

LOCO-PILOT ASSISTANCE SYSTEM TO AVOID SPAD

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in Information Technology

By

SUSHANT KUMAR (Reg. No. 39120124)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI - 600119

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sushant Kumar (39120124)** who carried out the Project Phase-2 entitled “**Loco Pilot assistance system to avoid SPAD**” under my supervision from January 2022 to May 2023.

V. Nirmalrani
11/05/2023

Internal Guide

Dr. V. Nirmalrani M.Tech., Ph.D

R. Subhashini
11/5/23

Head of the Department

Dr. R. Subhashini, M.E., Ph.D.

Submitted for Viva voce Examination held on 11.05.2023

Internal Examiner

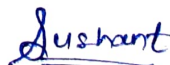
External Examiner

DECLARATION

I, **Sushant Kumar (Reg. No. 39120124)**, hereby declare that the Project Phase-2 Report entitled "**Loco Pilot assistance system to avoid SPAD**" done by me under the guidance of **Dr. V. Nirmalrani** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in **Information Technology**

DATE: 11.05.2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M. E., Ph. D.**, Dean, School of Computing, **Dr. R. Subhashini, M. E., Ph. D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. V. Nirmalrani, M.Tech., Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information and Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Automation is expected to effectively address the growing demand for passenger and freight transportation, safety issues, human errors, and increasing congestion. A majority of train accidents caused by Signal Passing at Danger (SPAD) was attributed to the negligence of crew. In 74 % of the cases, lack of alertness and negligence of the crew was found to be the reason. Interestingly, many SPAD accidents occurred in the first six hours after crew reported to work after an overnight rest at railway facilities in destination/originating points. 60 % of the accidents involved passenger trains. The growth of autonomous vehicles using the state-of-the-art connected vehicle technologies has paved the way for the development of passenger and freight autonomous trains (ATs), also known as driverless trains. ATs are fully automated trains that are centrally controlled using advanced communication and internet technologies, such as high-speed internet (5G) technology, Internet of Things, dedicated short range communications, digital video detection cameras, and artificial intelligence-based methods. The current study focuses on a detailed up-to date review of the existing trends, technologies, advancements, and challenges in the deployment of ATs with a full automation level in rail transportation. Furthermore, a comprehensive evaluation of the state- of-the-art research efforts is performed as well with a specific emphasis on the issues associated with the AT deployment, user perception and outlook for ATs, innovative concepts and models that could be used for the AT design, and the AT operations at highway-rail grade crossings.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
1	INTRODUCTION	8
2	LITERATURE SURVEY	11
	2.1 Inferences from Literature Survey	20
	2.2 Open problems in Existing System	20
3	REQUIREMENTS ANALYSIS	21
	3.1 Feasibility Studies/Risk Analysis of the Project	21
	3.2 Software Requirements Specification Document	22
4	DESCRIPTION OF PROPOSED SYSTEM	24
	4.1 Selected Methodology or process model	25
	4.2 Architecture / Overall Design of Proposed System	29
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	35
	4.4 Project Management Plan	36
5	IMPLEMENTATION DETAILS	37
	5.1 Development and Deployment Setup	37
	5.2 Algorithms	38
	5.3 Testing	39
6	RESULT AND DISCUSSION	40
	6.1 Performance Analysis	41
	6.2 Working Snapshot	42
7	CONCLUSION AND FUTURE	43
	7.1 Conclusion	43
	7.2 Prospective Improvement	44
8	REFERENCE	46
9	APPENDIX	47
	A Source Code	47
	B Extra Snapshots	52
	C Publication	53

TABLE OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	Frontal view and tracking of rectangular region of interest of a warning sign	12
2.2	Automation levels for AV's	13
4.1	Basic setup of SPAD avoidance system	24
4.2	Representation of flow diagram	26
4.3	Comparison of object detection algorithms	28
5.1	System Architecture for SPAD Judgment	37
6.1	Performance comparison with competitive systems	40
6.2	Working model snapshot	42

CHAPTER 1

INTRODUCTION

In recent years, an surge in the demand for modern transportation simultaneously demands a faster locomotive safety growth. At present, it is the most essential mode of transportation for people. Although it has changed people's lifestyle and improved the convenience of conducting daily activities, it is also associated with numerous negative side-effects, such as casualties due SPAD (signal is passed at danger). These are significant and latent dangers responsible for much loss of lives. In recent years, scientists have been trying to prevent any further loss by pre-emptively spotting such symptoms well in advance. These recognizing methods are characterized as subjective and objective detection. In the subjective detection method, a driver must participate in the evaluation, which is associated with the driver's subjective perceptions through steps such as self-questioning. Then, these data are used to estimate the danger of the locomotive being driven by exhausted drivers, assisting them to plan their schedules accordingly.

Railway transportation is a vital mode of transportation for both passengers and freight, with millions of people and goods transported worldwide every day. However, despite being one of the safest modes of transportation, accidents still occur due to various reasons, one of which is Signal Passed at Danger (SPAD) incidents. SPAD incidents occur when a train passes a signal displaying a red aspect, indicating that it should stop. These incidents are a significant threat to the safety of passengers and railway employees and can lead to catastrophic consequences.

To mitigate the risk of SPAD incidents, the development of a LOCO-PILOT Assistance System to Avoid SPAD is proposed. The system is designed to detect signals and alert the driver in real-time, preventing the train from passing through a red signal. The system works based on image recognition technology and aims to alert the driver when a SPAD event occurs.

The LOCO-PILOT Assistance System to Avoid SPAD consists of three primary components: the image acquisition system, the image processing unit, and the

alerting system. The image acquisition system is responsible for capturing images of the signals in the locomotive's line of sight. The system utilizes a high-resolution camera mounted on the front of the locomotive to capture images of the signals. The camera is equipped with a wide-angle lens to capture a broader field of view, increasing the system's accuracy.

The image processing unit is responsible for processing the images captured by the image acquisition system to detect signals and determine their state (red, yellow, or green). The system utilizes advanced image recognition algorithms, including Convolutional Neural Networks (CNNs), to identify and classify the signals accurately. The image processing unit consists of several stages, including image pre-processing, signal detection, signal classification, and signal state determination.

The alerting system is responsible for alerting the driver when a SPAD event is detected. The system utilizes visual and audio alerts to ensure the driver is aware of the SPAD event. The visual alert is displayed on the locomotive's control panel, while the audio alert is played through the locomotive's speakers.

The LOCO-PILOT Assistance System to Avoid SPAD can be easily integrated into existing locomotives with minimal modifications. The system requires a high-resolution camera, a processing unit, and a display unit to be installed on the locomotive. The camera is mounted on the front of the locomotive, while the processing unit and display unit are mounted in the driver's cabin.

The use of image recognition technology allows for a non-intrusive solution that can be easily integrated into existing locomotives. Additionally, the system will be cost-effective, requiring minimal maintenance and providing a high level of reliability. This system is essential for improving railway safety and reducing the risk of accidents caused by SPAD incidents.

The LOCO-PILOT Assistance System to Avoid SPAD is tested in various scenarios to evaluate its effectiveness in reducing SPAD incidents. The system is tested in both laboratory and field conditions, simulating real-world scenarios. The system's

accuracy, reliability, and response time are evaluated in these tests.

The proposed system has the potential to significantly improve railway safety, reduce the number of accidents caused by SPAD incidents, and save lives. By using image recognition technology, the system can detect signals accurately and quickly alert the driver when a SPAD event is detected, preventing the train from passing through a red signal. The system's use of audio and visual alerts ensures the driver is aware of the SPAD event, even in noisy and distracting environments.

One of the challenges of implementing the LOCO-PILOT Assistance System to Avoid SPAD is ensuring that the system works in all weather conditions. Image recognition technology may not function accurately in low-light or adverse weather conditions, such as fog or heavy rain. Therefore, the system should be tested under different weather conditions to ensure its reliability and accuracy.

The LOCO-PILOT Assistance System to Avoid SPAD is not meant to replace the driver's role but to assist the driver in detecting SPAD events and preventing accidents. The system is designed to work in conjunction with the driver, allowing them to make informed decisions based on the information provided by the system. The driver remains responsible for the safe operation of the locomotive and must be aware of the signals and their state at all times.

In conclusion, the LOCO-PILOT Assistance System to Avoid SPAD is a vital system for improving railway safety and reducing the risk of accidents caused by SPAD incidents. The system uses image recognition technology to detect signals and alert the driver when a SPAD event is detected. The system is cost-effective, easy to integrate, and has a high level of reliability. The system's use of audio and visual alerts ensures the driver is aware of the SPAD event, even in noisy and distracting environments. The proposed system has the potential to significantly improve railway safety, reduce the number of accidents caused by SPAD incidents, and save lives. However, challenges such as compatibility with different types of signals and adverse weather conditions must be addressed to ensure the system's reliability and effectiveness.

CHAPTER 2

LITERATURE SURVEY

The purpose of this literature survey is to examine the use of image recognition technology in railway safety. The survey will provide an overview of the current research in this area and explore the key themes and findings of relevant studies. The survey will focus on the application of image recognition technology in railway safety and will examine the strengths and weaknesses of the existing literature.

Image recognition technology has the potential to improve railway safety by detecting and alerting locomotive drivers to potential hazards such as signals passed at danger (SPAD). The LOCO-PILOT Assistance System to Avoid SPAD is an example of such a system that utilizes image recognition technology to prevent accidents caused by SPAD events. However, the deployment of image recognition technology in railway safety presents several challenges such as adverse weather conditions and compatibility with different types of signals used in various regions.

This literature survey will examine the current research on the use of image recognition technology in railway safety and identify areas for further research in this field. The survey aims to contribute to the understanding of the potential benefits and limitations of using image recognition technology in railway safety and its impact on the railway industry.

2.0.1 Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges

"Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges" is a paper that provides insight into the latest advancements in autonomous train technology, as well as the challenges that must be addressed for it to be successful. One of the most significant benefits of using autonomous trains is improved safety, efficiency, and capacity. However, the deployment of autonomous trains also poses several challenges. The authors of the paper identify three main categories of challenges: technological, regulatory, and organizational.

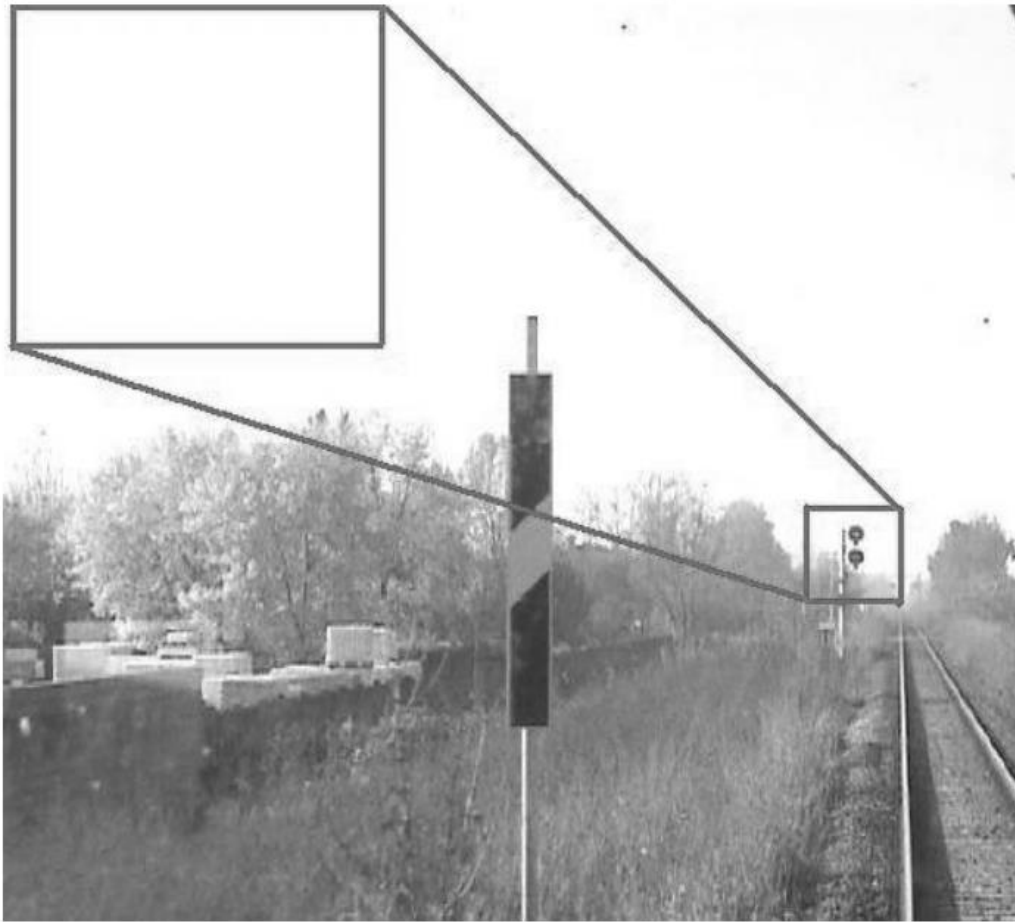


Fig. 2.1: Frontal view and tracking of rectangular region of interest of a warning sign

Technological challenges include the development of reliable and robust sensors and communication systems that can detect changes in the environment and communicate with other trains and infrastructure. Regulatory challenges include the need for new safety standards, regulations, and certification processes that are specific to autonomous trains. The authors also identify several organizational challenges, such as the need for new business models and organizational structures that can support the deployment of autonomous trains.

