chen Y, Khan S, Jamal A., Ijaz M & Ahmed T. *Predicting Risky and Aggressive Driving Behavior among Taxi Drivers: do Spatio-Temporal Attributes Matter?* **International Journal of Environmental Research and Public Health**. 17(11), Jun 2020: 3937.

Zebari R, Abdulazeez A, Zeebaree D, Zebari D & Saeed J. *A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction*. **J. Appl. Sci. Technol**. Trends. May 15 2020. 1(2):56-70.

Zhang C, Lei X & Liu L. *Predicting Metabolite–Disease Associations Based on LightGBM Model*. Frontiers in Genetics. Apr 13 2021;12:660275.

Zhang F, Yang X. *Improving Land Cover Classification in an Urbanized Coastal Area by Random Forests: The Role of Variable Selection*. Remote Sensing of Environment. 2020. 251:112105.

Zhang X, Qi S, Zheng A, Luo Y & Hao S. *Data-Driven Analysis of Fatal Urban Traffic Accident Characteristics and Safety Enhancement Research*. Sustainability. 2023. 15(4):3259.

Zou Q., Qu K., Luo Y., Yin D., Ju Y., & Tang H., "*Predicting Diabetes Mellitus with Machine Learning Techniques*," Frontiers in Genetics, vol. 9, 2018 p. 515

Appendices

Appendix I import

pandas as pd import

numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df = pd.read_csv('RTA Dataset.csv')

df.head() df.shape

print the dataset information df.info()

df.isnull().sum()/100

df['Accident_severity'].value_counts().plot(kind='bar')

import matplotlib.pyplot as plt

Plot the bar chart

ax = df['Accident_severity'].value_counts().plot(kind='bar', color=['#FF6347', '#4169E1', '#32CD32'])

Customize the plot

ax.set_xlabel('Accident Severity')

ax.set_ylabel('Count')

ax.set_title('Distribution of Accident

Severity') ax.legend(['Minor', 'Moderate',

'Severe']) ax.grid(axis='y', linestyle='--')

Save the plot

plt.savefig('accident_severity_plot.png') plt.show()

"""This shows imbalance multiclass label on the dataset"""

plot the bar plot of road_surface_type and accident severity feature plt.figure(figsize=(6,5))

sns.countplot(x='Road_surface_type', hue='Accident_severity', data=df)

plt.xlabel('Rode surafce type') plt.xticks(rotation=60)

plt.savefig('accident_severity_plot.png') plt.show()

convert object type column into datetime datatype column df['Time']

= pd.to_datetime(df['Time'])