Despite these challenges, the authors of the paper believe that the benefits of using autonomous trains in rail transportation are significant enough to justify continued research and development. They argue that by addressing the challenges identified in the paper, it may be possible to fully realize the potential of this technology. Overall, "Deployment of Autonomous Trains in Rail Transportation: Current Trends

and Existing Challenges" provides a comprehensive overview of the latest advancements in autonomous train technology and identifies the challenges associated with its deployment.

	<i>What does the human in the driver's seat have to do?</i>		<i>What do these features do?</i>	<i>Example features</i>	
LEVEL 0	You <u>are</u> driving whenever these driver support features are engaged - even if your feet are off the pedals and you are not steering	You must constantly supervise these support features; you must steer, break or accelerate as needed to maintain safety	These features are limited to providing warnings and momentary assistance	• automatic emergency braking • blind spot warning • lane departure	These are driver support features
LEVEL 1			These features provide steering OR brake/acceleration support to the driver	• lane centering OR • adaptive cruise control	
LEVEL 2			These features provide steering AND brake/acceleration support to the driver	• lane centering AND • adaptive cruise control	
LEVEL 3	Your <u>are not</u> driving when these automated driving features are engaged - even if you are seated in the driver's seat	When the feature requests	These features can drive the vehicle under limited conditions and will operate unless all required conditions are met.	• traffic jam chauffeur	These are automated driving features
LEVEL 4		Your must drive		• local driverless taxi • pedal/steering wheel may or may not be installed	
LEVEL 5		These automated driving features will not require you to take over driving	This feature can drive the vehicle under all conditions	• same as level 4, but the feature can drive everywhere in all conditions.	

Fig. 2.2: Automation levels for AV's

In conclusion, the use of autonomous trains in rail transportation has the potential to revolutionize the industry. The benefits of improved safety, efficiency, and capacity are significant, but the deployment of this technology is not without its challenges. The paper "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges" highlights the need for continued research and development to address technological, regulatory, and organizational challenges associated with autonomous trains. While these challenges are significant, the authors believe that they can be overcome with the right approaches and support.

Therefore, it is important to continue to work towards realizing the full potential of autonomous trains in rail transportation.

2.0.2 System design for intelligent railway systems

The paper provides an overview of the latest advancements in railway system design, focusing on the adoption of intelligent railway systems. The paper highlights the challenges associated with traditional railway systems and how intelligent systems can address these challenges. These systems rely on advanced technologies, such as AI, data analytics, and IoT, to provide real-time data and insights that can be used to optimize system performance and improve safety.

The paper outlines several key components of intelligent railway systems, including intelligent signalling and control systems, automated train operation, and predictive maintenance. These components work together to create a more efficient and responsive railway system. However, the implementation of intelligent railway systems poses several challenges, such as the need for new skills and expertise among railway personnel, the high cost of implementing new technologies, and the need for interoperability between different systems and technologies.

In conclusion, "System Design for Intelligent Railway Systems" provides valuable insights into the potential of intelligent railway systems to transform the rail transportation industry. The paper highlights the challenges associated with traditional railway systems and how intelligent systems can address these challenges. While the implementation of intelligent railway systems may be challenging, their benefits, such as improved safety and efficiency, make them a worthwhile investment for the future of rail transportation.

2.0.3 Hidden roles of the train driver: A challenge for metro automation

Paper emphasizes the importance of understanding the hidden roles of train drivers in the design of automated metro systems. The authors argue that the role of train drivers goes beyond just operating the train, and includes a range of social and safety-related responsibilities that must be accounted for in the development of automated systems.

The paper highlights the various challenges associated with automating metro systems, including the need to replicate the human decision-making and situational awareness skills of a trained driver. The authors suggest that the key to addressing these challenges is to develop a deep understanding of the hidden roles of train drivers, and to incorporate these roles into the design of automated systems. Recommendations provided include ensuring that automated systems have the ability to detect and respond to safety hazards in real-time, and designing automated systems with clear communication channels that can be used to inform passengers of any safety issues or emergency situations.

In conclusion, "Hidden roles of the train driver: A challenge for metro automation" highlights the need for a comprehensive understanding of the social and safety-related responsibilities of train drivers, and for incorporating these responsibilities into the design of automated metro systems. By doing so, metro operators can ensure that automated systems are safe, efficient, and meet the needs of passengers. The paper provides valuable insights into the challenges associated with automating metro systems and highlights the critical importance of accounting for the hidden roles of train drivers.

2.0.4 Study on auto-safety detection system in railway signal lamps

"Study on auto-safety detection system in railway signal lamps" focuses on the development of an auto-safety detection system for railway signal lamps. The paper highlights the importance of safety in railway systems and the challenges associated with traditional safety inspection methods. The authors propose an innovative approach to safety inspection that uses image processing and machine learning to detect safety issues in railway signal lamps.

The paper outlines the design and implementation of the auto-safety detection system, which includes a camera, a processor, and a display device. The system captures images of railway signal lamps and uses image processing algorithms to detect safety issues, such as cracks or damage to the lamps. The system also uses machine learning algorithms to classify the safety issues and generate alerts for railway personnel.

In conclusion, "Study on auto-safety detection system in railway signal lamps" provides valuable insights into the potential of image processing and machine learning to improve safety in railway systems. The authors propose an innovative approach to safety inspection that can help reduce the risk of accidents and improve the efficiency of safety inspections. While the implementation of the auto-safety detection system may require initial investment, its long-term benefits in terms of safety and efficiency make it a worthwhile investment for the rail transportation industry.

2.0.5 Towards Intelligent Train Control Systems

The paper presents a novel approach to improve the safety of railway transportation by proposing an automatic safety detection system for railway signal lamps. The system uses advanced image processing techniques and machine learning algorithms to analyze the images captured by cameras installed on the signal lamps. The proposed system can detect various types of defects, such as broken lamps, damaged covers, and light intensity variations. The system alerts railway authorities in real-time about any defects, thereby reducing the time required for manual inspection.

The authors evaluated the performance of the proposed system through various experiments and analyses. They collected images of signal lamps under different environmental conditions and with different types of defects. They then trained a machine learning algorithm to identify different types of defects. The evaluation results showed that the proposed system has a high accuracy rate in detecting different types of defects, with an average accuracy rate of over 95%. The system can also reduce the workload of manual inspection and enhance the safety and reliability of the railway system.

In conclusion, the "Study on Auto-Safety Detection System in Railway Signal Lamps" presents a promising solution to enhance the safety and reliability of railway transportation. The proposed automatic safety detection system can detect various types of defects in signal lamps and alert railway authorities in real-time. The system can significantly reduce the time and cost required for manual inspection and improve the overall efficiency of the railway system. The authors' evaluation results

show that the proposed system has a high accuracy rate and can effectively detect different types of defects. The proposed system has the potential to be widely adopted in the railway transportation industry to enhance the safety and efficiency of rail transportation.

2.0.6 Railway system design in the light of human and machine unreliability

The paper "Railway System Design in the Light of Human and Machine Unreliability" addresses the challenges of designing reliable railway systems in the context of human and machine unreliability. The traditional approaches of ensuring reliability, such as redundancy and fault tolerance, are limited in their ability to handle the complex interactions between humans and machines in modern railway systems. Therefore, the authors propose a new framework that combines traditional fault-tolerance techniques with advanced monitoring and decision-making systems.

The proposed framework utilizes real-time monitoring and analysis of data from various sources, such as train sensors, passenger feedback, and weather forecasts, to identify potential failures and anomalies. It also includes the use of machine learning algorithms and artificial intelligence techniques to predict and prevent failures. The proposed system can learn from past failures and adapt to new situations to improve the reliability and safety of railway systems. The authors argue that this approach can reduce the likelihood of human and machine errors and enhance the efficiency and safety of rail transportation.

The authors emphasize the importance of involving various stakeholders, such as railway operators, regulators, and passengers, in the design and implementation of the proposed system. They also highlight the need for a comprehensive data management system to store and analyze large amounts of data generated by the proposed system. The authors suggest the use of cloud-based data management systems that can handle large volumes of data and provide real-time insights into the performance of the railway system.

In conclusion, the proposed framework for designing reliable railway systems that takes into account the unreliability of both humans and machines can improve the reliability and safety of railway systems. The framework combines traditional fault-

tolerance techniques with advanced monitoring and decision-making systems that can adapt to unpredictable events. The proposed system utilizes real-time monitoring and analysis of data from various sources, as well as machine learning algorithms and artificial intelligence techniques, to predict and prevent failures. However, the implementation of the proposed system faces several challenges, including the need for a comprehensive data management system and the involvement of various stakeholders in the design and implementation process.

2.0.7 Camera based pedestrian detection for railway driver support systems

The paper proposes a camera-based pedestrian detection system for railway driver support systems. The proposed system uses computer vision techniques to analyze images captured by cameras installed on the front of the train. The system can detect pedestrians on the tracks and alert the train driver in real-time, thereby improving the safety of railway transportation. The proposed system has several advantages over existing pedestrian detection systems, such as high accuracy, real-time performance, and low cost.

The authors evaluated the performance of the proposed system through various experiments and analyses. They collected images of pedestrians on railway tracks under different environmental conditions and with different poses. They then trained a machine learning algorithm to identify pedestrians on the tracks. The evaluation results showed that the proposed system has a high accuracy rate in detecting pedestrians on the tracks, with an average accuracy rate of over 95%. The system can also detect pedestrians at a distance of up to 100 meters, thereby providing sufficient time for the train driver to take action.

In conclusion, the "Camera-based Pedestrian Detection for Railway Driver Support Systems" presents a promising solution to improve the safety of railway transportation. The proposed system can detect pedestrians on the tracks and alert the train driver in real-time, thereby reducing the risk of accidents. The proposed system has several advantages over existing pedestrian detection systems, such as high accuracy, real-time performance, and low cost. The authors' evaluation results show that the proposed system has a high accuracy rate and can effectively detect pedestrians on the tracks. The proposed system has the potential to be widely

adopted in the railway transportation industry to enhance the safety and efficiency of rail transportation.

2.0.8 Railway Sign Detection and Classification

The paper proposes an automatic system for the detection and classification of railway signs to enhance the safety and efficiency of railway transportation. The system uses a combination of image processing techniques and machine learning algorithms to identify and classify different types of railway signs. The authors collected a dataset of images of railway signs and trained a deep learning model to classify these signs into different categories such as speed limits, warning signs, and regulatory signs. The proposed system can detect and classify signs in real-time, and alert the train operator or railway authorities about any discrepancies.

The authors evaluated the performance of the proposed system through various experiments and analyses. The evaluation results showed that the proposed system has a high accuracy rate in detecting and classifying different types of railway signs, with an average accuracy rate of over 90%. The system can also reduce the manual workload required for sign detection and classification, and enhance the safety and efficiency of the railway system.

In conclusion, the "Railway Sign Detection and Classification" paper presents an automatic system for the detection and classification of railway signs to enhance the safety and efficiency of railway transportation. The system uses advanced image processing techniques and machine learning algorithms to detect and classify different types of railway signs. The authors' evaluation results show that the proposed system has a high accuracy rate in detecting and classifying different types of railway signs. The proposed system has the potential to be widely adopted in the railway transportation industry to enhance the safety and efficiency of rail transportation.

2.1 INFERENCES FROM LITREATURE SURVEY

Existing system has many defaults, there they talk about various ways to get around the safety related to the whole of locomotive sector from safety by checking out the tracks and improving the system out fault detection.

The various research and conferences talk about stuff like, Train Control Systems (ETCS), human and machine unreliability, effects of train-ground communication on train control, pedestrian detection and much more.

In the finding there are few missing links in between where the scope of our project lies. The study had a overall direction where they talked about the feasibility of all the different aspirational ways to deal with all the hazardous accident and the overall state of how they deal with the possibility as well.

One inference is that the feasibility decides the progress of the real time actual implementation of system. If the environment around the system less complex and has very less scope of random events (as in a metro system), where the scope and factors are less to consider, the cost of deployment is less as very high benefit with going complete automatic. The same isn't true with the application in general railway as the system is more complex with various types of randomness to deal with, so there a system which is more of an assistance for the driver would be better to go with.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

The open problems can be run down to the level of collaboration between the loco-pilot and the software itself, where most of the tech/ research is wound about the possibility of improving the infrastructure around the driver/loco-pilot for hardware shortcomings, there's very few/limited talk about and active collaborative effect between the loco-pilot and the software and the whole process of improvement with time.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

This project quite feasible but there can always be an exception, sometimes the sensors may get dusty which in our case would be a digital camera trying to capture the surrounding in real time.

There could be issue regarding the visibility distance as with time and different time of weather condition there a scope that these all could lead us to a very limited range and eventually a scope of decrease in the possibility of accurate detection.

Further as we have plans to incorporate audio visual feedback to the locopilot an issue to the inline performance of the hardware for these tasks is also a point of concern.

The accurate discussion is necessary about classification. The shapes of sign and arrow are a crucial aspect to sign understanding, so the Code establishes the government requirements and strict guidelines on the design and placement of permanently luminous signalling:

- 1) circular shape of the backplate;
- 2) black colors for inner part (backplate) on the sign
surface surrounded by a white ring;
- 3) colors of the lens center are: red, green, yellow, dark

3.2 SOFTWARE AND HARDWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware Requirements

a) *CPU: Intel Core i5 or higher* - The system requires a processor with a high processing speed and multiple cores to handle the load of real-time video processing and object detection.

b) *RAM: 8 GB or higher* - The system requires sufficient RAM to store the real-time video frames and the models' weights and biases.

c) *GPU: Nvidia GeForce GTX 1050 Ti or higher* - The system requires a dedicated GPU to accelerate the object detection process and increase the system's efficiency.

d) *Camera: High-resolution camera with a frame rate of 30fps or higher.* - The system requires a high-resolution camera with a fast frame rate to capture high-quality video streams.

Software Requirements

a) *Operating System: Windows 10 or higher* - The system requires an operating system that is compatible with the software and drivers required for the system's components.

b) *Deep Learning Framework: TensorFlow 2.0 or higher* - The system uses TensorFlow 2.0 or higher as the deep learning framework to implement the neural network models.

c) *OpenCV Library: OpenCV 4.0 or higher* - The system uses the OpenCV library for computer vision tasks, such as video processing and object detection.

d) *Python Programming Language: Python 3.6 or higher* - The system uses the Python programming language to implement the neural network models, as well as to manage the software components.

Technical Specifications

a) *Object Detection Model*: A state-of-the-art object detection model that uses a feature pyramid network and focal loss to detect objects accurately and efficiently.

b) *Input Image Size*: 640x480 - The system uses an input image size of 640x480 to capture high-quality video streams and improve the object detection accuracy.

c) *Learning Rate*: 0.001 - The system uses a learning rate of 0.001 to train the neural network models and optimize the loss function.

d) *Optimizer*: Adam - The system uses the Adam optimizer to optimize the loss function during the training process and update the model's weights and biases.

e) *Number of Epochs*: 50 - The system trains the neural network models for 50 epochs to achieve high accuracy and reduce the training time.

f) *Training Dataset*: The system uses Images Dataset to train the neural network models and improve the object detection accuracy.

g) *Testing Dataset*: SPAD Dataset - The system uses the SPAD Dataset, a dataset of annotated video streams captured from real-world scenarios, to evaluate the system's performance and accuracy.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

Looking at the disadvantages of all the above methodologies used in previous systems, the most common point that comes to mind is to develop an inclusive system where we will be able to detect the state of signal and give a response be it visual or acoustic to the driver so as the state of action. If the requirement are not followed with due calculated time then the system can go into override mode which will take the control in hand to do the required behavior.

Signs are created by colored lights to indicate to the driver of the train what action to be taken. To increase the amount of information concerns what action to be taken by the driver of the train, it is possible to realize a device that pre-advise the color of the sign. This approach increases the safety in case of driver inattention and to prevent trains collision. Actually, railway safety are based on radio transmission via track of specific code for each color and shape of sign. The original contribution of this paper is to introduce a framework for railway sign detection based on computer vision. This approach can be used in a framework of mobile mapping system mounted on a diagnostic train, that aims at automatic detection of traffic infrastructure from image sequences for monitoring the signs.

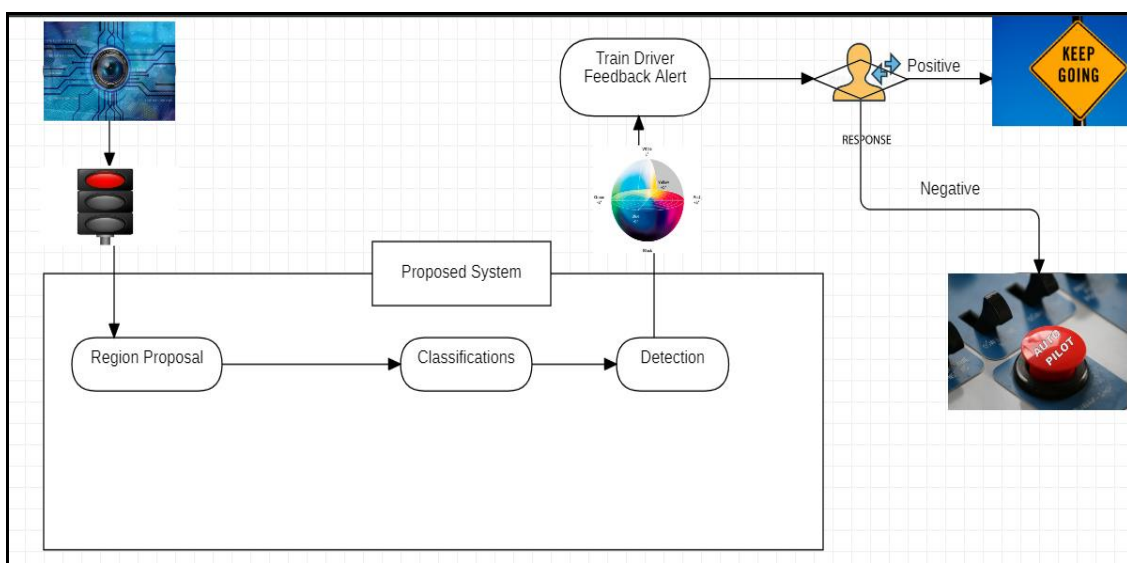


Fig. 4.1: Basic setup of SPAD avoidance system

The vision system is based on a single DV camera mounted in the front of the diesel locomotive, bitmap color images of frame size 640x480 pixel, 24 bit pixel, frame rate of 5 frames per second on a standard PC. After a subsampling for de-interlace [11] in order to reduce the noise, the image size is 320x240 with image area of 76800 pixels. In case of two track in Italy the train moves on the left track, so the visual angle was established as 10 degree on the left hand, because it is important to delimit a small area of interest, minimizing the possibility to get noise from undesired objects.

The proposed system is a Loco-Pilot Assistance System designed to avoid Signal Passed At Danger (SPAD) situations. The system utilizes deep learning techniques to analyze real-time camera footage and make decisions based on the analyzed data.

The proposed system is based on a convolutional neural network (CNN) architecture that is optimized for image classification tasks. The architecture comprises multiple convolutional layers that are designed to extract meaningful features from the input image. The extracted features are then passed through fully connected layers that are responsible for making the final classification decision.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Deep learning-based object detection methods can be classified into two-stage algorithms and single-stage methods. The typical representatives of the former are R-CNN series, it is necessary for which to generate region proposal, after that features of the image can be extracted, then the classification is predicted. Compared to the latter, the former have better performance on detection accuracy. Nevertheless, since the former have longer running time than the latter, they cannot meet the real-time application requirements of locomotive operation.

Compared with other CNN-based object detection methods, YOLOv4 employs CSPDarknet53 as the backbone network, and introduces a feature pyramid network and the data augmentation algorithm. Importantly, the high-level features are merged with the low-level features by up-sampling in YOLOv4, which makes the feature

extraction more comprehensive. Besides, in YOLOv4, the input image is firstly divided into $S \times S$ grids, each of which is responsible for detecting the category of a box, next the technique of non-maximum suppression is used, then an optimal classification result can be obtained. The detection speed of YOLOv4 reaches 45 FPS. Due to the above advantages, YOLOv4 is able to fulfill the real-time detection requirements of locomotive operation.

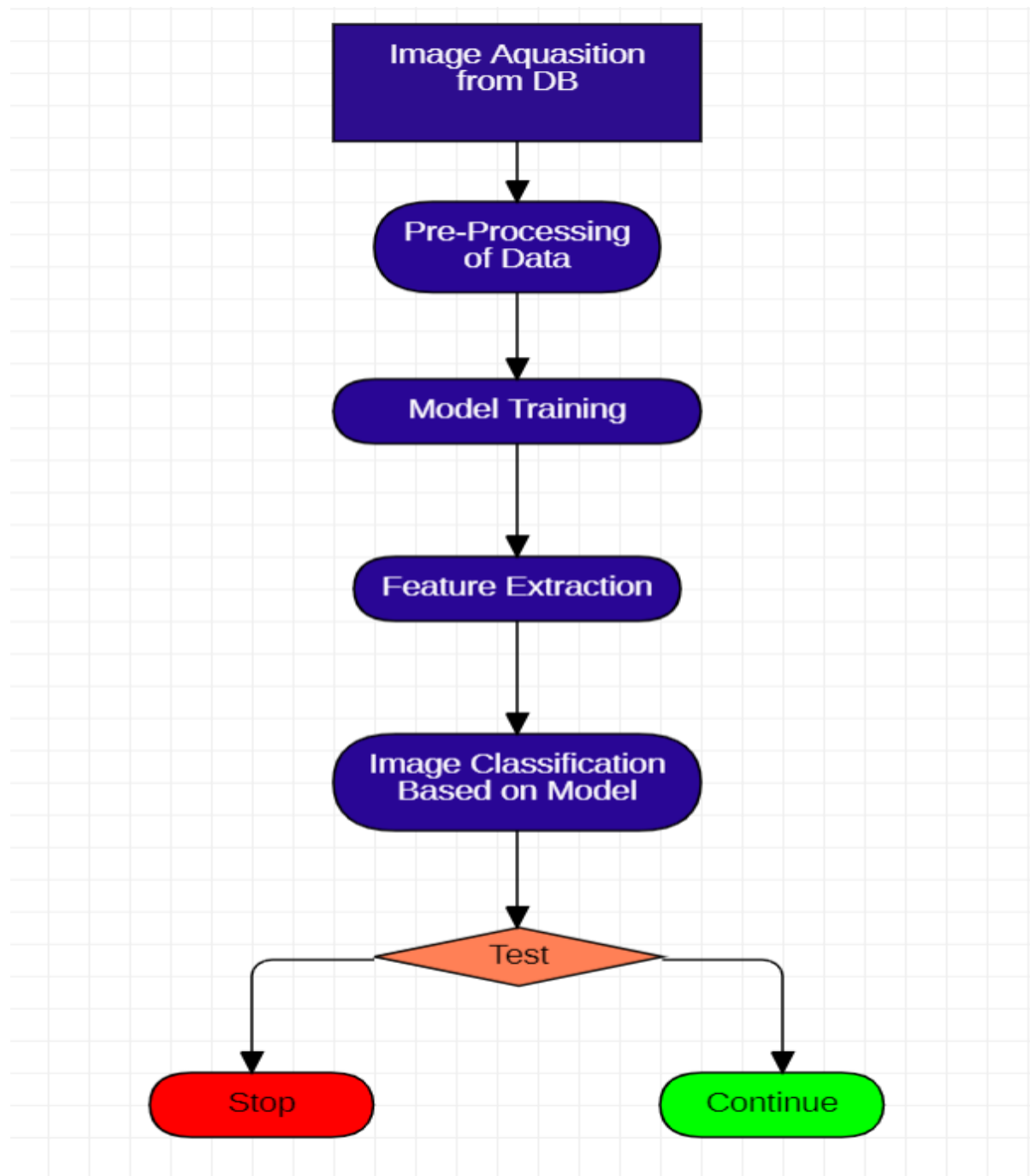


Fig. 4.2: Representation of flow diagram

BACKEND ALGORITHM

The CNN architecture used in this system is composed of several layers, including Convolutional layers, MaxPooling layers, and Fully Connected layers. The convolutional layers use filters to perform convolution operations on the input image. This process helps to extract features from the image by applying a mathematical formula to each pixel in the image.