Extrating 'Hour_of_Day' feature from the Time column

new_df = df.copy()

```
new_df['Hour_of_Day'] = new_df['Time'].dt.hour
df_new = new_df.drop('Time', axis=1)
df_new.head()
def fill_missing_values(df):
  # Loop over each column in the dataframe
  for col in df.columns:
    if df[col].dtype == 'float64' or df[col].dtype == 'int64': # Check if column is numeric
      # Fill missing values with mean
      df[col].fillna(df[col].mean(), inplace=True)
    else:
      # Fill missing values with mode
      df[col].fillna(df[col].mode()[0], inplace=True)
  return df

# Fill missing values using the function
df_new = fill_missing_values(df_new)

df_new.isnull().sum()

from sklearn.preprocessing import LabelEncoder

def label_encode_features(df): le = LabelEncoder() #
  create a label encoder object

  for col in df.columns:
    if df[col].dtype == 'object': # check if column is of type 'object'
      df[col] = le.fit_transform(df[col].astype(str)) # label encode the column
```

```
    return df
# Label encode the object-type features using the function
new_df = label_encode_features(new_df)

new_df.head()

df_new.columns

#handling imbalance multiclass
X = new_df.drop(['Accident_severity', 'Time'], axis=1)
y = new_df['Accident_severity']

X

!pip install imblearn

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.pipeline import Pipeline from imblearn.pipeline import
make_pipeline from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier,VotingClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

le = LabelEncoder() y
= le.fit_transform(y)
sc = StandardScaler()
X = sc.fit_transform(X)
```

```
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,  test_size=0.3,  random_state=42)
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

# modelling using random forest baseline
rf = RandomForestClassifier(n_estimators=800, max_depth=20, random_state=42)

rf.fit(X_train_res, y_train_res)

#   predicting   on   test   data
predics = rf.predict(X_test)

cm = confusion_matrix(y_test, predics)
ConfusionMatrixDisplay(cm).plot()

# classification report on test dataset classif_re
= classification_report(y_test,predics)
print(classif_re)

from sklearn import tree
from   sklearn.tree   import   DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix from
sklearn.metrics import classification_report from
sklearn.metrics import ConfusionMatrixDisplay

decisionTree = DecisionTreeClassifier(criterion='entropy')
```

```
print(decisionTree)

dtc_model = decisionTree.fit(X_train_res, y_train_res)

from  matplotlib  import  pyplot

# feature importance

importance    =    dtc_model.feature_importances_
for i,v in enumerate(importance):
 print('Feature: %0d, Score: %.5f' % (i,v))

# Barchat for feature importance

pyplot.bar([x for x in range(len(importance))], importance) pyplot.show()

prediction = dtc_model.predict(X_test)

cm = confusion_matrix(y_test, prediction)
ConfusionMatrixDisplay(cm).plot()
print(classification_report(y_test, prediction))
import pandas as pd
from    sklearn.model_selection    import    train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix # Load the dataset (replace 'your_dataset.csv' with your actual dataset file) data = pd.read_csv('your_dataset.csv')
```

```
# Feature columns (replace 'feature1', 'feature2', etc. with the actual feature column names)
features = data[['feature1', 'feature2', 'feature3', ...]]

# Target column (replace 'target' with the actual column containing the severity codes)
target = data['target']

# Split the data into training and testing sets (adjust the test_size as needed)
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.25, random_state=42)

# Initialize the Random Forest Classifier
rf_classifier = RandomForestClassifier()

# Train the model
rf_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred) print(f"Accuracy:
{accuracy}")
# Generate classification report and confusion matrix
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
import    pandas    as    pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

df = pd.read_csv('RTA Dataset.csv')
df.head() df.shape
# print the dataset information df.info()

df.isnull().sum()/100

df['Accident_severity'].value_counts().plot(kind='bar')

import matplotlib.pyplot as plt

# Plot the bar chart
ax  =        df['Accident_severity'].value_counts().plot(kind='bar',        color=['#FF6347',
    '#4169E1', '#32CD32'])

# Customize the plot
ax.set_xlabel('Accident Severity')
ax.set_ylabel('Count')
ax.set_title('Distribution of Accident
Severity') ax.legend(['Minor', 'Moderate',
'Severe']) ax.grid(axis='y', linestyle='--')
```

```
# Save the plot
plt.savefig('accident_severity_plot.png') plt.show()

"""This shows imbalance multiclass label on the dataset"""

# plot the bar plot of road_surface_type and accident severity feature plt.figure(figsize=(6,5))
sns.countplot(x='Road_surface_type', hue='Accident_severity', data=df)
plt.xlabel('Rode surafce type') plt.xticks(rotation=60)
plt.savefig('accident_severity_plot.png') plt.show()

# convert object type column into datetime datatype column df['Time']
= pd.to_datetime(df['Time'])

# Extrating 'Hour_of_Day' feature from the Time column
new_df = df.copy()
new_df['Hour_of_Day'] = new_df['Time'].dt.hour
df_new = new_df.drop('Time', axis=1)
df_new.head()
def fill_missing_values(df):
  # Loop over each column in the dataframe
  for col in df.columns:
    if df[col].dtype == 'float64' or df[col].dtype == 'int64': # Check if column is numeric
      # Fill missing values with mean
      df[col].fillna(df[col].mean(), inplace=True)
    else:
      # Fill missing values with mode
      df[col].fillna(df[col].mode()[0], inplace=True)
```

```
    return df

# Fill missing values using the function
df_new = fill_missing_values(df_new)

df_new.isnull().sum()

from sklearn.preprocessing import LabelEncoder

def label_encode_features(df): le = LabelEncoder() #
   create a label encoder object

    for col in df.columns:
        if df[col].dtype == 'object': # check if column is of type 'object'
            df[col] = le.fit_transform(df[col].astype(str)) # label encode the column

    return df
# Label encode the object-type features using the function
new_df = label_encode_features(new_df)

new_df.head()

df_new.columns

#handling imbalance multiclass
X = new_df.drop(['Accident_severity', 'Time'], axis=1)
y = new_df['Accident_severity']
```

X

!pip install imblearn

from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.pipeline import Pipeline from imblearn.pipeline import

make_pipeline from imblearn.over_sampling import SMOTE

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier,VotingClassifier

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

le = LabelEncoder() y

= le.fit_transform(y)

sc = StandardScaler()