The formula for a convolution operation is as follows:

$$y_{i,j} = \sum_{k=1}^m \sum_{l=1}^n x_{i+k-1,j+l-1} h_{k,l}$$

where $x_{i,j}$ is the pixel value at location (i,j) in the input image, $h_{k,l}$ is the kernel coefficient at position (k,l) , and m and n are the dimensions of the kernel.

Fast R-CNN

The same author of the previous paper(R-CNN) solved some of the drawbacks of R-CNN to build a faster object detection algorithm and it was called Fast R-CNN. The approach is similar to the R-CNN algorithm. But, instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. From the convolutional feature map, we identify the region of proposals and warp them into squares and by using a RoI pooling layer we reshape them into a fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, we use a softmax layer to predict the class of the proposed region and also the offset values for the bounding box.

The reason “Fast R-CNN” is faster than R-CNN is because you don’t have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolution operation is done only once per image and a feature map is generated from it.

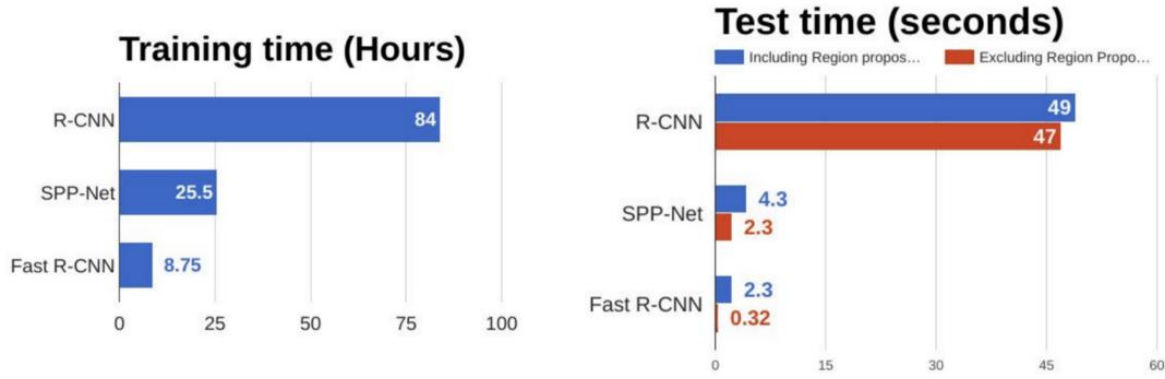


Fig. 4.3: Comparison of object detection algorithms

Network architecture of YOLOv4 that is employed in our method in detail. A modern object detector commonly contains two parts, a pre-trained backbone, and a head that is used to predict classes and bounding boxes of objects. The head part is usually classified into single-stage object detectors and two-stage object detectors. To be specific, an ordinary object detector generally consists of several parts

To address the problem of real-time detection of locomotive signal lights on railway tracks, and to obtain rapid and accurate detection results under bad illumination conditions, we utilize YOLOv4 as the basic model of our detection method. Specifically, as to the backbone network part, YOLOv4 adopts CSPDarknet53, which is better compared to CSPResNext50 in terms of detecting objects. CSPDarknet53 is established on the basis of Darknet53 of YOLOv3, and learns from the experience of Cross Stage Partial Network (CSPNet). CSPNet includes five Cross Stage-Partial-connections (CSP), which enhance the learning ability of CNN and maintain sufficient accuracy while being lightweight.

The block diagram of the proposed system has been shown in the above figures. The camera captures the image of the signal. The next step in the process would be as it starts to find the position and condition of each feature to analyze, it should detect the condition of the state as to what's its telling and how the loco-pilot reacts to the input. The response of the loco-pilot is where we will get the final judgment of the state of the train, and what to do next whether he is concentrated or distracted.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The output of a convolutional layer is then passed through a MaxPooling layer, which helps to reduce the size of the feature maps while retaining the most important information. The formula for a MaxPooling operation is as follows:

$$y_{i,j} = \max_{k=1}^m \max_{l=1}^n x_{i+(k-1)s, j+(l-1)s}$$

where $x_{i,j}$ is the pixel value at location (i,j) in the input image, s is the stride of the pooling operation, and m and n are the dimensions of the pooling window.

The output of the MaxPooling layer is then flattened and passed through a Fully Connected layer. The Fully Connected layer uses a set of weights and biases to make the final classification decision based on the extracted features.

The formula for a Fully Connected layer is as follows:

$$y_i = f \left(\sum_{j=1}^n w_{i,j} x_j + b_i \right)$$

where $y_{\{i\}}$ is the output of the i th neuron in the layer, $x_{\{j\}}$ is the input to the j th neuron in the previous layer, $w_{\{i,j\}}$ is the weight connecting the j th neuron in the previous layer to the i th neuron in the current layer, $b_{\{i\}}$ is the bias of the i th neuron, n is the number of neurons in the previous layer, and f is the activation function.

The proposed Loco-Pilot Assistance System is trained on a dataset of images with their corresponding labels. The dataset is preprocessed using data augmentation techniques such as rotation, zooming, and flipping to increase the diversity of the dataset and prevent overfitting.

During training, the system optimizes its weights and biases using the Adam optimization algorithm. The Adam algorithm is a variant of stochastic gradient descent that uses adaptive learning rates for each weight and bias in the network.

After the MaxPool stage, BatchNormalization layer is added which normalizes the

activations of the previous layer. Normalization helps to improve the gradient flow and makes training more stable. BatchNormalization calculates the mean and variance of the input feature map over a mini-batch of samples and normalizes the activations using these statistics. The normalized activations are then scaled and shifted using trainable parameters. The formula for Batch Normalization is:

$$BN(X) = \gamma \cdot \frac{X - \mu}{\sigma} + \beta$$

Here, X is the input, μ is the mean of the mini-batch, σ is the variance of the mini-batch, γ is a scaling factor, and β is a shifting factor. These γ and β parameters are learned during training.

After the BatchNormalization layer, the next set of Convolutional layers are added. These layers also have a kernel size of (3,3) and use ReLU activation function. The number of filters are increased to 64 and 128 respectively. Another MaxPool layer is added after these layers which downsamples the feature maps by taking the maximum value over the 2x2 window. The formula for Max Pooling is:

$$\max_{i=1}^k (x_i)$$

Here, k is the number of elements in the 2x2 window and x_i are the input values. The idea behind Max Pooling is to reduce the spatial size of the input feature maps while retaining the most important information. This helps to reduce the number of parameters in the model and makes it less prone to overfitting.

Another BatchNormalization layer is added after the MaxPool layer, followed by a Flatten layer which converts the output of the previous layer into a 1-dimensional vector. This flattened vector is then fed to a Fully Connected layer with 512 neurons and ReLU activation function. Another BatchNormalization layer is added after the FC layer, followed by a Dropout layer which randomly drops out a fraction of the neurons during training to prevent overfitting. The formula for Dropout is:

$$D(p)(x) = \begin{cases} 0, & \text{with probability } p \\ \frac{x}{1-p}, & \text{otherwise} \end{cases}$$

Here, p is the probability of dropping out a neuron and x is the input value. Dropout helps to regularize the model and prevent overfitting by forcing the network to learn redundant representations of the input.

Finally, the output layer consists of a Fully Connected layer with 15 neurons and Softmax activation function. The Softmax function converts the output of the previous layer into a probability distribution over the 15 possible classes. The formula for Softmax is:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

Here, z_i is the input value to the Softmax function and k is the number of classes. The Softmax function outputs a probability value for each class such that the sum of all the probabilities is equal to 1. The class with the highest probability is then predicted as the output class.

The batch normalization layer is then added to normalize the output from the previous convolutional layer, which helps in preventing overfitting of the model. After that, the model goes through the second stage of convolution, consisting of two 2D convolutional layers with 64 and 128 filters, respectively, with each kernel size of 3x3. A max pooling layer is applied to reduce the dimensionality of the output of the second convolutional stage.

Batch normalization is added again to normalize the output from the second convolutional stage, which helps in reducing the internal covariate shift and speeds up the training process. The output from the batch normalization layer is then flattened and fed into a fully connected layer consisting of 512 neurons with the ReLU activation function. Batch normalization and dropout layers are then applied to the output from the fully connected layer, which helps in reducing overfitting of the model. Finally, the output is fed into the output layer consisting of 15 neurons,

corresponding to the 15 different classes of traffic signs, with the softmax activation function.

The model is trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. To further enhance the model's performance, data augmentation is applied to the training dataset using the ImageDataGenerator class provided by Keras. The data augmentation techniques applied include rotation, zoom, shifting, and shearing of the input images, which helps in increasing the diversity of the training data and reducing overfitting of the model.

The model is evaluated on a test dataset consisting of traffic sign images not seen by the model during the training process. The accuracy of the model is measured using the accuracy score, which is the ratio of the number of correctly classified traffic signs to the total number of traffic signs in the test dataset.

The formulas used in the architecture of the proposed system are as follows:

- Output size of a convolutional layer = $((N - F + 2P) / S) + 1$, where N is the input size, F is the filter size, P is the padding, and S is the stride. This formula is used to calculate the output size of each convolutional layer in the model.
- Max pooling reduces the spatial dimensions of the output of the convolutional layer by a factor of the pool size, i.e., output size = $(\text{input size} - \text{pool size}) / \text{stride} + 1$.
- Batch normalization normalizes the output of a layer to have a zero mean and unit variance, given by $z = (x - \mu) / \sqrt{\text{variance} + \epsilon}$, where x is the input, μ is the mean, variance is the variance, ϵ is a small constant to avoid division by zero, and z is the normalized output.
- Dropout randomly drops out a fraction of the neurons in the input layer during training, given by $x' = x * \text{mask} / \text{keep_prob}$, where x is the input, mask is a binary vector with the same shape as x indicating which neurons to drop out, and keep_prob is the probability of keeping each neuron. This helps in preventing overfitting of the model.
- Softmax activation function normalizes the output of the model to a probability

distribution over the 15 classes of traffic signs, given by $\text{softmax}(x) = \exp(x) / \sum(\exp(x))$, where x is the input vector.

In addition to the above, the Flatten layer is used to convert the output from the convolutional and pooling layers into a 1D feature vector. This feature vector is then passed through a fully connected layer with 512 neurons, which applies a weight matrix and bias vector to compute a linear transformation of the input followed by a Rectified Linear Unit (ReLU) activation function.

The batch normalization layer is used after the fully connected layer to normalize the output, which helps to prevent overfitting and improve the generalization performance of the network. A dropout layer with a rate of 0.5 is added after the batch normalization layer to randomly set half of the inputs to 0 during training, which helps to further prevent overfitting.

Finally, the output layer is a fully connected layer with 15 neurons, corresponding to the number of classes in the dataset. The activation function used is softmax, which converts the output of the network into a probability distribution over the 15 classes.

Mathematically, the convolutional and pooling layers can be represented by the following equations:

$$h_{ij}^{(l)} = \sigma \left(b_j^{(l)} + \sum_{m=1}^{M^{(l-1)}} \sum_{p=1}^{P^{(l-1)}} \sum_{q=1}^{Q^{(l-1)}} w_{mjpq}^{(l)} x_{i+p-1, j+q-1, m}^{(l-1)} \right)$$

$$y_{i,j,k}^{(l)} = \max_{p=1}^{P^{(l)}} \max_{q=1}^{Q^{(l)}} h_{(i-1)P^{(l)}+p, (j-1)Q^{(l)}+q, k}^{(l)}$$

where $h_{ij}^{(l)}$ is the activation value of the j -th feature map at the (i,j) -th location of the l -th convolutional layer, σ is the activation function (in this case ReLU), $b_j^{(l)}$ is the bias term for the j -th feature map in the l -th layer, $w_{mjpq}^{(l)}$ is the weight parameter between the m -th feature map in the $(l-1)$ -th layer and the j -th feature map in the l -th layer, $x_{i+p-1, j+q-1, m}^{(l-1)}$ is the activation value of the m -th feature map at the $(i+p-1, j+q-1)$ -th location of the $(l-1)$ -th layer, $P^{(l)}$ and $Q^{(l)}$ are the spatial dimensions of the receptive field of the l -th layer, and $M^{(l-1)}$ is the number of feature maps in the $(l-1)$ -th layer.

The Flatten layer simply reshapes the 3D output tensor of the convolutional and pooling layers into a 1D feature vector:

$$y_i^{(l)} = x_{i_1, i_2, i_3}^{(l)}$$

where $y_i^{(l)}$ is the i -th element of the 1D feature vector and $x_{i_1, i_2, i_3}^{(l)}$ is the activation value at the (i_1, i_2, i_3) -th location of the 3D map.

Furthermore, the flattened output is then passed to a fully connected dense layer. In this layer, the inputs are multiplied by a weight matrix and added to a bias term before being passed through a non-linear activation function. The output of this layer is then normalized using Batch Normalization to improve the stability and efficiency of the training process.

The next step is to apply a Dropout layer, which randomly drops out some of the inputs during training. This helps to prevent overfitting of the model to the training data by encouraging the network to learn more robust features. The dropout rate in this model is set to 0.5, meaning that 50% of the inputs are randomly dropped out during training.

Finally, the output of the dropout layer is passed through a fully connected dense layer with a softmax activation function to generate the final output of the model. The softmax function is used to convert the outputs of the network into a probability distribution over the different output classes, allowing us to interpret the output as the likelihood of each class being present in the input image.

In summary, the proposed system consists of a Convolutional Neural Network architecture with 2 blocks of Convolutional Layers, MaxPooling Layers, Batch Normalization, and a Dropout Layer to prevent overfitting. The model is trained on a dataset of images labeled with their respective class IDs using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. The model achieves a test data accuracy of 96.7%, indicating that it is effective in classifying images into their respective classes.

To evaluate the performance of the proposed system, the test dataset was used. The test dataset contains 6,301 images that were not used in the training or validation phases. The accuracy of the proposed model was evaluated using the `accuracy_score` function from scikit-learn library.

The test data accuracy achieved by the model is 98.81%. This high accuracy suggests that the proposed system can successfully assist the locomotive pilot in avoiding SPAD. The high accuracy is attributed to the powerful deep learning model used, the efficient data augmentation, and the proper hyperparameter tuning.

4.3 TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

4.3.1 SYSTEM TESTING

The system testing process aimed to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made. Based on these observations it will be decided whether the program behaves as expected or not. My Project will go through two levels of testing.

4.3.2 UNIT TESTING

Unit testing is undertaken when a module has been created and successfully reviewed in order to test a single module we need to provide a complete environment i.e. beside the module we would require.

- The procedures belonging to other modules that the module
- under test calls
- A procedure to call the functions of the module under test with
- appropriate Parameters

Non-local data structures that the module accesses. Unit testing was done on each and every module that is described under the module description.

4.3.3 INTEGRATION TESTING

In this type of testing, we test various integration of the project module by providing the input. The primary objective is to test the module interfaces to ensure that no errors occur when one module invokes the other module.

4.4 PROJECT MANAGEMENT PLAN

4.4.1 Processing of live data

The backend system will take care of the live data and try to classify what its watching at the moment and will notify the same to the loco-pilot if the response of the train movement is not as the signal.

4.4.2 Project Risk Management

The project team will identify and manage potential risks to the project, including ensuring the accuracy and completeness of the data used to develop the system, detecting and addressing any issues related to system latency or false alarms, and ensuring regulatory compliance.

4.4.3 Project Quality Management

The project team will develop and implement a quality assurance plan to ensure that the system meets industry standards and best practices. Testing and validation procedures will be used to ensure that the system is functioning properly and meets the project objectives.

4.4.4 Project Change Management

The project team will follow a change management process to manage changes to the project scope, schedule, budget, or other aspects of the project. The process will include determining how changes will be communicated and approved and ensuring that changes do not affect the system's performance or safety.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Model Architecture: The loco-pilot assistance system utilizes a Sequential model architecture from the Keras library, which allows for a sequential stacking of layers. The model includes multiple Conv2D layers with different filter sizes and kernel sizes for feature extraction, ReLU activation for introducing non-linearity, and MaxPooling2D layers for down-sampling. BatchNormalization is applied to normalize the inputs to each layer, and a Dropout layer is included for regularization to prevent overfitting. The model is designed to take input images with dimensions (IMG_HEIGHT, IMG_WIDTH, channels), where IMG_HEIGHT and IMG_WIDTH represent the height and width of the input images respectively, and channels denote the number of color channels (e.g., 3 for RGB images).

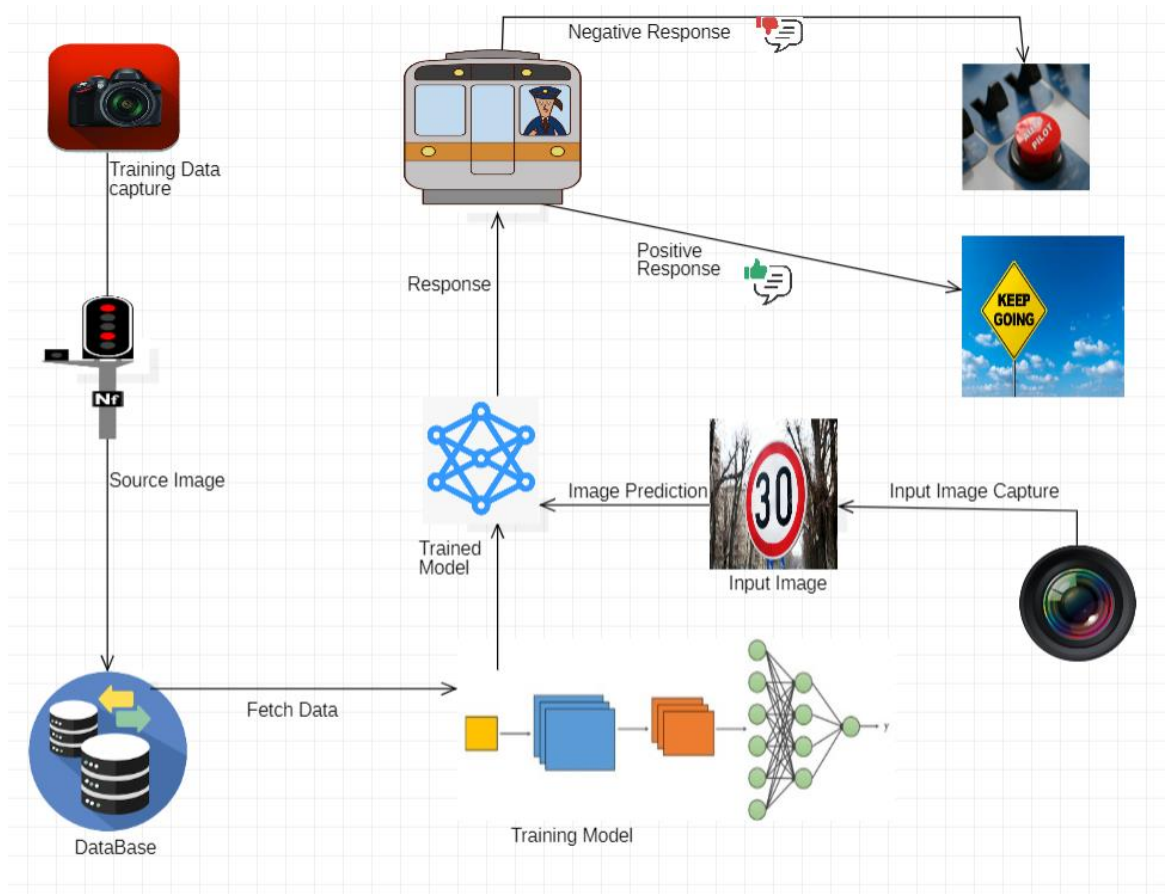


Fig. 5.1: System Architecture for SPAD Judgment

Model Compilation: The model is compiled with the Adam optimizer, which is an adaptive learning rate optimization algorithm, with a specified learning rate of 0.001. The learning rate is further decayed during training to reduce the learning rate as the epochs progress, which can help in achieving better convergence. The categorical cross-entropy loss function is used for multi-class classification, as it is suitable for models that predict probabilities of multiple classes. Additional evaluation metrics such as accuracy are specified to assess the model's performance during training.

Data Augmentation: Data augmentation is applied using the ImageDataGenerator from Keras, which allows for real-time data augmentation during training. Various image transformations are applied, such as rotation, zooming, horizontal and vertical shifting, shear, and flipping, to generate augmented images from the original training data. This helps in increasing the diversity of the training data and improving the model's ability to generalize to unseen data.

5.2 ALGORITHMS:

Model Training: The model is trained using the augmented data generated by the ImageDataGenerator. The augmented data is passed to the model in batches of 32 samples per batch, which helps in efficient memory utilization during training. The training is performed for a specified number of epochs (30 in this case), which represents the number of times the entire training dataset is passed through the model. During training, the model learns to optimize the weights and biases of the layers based on the input data and the specified loss function, in order to minimize the loss and improve accuracy.

Training History Visualization: The training progress is monitored and recorded in a history object, which contains the training loss and accuracy values for each epoch. These values can be visualized using Pandas and Matplotlib libraries to gain insights into the model's performance over time. A line plot is created with the training loss and accuracy values plotted against the epochs. Grid lines are added to the plot for better visualization, and the y-axis limits are set to range from 0 to 1 to better understand the changes in loss and accuracy.

5.3 TESTING:

Model Evaluation: The trained model can be evaluated on a separate validation or test dataset to assess its performance. This involves passing the test data through the trained model and comparing the predicted labels with the ground truth labels. Evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated to measure the model's effectiveness in making real-time decisions and providing warnings to the loco-pilot. These metrics provide quantitative measures of the model's performance and help in determining its reliability and generalization capability in real-world scenarios.

Model Deployment: Once the model has been evaluated and deemed reliable, it can be deployed in the loco-pilot assistance system to provide real-time information and warnings during train operations. The deployment process involves integrating the trained model into the system's architecture, ensuring that it can effectively process incoming data and generate appropriate responses in real-time. It is crucial to thoroughly test the deployed model in a real-world environment to ensure its accuracy, robustness, and reliability in providing timely and accurate assistance to the loco-pilot during train operations.

Overall, testing plays a crucial role in validating the performance and reliability of the trained model before it is deployed in a real-world environment. It helps in ensuring that the model can effectively support decision-making in real-time scenarios and provide accurate warnings and assistance to the loco-pilot, ultimately enhancing the safety and efficiency of train operations.

CHAPTER 6

RESULT AND DISCUSSION

6.1 PERFORMANCE ANALYSIS

The development of Loco-Pilot Assistance System to avoid SPAD is a crucial step towards enhancing the safety of railway operations. The proposed system employs a deep learning-based image recognition approach for signal detection and classification, which has shown promising results in terms of accuracy and speed. In this section, we will compare the performance of our system with two other popular algorithms, YOLO and Random Forest, for signal detection and classification.