X = sc.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

smote = SMOTE(random_state=42)

X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

modelling using random forest baseline

rf = RandomForestClassifier(n_estimators=800, max_depth=20, random_state=42)

rf.fit(X_train_res, y_train_res)

predicting on test data

predics = rf.predict(X_test)

```
cm = confusion_matrix(y_test, predics)
ConfusionMatrixDisplay(cm).plot()

# classification report on test dataset classif_re
= classification_report(y_test,predics)
print(classif_re)

from sklearn import tree
from   sklearn.tree   import   DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix from
sklearn.metrics import classification_report from
sklearn.metrics import ConfusionMatrixDisplay

decisionTree = DecisionTreeClassifier(criterion='entropy')
print(decisionTree)

dtc_model = decisionTree.fit(X_train_res, y_train_res)

from  matplotlib  import  pyplot

# feature importance

importance   =   dtc_model.feature_importances_
for i,v in enumerate(importance):
```

```
print('Feature: %0d, Score: %.5f' % (i,v))

# Barchat for feature importance

pyplot.bar([x for x in range(len(importance))], importance) pyplot.show()

prediction = dtc_model.predict(X_test)

cm = confusion_matrix(y_test, prediction)
ConfusionMatrixDisplay(cm).plot()
print(classification_report(y_test, prediction))
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix import
pandas as pd import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

df = pd.read_csv('RTA Dataset.csv')
df.head() df.shape
# print the dataset information df.info()

df.isnull().sum()/100

df['Accident_severity'].value_counts().plot(kind='bar')
```

```
import matplotlib.pyplot as plt

# Plot the bar chart

ax  =        df['Accident_severity'].value_counts().plot(kind='bar',        color=['#FF6347',
    '#4169E1', '#32CD32'])

# Customize the plot

ax.set_xlabel('Accident Severity')

ax.set_ylabel('Count')

ax.set_title('Distribution of Accident Severity')

ax.legend(['Minor', 'Moderate', 'Severe'])

ax.grid(axis='y', linestyle='--')

# Save the plot

plt.savefig('accident_severity_plot.png') plt.show()

"""This shows imbalance multiclass label on the dataset"""

# plot the bar plot of road_surface_type and accident severity feature plt.figure(figsize=(6,5))

sns.countplot(x='Road_surface_type', hue='Accident_severity', data=df)

plt.xlabel('Rode surafce type') plt.xticks(rotation=60)

plt.savefig('accident_severity_plot.png') plt.show()

# convert object type column into datetime datatype column df['Time']

= pd.to_datetime(df['Time'])

#  Extrating  'Hour_of_Day'  feature  from  the  Time  column

new_df = df.copy()
```

```
new_df['Hour_of_Day'] = new_df['Time'].dt.hour
df_new = new_df.drop('Time', axis=1)
df_new.head()

def fill_missing_values(df):
  # Loop over each column in the dataframe
  for col in df.columns:
    if df[col].dtype == 'float64' or df[col].dtype == 'int64': # Check if column is numeric
      # Fill missing values with mean
      df[col].fillna(df[col].mean(), inplace=True)
    else:
      # Fill missing values with mode
      df[col].fillna(df[col].mode()[0], inplace=True)
  return df

# Fill missing values using the function
df_new = fill_missing_values(df_new)

df_new.isnull().sum()

from sklearn.preprocessing import LabelEncoder

def label_encode_features(df): le = LabelEncoder() #
  create a label encoder object

  for col in df.columns:
    if df[col].dtype == 'object': # check if column is of type 'object'
```

```
        df[col] = le.fit_transform(df[col].astype(str)) # label encode the column

    return df

# Label encode the object-type features using the function
new_df = label_encode_features(new_df)
new_df.head()

df_new.columns

#handling imbalance multiclass
X = new_df.drop(['Accident_severity', 'Time'], axis=1)
y = new_df['Accident_severity']

X

!pip install imblearn

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.pipeline import Pipeline from imblearn.pipeline import
make_pipeline from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier,VotingClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
le = LabelEncoder() y
= le.fit_transform(y)
sc = StandardScaler()
X = sc.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)
# modelling using random forest baseline
rf = RandomForestClassifier(n_estimators=800, max_depth=20, random_state=42)

rf.fit(X_train_res, y_train_res)

# predicting on test data
predics = rf.predict(X_test)