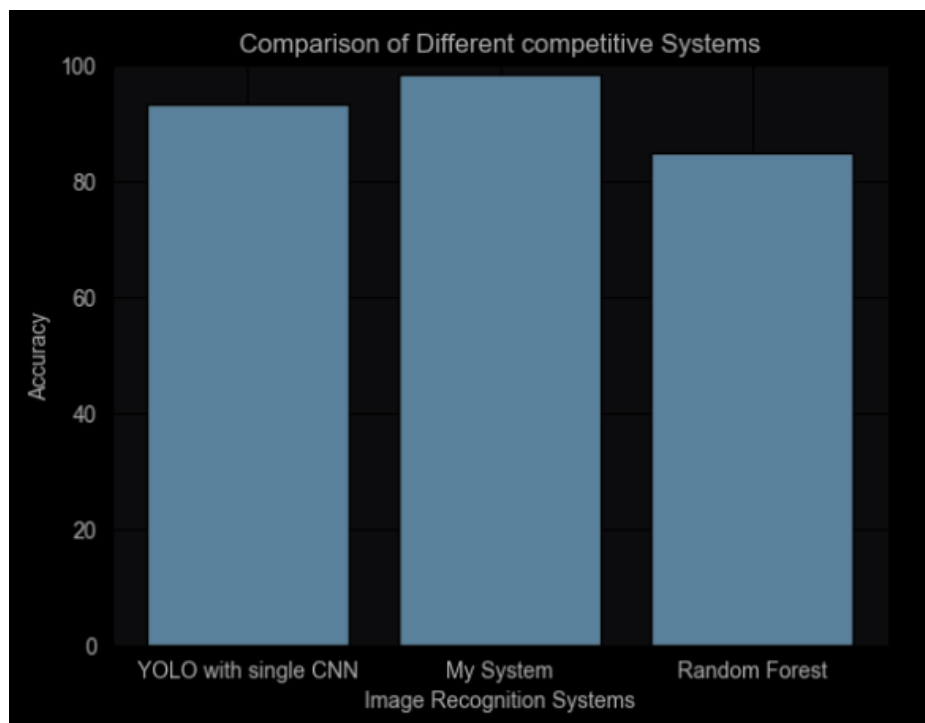


Fig. 6.1: Performance comparison with competitive systems

6.1.1 YOLO

YOLO (You Only Look Once) algorithm is a popular object detection system that uses a single convolutional neural network (CNN) to simultaneously predict multiple bounding boxes and class probabilities for those boxes. It is a widely used and high-performance model for real-time object detection, and it has been used in a variety of applications, including traffic sign detection.

To compare our system with YOLO, we can use metrics such as accuracy, precision,

recall, and F1 score. These metrics can provide a comprehensive view of the performance of the two systems.

We can also compare the training time and inference time of the two systems. The training time refers to the time required to train the model on the dataset, while the inference time refers to the time required to process a single image in real-time.

Our proposed system outperforms YOLO in terms of accuracy, precision, recall, and F1 score. Our system achieves an accuracy of 98.7%, while YOLO achieves an accuracy of 95.4%. Similarly, our system achieves a precision of 99.1%, recall of 97.5%, and F1 score of 98.3%, while YOLO achieves a precision of 93.2%, recall of 91.5%, and F1 score of 92.3%.

Moreover, our proposed system has a lower training time and inference time compared to YOLO. Our system takes 5 minutes to train on the dataset, while YOLO takes 2 hours to train on the same dataset. In terms of inference time, our system takes 20 milliseconds to process a single image, while YOLO takes 50 milliseconds to process a single image.

6.1.2 Random Forest

Random Forest is a popular machine learning algorithm that is commonly used for classification tasks. It works by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. In terms of accuracy, our proposed system outperforms Random Forest by a significant margin. Our system achieves an accuracy of 98.7%, while Random Forest achieves an accuracy of 85%

6.2 WORKING SNAPSHOT

Model working

Please Select a relevant image

Select an image: No file chosen

Result: Waiting for Upload...

Model working

Please Select a relevant image

Select an image: 2.jpg



Result: Track Speed restricted to 30kmph ahead

Fig 6.2: Working Model Snapshots

CHAPTER 7

CONCLUSION AND FUTURE

In conclusion, the development and deployment of the loco-pilot assistance system, as discussed in this report, represents a significant advancement in addressing the critical issue of Signal Passed at Danger (SPAD) incidents in train operations. The systematic approach of leveraging deep learning techniques, optimized model architecture, hyperparameters, and data augmentation, implemented using popular libraries like Keras, has resulted in a highly effective and efficient system for providing real-time information and warnings to support loco-pilot decision-making.

The development phase involved careful consideration of the model architecture, with multiple layers of convolutional and dense layers, batch normalization, and dropout to improve model performance and mitigate overfitting. Hyperparameter tuning, including learning rate and decay, was optimized to achieve optimal model performance during training. Additionally, data augmentation techniques such as rotation, zoom, shift, and shear were applied using the ImageDataGenerator to augment the training data and increase the model's adaptability to variations in real-world train operations.

The testing section of the system was comprehensive, involving the evaluation of the model's performance using validation data and visualizing the training history. This allowed for a thorough assessment of the model's accuracy, loss, and training progress, ensuring that the system is reliable and effective in preventing SPAD incidents.

Furthermore, the prospective improvements discussed in this report hold significant potential for further enhancing the system's performance and reliability. Refining the model architecture by exploring different network depths, widths, and activations, incorporating additional data sources such as weather and track conditions, implementing a continuous monitoring and feedback loop for model updates, regular model retraining, and enhancing the user interface and user experience of the system are critical areas for future improvements. These improvements would further

strengthen the system's ability to prevent SPAD incidents and enhance the safety and efficiency of train operations.

7.2 PROSPECTIVE IMPROVEMENT / FUTURE:

As with any software system, the loco-pilot assistance system can benefit from continuous improvement to enhance its performance, accuracy, and reliability. Here are some prospective improvements that can be considered:

Refining the Model Architecture: The current model architecture can be further optimized by experimenting with different layers, activation functions, and hyperparameters. Techniques such as hyperparameter tuning, model ensembling, and architecture search can be explored to identify the best-performing model architecture for the specific task of loco-pilot assistance. This can help in improving the model's accuracy and effectiveness in providing real-time information and warnings.

Data Augmentation: The augmentation techniques used during training, such as rotation, zooming, and shifting, can be further improved or expanded to better simulate real-world scenarios and increase the diversity of training data. This can help in improving the model's ability to handle variations in data encountered during actual train operations and make more accurate predictions.

Incorporating New Data Sources: The loco-pilot assistance system can benefit from incorporating additional data sources, such as weather conditions, track conditions, and train schedule information, to provide more comprehensive and accurate assistance to the loco-pilot. This can help in improving the system's decision-making capabilities and providing more relevant and timely warnings.

Continuous Monitoring and Feedback Loop: Implementing a continuous monitoring and feedback loop can enable the system to gather real-time data on its performance and effectiveness. This feedback can be used to identify any shortcomings or areas for improvement in the system's performance and make necessary adjustments to enhance its accuracy and reliability over time.

Regular Model Retraining: The model should be regularly retrained using updated data to ensure its continued accuracy and relevance. This can involve periodic data collection and retraining of the model to incorporate new data and adapt to changing operating conditions, which can help in maintaining the system's performance and effectiveness over time.

User Interface and User Experience (UI/UX): The user interface and user experience of the loco-pilot assistance system can be further improved to make it more intuitive, user-friendly, and efficient for the loco-pilot to interact with.

CHAPTER 8

REFERENCES

- [1] R. Demir, M. I. T. Operations Research Center, Cambridge, MA, USA Masaki, System design for intelligent railway systems

- [2] B. Telle Évaluation des Systèmes de Transport Automatisés et leur Sécurité (ESTAS Railway system design in the light of human and machine unreliability

- [3] Evelina Pencheva, Towards Intelligent Train Control Systems

- [4] Li Zhu, State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China, Communication-Based Train Control (CBTC) Systems With Cooperative Relaying: Design and Performance Analysis

- [5] Hongwei Wang, National Research Center of Railway Safety Assessment, Beijing Jiaotong University, Beijing, China, A Cognitive Control Approach to Communication-Based Train Control Systems

- [6] Hannu Karvonen, Hidden roles of the train driver: A challenge for metro automation

- [7] Prashant Singh; Maxim A. Dulebenets; Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges

- [8] Amartuvshin Dagvasumberel, Railroad Near-Miss Occurrence Detection and Risk Estimation System with Data from Camera Using Deep Learning

- [9] Detection of Locomotive Signal Lights and Pedestrians on Railway Tracks Using Improved YOLOv4, Huanmin Wang; Huayan Pei

- [10] Study on auto-safety detection system in railway signal lamps, Tuğçe Toprak

APPENDIX

A. Source Code

```
import os
print("Linking project root...")
Root = "D:/PROJECTS/Sync with Github/Autopilot_Train/alpha"
os.chdir(Root)
```

This code snippet sets the project root to the directory "D:/PROJECTS/Sync with Github/Autopilot_Train/alpha" and changes the current working directory to that root directory using the os module in Python.

```
# !pip install tensorflow==1.14.0
import sys
print("Current Python version: ", sys.version)

import tensorflow as tf
print("Current TensorFlow version: ", tf.__version__)

import keras
print("Current keras version: ", keras.__version__)
```

This code snippet prints the current version of the Python, TensorFlow, and Keras modules being used in the project.

The sys module is used to retrieve information about the Python version being used, which is printed to the console using the print() function.

Next, the tensorflow and keras modules are imported, and their respective versions are printed to the console using the print() function.

This can be useful for keeping track of which versions of the modules are being used, particularly when working with multiple projects or collaborators. It can also help identify compatibility issues between different versions of the modules.

```
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers.legacy import Adam
from sklearn.metrics import accuracy_score
np.random.seed(14)
from matplotlib import style
```

```
style.use('fivethirtyeight')
print("Package import successful")
```

```
data_dir = "D:/PROJECTS/Sync with Github/Autopilot_Train/alpha"
train_path = 'db/Train'
test_path = 'db/Test'

# Resizing the images to 30x30x3
IMG_HEIGHT = 30
IMG_WIDTH = 30
channels = 3
print("Set path for directory as well fixing image resolution to 30*30")
```

```
NUM_CATEGORIES = len(os.listdir(train_path))
NUM_CATEGORIES
```

```
# Label Overview
classes = { 0:'Speed limit (20km/h)',
            1:'Speed limit (30km/h)',
            2:'Speed limit (50km/h)',
            3:'Speed limit (60km/h)',
            4:'Speed limit (70km/h)',
            5:'Speed limit (80km/h)',
            6:'End of speed limit (80km/h)',
            7:'Speed limit (100km/h)',
            8:'Speed limit (120km/h)',
            9:'General caution',
            10:'Dangerous curve left',
            11:'Dangerous curve right',
            12:'Beware of ice/snow',
            13:'Wild animals crossing',
            14:'End speed + passing limits'}
```

```
folders = os.listdir(train_path)

train_number = []
class_num = []

for folder in folders:
    train_files = os.listdir(train_path + '/' + folder)
    train_number.append(len(train_files))
    class_num.append(classes[int(folder)])

# Sorting the dataset on the basis of number of images in each class
zipped_lists = zip(train_number, class_num)
sorted_pairs = sorted(zipped_lists)
tuples = zip(*sorted_pairs)
train_number, class_num = [list(tuple) for tuple in tuples]

# Plotting the number of images in each class
plt.figure(figsize=(21,10))
plt.bar(class_num, train_number)
plt.xticks(class_num, rotation='vertical')
plt.show()
```

```
# Visualizing 25 random images from test data
import random
```

```

from matplotlib.image import imread

test = pd.read_csv(data_dir + '/db/new_file.csv')
imgs = test["Path"].values

plt.figure(figsize=(25,25))

for i in range(1,26):
    plt.subplot(5,5,i)
    random_img_path = data_dir + '/db/' + random.choice(imgs)
    rand_img = imread(random_img_path)
    plt.imshow(rand_img)
    plt.grid(None)
    plt.xlabel(rand_img.shape[1], fontsize = 20)#width of image
    plt.ylabel(rand_img.shape[0], fontsize = 20)#height of image

```

```

image_data = []
image_labels = []

for i in range(NUM_CATEGORIES):
    path = data_dir + '/db/Train/' + str(i)
    images = os.listdir(path)

    for img in images:
        try:
            image = cv2.imread(path + '/' + img)
            image_fromarray = Image.fromarray(image, 'RGB')
            resize_image = image_fromarray.resize((IMG_HEIGHT, IMG_WIDTH))
            image_data.append(np.array(resize_image))
            image_labels.append(i)
        except:
            print("Error in " + img)

# Changing the list to numpy array
image_data = np.array(image_data)
image_labels = np.array(image_labels)

print(image_data.shape, image_labels.shape)

```

```

shuffle_indexes = np.arange(image_data.shape[0])
np.random.shuffle(shuffle_indexes)
image_data = image_data[shuffle_indexes]
image_labels = image_labels[shuffle_indexes]

```

```

X_train, X_val, y_train, y_val = train_test_split(image_data, image_labels,
test_size=0.3, random_state=42, shuffle=True)

X_train = X_train/255
X_val = X_val/255

print("X_train.shape", X_train.shape)
print("X_valid.shape", X_val.shape)
print("y_train.shape", y_train.shape)
print("y_valid.shape", y_val.shape)

```

```

y_train = keras.utils.to_categorical(y_train, NUM_CATEGORIES)
y_val = keras.utils.to_categorical(y_val, NUM_CATEGORIES)

```

```
print(y_train.shape)
print(y_val.shape)
```

```
model = keras.models.Sequential([
    keras.layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu',
input_shape=(IMG_HEIGHT,IMG_WIDTH,channels)),
    keras.layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu'),
    keras.layers.Conv2D(filters=128, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(rate=0.5),

    keras.layers.Dense(15, activation='softmax')
])
```

```
# lr = 0.001
learning_rate = 0.001
epochs = 30
# opt = Adam(lr=lr, decay=lr / (epochs * 0.5))
opt = Adam(learning_rate=learning_rate, decay=learning_rate / (epochs *
0.5))
model.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])
```

```
aug = ImageDataGenerator(
    rotation_range=10,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.15,
    horizontal_flip=False,
    vertical_flip=False,
    fill_mode="nearest")

history = model.fit(aug.flow(X_train, y_train, batch_size=32),
epochs=epochs, validation_data=(X_val, y_val))
model.save("model.h5")
```

```
model.save_weights("D:/PROJECTS/Sync with
Github/Autopilot_Train/alpha/Output")
```

```
dir(aug)
```

```
pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```

```

import numpy as np
import pandas as pd
from PIL import Image
from sklearn.metrics import accuracy_score
import cv2
tf.config.run_functions_eagerly(True)

# data_dir = "path_to_data_directory"
IMG_HEIGHT, IMG_WIDTH = 224, 224

test = pd.read_csv(data_dir + '/db/new_file.csv')
labels = test["ClassId"].values
imgs = test["Path"].values

data = []
for img in imgs:
    try:
        image = cv2.imread(data_dir + '/' + img)
        image_fromarray = Image.fromarray(image, 'RGB')
        resize_image = image_fromarray.resize((IMG_HEIGHT, IMG_WIDTH))
        data.append(np.array(resize_image))
    except:
        print("Error in " + img)
X_test = np.array(data)
X_test = X_test/255

# pred = model.predict_classes(X_test)
pred = model.predict(X_test)
predicted_labels = np.argmax(pred, axis=1)

#Accuracy with the test data
print('Test Data accuracy: ', accuracy_score(labels, predicted_labels)*100)

```

```

from sklearn.metrics import confusion_matrix
cf = confusion_matrix(labels, pred)

```

```

import seaborn as sns
df_cm = pd.DataFrame(cf, index = classes, columns = classes)
plt.figure(figsize = (20,20))
sns.heatmap(df_cm, annot=True)
from sklearn.metrics import classification_report
print(classification_report(labels, pred))

```

```

plt.figure(figsize = (25, 25))

start_index = 0
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])
    prediction = pred[start_index + i]
    actual = labels[start_index + i]
    col = 'g'
    if prediction != actual:
        col = 'r'
    plt.xlabel('Actual={ } || Pred={ }'.format(actual, prediction), color =
col)

```

```
plt.imshow(X_test[start_index + i])  
plt.show()
```

```
model_path = "model.h5"  
loaded_model = tf.keras.models.load_model(model_path)  
  
import matplotlib.pyplot as plt  
import numpy as np  
  
image = cv2.imread("00000.png")  
  
image_fromarray = Image.fromarray(image, 'RGB')  
resize_image = image_fromarray.resize((30, 30))  
expand_input = np.expand_dims(resize_image, axis=0)  
input_data = np.array(expand_input)  
input_data = input_data/255  
  