cm = confusion_matrix(y_test, predics)
ConfusionMatrixDisplay(cm).plot()

# classification report on test dataset classif_re
= classification_report(y_test,predics)
print(classif_re)

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier from
sklearn import metrics from sklearn.metrics import
confusion_matrix from sklearn.metrics import
```

```
classification_report from sklearn.metrics import
ConfusionMatrixDisplay

decisionTree = DecisionTreeClassifier(criterion='entropy') print(decisionTree)

dtc_model = decisionTree.fit(X_train_res, y_train_res)
from matplotlib import pyplot

# feature importance

importance = dtc_model.feature_importances_
for i,v in enumerate(importance):
 print('Feature: %0d, Score: %.5f' % (i,v))

# Barchat for feature importance

pyplot.bar([x for x in range(len(importance))], importance) pyplot.show()

prediction = dtc_model.predict(X_test)

cm = confusion_matrix(y_test, prediction)
ConfusionMatrixDisplay(cm).plot()
print(classification_report(y_test, prediction))
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Load the dataset (replace 'your_dataset.csv' with your actual dataset file)
data = pd.read_csv('your_dataset.csv')
# Feature columns (replace 'feature1', 'feature2', etc. with the actual feature column names)
features = data[['feature1', 'feature2', 'feature3', ...]]

# Target column (replace 'target' with the actual column containing the severity codes)
target = data['target']

# Split the data into training and testing sets (adjust the test_size as needed)
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.25,
random_state=42)

# Initialize the Random Forest Classifier
rf_classifier = RandomForestClassifier()

# Train the model
rf_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred) print(f"Accuracy:
{accuracy}")
# Generate classification report and confusion matrix
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
print("\nConfusion Matrix:") print(confusion_matrix(y_test,
y_pred))
import    pandas    as    pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

df = pd.read_csv('RTA Dataset.csv')
df.head() df.shape
# print the dataset information df.info()

df.isnull().sum()/100

df['Accident_severity'].value_counts().plot(kind='bar')

import matplotlib.pyplot as plt

# Plot the bar chart
ax  =    df['Accident_severity'].value_counts().plot(kind='bar',    color=['#FF6347',
    '#4169E1', '#32CD32'])

# Customize the plot
ax.set_xlabel('Accident Severity')
ax.set_ylabel('Count')
```

```
ax.set_title('Distribution of Accident Severity')
ax.legend(['Minor', 'Moderate', 'Severe'])
ax.grid(axis='y', linestyle='--')
# Save the plot
plt.savefig('accident_severity_plot.png') plt.show()

"""This shows imbalance multiclass label on the dataset"""

# plot the bar plot of road_surface_type and accident severity feature plt.figure(figsize=(6,5))
sns.countplot(x='Road_surface_type', hue='Accident_severity', data=df)
plt.xlabel('Rode surafce type') plt.xticks(rotation=60)
plt.savefig('accident_severity_plot.png') plt.show()

# convert object type column into datetime datatype column df['Time']
= pd.to_datetime(df['Time'])

# Extrating 'Hour_of_Day' feature from the Time column
new_df = df.copy()
new_df['Hour_of_Day'] = new_df['Time'].dt.hour
df_new = new_df.drop('Time', axis=1)
df_new.head()

def fill_missing_values(df):
  # Loop over each column in the dataframe
  for col in df.columns:
    if df[col].dtype == 'float64' or df[col].dtype == 'int64': # Check if column is numeric
      # Fill missing values with mean
```

```
            df[col].fillna(df[col].mean(), inplace=True)
        else:
            # Fill missing values with mode
            df[col].fillna(df[col].mode()[0], inplace=True)
    return df

# Fill missing values using the function
df_new = fill_missing_values(df_new)

df_new.isnull().sum()

from sklearn.preprocessing import LabelEncoder

def label_encode_features(df): le = LabelEncoder() #
    create a label encoder object

    for col in df.columns:
        if df[col].dtype == 'object': # check if column is of type 'object'
            df[col] = le.fit_transform(df[col].astype(str)) # label encode the column

    return df

# Label encode the object-type features using the function
new_df = label_encode_features(new_df)
new_df.head()

df_new.columns
```

```
#handling imbalance multiclass

X = new_df.drop(['Accident_severity', 'Time'], axis=1)

y = new_df['Accident_severity']


X


!pip install imblearn


from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.pipeline import Pipeline from imblearn.pipeline import

make_pipeline from imblearn.over_sampling import SMOTE

from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier,VotingClassifier

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score


le = LabelEncoder() y

= le.fit_transform(y)

sc = StandardScaler()

X = sc.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

smote = SMOTE(random_state=42)

X_train_res, y_train_res = smote.fit_resample(X_train, y_train)
# modelling using random forest baseline

rf = RandomForestClassifier(n_estimators=800, max_depth=20, random_state=42)


rf.fit(X_train_res, y_train_res)
```

```
#   predicting   on   test   data
predics = rf.predict(X_test)

cm = confusion_matrix(y_test, predics)
ConfusionMatrixDisplay(cm).plot()

# classification report on test dataset classif_re
= classification_report(y_test,predics)
print(classif_re)

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier from
sklearn import metrics from sklearn.metrics import
confusion_matrix from sklearn.metrics import
classification_report from sklearn.metrics import
ConfusionMatrixDisplay

decisionTree = DecisionTreeClassifier(criterion='entropy') print(decisionTree)

dtc_model = decisionTree.fit(X_train_res, y_train_res)
from  matplotlib  import  pyplot

# feature importance

importance   =   dtc_model.feature_importances_
for i,v in enumerate(importance):
```

```
 print('Feature: %0d, Score: %.5f' % (i,v))

# Barchat for feature importance

pyplot.bar([x for x in range(len(importance))], importance) pyplot.show()

prediction = dtc_model.predict(X_test)

cm = confusion_matrix(y_test, prediction)
ConfusionMatrixDisplay(cm).plot()
print(classification_report(y_test, prediction))
import pandas as pd
from   sklearn.model_selection   import   train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

#  Extrating  'Hour_of_Day'  feature  from  the  Time  column
new_df = df.copy()
new_df['Hour_of_Day'] = new_df['Time'].dt.hour
df_new = new_df.drop('Time', axis=1) df_new.head()

def fill_missing_values(df):
   # Loop over each column in the dataframe
   for col in df.columns:
      if df[col].dtype == 'float64' or df[col].dtype == 'int64': # Check if column is numeric
         # Fill missing values with mean
         df[col].fillna(df[col].mean(), inplace=True)
```

```
        else:
            # Fill missing values with mode
            df[col].fillna(df[col].mode()[0], inplace=True)
    return df

# Fill missing values using the function
df_new = fill_missing_values(df_new)

df_new.isnull().sum()

from sklearn.preprocessing import LabelEncoder

def label_encode_features(df): le = LabelEncoder() #
    create a label encoder object

    for col in df.columns:
        if df[col].dtype == 'object': # check if column is of type 'object'
            df[col] = le.fit_transform(df[col].astype(str)) # label encode the column
    return df

# Label encode the object-type features using the function
new_df = label_encode_features(new_df)

new_df.head()

df_new.columns
```

```
#handling imbalance multiclass

X = new_df.drop(['Accident_severity', 'Time'], axis=1)

y = new_df['Accident_severity']


X


!pip install imblearn


from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.pipeline import Pipeline from imblearn.pipeline import
make_pipeline from imblearn.over_sampling import SMOTE

from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier,VotingClassifier

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score


le = LabelEncoder()
y  =  le.fit_transform(y)

sc = StandardScaler()

X = sc.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

smote = SMOTE(random_state=42)

X_train_res, y_train_res = smote.fit_resample(X_train, y_train)


# modelling using random forest baseline

rf = RandomForestClassifier(n_estimators=800, max_depth=20, random_state=42)


rf.fit(X_train_res, y_train_res)
```

```
# predicting on test data
predics = rf.predict(X_test)

cm = confusion_matrix(y_test, predics)
ConfusionMatrixDisplay(cm).plot()

# classification report on test dataset classif_re
= classification_report(y_test,predics)
print(classif_re)

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import
ConfusionMatrixDisplay

decisionTree = DecisionTreeClassifier(criterion='entropy') print(decisionTree)

dtc_model = decisionTree.fit(X_train_res, y_train_res)

from matplotlib import pyplot

# feature importance
```

importance = dtc_model.feature_importances_

for i,v in enumerate(importance):

print('Feature: %0d, Score: %.5f' % (i,v)) #

Barchat for feature importance

pyplot.bar([x for x in range(len(importance))], importance) pyplot.show()

prediction = dtc_model.predict(X_test)

cm = confusion_matrix(y_test, prediction)

ConfusionMatrixDisplay(cm).plot()

print(classification_report(y_test, prediction))

import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df = pd.read_csv('RTA Dataset.csv')

df.head() df.shape

print the dataset information df.info()

```
df.isnull().sum()/100
```

```
df['Accident_severity'].value_counts().plot(kind='bar')
```

```
import matplotlib.pyplot as plt
```

```
# Plot the bar chart
ax = df['Accident_severity'].value_counts().plot(kind='bar', color=['#FF6347', '#4169E1',
```

Biodata

A. Personal Data 1. Full Name: Fashina Temilade Temitope 2. **Date and Place of Birth: 2nd** September 1982, Ibadan **3. Nationality:** Nigerian **4. Marital Status: Married 5. No. of Children & their ages: 2 (10years and 7 years)** 6. **Name and Address of Spouse:** Engr. Peter Fashina

7. **Name and Address of Next of Kin:** Engr. Peter Fashina, Hwawei Technology, Ibadan
8. **Faculty:** Natural and Applied Sciences
9. **Department:** Computer Science

B. Educational Institutions Attended with Dates and Qualification:

- **2008– 2009 PGDE EDUCATION**
  University Of Ibadan, Ibadan, Oyo State, Nigeria
- **2001 – 2006 BTECH ELECTRICAL ELECTRONICS ENGINEERING**
  Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria
- **1993 - 1999 WASSCE's (A* - C) including English and Mathematics**
  L.A Secondary Grammar School, Ipetumodu, Osun State

C. Work Experience: With Dates February 2012–Till date. New Horizons systems solution Business manager/Database Administrator

- Managing networking department of New Horizons Bowen center
- Promoting and developing business growth through training Bowen students on various IT Professional courses like Oracle database, Microsoft office packages CCNA, configuration of routers, switches and other networking devices
- Facilitating smooth relationship between Bowen our client and the organization

July 2010 - Vanfrank Limited, Akin Osiyemi Street, Ikeja Lagos Business manager/Administrator.

- Managing the client's equipment for smooth operations   Administering and maintaining Microsoft windows servers.
- Deploying new IT equipment when necessary
- Routine maintenance of server room, antivirus and other security software on the server and client systems.
- Managing the server room of ACCA office

September 2009 –July 2010. New Horizons, 5, Babatola Drive, Off Awolowo way, Ikeja Lagos

Database Administrator/Instructor

Instructing students in the following courses:

- ORACLE 10G (OCA &OCP)
- Microsoft Access for database administration
- Dreamweaver for website designing
- Microsoft packages (word, excel, PowerPoint)

February 2008 – September 2009 (ICT Representative) KarRox Technologies, Mokola Ibadan. ICT Instructor.

Instructing student in the following courses

- Microsoft Access for database design
- SQL for database administration

March 2007 – February 2008 Wema Bank PLC, Wema Tower, Marina Lagos (ATM Administrator/Technical support Engineer)

- Administering ATM transaction on the SQL server in conjunction with Interswitch network
- Adding route on the router whenever there is an integration of a new network on the network
- Managing users on the Wema bank domain.

April 2005 – October 2005 (Technical support Engineer) HYPERIA Motor phone) Saka Tinubu, Vitoria Island, Lagos.

- Supporting clients for dial up and internet subscription by troubleshooting using a monitoring software called WATSUP GOLD
- Installation of VSAT (C band or KU band) in client sites

April 2003 – September 2003 (Long distance communication/Technical support Engineer) NITEL Telecommunication (Internship)

- Monitoring of long distant Signals on a monitoring software
- Maintenance of NITEL generator room and other gadget including AC, monitoring

D. Names and Addresses of Referees 1.Mr Ologbon Seun Bayo

Network administrator
New horizons plc
Adamasingba, Ibadan.
TEL:07062556013

2. **ENGR Peter Fasina**
Huawei Technology
Ibadan Office, Ibadan.

3. **Reverend Ademuyiwa**
Alma Rohm Baptist church
Ile marun, Iwo, Osun State
TEL:08034731026

The University Compliance Certification

This is to certify that this thesis written by Temilade Temitope FASHINA with the Matriculation

Number LCU/PG/002661 in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approved University's format and style.

……………………………… ……………………………..

Signature Date

Temilade_Temitope_FASINA_LCU LIBRARY

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	academic.oup.com Internet Source	<1%
2	businessdocbox.com Internet Source	<1%
3	www.research-collection.ethz.ch Internet Source	<1%
4	"Handbook on Public and Private Security", Springer Science and Business Media LLC, 2023 Publication	<1%

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		