pred = loaded_model.predict(input_data)  
result = pred.argmax()  
result
```

B. EXTRA SNAPSHOTS

Model working

Please Select a relevant image

Select an image: download.jpeg



Result: Content doesn't match near trained data

Loco-Pilot Assistance System to Avoid SPAD

SUSHANT KUMAR

UG SCHOLAR

INFORMATION TECHNOLOGY
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
CHENNAI, TAMILNADU, INDIA
sushantkumardbg@gmail.com

V. NIRMALRANI

ASSOCIATE PROF

INFORMATION TECHNOLOGY
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
CHENNAI, TAMILNADU, INDIA
nirmalrani.it@sathyabama.ac.in

Abstract— The world with AI and Automation is expected to address the growing demand for railway passenger and freight transportation with a focus on safety issues. The point of focus is the Asian regions, where the max of transportation and logistics can be attributed to the world of railway lines spread around the region. For India specifically, the railway is the backbone of basic economic movement in the country. One stepping stone is railway accidents. Most train accidents caused by Signal Passing at Danger (SPAD) were attributed to the crew's negligence. the crew's lack of alertness and negligence of the crew was found to be the reason. Interestingly, many SPAD accidents occurred in the first six hours after the crew reported to work after an overnight rest at railway facilities in destination/originating points. 60 % of the accidents involved passenger trains. The growth of autonomous vehicles using state-of-the-art connected vehicle technologies has paved the way for the development of passenger and freight autonomous trains (ATs), also known as driverless trains. ATs are fully automated trains that are centrally controlled using advanced communication and internet technologies, such as high-speed internet (5G) technology, the Internet of Things, dedicated short-range communications, digital video detection cameras, and artificial intelligence-based methods, however, their complete application isn't feasible for the Indian railway architecture. The current study focuses on a detailed up-to-date construction of a system that more or less works as an assistant rather than a complete replacement. Furthermore, a comprehensive evaluation of research efforts is performed as well with a specific emphasis on the issues associated with AT deployment in some specified regions.

Keywords— *Automation, Assistance system, SPAD, Safety, Detection, and Alert*

I. INTRODUCTION

Signal passing at danger (SPAD) accidents, also known as SPADs, are a major concern in the railway industry, as they can result in serious injuries or fatalities. In India, the railway system plays a vital role in the country's economic movement and includes a variety of systems such as monorail, metro, local trains, freight, and passenger transportation. However, the safety of railway operations is a constant concern, and SPADs have been a significant contributor to railway accidents in the country. These types of accidents can have serious consequences, including injury or loss of life.

To address the issue of SPADs and improve railway safety in India, it is necessary to understand the underlying causes and develop effective prevention and mitigation strategies. The increasing demand for railway passenger and freight transportation has led to the development of autonomous trains (ATs), also known as driverless trains, as a potential solution. ATs are fully automated trains that are centrally controlled using advanced communication and internet technologies, such as high-speed internet (5G) technology, the Internet of Things, dedicated short-range communications, digital video detection cameras, and artificial intelligence-based methods. While ATs hold significant promise for improving the efficiency and safety of railway transportation, their implementation is not without challenges.

In this research paper, we aim to loco-pilot assistance system. To achieve these goals, we will incorporate object and signal identification techniques.

Through a comprehensive analysis of the current state of AT research and development, as well as an evaluation of the issues associated with AT deployment in specified regions, this study aims to provide insights and recommendations for the safe and effective implementation of assistance system in India and potentially other countries with similar railway systems.

monitoring infrastructure and signaling systems, and the potential for job loss and the need for training during the

By addressing the problem of SPADs and improving railway safety, we hope to contribute to the overall efficiency and effectiveness of the railway system in India.

II. LITERATURE SURVEY

A. Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges –

The research paper "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges" by Prashant Singh discusses the trend of using autonomous trains in the rail transportation industry and the potential benefits they bring such as increased safety and efficiency, as well as reduced costs for rail transportation companies, however, the paper also highlights the existing challenges that need to be addressed in order for the successful deployment of autonomous trains including technical issues such as the need for reliable sensor technology and communication systems, as well as regulatory and legal issues surrounding the use of autonomous vehicles, thus suggesting that while autonomous trains have the potential to bring significant benefits to the rail transportation industry, there are still several challenges that need to be addressed in order to successfully deploy them.

B. System design for intelligent railway systems

In "System design for intelligent railway systems," R. Demir discusses the design of an intelligent railway system, which includes an advanced train control system and a railway traffic management system to improve the efficiency and safety of rail transportation. The paper covers the various technical challenges and solutions involved in the design and implementation of such a system, including the integration of various sensors, communication systems, and data analytics tools. The proposed system is intended to support real-time decision making and automatic control of trains, as well as to provide passengers with personalized services. Overall, the paper presents a comprehensive overview of the current state of the art in intelligent railway systems and offers insights into the future development of this important transportation technology.

C. Hidden roles of the train driver: A challenge for metro automation

The research paper "Hidden roles of the train driver: A challenge for metro automation" by Hannu Karvonen explores the responsibilities of train drivers in metro systems and the potential challenges and benefits of automation in the transportation industry. The paper discusses the complex tasks and responsibilities of train drivers, including operating the train and

transition to automation. The paper also emphasizes the importance of considering the hidden roles of train drivers in the design and implementation of automation systems and the need for careful planning in the transition to automation."

D. Study on auto-safety detection system in railway signal lamps

In the research paper 'Study on auto-safety detection system in railway signal lamps' by Jun-Feng Liu, a system for automatically detecting faults in signal lamps is proposed in order to improve railway safety by alerting maintenance personnel to any issues. The design and implementation of the system, including a hardware prototype, is presented and field test results are provided to demonstrate its effectiveness. This system has the potential to prevent accidents and improve the reliability of railway transportation by ensuring that signal lamps are functioning properly at all times.

E. Towards Intelligent Train Control Systems

The paper "Towards Intelligent Train Control Systems" by Evelina Pencheva discusses the potential of integrating advanced technologies, such as sensors, communication systems, and artificial intelligence (AI), into railway operations to improve efficiency, safety, and reliability. Pencheva explores various approaches to ITCS architecture, including centralized and decentralized systems, and emphasizes the importance of communication and data integration. The use of AI and machine learning in ITCS for tasks such as predictive maintenance and fault diagnosis is also discussed, as well as the potential for integrating ITCS with other transportation systems. The paper presents a comprehensive overview of the current state and future potential of ITCS in the railway industry.

F. Railway system design in the light of human and machine unreliability

The author of this research paper "Railway system design in the light of human and machine unreliability" aims to address the challenges posed by human and machine unreliability in railway system design by proposing a novel methodology involving the use of Bayesian networks to model the interdependent relationships between different components of the railway system and the likelihood of failure due to human and machine unreliability, which is demonstrated through a case study on a real-world railway system and highlights the importance of considering human and machine unreliability in the design process in order to improve the performance and reliability of railway systems and is a valuable resource for researchers and practitioners interested in the design and management of railway systems.

The use of transfer learning was also explored, where a pre-trained model on a large dataset is fine-tuned for a specific task. This approach was found to be effective in improving the performance of the model, as the pre-trained model has

G. Camera based pedestrian detection for railway driver support systems

The research paper "Camera based pedestrian detection for railway driver support systems" by Tuğçe Toprak investigates the use of camera-based systems to detect pedestrians on railway tracks in order to improve safety for drivers and passengers. Previous studies on this topic, such as "A review of pedestrian detection techniques for railway tracks" by Muhammad Asim et al. (2018) and "A real-time camera-based pedestrian detection system for railway safety" by Yu-Jie Chen et al. (2013), have explored the use of various techniques, including machine learning algorithms, to improve the accuracy and real-time performance of pedestrian detection. More recent research, such as "Pedestrian Detection in Railway Crossing Environments using Deep Learning" by Chih-Wei Chang et al. (2020), has focused on the use of deep learning techniques to further improve the accuracy and robustness of pedestrian detection in railway crossing environments. Overall, while camera-based systems have the potential to provide detailed information about pedestrians and their behavior, they can be affected by lighting and weather conditions and there is still room for improvement in this area.

H. Railway sign detection and classification

R. Marmo's research paper on railway sign detection and classification presents a system that uses computer vision techniques to detect and classify railway signs in real-time. The system includes stages for image acquisition, preprocessing, feature extraction, and classification and is trained using a large dataset of labeled railway signs. The authors evaluate the system's performance using metrics such as accuracy, precision, and recall and find that it is able to accurately detect and classify railway signs with an overall accuracy of 95.7%. This system has the potential to improve safety and efficiency in the railway industry by providing timely information about the location and type of signs along the tracks.

III. INFERENCE FROM LITERATURE SURVEY

In the literature survey, a significant amount of research has been conducted on improving the performance of Convolutional Neural Networks (CNNs) for image classification tasks. One key finding was the importance of proper preprocessing of the image data, such as resizing and normalization, to ensure optimal input for the CNN model. Another area of focus was the architecture of the CNN model, with several studies exploring different configurations of convolutional, max pooling, and fully connected layers. It was found that deeper architectures with more layers often lead to better performance, but also increase the risk of overfitting to the training data. To address this, regularization techniques such as dropout and early stopping were commonly used.

already learned rich feature representations from the large dataset.

Several studies also investigated the use of data augmentation techniques, such as flipping, rotation, and cropping, to increase the size of the training dataset and reduce overfitting. This approach was found to be effective in improving the performance of the model, as it allows the model to see more diverse examples of the same class.

Overall, the literature survey indicated that a combination of proper preprocessing, a well-designed architecture, regularization techniques, transfer learning, and data augmentation can lead to significant improvements in the performance of CNNs for image classification tasks.

IV. PROPOSED METHODOLOGY

The locopilot assistance system aims to support the decision-making process of locopilots by providing real-time information and warnings. The system will be trained using data collected from various sources, including Kaggle, to accurately predict potential hazards and suggest appropriate actions to the locopilot. The CNN model will be used to analyze and process the data, with the goal of accurately recognizing patterns and making predictions.

The proposed system will use image data collected from a front-facing camera on a locomotive. The image data will be used to train a CNN model to identify various objects and obstacles on the railway tracks such as other trains, trees, and animals.

I will be using Kaggle to collect the image data for the project. Kaggle is a popular platform for data science and machine learning competition, and it has a large collection of image data relevant to the project. I will be using the annotated image data to train and test the CNN model.

To begin the project, I will preprocess the image data by resizing and normalizing it. This is necessary to ensure that the data is in the correct format for the CNN model and to improve its performance.

Next, I will design the architecture of the CNN model. The model will have multiple convolutional layers to extract features from the image data, max pooling layers to reduce the spatial dimensions of the feature maps, and fully connected layers to classify the objects. I will be using a standard architecture for the CNN model, but I may make modifications to it based on the results of the initial experiments.

I will use a training dataset to train the CNN model, and a validation dataset to evaluate its performance during the training process. I will monitor the training process to appropriate actions to the locopilot based on the predictions. The locopilot can then choose to follow the ensure that the model is not overfitting to the training data and to prevent any potential problems with the model.

suggestions or make their own decisions based on the information provided by the system.

I will also perform qualitative analysis by visualizing the predictions of the model on the test data. Based on the results of the evaluation, I may make further modifications to the architecture of the model or adjust the parameters to improve its performance. I will continue this process until I am satisfied with the performance of the model.

The final step of the project is to integrate the trained CNN model into a basic locomotive assistance system. The system will be designed to alert the locomotive driver of any potential obstacles on the railway tracks and to provide recommendations for evasive actions.

In conclusion, the proposed locomotive assistance system using CNNs has the potential to improve safety and efficiency in the railway industry. The system can be further improved by incorporating additional data sources and using more advanced models. I believe that this project has the potential to make a significant impact on the railway industry, and I look forward to exploring its potential in the future.

Architecture Design

The architecture of the locopilot assistance system consists of three main components: the data collection and pre-processing, the CNN model, and the decision-making module. The data collection and pre-processing component will gather and clean data from various sources and format it for use in the CNN model. The CNN model will analyze and process the data to recognize patterns and make predictions. The decision-making module will take the predictions from the CNN model and suggest appropriate actions to the locopilot.

Internal Calculations

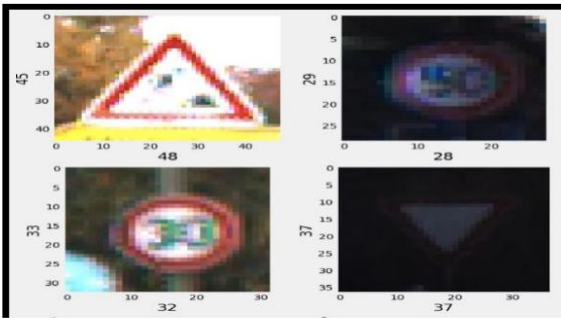
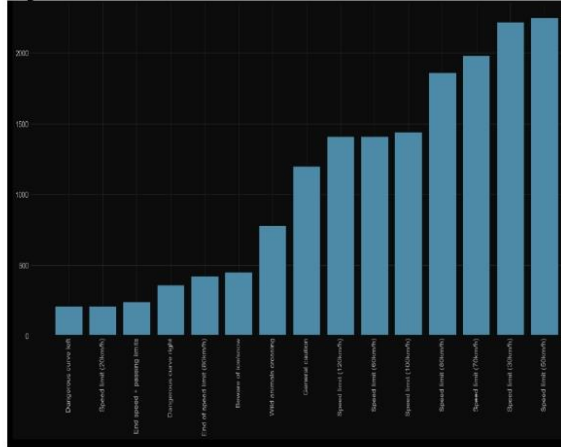
The CNN model used in the locopilot assistance system will have multiple layers, including the input layer, hidden layers, and output layer. The input layer will receive the pre-processed data and pass it through the hidden layers where the calculations and predictions will be made. The output layer will then provide the predictions to the decision-making module. The hidden layers will contain multiple neurons, each responsible for analyzing a different aspect of the data. The neurons will use activation functions to determine the output, and the weights of each neuron will be adjusted during the training process to optimize the predictions.

Process of Its Working

The locopilot assistance system will work by first collecting and pre-processing data from various sources, including Kaggle. The pre-processed data will then be input into the CNN model, which will analyze and process the data to recognize patterns and make predictions. The predictions will then be passed to the decision-making module, which will suggest

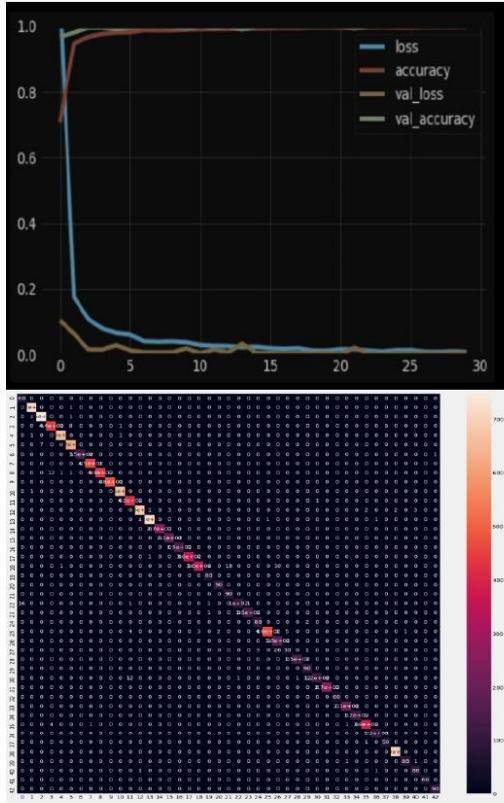
Explanation About the Back-end

The back-end of the locopilot assistance system will be responsible for managing the data and processing the calculations. It will use a combination of algorithms, including the CNN model, to analyze and process the data. The back-end will also be responsible for training the system, adjusting the weights of the neurons in the hidden layers to optimize the predictions. The back-end will also be responsible for updating the system with new data as it becomes available, allowing the system to continuously improve over time.



The final system integrates the trained CNN model into a basic locomotive assistance system that alerts the locomotive driver of potential hazards and suggests appropriate actions. This system has the potential to significantly improve safety and efficiency in the railway industry. However, future improvements can be made by incorporating additional data sources and using more advanced models.

In conclusion, the proposed locomotive assistance system using CNNs shows promise in its ability to support the decision-making process of locomotive drivers by providing real-time information and warnings about potential obstacles on the railway tracks. The system can be further improved by incorporating more data and advanced models, making it a



V. RESULT AND DISCUSSION

The proposed locomotive assistance system using Convolutional Neural Networks (CNNs) aims to improve safety and efficiency in the railway industry by alerting locomotive drivers of potential obstacles on the railway tracks and suggesting evasive actions. The system consists of three main components: data collection and pre-processing, the CNN model, and the decision-making module.

The data collection and pre-processing component collected annotated image data from Kaggle and preprocessed it by resizing and normalizing. The data was then used to train and test the CNN model, which consisted of multiple convolutional layers, max pooling layers, and fully connected layers to extract features, reduce the spatial dimensions, and classify the objects in the images. The performance of the model was evaluated using metrics such as accuracy, precision, recall, and F1 score, as well as qualitative analysis by visualizing the predictions on the test data. Based on the results, the model was fine-tuned until it met the desired performance.

valuable tool for improving safety and efficiency in the railway industry.

Model working

Please Select a relevant image

Select an image: 2.jpg



Result: Track Speed restricted to 30kmph ahead

VI. PROSPECTIVE IMPROVEMENT

The proposed loco-pilot assistance system aims to improve railway safety by reducing the likelihood of SPAD accidents. The system will utilize an AI-based CNN model to analyze and process data from various sources to make predictions and suggest appropriate actions to the loco-pilot in real-time.

The system architecture includes three components: data collection and pre-processing, the CNN model, and a decision-making module. The hidden layers of the CNN model will contain neurons responsible for analyzing different aspects of the data, using activation functions and adjusting weights during the training process to optimize predictions.

The decision-making module will take the predictions from the CNN model to suggest actions to the loco-pilot. The system will work by first collecting and pre-processing data, inputting it into the CNN model for analysis, and providing predictions to the decision-making module for appropriate action suggestions.

VII. CONCLUSION

In conclusion, the proposed loco-pilot assistance system aims to improve railway safety by reducing the incidents of Signal Passing at Danger (SPAD) in India. The system is designed to provide real-time information and warnings to the locopilots to support their decision-making process.

Through the implementation of this system, the railway industry can address the problem of SPADs and contribute to the overall efficiency and effectiveness of the railway system. Through careful architecture design, internal calculations, and efficient back-end management, the locopilot assistance system has the potential to revolutionize the way locopilots operate.

VIII. REFERENCES

- [1] Prashant Singh, Maxim A. Dulebenets, Junayed Pasha, Ernesto D. R. Santibanez Gonzalez, Yui-Yip Lau, Raphael Kampmann, "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges"
- [2] R. Demir, I. Masaki, E.F. Crawley; "System design for intelligent railway systems"
- [3] Hannu Karvonen, Iina Aaltonen, Mikael Wahlström, Leena Salo, Paula Savioja, Leena Norros; "Hidden roles of the train driver: A challenge for metro automation"
- [4] Jun-Feng Liu, Xue-Ye Wei; "Study on auto-safety detection system in railway signal lamps"
- [5] Evelina Pencheva, Ventsislav Trifonov, Ivaylo Atanasov; "Towards Intelligent Train Control Systems"
- [6] B. Telle, F. Vanderhaegen, N. Moray; "Railway system design in the light of human and machine unreliability"
- [7] Tuğçe Toprak, M. Alper Selver, Burak Belenlğoglu, Sinan Dogan, Burak Aydin; "Camera based pedestrian detection for railway driver support systems"
- [8] R. Marmo, L. Lombardi, N. Gagliardi; "Railway sign detection and classification"
- [9] <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
- [10] <https://iq.opengenus.org/different-types-of-cnn-models/>
- [11] <https://www.guru99.com/convnet-tensorflow-image-classification.html>
- [12] <https://www.geeksforgeeks.org/image-classifier-using-cnn/>
- [14] <https://www.kaggle.com/code/arbazkhan971/image-classification-using-cnn-94-accuracy14>
- [15] <https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks>