

Emergency vehicle detection in Heavy Traffic using Deep ConvNet2D and CV

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Saminepalli Sai Charan(Reg.No - 40111118)

G D C H Sai Phani Santosh (Reg.No - 40111100)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

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

Accredited with Grade "A++" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL - 2024

Emergency vehicle detection in Heavy Traffic using Deep ConvNet2D and CV

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Saminepalli Sai Charan(Reg.No - 40111118)

G D C H Sai Phani Santosh (Reg.No - 40111100)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

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

Accredited with Grade "A++" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

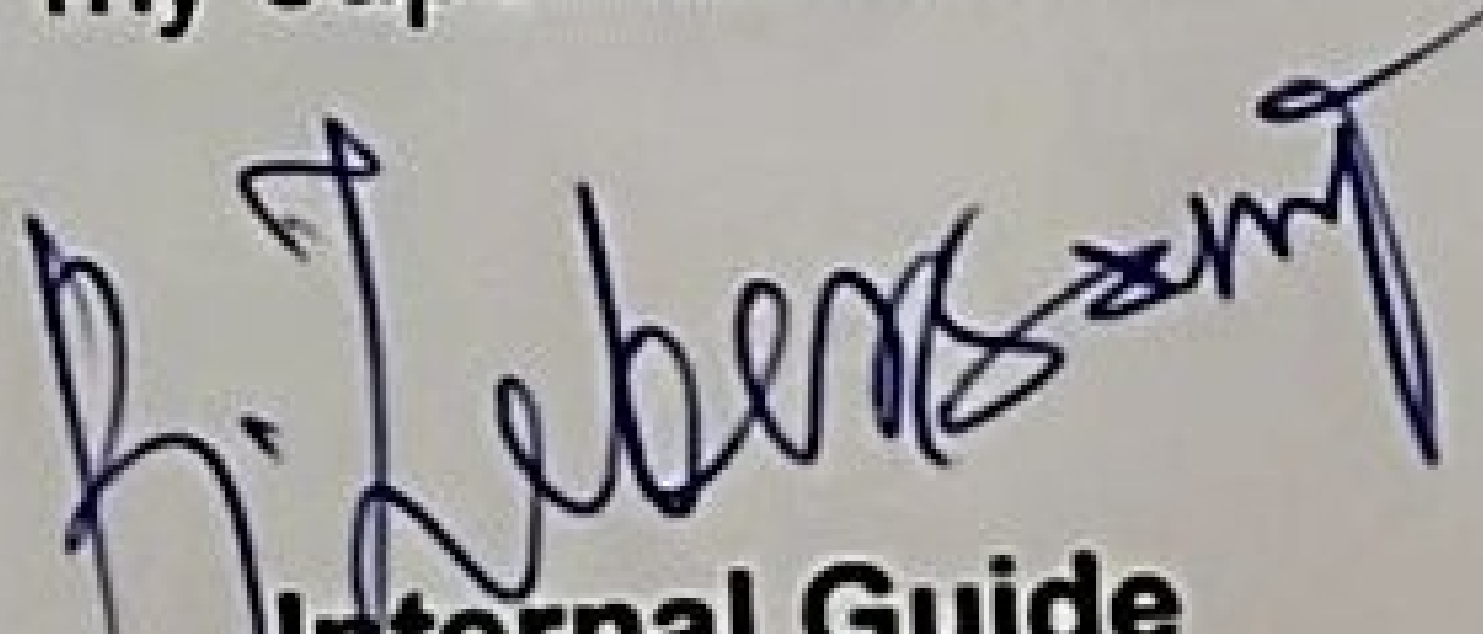
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of G D C H Sai Phani Santosh (40111100) who have done the Project work as a team Saminepalli Sai Charan (40111118), G D C H Sai Phani Santosh (40111100), who carried out the project entitled "Emergency vehicle detection in Heavy Traffic using Deep ConvNet2D and CV" under my supervision from June 2023 to April 2024.



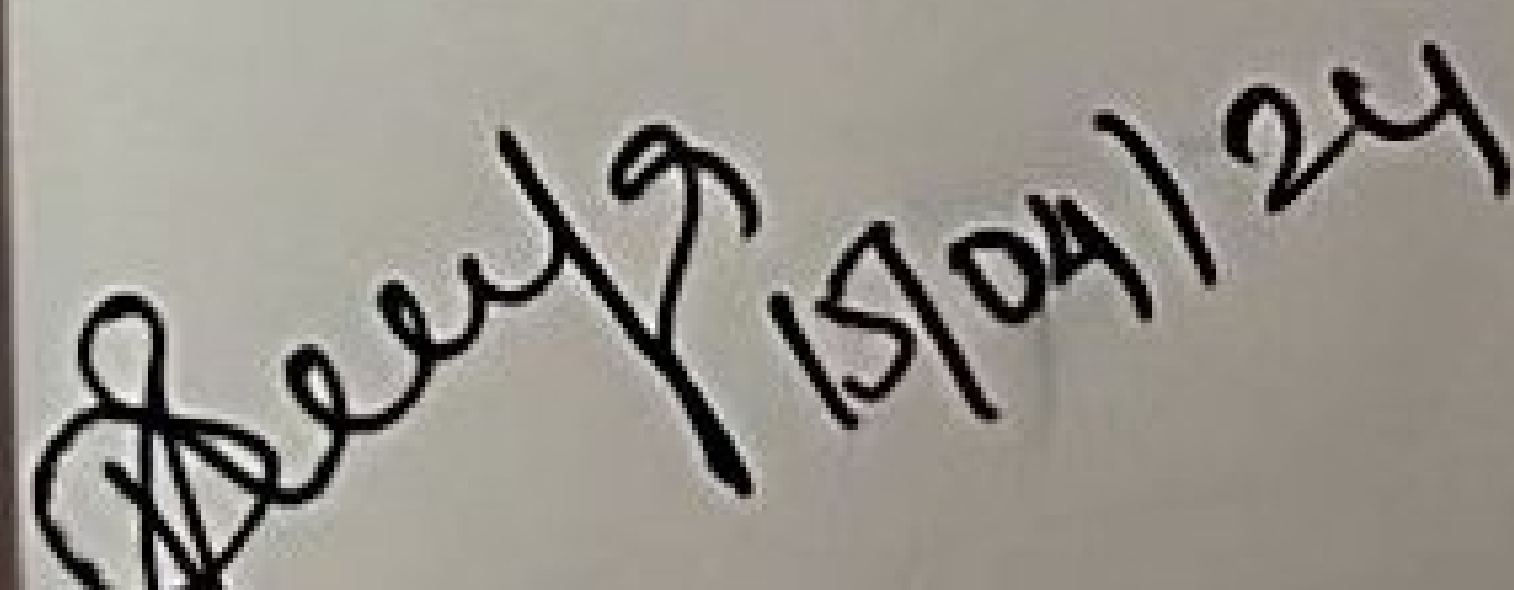
Internal Guide

Dr. R. JEBERSON RETNA RAJ, M.Tech., Ph.D.,

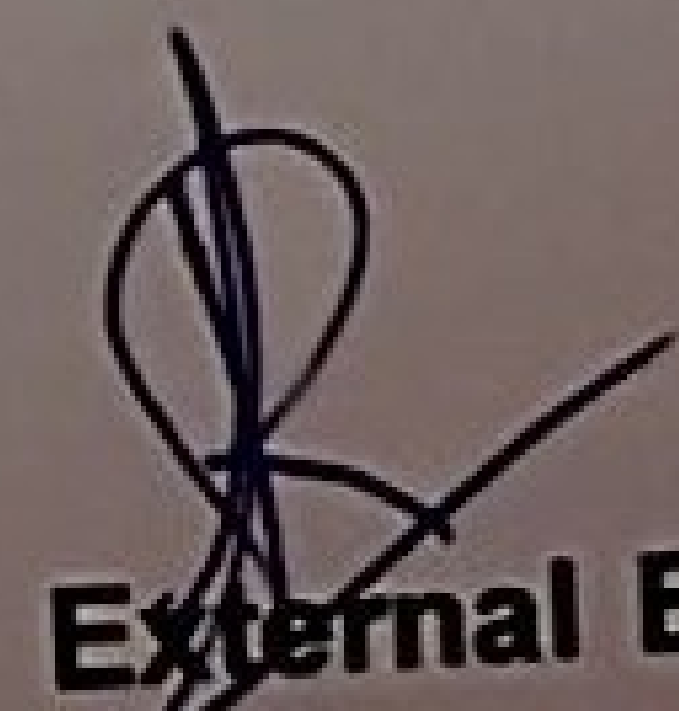


Head of the Department
DEPARTMENT OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jepplaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

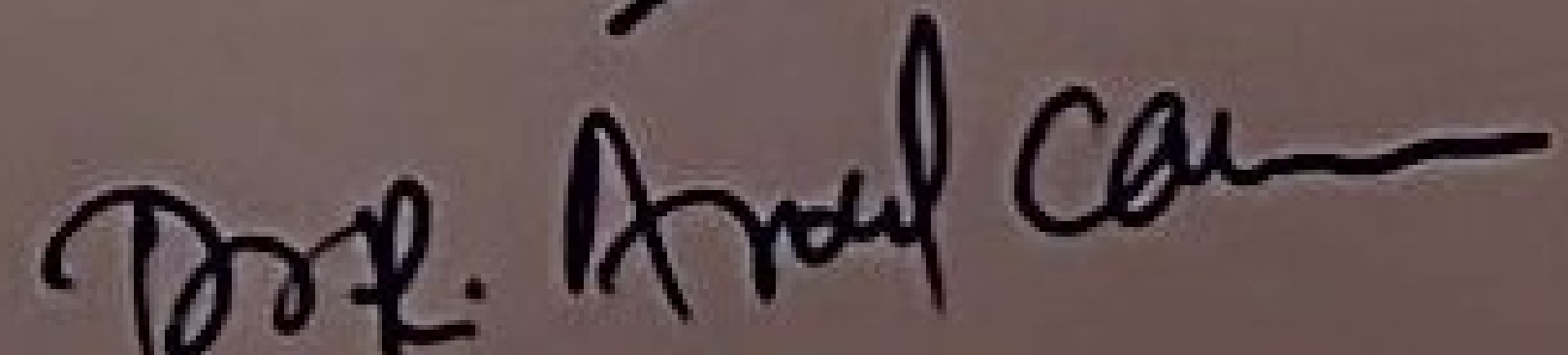
Submitted for Viva voce Examination held on 15/04/2024



Internal Examiner



External Examiner



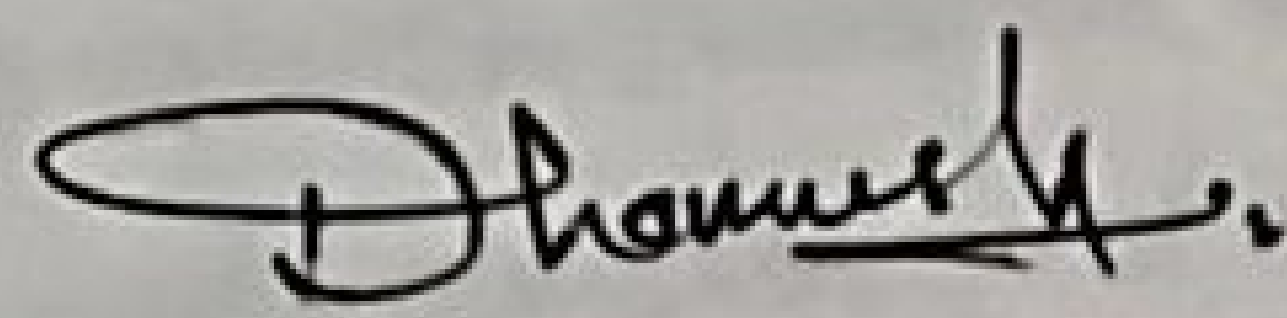
Dr. Arul

DECLARATION

We, G D C H Sai Phani Santosh, Saminepalli Sai Charan hereby declare that the Project Report entitled **Emergency vehicle detection in Heavy Traffic using Deep ConvNet2D and CV** done by us under the guidance of **Dr.R.Jeberson Retna Raj**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 15/04/2024

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** 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.L.Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering 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.R.Jeberson Retna Raj, MTech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** 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.L.Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering 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.R.Jeberson Retna Raj, MTech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

The vast number of traveling vehicles on the roads and the daily traffic crises that everyone faces produce problems and issues that require professional treatment and traffic solutions. The movement of emergency vehicles, such as ambulances, police vehicles, firefighting vehicles, etc., is one of the most current issues that require investigations. Obstacles in front of these vehicles may threaten the lives of some people, institutions, or property. These emergency vehicles usually use sirens to announce their existence on the road networks. Recently, most drivers have closed the windows and switched on stereos during their trips. Thus, hearing the sirens and reacting to the existing emergency vehicles becomes problematic. Fortunately, modern vehicles are equipped with cameras, sensors, and advanced technologies that help surveillance and monitor the surrounding road environment. Machine learning and deep learning techniques generate an intelligent model to configure emergency vehicles from pictures. Then, vehicles use this model to analyze regularly taken photos of their surrounding road environment. Instant and critical actions are required upon detecting emergency vehicles to achieve safety conditions on road networks. This includes opening the roads in front of these emergency vehicles by changing lanes or traveling speeds. The developed model's accuracy highly affects road scenarios' safety conditions. The accuracy of these models highly depends on the dataset used to train the intelligent model. In this thesis, we analyzed available datasets for emergency vehicles and collected a large dataset of real world images of emergency vehicles on the streets. This work then used GANs to augment the dataset, resulting in a more comprehensive and accurate representation of emergency vehicles. This built a vehicle detection model using CNN, which yielded satisfactory and reliable results in detecting real emergency vehicles. Our work highlights the importance of accurate datasets in the development of intelligent models for emergency vehicle detection. Finally, this project validated the dataset and the model's accuracy using a realistic data.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	
	TABLE OF CONTENTS	VI
	LIST OF ABBREVIATIONS	VIII
	LIST OF TABLES	IX
	LIST OF FIGURES	X
1	INTRODUCTION	1
	1.1. Introduction	1
	1.2. Problem Statement	3
	1.3. Research Contributions	3
	1.4. Research Methodology	4
	1.5. Research Layout	6
2	LITERATURE SURVEY	8
	2.1. Introduction	8
	2.2. Vehicle and Object Detection Methods	8
	2.3. Using Generative Adversarial Networks (GANs) To generate Augmented Datasets	12
3	AIM AND SCOPE OF PRESENT INVESTIGATION	16
	3.1. Introduction	16
	3.2. Emergency Vehicle Detection	17
	3.3. “JanataHack AV ComputerVision” and ”Emergency vs Non-Emergency Vehicle Classification”	17
	3.4. Ambulance Regression	17
	3.5. Dataset Used	18
	3.6. Ambulance Detect	18
	3.7. Emergency Vehicle Detection	18
	3.8. FALCK	19
	3.9. Sirens	19
	3.10. Smart Car	19

4	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	23
	4.1. Introduction	23
	4.2. Methods for Detecting Emergency Vehicles	23
	4.3. Convolutional Neural Networks (CNNs)	24
	4.4. CNN, R-CNN and Fast R-CNN	26
	4.5. Generate New Emergency Vehicle Images Using GANs	26
	4.6. Performance Evaluation	28
	4.7. Generating A Newly Balanced Dataset Introduction	31
	4.8. Methodology	31
	4.9. Gathering A Dataset For Emergency Vehicle Detection	33
	4.10. Generating New Images Using GANs	38
	4.11. The Generator	41
	4.12. The Discriminator	43
	4.13. Hyper Parameters Adjustment	46
5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	54
	5.1. Introduction	54
	5.2. Emergency Vehicle Detection Model using CNN	54
	5.3. Robustness and Reliability Evaluation using 10-fold Cross-Validation	59
	5.4. Comparative Performance Study	61
	5.5. Graphical User Interface (GUI) for Easy Model Testing and Evaluation	65
6	SUMMARY AND CONCLUSIONS	67
	6.1. Conclusion	67
	6.2. Future Work	67
	REFERENCES	69
	APPENDIX	74
	A. SOURCE CODE	74
	B. SCREENSHOTS	80
	C. RESEARCH PAPER	82

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
YOLO	You Only Look Once
CCTV	Closed-Circuit Television
DL	Deep Learning
GANs	Generative Adversarial Networks
ML	Machine Learning
AI	Artificial Intelligence
DT	Decision Tree Algorithm
DCGAN	Deep Convolutional Generative Adversarial Network
LR	Logistic Regression
TP	True Positives
TN	True Negatives
FP	False Positives

LIST OF TABLES

Table Number	Table Title	Page No.
2.1	Main Characteristics of Previous Emergency Vehicles' Detection Mechanisms	11
3.1	Summary the Most Important Findings from the Available Datasets	
4.1	Summary of generator layers	20
4.2	Summary of Discriminator layers	43
4.3	Summary of GANS model parameters	45
4.4	Samples images generated during various epochs	47
5.1	CNN results on initial dataset	52
5.2	CNN on initial dataset training confusion matrix	56
5.3	CNN on initial dataset validation confusion matrix	56
5.4	CNN results on final dataset	57
5.5	CNN on final dataset training confusion matrix	57
5.6	CNN on final dataset validation confusion matrix	58
5.7	10-fold cross-validation results on the initial dataset	58
5.8	10-fold cross-validation results on the final dataset	60
5.9	Comparative Performance of Emergency Vehicle Detection Models	60
		63

LIST OF FIGURES

Chapter No.	Figure Name	Page No.
1.1	Proposed Approach Methodology for Emergency Vehicle Detection	6
2.1	The image after merging the generated images by GANs	14
2.2	The Proposed GANs	15
4.1	Features Extraction by CNN with layers	26
4.2	GANs Image Generation General Process	27
4.3	GANs Generator and Discriminator Layers Process	28
4.4	K-fold Cross-Validation form first fold to K fold.	29
4.5	Confusion Matrix	30
4.6	Samples images collected for our initial dataset	34
4.7	Image Collection through Video Recording and Frame Selection Methodology	35
4.8	Samples of images collected during Video Recording and Frame Selection	36
4.9	Samples of images after augmentation process	38
4.10	GANs Image Generation Process	39
4.11	GANs Image Generation Process	40
4.12	The Designing of GANs	41
4.13	Generator Layers	42
4.14	Discriminator Layers	46
4.15	The DCGANs model total parameters	48
4.16	The generator model parameters	49
4.17	The discriminator model parameters	50
4.18	Discriminator and generator improvement through epochs	51
5.1	Comparative Analysis of CNN Models Performance	58
5.2	Comparative analysis using 10-fold Cross Validation	61
5.3	Examples of Vehicle Detection Results	62
5.4	Comparison of Emergency Vehicle Detection Models	64
5.5	The GUI's main interface	65

CHAPTER 1

1.INTRODUCTION

1.1. INTRODUCTION

Smart cities are widely developed and have been investigated differently recently. Technological development helps to build and monitor these cities. They aim to improve the quality of life by improving the quality of services such as education, healthcare, and transportation. These services have been linked to technological innovation (Campisi et al., 2021).

Traffic crises are the most critical challenges in traditional cities, especially crowded ones. Modern technologies emerge and provide solutions, especially in the matter of enhancing the safety conditions on road networks. Emergency vehicle management is one of the most critical problems that require sensitive and real-time solutions. Special driving rules are announced to vehicles around emergency vehicles, such as opening the way for them and giving them the highest priority to traverse a signalized road intersection.

Detecting emergency vehicles on the road network is the first step to react according to location, speed, and other parameters. Drivers used to be able to see these emergency vehicles and respond accordingly visually. Emergency vehicles produce sirens in more crowded and fast road scenarios to alert the surrounding drivers. Several drivers panic to react to these sudden sirens. Sometimes they fail to determine these emergency vehicles' exact location, speed, or other characteristics. Thus, they fail to respond promptly and adequately, threatening the road network's safety conditions. Benefiting from modern vehicles' advanced equipment and technologies, it is necessary to develop an intelligent emergency vehicle detection system. Models that use artificial intelligence technology can improve and increase the efficiency of logistics transportation services, reduce response time, choose the healthiest and fastest routes, and alert vehicles that will open the way for them (Eliyi, 2022).

Artificial intelligence (AI) offers many varied solutions in this field. It has excellent potential in Internet applications to monitor and manage traffic and

predict future events through machine learning (ML). Many scientific research studies tried to handle this topic to find a way to detect and classify vehicles on the roads. Some of these researches specifically studied the issue of detecting emergency vehicles through photos and videos to inform and alert the concerned authorities (Haque et al., 2022 ; Roy et al., 2019). Some problems are reported regarding these previous studies that consider either the used method, the tested dataset, or the accuracy of the obtained model.

Several studies have introduced intelligent emergency vehicle detection models in the literature to classify vehicles on road networks. The main weaknesses in the previous studies lie in the used datasets. The unbalanced dataset rarely involves an emergency vehicle and is not the best choice for training the machine learning mechanisms and predicting the existing emergency vehicles (Patel et al., 2022). We have noticed that the dataset used is unrealistic in several scenarios. These datasets considered photos or videos are not taken from road scenarios (Shravan, 2023 ; Folio3, 2022). In addition, the method of determining the accuracy as a criterion is not always the most appropriate criterion to rely on to accept the results (Shung, 2018). The real problem we may face in applying the proposed model is the same dataset on which we will train the model. It turns out that several conditions must be achieved in the dataset to obtain a precise and accurate detection of emergency vehicles. Among these conditions are the size of the dataset and the quality of the images, in addition to the balance characteristics.

This project will analyze available datasets for emergency vehicles and collect a large real-world dataset of emergency vehicle images on streets. This work plans to use GANs to augment the dataset, resulting in a more comprehensive and accurate representation of emergency vehicles. Our goal is to produce a new suitable dataset using GANs that achieves acceptable accuracy levels for emergency vehicle detection. We will build a vehicle detection model using CNN, which we hope will yield satisfactory and reliable results in detecting real emergency vehicles. Our work will highlight the importance of accurate datasets in the development of intelligent models for emergency vehicle

detection. Finally, we will test and verify the generated dataset's accuracy using a realistic dataset to ensure the model's effectiveness.

1.2. PROBLEM STATEMENT

The development of an intelligent model that accurately detects emergency vehicles on road networks remains a challenge due to the lack of realistic and balanced datasets. Most existing studies have achieved high accuracy levels, but these models fail to perform adequately in real-world scenarios due to issues such as data imbalance, unrealistic image data, and inadequate evaluation criteria. Therefore, there is a need to develop a new approach that can produce a realistic dataset with balanced classes of emergency and non-emergency vehicles. Additionally, the model should be evaluated using appropriate metrics such as precision, recall, and F1-score to ensure its practicality in real-world applications. Furthermore, data augmentation techniques should be explored to preserve the integrity of the data and avoid duplicating images, leading to higher accuracy levels. The ultimate goal is to develop an intelligent model that can accurately detect emergency vehicles on the road, assisting drivers and autonomous vehicles in critical situations.

1.3. RESEARCH CONTRIBUTIONS

Our work aims to make the following contributions to the field of intelligent emergency vehicle detection:

- The creation of a large, balanced, and realistic dataset of emergency vehicle images that can be used by other researchers to conduct their own experiments. This dataset includes real-world images of emergency vehicles on the streets, and we used GANs to generate additional images to increase the dataset's size and diversity.
- The development of a new CNN-based model for emergency vehicle detection that achieves real and acceptable accuracy. We also evaluated the model using other measurement tools, such as Precision, Recall, and F1-Score, to ensure its reliability and effectiveness.

By providing a new dataset and model for emergency vehicle detection, our work aims to support the development of more intelligent transportation systems and assist drivers and autonomous vehicles in detecting emergency vehicles with greater accuracy and efficiency.

1.4. RESEARCH METHODOLOGY

To develop our intelligent model for detecting emergency vehicles, we followed a comprehensive research methodology designed to produce a reliable and accurate model.

First, the proposed work collected and analyzed available datasets for emergency vehicles. We filmed realistic emergency vehicles on the streets with our cameras and compiled additional images of emergency vehicles from several datasets. We standardized the size of the images to $416 * 416$ and removed duplicate and unrealistic images. This work used various augmentation operations such as rotation, shifting, shearing, zooming, flipping, and fill modes. We also utilized DCGANs to generate new images, trained the model on the dataset we collected, and built several layers of the generator and discriminator for a large number of epochs. Ultimately, we generated a dataset of around 20,000 realistic images. Next, we preprocessed the dataset to prepare it for training the machine learning model. We used various techniques, including data augmentation and normalization, to improve the quality and diversity of the dataset.

Then divided the data into 20% test, 10% validation, and 70% training datasets and employed CNN to build our vehicle detection model. We trained the model using the augmented and preprocessed dataset and evaluated its performance using various metrics, including accuracy, Precision, Recall, and F1-Score. We used a validation dataset to finetune the model's hyperparameters and ensure that it generalized well to new data. Finally, we tested the model on a realistic dataset and compared its performance to state-of-the-art methods. Our model outperformed the existing models and provided more realistic results. Through this methodology, we developed a realistic and accurate model for

detecting emergency vehicles that other researchers can use to conduct their own experiments.

Figure 1 depicts the proposed approach methodology for detecting emergency vehicles. The methodology involves several steps outlined as follows:

- **Initial Dataset Collection:** Thorough research on available datasets for emergency vehicle detection is conducted. An initial dataset is collected, serving as a starting point for further dataset construction. Public street images of both emergency and nonemergency vehicles are utilized to obtain a realistic dataset.
- **Data Augmentation:** Two methods are employed for dataset augmentation. The first method utilizes standard augmentation techniques such as flipping, mirroring, and adjusting brightness. The second method employs Generative Adversarial Networks (GANs) to generate new images based on the initial dataset. GANs consist of a generator, responsible for generating new images, and a discriminator, which categorizes the generated images as real or synthetic.
- **Dataset Pre-processing:** The newly obtained dataset undergoes pre-processing, which involves removing duplicate and unrealistic images, among other necessary operations.
- **CNN Algorithm and Model Construction:** The Convolutional Neural Network (CNN) algorithm is employed to build an emergency vehicle detection model. Images are classified into two categories: emergency vehicles and non-emergency vehicles. The dataset is divided into training, validation, and test sets. Model accuracy is evaluated using various metrics, such as F1 score, precision, recall, and accuracy.
- **Dataset Validation:** The dataset is validated using a 10-fold cross-validation technique. This method is commonly used to assess the performance of machine learning models, particularly in classification and prediction problems. The dataset is divided into multiple training and test sets in an equal and heterogeneous manner.

-

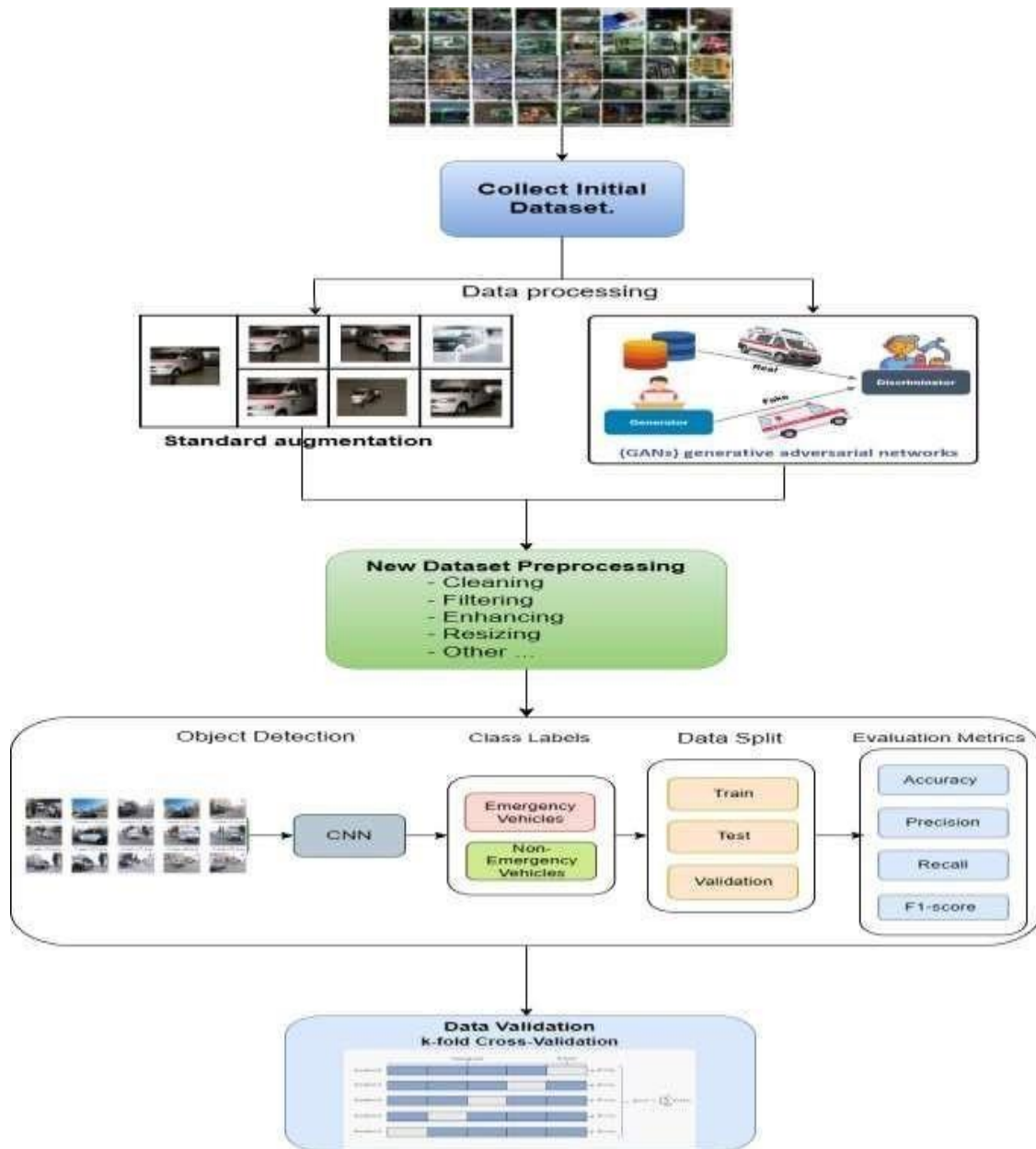


Figure 1.1: Proposed Approach Methodology for Emergency Vehicle Detection

1.5. RESEARCH LAYOUT

This project has presented a comprehensive framework for detecting emergency vehicles on road networks. The seven chapters have addressed various aspects of the research problem, providing valuable insights and contributions to the field.

Chapter 1 introduced the research problem and objectives, highlighting the significance of detecting emergency vehicles on road networks. Chapter 2 conducted a thorough literature review. Chapter 3 provides ai, scope and examining previous work related to the detection of emergency vehicles. It analyzed related technologies

and their applications in this field, including the utilization of Generative Adversarial Networks (GANs) in vehicle detection.

Chapter 4 provided a background on the essential aspects of the research, including the algorithms used and other relevant information and analyzed existing datasets, assessing their strengths and weaknesses for emergency vehicle detection. It focused on the generation of a newly balanced dataset. It involved gathering data from available datasets, collecting images through video recording and frame selection, dataset augmentation techniques, and necessary preprocessing steps.

Chapter 5 delved into the performance evaluation of the new dataset. It developed an emergency vehicle detection model using the Convolutional Neural Network (CNN) algorithm. The model's robustness and reliability were ensured through the use of the 10-fold cross-validation technique. Comparative analysis with previously developed models and online available detection models was conducted to assess the performance of the proposed model.

Finally, Chapter 6 concludes the project and discusses future research directions. The findings of this research contribute to the field of detecting emergency vehicles on road networks, providing a foundation for further advancements. The creation of a new dataset opens up possibilities for future work, such as dataset expansion, fine-tuning and transfer learning, multi-modal fusion, real-time implementation, and adapting to new emergency vehicle types.

In conclusion, this project has provided valuable insights into the detection of emergency vehicles on road networks. It has laid the groundwork for further research and development in this field, with the aim of enhancing traffic safety and management.

CHAPTER 2

LITERATURE SURVEY

2.1. INTRODUCTION

With the exponential growth of vehicles and the consequent increase in traffic congestion, there has been a growing interest among researchers in addressing the challenges related to the mobility of emergency vehicles. In response to this issue, many researchers have explored innovative approaches and technologies, such as the Internet of Things (IoT), to develop new tools and models aimed at mitigating the impact of these problems. In this chapter, we delve into two distinct areas of scientific research. Firstly, we examine previous studies focused on vehicle and object detection methods employed in the context of road networks. Secondly, we explore the utilization of Generative Adversarial Networks (GANs) models to enhance or augment existing datasets, emphasizing their role in improving the accuracy of vehicle detection systems.

2.2. VEHICLE AND OBJECT DETECTION METHODS

The development of self-driving cars aims to enhance safety, minimize accidents, and provide efficient and reliable transportation (Gupta et al., 2021). To achieve these goals, specific object recognition is crucial in accurately identifying and assigning distinguishing attributes to objects, including vehicles, using appropriate names (Wang et al., 2021). This section delves into the existing literature on vehicle and object detection methods employed within road networks. These methods encompass diverse approaches, spanning traditional computer vision techniques to modern deep learning algorithms. Researchers have investigated various features, such as shapes, colors, and textures, to enable precise detection and classification of vehicles and other objects in real-world scenarios. We thoroughly examine the strengths and limitations of these methods and highlight significant findings and advancements in this field of study.

Shuvendu Roy and Sakif Rahman (2019) have proposed an automated detection system to detect emergency vehicles from a closed-circuit television camera (CCTV) using a neural network. They have used a pre-trained model to detect objects containing 80 different categories (e.g., bike, car, motorcycle). This

study aimed to classify the detected vehicles into two types: emergency vehicles and non-emergency vehicles. Then, a convolutional neural network (CNN) technology was used for the emergency vehicle classification class based on analyzing the visually detected images. They have augmented the dataset by changing some properties, such as mirroring the images, zooming in, zooming out, rotating, etc. (Shung, 2018). After that, they used the YOLOv3 algorithm to detect emergency vehicles (Redmon et al., 2018). In the obtained results, the accuracy of classifying the vehicles is more than 97%. This is a better accuracy than the obtained of the traditional system that had previously been used. However, this study never determined the size of the used dataset before or after the augmentation process. Moreover, the confusion matrix has not been investigated to show if the data is balanced or not and to ensure that the accuracy standard used can be adopted or needs more improvement (Roy and Rahman, 2019).

On the other hand, two recent works we proposed by Ganesh D and Punyavathi et al. The first study (Ganesh D, 2022) presented an automated system to detect emergency vehicles using Machine Learning and CNN. The dataset used is from UCI Repository (2000 Images) with two types to classify emergency vehicles (i.e., emergency vehicles and non-emergency vehicles). The researcher standardized the dataset by some algorithms and adjusted the images such as `do_flip`, `flip_over`, etc. He develops a new model by the neural network, using the pre-trained CNN model “ResNet152 CNN Model”. This is done by using Pandas and other libraries in Python. In his model, the training data is divided into emergency vehicles and non-emergency vehicles (439 images Class A, 372 images Class B), the rest data is 20% for validation and 53 images for testing. In the results, the researcher proposed an automated system model to detect emergency vehicles using Machine Learning and CNN. However, this is without exposing the actual application of the model or giving any results or accuracy of the detection. Instead, the research describes how the model works and provides advice and the mechanism for building the model.

The latter study proposed by Punyavathi et al. (Punyavathi et al., 2022) investigated various methods for detecting vehicles, including traditional statistical approaches, a novel algorithm for vehicle classification, and deep

learning using the YOLOv3 algorithm. The study demonstrated the effectiveness of the YOLOv3 algorithm in detecting and tracking vehicles, achieving an average precision of 95.8% and a recall of 97.5% on a dataset consisting of vehicles captured under different lighting and weather conditions. The dataset used in the study was collected using cameras placed on vehicles and fixed locations. Although the paper did not explicitly mention whether the dataset was balanced or imbalanced, an analysis of the confusion matrix results presented in Table 1 suggests that it is unbalanced. Therefore, relying solely on accuracy and recall as evaluation measures may not be appropriate, as mentioned earlier.

Moreover, Pillai and Valles, 2020, designed a new model to classify and detect vehicles using CNN and YOLO by their types and colors and presents two methodologies. In the first one, the researcher designs two CNN models, one to classify the vehicle's types and to classify the vehicle's colors. Indeed, CNN provides good results in image recognition tasks. In the second methodology, the researcher designed a faster model to detect vehicles using YOLO because of the high accuracy and speed of this algorithm, especially in all weather conditions. In this work, the researcher has used two different datasets. Comprehensive Cars (CompCars) Dataset, which contains about 50,000 images of different types of cars (12 classes, such as sedan, hatchback, bus, etc.). This dataset has been used to evaluate the type of vehicle. The other dataset is Vehicle Color Recognition consists of 15,601 images for vehicle colors and includes eight classes (black, blue, etc.). This work has enhanced the images by removing fog, improving contrast, and filtering noise, using several tools such as DCP or CLAHE (Park et al., 2018). This improves accuracy in detecting vehicles. The object was extracted in several ways, such as a colored graph, HOG, and pre-trained CNN, using the coordinates of the four corners of the image perimeter and the proportion of the presence of the vehicle relative to the square. However, the other proposed methodology uses YOLOv3 to classify the vehicle by detecting the type of vehicle as well as the color of the vehicle. Despite the precise sequence of the work method, this study has presented a proposal for the detection method without testing or discussing the results of the experiments (Pillai and Valles, 2020).

Table 2.1. summarizes some previous studies that have mainly proposed to detect emergency vehicles on the road network. The main objectives, used algorithm, datasets, obtained results, and drawbacks of each study have been illustrated in this table.

Table 2.1: Main Characteristics of Previous Emergency Vehicles' Detection Mechanisms

Related Work	Algorithm Used	Dataset Used	Performance	Limitations
(Roy and Rahman, 2019) Detect emergency vehicles	Convolutional Neural Network (CNN) YOLO v3	COCO	Accuracy 97%	The researcher did not explain the amount of data after processing, nor did he indicate the confusion matrix to show whether the data was balanced or not
(Ganesh D, 2022) Proposes an automated system to detect emergency vehicles	CN6N	UCI 75% train 25% test	The proposed system, according to the researcher, is better than the traditional	The researcher did not show any actual results showing the success of the experiment and the accuracy values
(Punyavathi et al., 2022) Detect vehicles	Traditional algorithms YOLO v3	The source of the dataset is not mentioned	ACC and Recall>97%	According to the number of data in the confusion matrix, the data was not balanced, and acc and Recall could not be adopted as real indicators of accuracy
(Pillai and Valles, 2020)	CNN YOLO 9000	CompCar Vehicle color recognition	New model for Classification and detection of the type and color of	The author presented only a proposal without offering any results of the experiments

2.3. USING GENERATIVE ADVERSARIAL NETWORKS (GANs) TO GENERATE AUGMENTED DATASETS

In recent years, Generative Adversarial Networks (GANs) have become a powerful technique for generating augmented datasets (Tanaka &Aranha, 2019). Researchers have recognized the potential of GANs in enhancing the training process and improving the performance of detection models by leveraging these augmented datasets. GANs have shown promising results in various computer vision tasks, including vehicle and object detection. Consequently, several research studies have been conducted to explore the application of GANs in generating new and suitable datasets to enhance the accuracy and effectiveness of detection models.

The study conducted by Tanaka &Aranha, 2019, has used the generative adversarial networks (GANs) model to generate new artificial datasets. These datasets have been used to train machine learning. The trained machine learning algorithms have been evaluated to prove the usefulness of GANs in generating new datasets. For training a classifier using a dataset or balancing a dataset by oversampling lower classes, the researcher used the same general GANs structure to generate the synthetic data. In this work, the training did not use all the synthetic datasets and the decision tree classifier was used to examine the quality of the training data; the trained datasets are as in Table 3. The result of evaluating the performance of the generated GANs data was better than the original data. As for the balanced scenario using the generated GANs data only, the assessment showed that the GANs data had better accuracy and recall than the original dataset. Table 4 is one of the results of the evaluation. The above results confirm the need to oversample data using the GANs method for images specifically, which proves necessary and essential to over-sample the images in datasets, and that the new data generated from the original data gives results of the same or better assessment than the original data (Tanaka &Aranha, 2019). Kumar and Kaur, 2020, proposed the creation of a new model using GANs to identify vehicle plates with accuracy and speed. After collecting the images and changing their size, they lost many images that had a size less than required to experiment. Thus, this study suggested using GANs to oversample the images.

The proposed model is that after the camera captures the image of the vehicle, it sends the image to the GANs. In turn GANs generates new artificial images. Then, the CNN algorithm using deep learning is used to detect the images. The generator in GANs generates new images that are similar to authentic images, and the discriminator (classifier) tries to differentiate between fake and real images. The dataset was used from Pascal, increased from other datasets and images taken by the camera. The classifier is used to train dataset and create a noisy subset sample with size m , then the generator training the data. The proposed method achieves an accuracy of up to 99% in recognizing car plates compared to other studies that did not use the GANs to increase the data sample. In this study, the work notices the vehicle great importance of data augmentation using GAN, as data is generated in real-time and with high accuracy, and this is an important thing (Kumar and Kaur, 2020).

Furthermore, Yuan Gao et al., 2021, prove that data augmentation using GANs is a suitable method for generating new images of unmanned aerial vehicles (UAV). Using infrared data collection as it is the primary method for detecting all-weather UAVs that are detected through CNN. To obtain better detection effects with better accuracy, the researcher increased the infrared dataset of UAVs through the hostile generation network (GANs). The dataset used 760 images, 75% was for training and 25% for testing, as experiments proved that the detection of the enlarged data using GANs was more accurate than the one trained on the original data. This is due to the fact that the infrared images of the UAV are relatively simple and they do not have any complexity, the researcher chose the DCGANs with fully convolutional layers, as it is the most suitable for generating new artificial images. The work was divided into two parts, the first is to train the GANs to generate new images that are identical to reality by entering the original set of images for the GANs, and the second, after testing, the generated images are combined with authentic images to be trained now to detect UAV through CNN as shown in the figure 2.1.

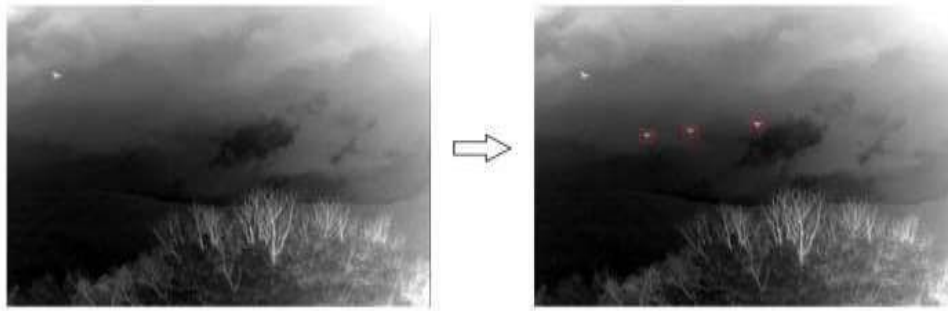


Fig 2.1. The image after merging the generated images by GANs

After the GANs was tested on the datasets, and the detection of UAV was examined on the trained dataset after the trained GANs on the original data and the results were compared. The researcher proved that the results of the detection of UAV that were trained by the dataset after the GANs give better results and accuracy according to Standard accuracy (map@.5 & F1 Score) as shown in Table 5 (Yuan et al., 2021).

Finally, Sathananthavathi et al., 2022, merged the YOLO to detect vehicles with GANs algorithms to improve the night images and increase the accuracy of detecting vehicles at night. Thus, night images are introduced to GANs, which in turn generate new daytime images to increase the accuracy of vehicle detection. That is by translating night images into daytime images using (Cycle GANs) to improve the features of images and vehicles and improve lighting conditions and bad weather. Two GANs have been used in this work, for each GAN, one generator and one discriminator. Generator A is responsible of translating the nighttime images into daytime images and then checking the fake "generated" daytime images for the real-time image in Discriminator. Generator B takes real daytime images and generates nighttime images. Discriminator B checks the output to see if it is a real or fake image. Cycle GANs, suggested by the researchers, is the first proposed model for improving the images. In contrast, the second model is the detection of vehicles in the images using the neural network and YOLOv5, which is trained on the entire image. The images were trained and tested by the GANs provided by the researcher to detect the vehicles by YOLOv5, where the Unet (uses a segmentation network) architecture was adopted in GANs A, which generates more detailed information than the Resnet architecture that

was used in GANs B. The dataset was used from Berkeley Deep Drive. The focus here was on deep learning because it is characterized by permanent and automatic learning in improving classification and prediction (Sathananthavathi et al., 2022). Table 6 shows results for using the proposed GANs and high accuracy in detecting vehicles.

Input image	Shallow unet as Generator A(proposed architecture)	Resnet as Generator A
		
		
		

Figure 2.2 The proposed GANs

From this research, this work came to the certainty that the accurate detection of vehicles needs to choose a correct mechanism to improve the images before thinking of starting a vehicle detection. This helps to reach a very high accuracy in the accurate detection of vehicles, as well as in the monitoring systems.

All previous research shows that used the GANs model to pre-process data either to enhance or augment the images or balance the datasets. Table 7 summarizes the main previous studies that have used GANs to generate new datasets. It illustrates the main objective, used algorithm, original dataset, obtained results, drawbacks, and limitations.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1. INTRODUCTION

The background section provides a concise and comprehensive overview of the essential aspects of our research. It encompasses the methods employed to construct the dataset, including traditional augmentation techniques, the utilization of Generative Adversarial Networks (GANs), and data processing procedures. Additionally, it elucidates the specialized algorithms used in developing the Convolutional Neural Network (CNN) model for emergency vehicle detection. Furthermore, it discusses the approaches employed to verify the dataset, such as the application of 10-fold cross-validation. Lastly, the evaluation of the model is conducted using performance measurement tools, including the F1 score, accuracy, and recall metrics. By incorporating these key elements into the background section, a clear and scientifically sound foundation is established for the subsequent chapters. This ensures that the reader gains a comprehensive understanding of the fundamental methodologies and techniques implemented in our research.

In this chapter, the work delve into different datasets that encompass images of emergency vehicles, specifically ambulances, fire trucks, and police vehicles. Our assessment revolves around examining the datasets in terms of their type, quantity, and quality of images. The project mainly focus on the realism of the images and their capture of real-world scenarios. Additionally, this proposed work explore the image augmentation techniques employed in these datasets and how researchers have utilized them for various purposes. This analysis aimsto identify any limitations or shortcomings in the existing datasets. To validate our findings, the project tests real-world images using previously proposed models in this domain to determine their effectiveness. Through this process, the proposed work aim to gain insights into the available traffic datasets containing emergency vehicles and contribute to understanding their strengths and weaknesses.

3.2. EMERGENCY VEHICLE DETECTION

The “emergency vehicle detection” dataset (project-dbqtw, 2023). from Roboflow (Dwyer and Nelson, 2022) contains 365 training images and 158 testing and validation images with a medium quality of 640x640 pixels. However, all the images in the training set are of ambulance vehicles, and there are no other emergency vehicles, such as police or fire truck vehicles. It is an unbalanced dataset; only a few images contain ambulance vehicles. Additionally, no image augmentation techniques were applied. Dissanayake et al., 2022, used this dataset by the YOLO detection algorithm. The dataset was divided into 80% training and 20% testing and validation, aiming to detect emergency vehicles upon their arrival at the traffic light and give them a higher priority to pass through the signalized intersection. This study has obtained an accuracy of 82%

(Dissanayake et al., 2022). After testing the online model available on Roboflow (project-dbqtw,

2023) with several realistic images of emergency vehicles, it was observed that the model’s detection performance was poor. This suggests that the model is not trained to detect many types of emergency vehicles outside of its limited dataset.

3.3. “JANATAHACK AV COMPUTERVISION” AND “EMERGENCY VS NON-EMERGENCY VEHICLE CLASSIFICATION”

The second dataset, found in multiple locations on Kaggle” JanataHack AV Computer Vision” (Shravan, 2021), “Emergency vs Non-Emergency Vehicle Classification” (Chokhra, 2020), contains approximately 3300 images, including 1000 emergency vehicles such as ambulances, fire trucks, and police vehicles. It also contains around 1300 images of other vehicles, making it a nearly balanced dataset. However, the images are of poor quality, with a resolution of 224x224 pixels. Kherraki and El Ouazzani, 2022, used this dataset for emergency vehicle classification, achieving over 90% accuracy, but the primary issue remains the quality of the images and the limited use of data augmentation (Kherraki and El Ouazzani, 2022).

3.4. AMBULANCE REGRESSION

The” Ambulance Regression” dataset (SipalingAI, n.d.) contains 307 images of ambulances in YOLOv8 format, with 294 training images and 13 testing

images. Standard augmentation techniques were applied, including rotation, shear, and brightness adjustment.

However, the dataset lacks real road images, and there are too few test images for training images.

3.5. DATASET USED

The “Ambulans” dataset (SipalingAI, 2023) contains 2134 images of ambulances in YOLOv8 format, with standard augmentation techniques applied, including rotation, shear, brightness adjustment, exposure adjustment, and Gaussian blur. However, the quality of some images may have been reduced due to augmentation. Many of the images are not from real roads, but rather from exhibitions or the internet.

3.6. AMBULANCE DETECT

The “ambulance detect” dataset (hacknjill, 2023) contains 1400 images of ambulances in YOLOv8 format, with no augmentation techniques applied. The dataset has 980 training images, 140 testing images, and 280 validation images. However, as mentioned earlier, there is a lack of real road images in the dataset. Additionally, some of the images in the dataset are taken from car exhibitions, while others are captured from real-world scenarios. Nevertheless, the number of testing images is relatively small compared to the number of training images, which may affect the model’s generalization ability.

3.7. EMERGENCY VEHICLE DETECTION

The “Emergency Vehicle Detection” dataset (John, 2023) contains 1680 images of cars in YOLOv8 format, with standard augmentation techniques applied, including horizontal flip, random shear, and salt and pepper noise. The dataset has 1470 training images, 71 testing images, and 139 validation images. However, it does not contain images that only feature emergency vehicles for training purposes. Most images in the “Emergency Vehicle Detection” dataset were captured using video cameras placed in specific locations. This could be attributed to the dataset’s purpose of detecting emergency vehicles in a particular context, such as a specific city or region.

However, this may limit the dataset's ability to train the model to detect emergency vehicles in other locations. Additionally, some of the captured images may not show detailed features of emergency vehicles due to the cameras being positioned at a distance from them. This could potentially create challenges for the model when attempting to classify and detect emergency vehicles accurately.

3.8. FALCK

The “FALCK” dataset (Folio3, 2023) contains 176 images of ambulances and firefighting vehicles in YOLOv8 format, with no augmentation techniques applied. The dataset has 140 training images, one testing image, and 35 validation images. The images are of good quality, but there is a lack of real road images, and the number of test images is too small for training images.

3.9. SIRENS

Similarly, the “Sirens” dataset (Martin, 2023) comprises 213 medium-quality images of ambulances, firefighting, and police vehicles in YOLOv8 format, including 145 training images, 22 testing images, and 44 validation images. However, this dataset is very small, which limits its ability to build high-accuracy and realistic detection models. Additionally, there is also a lack of real-world road images in this dataset.

3.10. SMART CAR:

The “Smart car” dataset (Ahmed, 2022) was designed for detecting emergency vehicles, including ambulances, firefighting, and police vehicles. It includes 1152 images pre-processed with auto-orientation and resizing to 640x640 (stretch), with no augmentation techniques. The dataset is split into 921 training and 231 testing images, with medium-quality images. However, the dataset lacks real road images, and some of the images are unrealistic. Moreover, most of the images were not taken on the road. Furthermore, the dataset does not contain any images of vehicles from other classes. The datasets of the proposed work are reviewed exhibit various limitations and inadequacies. Many available datasets lack realism. They comprise images not obtained from real world scenarios. This may affect the models' ability to generalize to practical situations. Available datasets often suffer from a class of imbalance.

Some types of emergency vehicles have a disproportionate number of images compared to others. Consequently, the models' performance may be biased toward certain classes and suboptimal for others.

Most datasets have limited test data, making it challenging to accurately assess the models' performance. Poor test data quality also complicates developing models that work effectively in real-world scenarios. Some datasets have few images and lack comprehensive data augmentation techniques, hindering the model's generalization of different scenarios. Excessive augmentation techniques may also reduce image quality.

The limited usage of these datasets in published research suggests that they are not widely recognized or effective for ambulance detection.

Table 3.1: Summary of the Most Important Findings from the Available Datasets

Dataset	Number of Images	Quality	Vehicle Types	Limitations	Sample
Emergency Vehicle Detection. (project-dbqtw, 2023)	523 Images 365 (training) 158 (testing) and (validation)	Medium (640x640 Pixels)	Only ambulance	Unbalanced dataset, cannot detect emergency vehicles outside of the training dataset	
Janata Hack AV Computer Vision (Shravan, 2021) Emergency vs Non-Emergency Vehicle Classification (Chokhra, 2020)	Approx. 3300 Images	Poor (224x224 Pixels)	Ambulances, fire, police, and other vehicles	Low-quality images, limited augmentation	

Ambulance Regression (SipalingAI, n.d.)	333 Images 307 (training) 26 (testing) and (validation)	Medium (416x416 Pixels)	Only ambulances	Lack of real road images, too few test images for training images, overfitting	
Ambulans (SipalingAI, 2023)	2134 Images	Medium (416x416 Pixels)	Only ambulance	Some images may be low quality due to augmentation, lack of test data, unrealistic images	
Ambulance detect (hacknjill, 2023)	1400 Images 980 (training) 420 (testing) and (validation)	Medium (640x640)	Only ambulances	Lack of real road images, too few test images for training images	
Emergency Vehicle Detection (John, 2023)	1680 Images 1470 (training) 210 (testing) and (validation)	Medium (416x416)	Cars (Includes emergency vehicles)	Does not contain images of only. Emergency vehicles for training	
FALCK dataset (Folio3, 2023)	176 Images 140 (training) 36 (testing) and (validation)	High (1280x720)	Ambulances and firetruck Vehicles	Lack of real road images, too few test images for training images, unrealistic images	
Siren's dataset (Martin, 2023)	213 Images 145 (training) 66 (testing) and (validation)	Medium (416x416 Pixels)	Ambulances, firetrucks , and police Vehicles	Lack of real road images, small dataset images, unrealistic images	

In conclusion, the analysis of the available traffic datasets containing images of emergency vehicles has provided valuable insights into their

characteristics and limitations. While these datasets contribute to the advancement of emergency vehicle detection methods, they also exhibit certain shortcomings. The datasets vary in terms of image quantity, quality, balance, and representation of different types of emergency vehicles. Several datasets lack real road images, limiting their ability to train models for real-world scenarios. Furthermore, some datasets suffer from a small number of test images, which can affect the generalization ability of the models.

To address these limitations and further enhance the field of emergency vehicle detection, the next chapter of this project will focus on building a new dataset. This dataset will leverage real world imaging, research, and the use of Generative Adversarial Networks (GANs) to generate augmented data. By capturing images from reality and utilizing GANs for dataset generation, the aim is to create a more comprehensive and diverse dataset that closely represents real-world traffic scenarios. The potential of GANs for augmentation will be explored in the next section, offering a promising approach to overcome the limitations identified in the available datasets.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1. INTRODUCTION

The background section provides a concise and comprehensive overview of the essential aspects of our research. It encompasses the methods employed to construct the dataset, including traditional augmentation techniques, the utilization of Generative Adversarial Networks (GANs), and data processing procedures. Additionally, it elucidates the specialized algorithms used in developing the Convolutional Neural Network (CNN) model for emergency vehicle detection. Furthermore, it discusses the approaches employed to verify the dataset, such as the application of 10-fold cross-validation. Lastly, the evaluation of the model is conducted using performance measurement tools, including the F1 score, accuracy, and recall metrics. By incorporating these key elements into the background section, a clear and scientifically sound foundation is established for the subsequent chapters. This ensures that the reader gains a comprehensive understanding of the fundamental methodologies and techniques implemented in our research.

4.2. METHODS FOR DETECTING EMERGENCY VEHICLES

Vehicle detection methods have evolved over time, encompassing both traditional approaches that have been employed since the inception of emergency vehicles and more recent smart methods utilizing artificial intelligence techniques. This section will outline these methods, categorizing them as traditional methods and AI-based techniques.

4.2.1. Traditional Methods :

Traditional methods for detecting emergency vehicles rely on established rules and characteristics associated with emergency vehicles, such as flashing lights and sirens, and distinctive colors. These methods have been in use since the emergence of emergency vehicles but may have limitations in terms of adaptability and accuracy (De Lorenzo et al., 1991).

4.2.2. AI-based techniques :

Which is based on machine learning and deep learning Machine Learning Methods. In the context of detecting emergency vehicles, traditional machine learning algorithms such as Support Vector Machines (SVM) and Random Forests have been widely employed. These methods rely on manually engineered features extracted from the data, which are then used to classify emergency vehicles. However, the performance of these methods heavily depends on the quality and representativeness of the chosen features (Razalli et al., 2020).

Deep Learning Methods: Deep learning methods, particularly Convolutional Neural Networks (CNNs), have revolutionized vehicle detection, including the detection of emergency vehicles. CNNs have the ability to automatically learn hierarchical representations of the data, capturing complex features and patterns. They exhibit robustness to variations in appearance, lighting conditions, and backgrounds, enabling better generalization and adaptability (Tran et al., 2020).

In summary, machine learning and deep learning methods, specifically CNNs, offer powerful and effective approaches for detecting emergency vehicles by automatically learning discriminative features from the data. These methods outperform traditional techniques by providing superior adaptability, generalization, and robustness in detecting emergency vehicles across diverse real-world scenarios.

4.3. CONVOLUTIONAL NEURAL NETWORKS (CNNs)

CNNs are powerful models used for detecting emergency vehicles in images or video streams. They are designed to mimic how the human brain processes visual information, CNN networks work to detect vehicles in emergency as follows (AjitArohan et al., 2020):

- **Image Analysis:** CNNs analyze images by breaking them down into smaller parts called patches and examining them closely.
- **Feature Extraction:** At each level of analysis, the CNN looks for specific patterns or features that are important for identifying emergency vehicles.

These features can include shapes, colors, textures, or other visual characteristics.

- **Learning and Training:** CNNs learn to recognize these important features by being trained on a large dataset of labeled images. During training, the CNN adjusts its internal parameters to optimize its ability to correctly classify images based on the presence or absence of emergency vehicles.
- **Detection and Classification:** Once trained, the CNN can be used to detect emergency vehicles in new, unseen images. It scans the input image using its learned features and makes predictions about the presence or absence of an emergency vehicle.

Convolutional Neural Networks (CNNs) consist of several layers that work together to detect emergency vehicles in images. Here's how the steps mentioned earlier align with the names of the common layers in a CNN (Liu Yao et al., 2021):

- **Convolutional Layer:** Applies filters to extract features from the input image.
- **ReLU Layer:** Introduces non-linearity to capture complex relationships between features.
- **Pooling Layer:** Downsamples the feature maps, reducing spatial dimensions.
- **Flattening Layer:** Converts the output into a one-dimensional vector.
- **Fully Connected Layers:** Learn global patterns and relationships for classification.

In summary, the convolutional layer extracts features, the ReLU layer introduces nonlinearity, the pooling layer downsamples the features, the flattening layer prepares the data for fully connected layers, and the fully connected layers learn the relationships for classification, Figure 1.1 shows how to extract features from images.

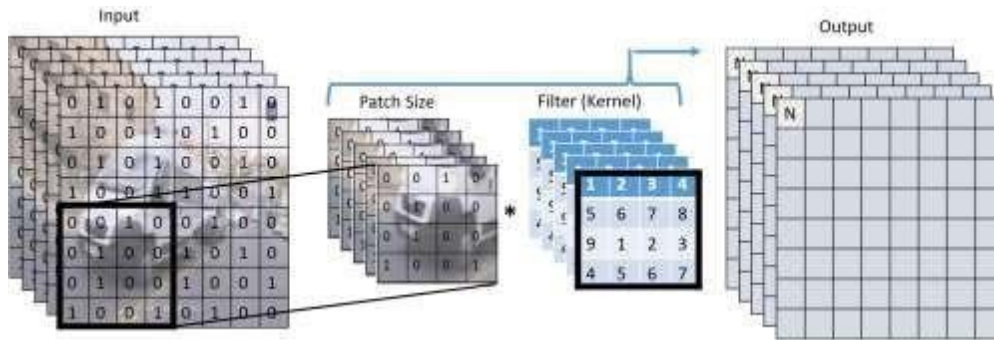


Figure 4.1: Features Extraction by CNN with layers

4.4.CNN, R-CNN, AND FAST R-CNN

CNN, R-CNN, and Fast R-CNN are popular deep learning architectures used in image analysis and object detection tasks. They have different characteristics and performance trade-offs, with Fast R-CNN being a faster and more efficient version of R-CNN (Jiang et al., 2017):

- CNN stands for Convolutional Neural Network, which is a type of neural network commonly used for image processing and computer vision tasks. It is designed to automatically learn and extract features from input images through convolutional and pooling layers.
- R-CNN stands for Region-based Convolutional Neural Network. It is an extension of CNN that focuses on object detection. R-CNN proposes regions of interest in an image and applies a CNN to each region to classify and localize objects.
- Fast R-CNN is an improved version of R-CNN that addresses some of its limitations. It introduces a shared feature extraction network and performs region-wise pooling, enabling faster and more efficient object detection.

4.5.GENERATE NEW EMERGENCY VEHICLE IMAGES USING GANs

Generating new emergency vehicle images using GANs involves leveraging Generative Adversarial Networks (GANs), which are a type of generative model in machine learning. GANs consist of a generator network and a discriminator network. The generator network creates new samples, such as images, while the discriminator network attempts to differentiate between real

samples from a training dataset and generated samples from the generator (Navidan et al., 2021).

Specifically, Deep Convolutional GANs (DCGANs) are a variant of GANs that utilize deep convolutional neural networks. DCGANs enhance the traditional GANs framework by replacing fully connected layers with convolutional layers in both the generator and discriminator networks. This modification enables DCGANs to capture complex patterns and spatial dependencies in the data, resulting in the generation of more sophisticated and realistic images (Jain et al., 2022).

GANs and DCGANs have significantly impacted the field of generative modeling and have found successful applications in various domains, including computer vision, art generation, and data augmentation. They offer a powerful approach for generating new emergency vehicle images, providing opportunities for data expansion and creative exploration.

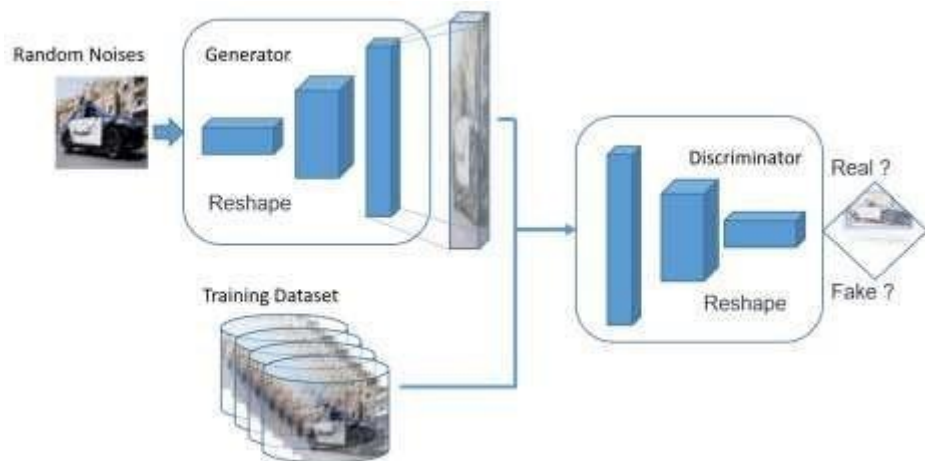


Figure 4.2: GANs Image Generation General Process

Figure 4.2 visually represents the general process of image generation using GANs. It illustrates how the generator network takes random noise as input and progressively transforms it through multiple layers. As the input noise passes through these layers, it gradually generates an image that becomes more refined and closely resembles the desired output.

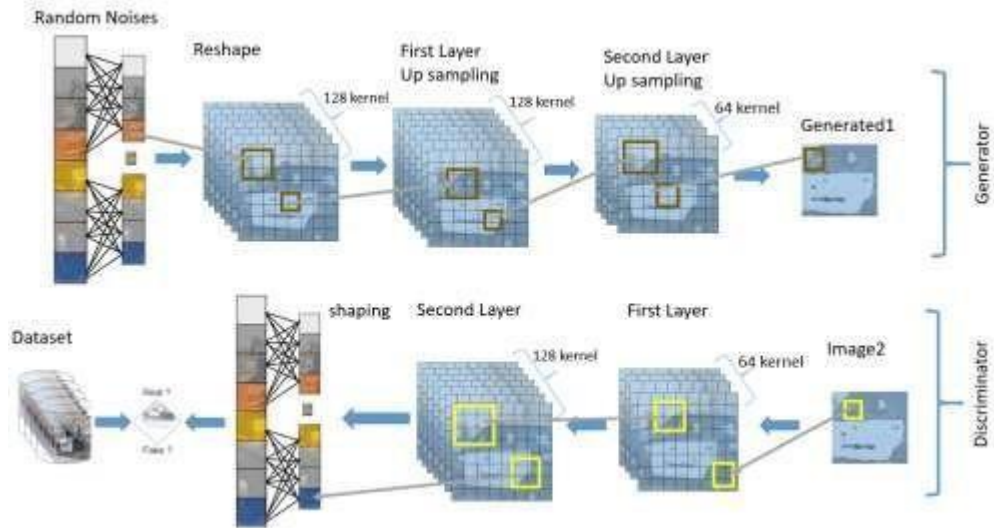


Figure 4.3: GANs Generator and Discriminator Layers Process

Figure 4.3 illustrates the process of GANs at the level of the generator and discriminator layers. The generator takes random noise as input and passes it through a series of layers, such as convolutional and fully connected layers. These layers gradually learn and generate complex features, patterns, and details, ultimately producing the desired image.

Together, these figures visually depict the key steps and processes involved in the image generation using GANs, emphasizing the interplay between the generator and discriminator networks in achieving the desired output.

4.6. PERFORMANCE EVALUATION

Validation involves evaluating the performance of a model or system on an independent dataset to determine its generalization capabilities. It ensures that the model can effectively handle unseen data and make accurate predictions or classifications. By validating the model, the work gains confidence in its ability to perform well in real-world scenarios beyond the training data (Gabler et al., 2020).

4.6.1.K-fold Cross-Validation:

To validate the performance of our system, the proposed work utilize k-fold cross-validation. This technique involves dividing the dataset into k equal-sized subsets, or folds. The system is then trained and evaluated k times, with each fold serving as the validation set once, while the remaining folds are used for

training. This process helps to provide a more robust and reliable estimate of the system's performance by utilizing different subsets of the data for training and validation (Wong et al., 2019).



Figure 4.4: K-fold Cross-Validation form first fold to K fold

In Figure 4.6, the original dataset is represented by the large rectangular box. The dataset is divided into K smaller subsets, denoted by the numbered rectangles (1 to K). During each iteration of K-fold Cross-Validation, one of these subsets is chosen as the evaluation set (highlighted in a darker shade), while the remaining K-1 subsets are combined to form the training set.

4.6.2. Evaluation Metrics:

The proposed work employ several standard evaluation metrics to measure the performance of our emergency vehicle detection system. These metrics include accuracy, precision, recall, F1 score, and the confusion matrix (Yacouby et al., 2020).

- Accuracy: It measures the overall correctness of the system's predictions by calculating the ratio of correctly classified instances to the total instances.

$$\text{Accuracy} = \frac{\text{Total number of corrected predictions}}{\text{Total number of predictions}} \quad (1)$$

- Precision: It quantifies the proportion of correctly predicted emergency vehicle instances out of all the instances predicted as emergency vehicles.

$$\text{Precision} = \frac{\text{Total number of True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

- Recall: It measures the proportion of correctly predicted emergency vehicle instances out of all the actual emergency vehicle instances.

$$\text{Recall} = \frac{\text{Total number of True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

- F1 score: It provides a balanced measure of precision and recall, taking into account both metrics to evaluate the system's performance.

$$\text{F1 score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

- Confusion Matrix: It presents a tabular representation of the system's predictions, comparing them with the actual labels. It provides insights into true positives, true negatives, false positives, and false negatives, aiding in the assessment of the system's performance.

Confusion Matrix	Class A	Class B
Class A	True Positives	False Positives
Class B	False Negatives	True Negatives

Figure 4.5: Confusion Matrix

Figure 4.7 illustrates a Confusion Matrix representing the classification performance between two classes, Class A and Class B. The matrix consists of four cells displaying the counts of True Positives, False Positives, False Negatives, and True Negatives.

- True Positives: This cell represents the instances where the model correctly predicted Class A as Class A.
- False Positives: This cell represents the instances where the model incorrectly predicted Class B as Class A.
- False Negatives: This cell represents the instances where the model incorrectly predicted Class A as Class B.
- True Negatives: This cell represents the instances where the model correctly predicted Class B as Class B.

In Figure 4.7, the True Positives and True Negatives cells are highlighted in green, indicating correct predictions within their respective classes. The False Positives and False Negatives cells are highlighted in red, indicating incorrect predictions.

4.7. GENERATING A NEWLY BALANCED DATASET INTRODUCTION

In the previous chapter, the project examined the limitations and shortcomings of existing traffic datasets used for emergency vehicle detection. These limitations include imbalanced representation of emergency vehicle types, insufficient test images, lack of real road images, and limited diversity within the datasets. To overcome these challenges and further enhance the field of emergency vehicle detection, this chapter focuses on the generation of a newly balanced dataset.

For the emergency vehicle detection model to be effective and accurate, it is essential to have a large and balanced dataset that encompasses realistic images of each category of emergency vehicles, including ambulances, firefighting vehicles, police vehicles, as well as non-emergency vehicles. In this section, this project has collected images from available datasets and supplemented them with real-life images captured from public streets. These images have undergone standard augmentation techniques to enhance their diversity and quality. Additionally, we have employed Generative Adversarial Networks (GANs) to generate new images by training them on the existing dataset. This approach aims to augment the dataset, improve its balance, and address the limitations observed in previous datasets, ultimately enhancing the model's ability to accurately detect emergency vehicles.

4.8. METHODOLOGY

A combination of real-world imaging, research, and the utilization of Generative Adversarial Networks (GANs) was employed to generate the newly balanced dataset for emergency vehicle detection. The methodology consisted of the following steps:

4.8.1. Gathering a Dataset for Emergency Vehicle Detection:

- Initial Data Collection: The project collected existing datasets that contain images of emergency vehicles, including ambulances, firefighting vehicles, and police vehicles, as well as non-emergency vehicles.
- Image Collection through Video Recording and Frame Selection: To supplement the existing datasets, the work conducted video recordings of public streets capturing real-life scenarios.
- Dataset Pre-processing: The collected images underwent pre-processing steps to ensure consistency and quality.
- Dataset Augmentation: To increase the diversity and variability of the dataset, this project applied augmentation techniques.

4.8.2. Generating New Images Using GANs

- Dataset Preparation and Processing: Preparing and processing the dataset ensures that the input data for the GANs model is clean, consistent, and properly annotated. This lays the foundation for generating new images using GANs that closely resemble real emergency vehicles.
- Generator and Discriminator layers: The construction of the generator and discriminator layers in the generator and discriminator networks is crucial for the successful generation of new images and the effective discrimination between real and generated images.
- Hyperparameters Adjustment: Adjusting hyperparameters is essential in training GANs to ensure optimal performance and generate high-quality vehicle images. Some of the key hyperparameters that need to be changed include (Learning Rate, Batch Size, and Number of Training Iterations).
- GANs Training and Generating New Images (Implementation): The GANs model is trained by alternating between updating the generator and discriminator networks. and it can generate new images of

emergency vehicles. Through the implementation of the GANs model and adjusting the hyperparameters, new images of emergency vehicles can be generated, addressing the limitations of the original dataset and enhancing the capabilities of the emergency vehicle detection model.

4.9. GATHERING A DATASET FOR EMERGENCY VEHICLE DETECTION

In the process of gathering a dataset for emergency vehicle detection, several steps were followed to ensure the inclusion of diverse and representative images. The methodology for gathering the dataset is as follows:

4.9.1. Initial Data Collection:

In the initial data collection step, our focus was on gathering a dataset that includes both emergency and non-emergency vehicle images. The proposed work employed various strategies to ensure the dataset's diversity and representativeness. Here is an overview of the initial data collection process:

- **Identifying Relevant Sources:** This work searched for publicly available datasets that contain images of emergency vehicles. Datasets such as (SipalingAI, n.d.; SipalingAI, 2023; Martin, 2023; Ahmed, 2022) were identified as valuable sources and provided us with approximately 500 images of emergency vehicles. Additionally, this work has obtained around 500 images of non-emergency vehicles from the dataset (Sapra, n.d.). These datasets formed the foundation of our initial dataset.
- **Expanding the Dataset:** To further enrich the dataset and increase its diversity, the proposed work conducted an extensive search for additional datasets specifically focused on emergency vehicles. During our search, the proposed work discovered the Firetruck Dataset (Maleki, 2022). and the Police Cars Dataset (FYP, 2022). These datasets were particularly useful for augmenting our initial dataset, contributing approximately 700 additional images.
- **Dataset Composition:** After the collection process, our initial dataset consisted of approximately 1,700 images. These images

encompassed various types of emergency vehicles, including ambulances, firefighting vehicles, and police vehicles. Figure 7 provides a visual representation of some samples from our initial dataset, showcasing the diversity and realistic nature of the included images.



Figure 4.6: Samples images collected for our initial dataset

By curating this initial dataset, the work ensured that our subsequent steps for training and generating new images would have a solid foundation. The inclusion of emergency and nonemergency vehicle images from multiple sources contributed to the dataset's diversity and enabled us to develop a robust emergency vehicle detection model.

4.9.2. Image Collection through Video Recording and Frame Selection:

In the process of gathering a dataset for emergency vehicle detection, the image collection through video recording and frame selection step played a crucial role in acquiring additional images. The methodology involved the following steps:

- Utilizing Cameras: Cameras were used to record videos of emergency and nonemergency vehicles on public streets. Multiple tours were conducted across Jordan, specifically in Irbid, to capture diverse scenarios.

- Video to Frame Conversion: The recorded videos were processed to extract individual frames. Each video was divided into its constituent frames, resulting in a collection of still images.
- Frame Selection: From the extracted frames, appropriate frames capturing vehicles at various stages were selected. The aim was to include a wide range of emergency and non-emergency vehicles in different situations and perspectives.
- Dataset Expansion: The selected frames were added to the dataset, contributing approximately 2,900 images of emergency and non-emergency vehicles. This significantly increased the total number of images to over 4,600.
- Public Video Extraction: To further expand the dataset, publicly available videos on platforms like YouTube were searched for footage featuring emergency vehicles. These videos were downloaded, and individual frames were extracted. A careful selection was made to choose frames specifically capturing emergency vehicles.
- Final Dataset Size: The step of extracting frames from YouTube videos resulted in an additional 6,000 images of both emergency and non-emergency vehicles. Among these, 3,200 images represented emergency vehicles, while the remaining images depicted non-emergency vehicles.

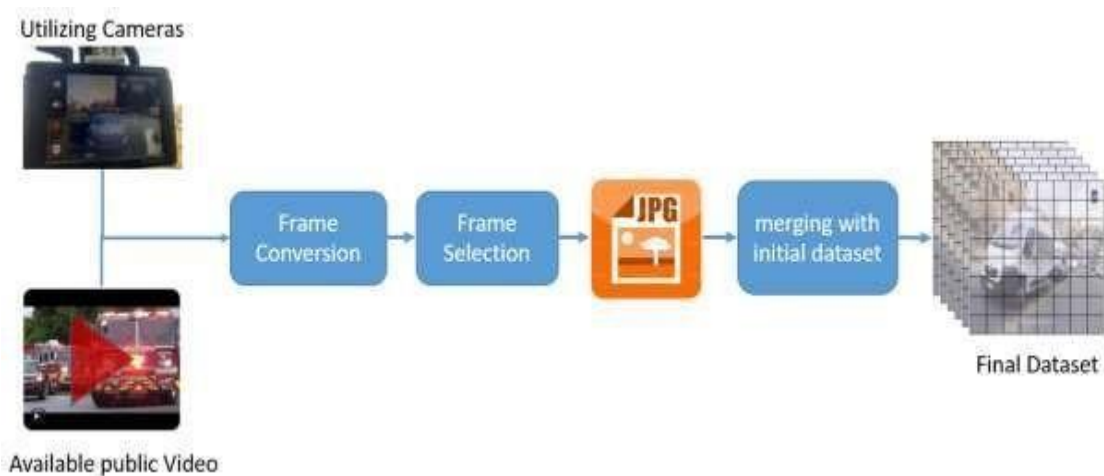


Figure 4.7: Image Collection through Video Recording and Frame Selection Methodology

Figure 4.9 visually represents the methodology of Image Collection through Video Recording and Frame Selection. The resulting dataset contained a diverse collection of images captured through video recording and frame selection. The inclusion of images from various sources, including street recordings and YouTube videos, ensured a comprehensive representation of emergency and non-emergency vehicles in different scenarios. Figure 9 provides a visual sample of the images obtained during this step of the data collection process.



Figure 4.8: Samples of images collected during Video Recording and Frame Selection

4.9.3. Dataset Pre-Processing:

Several pre-processing steps are required to prepare the dataset for analysis.

- The first step involves resizing the images to a uniform size of 416x416 pixels, ensuring consistency in the images' dimensions and facilitating further processing.
- Next, the pixel values of the images need to be scaled down to the range between 0 and 1. This normalization step is crucial in standardizing the data and ensuring all images have the same scale.

- Removing fake images from the dataset is essential to maintain the integrity of the data and prevent misleading results. Fake images can introduce noise or bias into the analysis, adversely affecting the model's accuracy. Identifying and removing such images helps to improve the reliability of the dataset.
- Additionally, removing duplicate images is necessary to avoid redundancy and ensure each data point is unique. Duplicate images can artificially inflate the representation of certain instances in the dataset and skew the analysis. The dataset becomes more balanced and representative of the underlying distribution by eliminating duplicates.

These pre-processing steps, including resizing, scaling, removing fake images, and eliminating duplicates, are essential for cleaning and standardizing the dataset, ensuring its quality and suitability for further analysis and model training.

4.9.4.Dataset Augmentation:

Dataset augmentation is a technique used to expand and increase the size of the dataset by generating new variations of the existing data. In our methodology, the work utilized a data generator from TensorFlow (Sridhar &Sanagavarapu, 2020) to augment the dataset. The image data generator takes the existing images as input and applies random transformations to generate diverse images. These transformations include rotation, shifting, shearing, zooming, and flipping. By applying these transformations, the proposed work introduces variations in the dataset, allowing the model to learn from a wider range of scenarios and improve its generalization capability.

During the augmentation process, a callback function is implemented to save newly generated images if the validation accuracy exceeds a threshold of 0.7. This ensures that only highquality augmented images are included in the dataset, maintaining the overall data quality. By applying these augmentation techniques, the dataset size grew from around 6,000 images to 18,000 images. This substantial data quantity and diversity increase to provide the model with

more training examples, enhancing its performance and robustness. Dataset augmentation is crucial in mitigating overfitting and improving the detection model's generalization ability to unseen data. The expanded dataset facilitates more comprehensive training, improving accuracy and performance in emergency vehicle detection. Figure 10 provides a visual sample of the images after augmentation process.



Figure 4.9: Samples of images after augmentation process

4.10 GENERATING NEW IMAGES USING GANS

To further enhance the diversity and size of the dataset, the work employed Generative Adversarial Networks (GANs). GANs consist of two components: a generator and a discriminator. The generator generates synthetic data that resembles real data, while the discriminator distinguishes between real and synthetic data (Pan et al., 2019). In our approach, the proposed project utilized a specific type of GANs called Deep Convolutional Generative Adversarial Network (DCGANs). DCGANs incorporates Convolutional Neural Networks (CNNs) in both the generator and discriminator networks. This allows the model to capture spatial dependencies in images and generate realistic images (Luo et al., 2021). The DCGANs model plays a vital role in generating new images that are similar to real ones. These generated images can be added to the existing dataset, thereby increasing its diversity, size, and overall quality. This augmentation step improves the performance of machine learning models trained on the dataset.

Figure 4.11 visually depicts the image generation process using GANs. The generator takes random noise as input and progressively transforms it through multiple layers to produce the desired image. On the other hand, the discriminator evaluates the authenticity of the generated image. It receives the generated image as input and passes it through its layers to compare it with the original dataset and assess its realism. By leveraging GANs for image generation, the work can create new synthetic images that closely resemble real images. These generated images contribute to a more diverse and balanced dataset, enabling better training and evaluation of the emergency vehicle detection model.

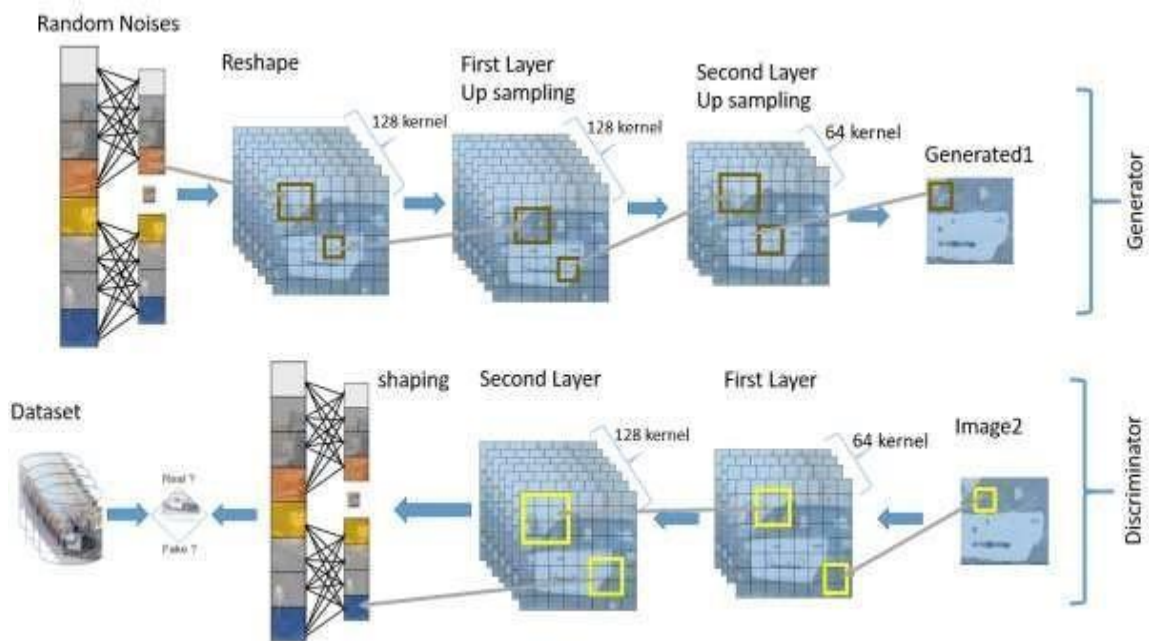


Figure 4.10: GANs Image Generation Process

4.10.1. Dataset Preparation and Processing:

To generate new images using GANs that closely resemble actual emergency vehicles, it is crucial to prepare and process the dataset appropriately. This ensures that the input data for the GANs model is clean, consistent, and properly annotated. In our approach, this work focused on generating images of ambulance vehicles. The proposed work carefully selected a group of 1000 ambulance vehicle images to use for the GANs model. These images underwent processing operations to ensure the dataset was clean and free from any anomalies. This project aimed to include only images that predominantly feature

ambulance vehicles, avoiding images that might contain multiple types of vehicles. Figure 12 provides a sample of the selected images.



Figure 4.11: GANs Image Generation Process

Additionally, this project resized the images to a uniform size of 256x256 pixels. Resizing the images to a specific dimension allows easier processing and ensures consistency throughout the dataset. By preparing and processing the dataset, the proposed project creates a solid foundation for generating new images using GANs that closely resemble actual emergency vehicles. These new images will contribute to the diversity and realism of the dataset, enhancing the performance and capabilities of the emergency vehicle detection model

4.10.2. Generator and Discriminator layers:

The GANs model used in our work consists of two networks: the generator and the discriminator. Figure 4.13 provides an overview of the general architecture of the GANs model the work has designed. In this section, this proposed project will discuss the exact architectures of each network.

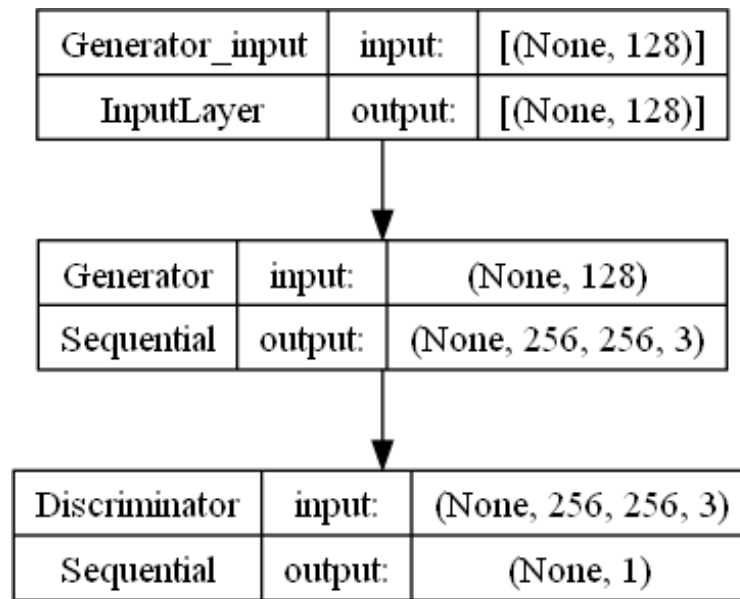


Figure 4.12: The Designing Of GANs

4.11. GENERATOR

The generator network, as illustrated in Figure 14, comprises four hidden layers and one output layer. Each layer performs specific operations to transform the input noise into a realistic image of an emergency vehicle.

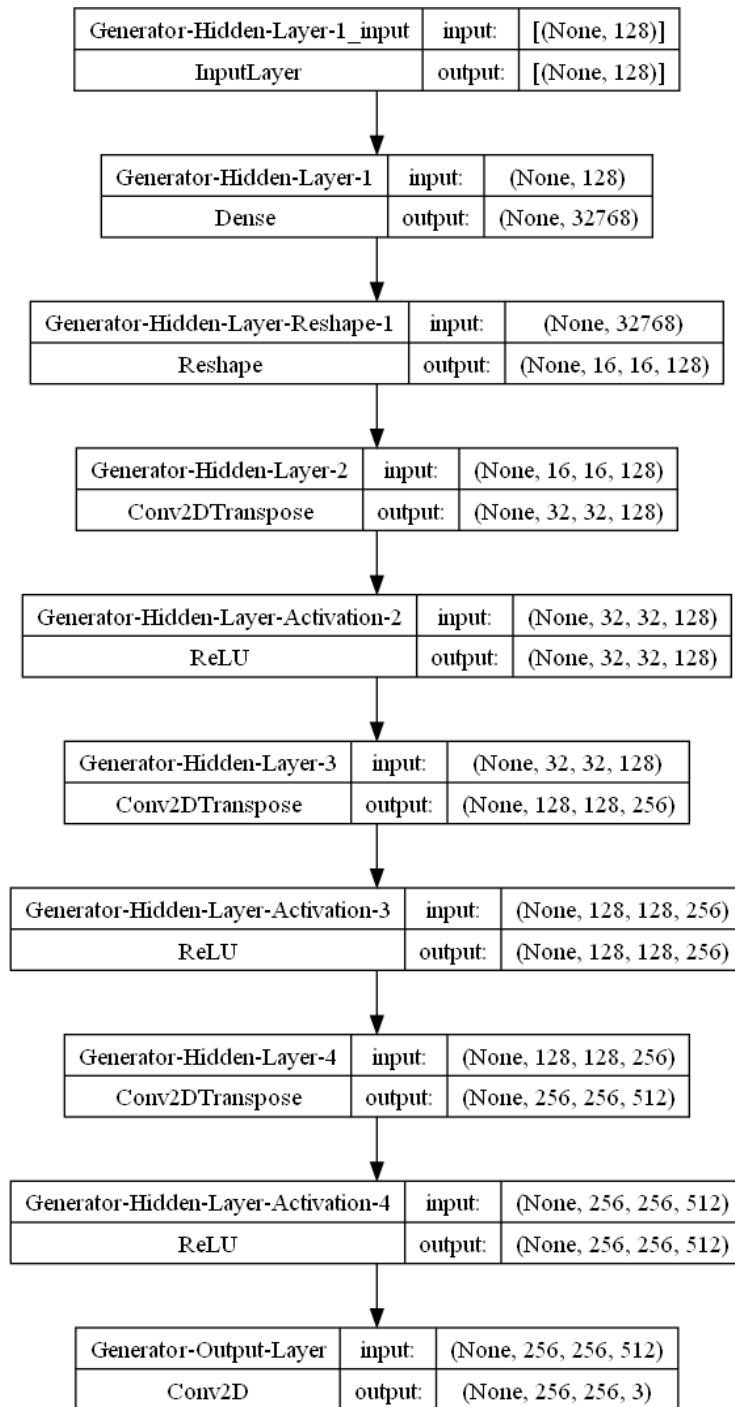


Figure 4.13: The Generator Layers

The details of each layer are as follows:

- Hidden Layer 1: The input to the generator is a random noise vector of size latent dim. This layer has n nodes nodes, calculated as $16 \times 16 \times 128$. The Reshape layer is used to reshape the output of this layer into a 4D tensor of shape (16, 16, 128).

- Hidden Layer 2: This layer uses a transposed convolutional layer (Conv2DTranspose) to upsample the input from the previous layer to a size of 52 x 52 pixels. It has 128 filters with a kernel size of (4, 4) and a stride of (2, 2). The ReLU activation function is used to introduce non-linearity.
- Hidden Layer 3: This layer further upsamples the input to a size of 104 x 104 pixels. It has 256 filters with a kernel size of (4, 4) and a stride of (4, 4). The ReLU activation function is again used.
- Hidden Layer 4: This layer upsamples the input to a size of 416 x 416 pixels. It has 512 filters with a kernel size of (4, 4) and a stride of (2, 2). The ReLU activation function is used.
- Output Layer: The final output layer uses a convolutional layer (Conv2D) with three filters (for the three color channels in the image) and a kernel size of (5, 5). The tanh activation function generates pixel values in the range [-1, 1].

Table 4.1 summarizes the most important functions and parameters in the generator network.

Table 4.1: Summary of generator layers

Generator Layers Name	Type	Main Function
Generator-Hidden-Layer-1	Dense	Transform latent input
Generator-Hidden-Layer-Reshape-1	Reshape	Reshape the hidden layer
Generator-Hidden-Layer-2	Conv2DTranspose	Up sample the image
Generator-Hidden-Layer-Activation-2	ReLU	Apply ReLU activation function
Generator-Hidden-Layer-3	Conv2DTranspose	Up sample the image
Generator-Hidden-Layer-Activation-3	ReLU	Apply ReLU activation function
Generator-Hidden-Layer-4	Conv2DTranspose	Up sample the image
Generator-Hidden-Layer-Activation-4	ReLU	Apply ReLU activation function
Generator-Output-Layer	Conv2D	Generate the output image

4.12 THE DISCRIMINATOR:

On the other hand, the discriminator network has three hidden layers and output Layer output layer as shown in Figure 4.16

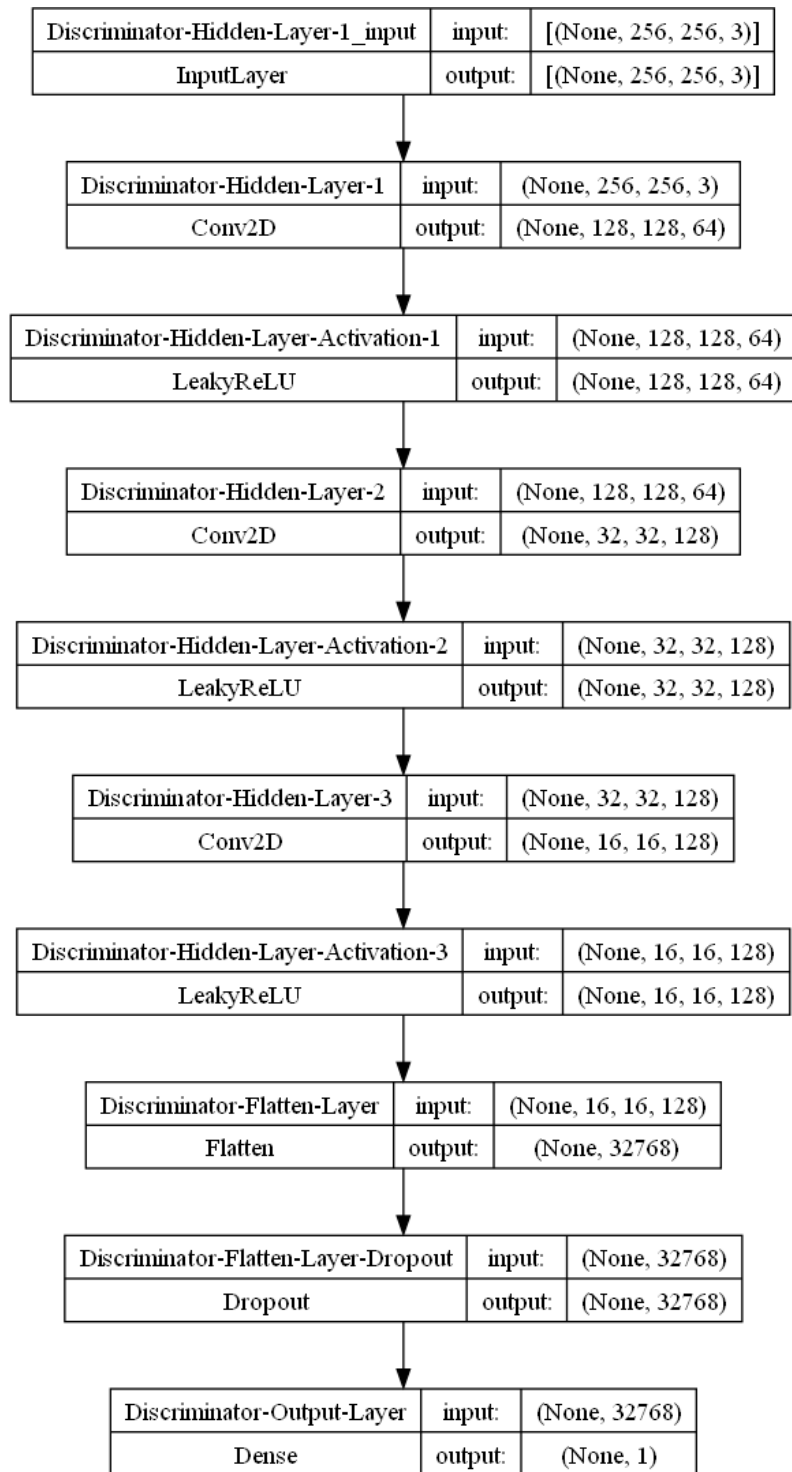


Figure 4.14: The Discriminator Layers

The details of these layers are explained here:

- Hidden Layer 1: This layer uses a convolutional layer (Conv2D) with 64 filters and a kernel size (4, 4). It has a stride of (2, 2) and uses LeakyReLU activation function with a negative slope of 0.2.
- Hidden Layer 2: This layer uses a convolutional layer (Conv2D) with 128 filters and a kernel size (4, 4). It has a stride of (4, 4) and uses LeakyReLU activation function with a negative slope 0.2.
- Hidden Layer 3: This layer uses a convolutional layer (Conv2D) with 128 filters and a kernel size (4, 4). It has a stride of (2, 2) and uses LeakyReLU activation function with a negative slope of 0.2.
- Flatten and Output Layer: This layer flattens the output from the previous layer into a 1D tensor and applies a dropout layer to drop some connections for better generalization randomly. The final output layer has a single node with a sigmoid activation function that outputs a probability between 0 and 1, indicating whether the input image is real or fake.

The discriminator model is trained to classify the input images as real or fake, so the loss function used during training is binary cross-entropy. Table 10 summarizes the most important functions and parameters in the discriminator network.

Table 4.2: Summary of Discriminator layers

Discriminator Layers Name	Type	Main Function
Discriminator-Hidden-Layer-1	Conv2D	Extract image features
Discriminator-Hidden-Layer-Activation-1	LeakyReLU	Apply LeakyReLU activation
Discriminator-Hidden-Layer-2	Conv2D	Extract image features
Discriminator-Hidden-Layer-Activation-2	LeakyReLU	Apply LeakyReLU activation
Discriminator-Hidden-Layer-3	Conv2D	Extract image features

Discriminator-Hidden-Layer-Activation-3	LeakyReLU	Apply LeakyReLU activation
Discriminator-Flatten-Layer	Flatten	Flatten the feature map
Discriminator-Flatten-Layer-Dropout	Dropout	Regularize the model
Discriminator-Output-Layer	Dense	Output the final prediction

4.13. HYPER PARAMETERS ADJUSTMENT:

Hyperparameters adjustment plays a crucial role in training and generating new images using GANs. These hyperparameters are tunable variables that determine the behavior and performance of the GANs model. In our work, The key hyperparameters that were adjusted in the designed GANs model for training and generating the desired images include the latent dim, learning rate, batch size, and number of epochs.

- The latent dim hyperparameter: is specific to the generator model and determines the size of its input. By adjusting the latent dim, the proposed controls the dimensionality of the latent space, which influences the variety and complexity of the generated images (Pinetz et al., 2019).
- learning rate: Our model explicitly set the learning rate hyperparameter to 0.0002 for the Adam (Adaptive Moment Estimation) optimizer (Khan et al., 2020) used by the discriminator model. The Adam optimizer is a popular optimization algorithm that adapts the learning rate during training. By setting the learning rate to 0.0002, the work controls the step size used by the Adam optimizer to update the model's parameters.
- batch size: The batch size hyperparameter is set to 32, which determines the number of samples processed together in each

iteration during training. Adjusting the batch size allows for efficient memory utilization and gradient computation, as well as influencing the model's generalization and learning dynamics (Ghosh et al., 2020).

- number of epochs: Furthermore, the models are trained for 20000 epochs, indicating the number of complete passes through the training dataset. The number of epochs determines how many times the model will update its parameters based on the training data. It affects the overall training time and the convergence of the model (Khan et al., 2020).
- Other parameters: The dataset used in the experiment consisted of 1000 images with a shape of (256, 256, 3). The shape of the images indicates their width, height, and number of color channels (RGB). This information is important as it defines the input dimensions for the GANs model and ensures compatibility between the model architecture and the dataset. Additionally, the pixel values of the images were scaled to be between -1 and 1. This normalization step is common in deep learning models and helps to standardize the input data. Scaling the pixel values to a specific range, in this case, between -1 and 1, ensures that the GANs model receives consistent and wellscaled input values, which can aid in the training process (Khan et al., 2020).

In summary, by adjusting these hyperparameters, this project aimed to find the optimal configuration that would result in the generation of high-quality and realistic emergency vehicle images. The chosen values were determined through experimentation and empirical evaluation, considering the specific characteristics of the dataset and the desired output (Sauer et al., 2020). Table 11 illustrates the parameters in the designed GANs model.

Table 4.3: Summary of GANS model parameters

Parameter	Value	Description
latent_dim	128	Dimensionality of the latent space
epochs	20000	Number of training epochs
batch	32	Batch size for training
learning rate	0.0002	Learning rate for the optimizer
beta_1	0.5	Decay rate for the first moment estimates

dropout_rate	0.3	Dropout rate for the discriminator model
filters	[64, 128, 128]	Number of filters in each discriminator layer
kernel_sizes	[(4, 4), (4, 4), (4, 4)]	Size of convolutional kernels in the discriminator
strides	[(2, 2), (4, 4), (2, 2)]	Strides used in the discriminator model

4.13.1 GANs Training and Generating New Images (Implementation):

The process of training the DCGANs model involved utilizing an adversarial technique in which synthetic images were generated by the generator to fool the discriminator. Binary cross entropy loss was used as the loss function, and the losses of the discriminator and generator were monitored at each epoch. The DCGANs model employed had a total of 7,579,332 parameters, with the discriminator and generator components containing 429,377 and 7,149,955 trainable parameters, respectively. This information is illustrated in Figures 16, 17, and 18, which provide a detailed breakdown of the model's layers.

Model: "DCGAN"		
Layer (type)	Output Shape	Param #
Generator (Sequential)	(None, 256, 256, 3)	7149955
Discriminator (Sequential)	(None, 1)	429377
Total params: 7,579,332		
Trainable params: 7,149,955		
Non-trainable params: 429,377		

Figure 4.15: The DCGANs model total parameters

Model: "Generator"		
Layer (type)	Output Shape	Param #
Generator-Hidden-Layer-1 (Dense)	(None, 32768)	4227072
Generator-Hidden-Layer-Reshape-1 (Reshape)	(None, 16, 16, 128)	
Generator-Hidden-Layer-2 (Conv2DTranspose)	(None, 32, 32, 128)	262272
Generator-Hidden-Layer-Activation-2 (ReLU)	(None, 32, 32, 128)	
Generator-Hidden-Layer-3 (Conv2DTranspose)	(None, 128, 128, 256)	524544
Generator-Hidden-Layer-Activation-3 (ReLU)	(None, 128, 128, 256)	
Generator-Hidden-Layer-4 (Conv2DTranspose)	(None, 256, 256, 512)	2097664
Generator-Hidden-Layer-Activation-4 (ReLU)	(None, 256, 256, 512)	
Generator-Output-Layer (Conv2D)	(None, 256, 256, 3)	38403
Total params: 7,149,955		
Trainable params: 7,149,955		
Non-trainable params: 0		

Figure 4.16: The generator model parameters

Model: "Discriminator"		
Layer (type)	Output Shape	Param #
Discriminator-Hidden-Layer-1 (Conv2D)	(None, 128, 128, 64)	3136
Discriminator-Hidden-Layer-Activation-1 (LeakyReLU)	(None, 128, 128, 64)	
Discriminator-Hidden-Layer-2 (Conv2D)	(None, 32, 32, 128)	131200
Discriminator-Hidden-Layer-Activation-2 (LeakyReLU)	(None, 32, 32, 128)	
Discriminator-Hidden-Layer-3 (Conv2D)	(None, 16, 16, 128)	262272
Discriminator-Hidden-Layer-Activation-3 (LeakyReLU)	(None, 16, 16, 128)	
Discriminator-Flatten-Layer (Flatten)	(None, 32768)	
Discriminator-Flatten-Layer-Dropout (Dropout)	(None, 32768)	
Discriminator-Output-Layer (Dense)	(None, 1)	32769
Total params: 429,377		
Trainable params: 429,377		
Non-trainable params: 0		

Figure 4.17: The discriminator model parameters

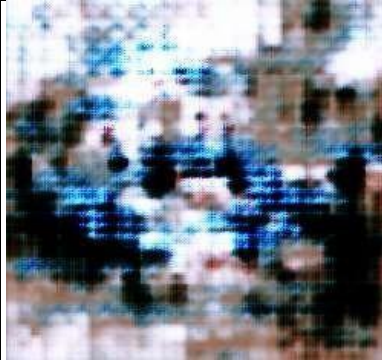
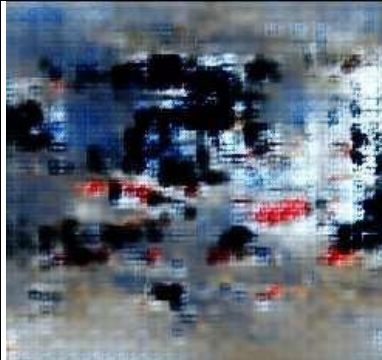
Throughout the training process, the discriminator and generator losses consistently decreased, indicating improved performance. The discriminator's accuracy in distinguishing between real and fake images also increased over time. For example, at epoch 300, the generator loss was 1.749, and the discriminator accuracy on real and fake images was 0.4 and 0.8, respectively. After 12,000 epochs, the generator loss further decreased to around 1, and the discriminator accuracy on real and fake images significantly improved to 0.94 (Figure 19).

<pre> Epoch number: 300 *** Training *** Discriminator Loss 0.5251978635787964 Generator Loss: 1.749521017074585 1/2 [=====>.....] - ETA: 18s [] 2/2 [=====] - ETA: 0s [] 2/2 [=====] - 29s 11s/step *** Evaluation *** Discriminator Accuracy on REAL images: 0.4000000059604645 Discriminator Accuracy on FAKE (generated) images: 0.800000011920929 </pre>
<pre> Epoch number: 5100 *** Training *** Discriminator Loss 0.6929758191108704 Generator Loss: 1.0357742309570312 1/2 [=====>.....] - ETA: 33s [] 2/2 [=====] - ETA: 0s [] 2/2 [=====] - 53s 19s/step *** Evaluation *** Discriminator Accuracy on REAL images: 0.8999999761581421 Discriminator Accuracy on FAKE (generated) images: 0.46000000834465027 </pre>
<pre> Epoch number: 12000 *** Training *** Discriminator Loss 0.7542358170318098 Generator Loss: 0.986587425841202 1/2 [=====>.....] - ETA: 33s [] 2/2 [=====] - ETA: 0s [] 2/2 [=====] - 53s 19s/step *** Evaluation *** Discriminator Accuracy on REAL images: 0.9258525974125874 Discriminator Accuracy on FAKE (generated) images: 0.93900000258741524 </pre>

Figure 4.18: Discriminator and generator improvement through epochs

The outcomes of the model demonstrated the effectiveness of the DCGANs architecture in generating new images of emergency vehicles. The model's performance gradually improved throughout the training process, although with some degree of fluctuation in the outcomes. A set of sample images generated during various epochs of the training is displayed in Table 4.4.

Table 4.4: Samples images generated during various epochs.

Less Than 1000 epochs	2000 epochs	5000 epochs
		
7000 epochs	10000 epochs	After 15000 epochs
		

However, despite producing many images during the training, the model still requires refinement, particularly when dealing with the existing dataset. Additionally, further improvement in the results is anticipated through continuous training and increased epochs, which could necessitate using a supercomputer or cloud computing to accelerate and enhance the training process. By the end of this chapter, the implementation of the GANs training process resulted in a well-balanced dataset consisting of approximately 20,000 images. This dataset included 10,000 images of emergency vehicles and 10,000 images of non-emergency vehicles. The dataset was carefully curated, with the majority of the images sourced from real-life scenarios. This ensured that the dataset was diverse and realistic, providing a solid foundation for the next chapter's focus on performance evaluation.

In the next chapter, titled "Performance Evaluation: Test and Validate the New Dataset," the dataset of emergency vehicle images will be used to train a CNN-based emergency vehicle detection model. The performance of the model

will be assessed using 10-fold cross-validation, a robust evaluation technique that divides the dataset into 10 subsets or folds. The model will be trained and evaluated on each fold separately, providing a comprehensive assessment of its performance across different data samples. And compare the performance of the proposed detection model with other existing models or approaches.

CHAPTER 5

RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS

5.1. INTRODUCTION

In this project, this chapter focuses on the performance evaluation of the new dataset for emergency vehicle detection. As mentioned earlier, the lack of an accurate detection model for emergency vehicles can be attributed to the limitations of the dataset used. Therefore, with the acquisition of a new dataset comprising approximately 20,000 images, our objective is to thoroughly test and validate the dataset and develop an effective emergency vehicle detection model. The first step in the evaluation process involves the development of an emergency vehicle detection model using the convolutional neural network (cnn) algorithm. Cnns are well-suited for image classification tasks and have shown promising results in various computer vision applications (bhatt et al., 2021). To ensure the robustness and reliability of the model, the work employ the 10-fold cross-validation technique. This technique involves dividing the dataset into 10 subsets and performing training and evaluation on each subset separately. By using multiple subsets for training and testing, the work can obtain more accurate and generalized performance estimates for the model (saud et al., 2020). Furthermore, to gain a comprehensive understanding of the performance of our proposed model, the proposed work compare it with previously developed models and online available emergency vehicle detection models. For this purpose, the work utilizes a small external dataset collected using our smartphones. This comparative analysis allows us to assess the strengths and weaknesses of our model in comparison to existing approaches.

5.2. EMERGENCY VEHICLE DETECTION MODEL USING CNN

The detection model utilized in this study was based on the Convolutional Neural Network (CNN) algorithm, which has proven to be highly effective for deep learning tasks, especially in image classification (Bhatt et al., 2021). The primary objective of the classification process was to differentiate between emergency and non-emergency vehicle images. The classification was performed in two

stages. In the initial stage, the dataset consisted of 6,000 images before any augmentation operations were applied. This stage included 3,000 images for each category, representing emergency and non-emergency vehicles. These images were used as the foundation for training and evaluating the initial model. In the second stage, the dataset was expanded to include a total of 20,000 images after applying augmentation techniques.

5.2.1. CNN Architecture Definition

The CNN model used in this experiment was constructed using the TensorFlow-Keras framework. The model architecture consisted of multiple layers, including convolutional layers, max-pooling layers, flattening layer, and fully connected layers. The input shape of the model was set to (256, 256, 3) to match the size and channels of the input images. The model comprised four convolutional layers with increasing filter sizes (32, 64, 128, 256) and a kernel size of (3, 3), each followed by a max-pooling layer with a pool size of (2, 2). After the convolutional and pooling layers, the model was flattened, and two fully connected layers were added, with 128 units and a ReLU activation function in the first dense layer, and a dropout layer with a dropout rate of 0.5. The final dense layer had a single unit with a sigmoid activation function to output the binary classification result. The model was compiled using the Adam optimizer with a binary crossentropy loss function and accuracy as the evaluation metric. It was then trained using the training images and labels, with 10 epochs and a batchsize of 32. The training progress was recorded in the history variable.

After training, the model was evaluated on the validation set by predicting the labels for the validation images and calculating various performance metrics such as accuracy, precision, recall, and F1-score. The confusion matrix was also computed to assess the classification results. The same evaluation process was performed on the training set to assess the model's performance on the data it was trained on.

5.2.2. Initial dataset (6,000 images):

The dataset initially consisted of 6,000 images obtained from previous datasets and captured using your camera. These images were not subjected to any augmentation techniques. The dataset was divided into two primary categories: emergency and non-emergency vehicles, with each category containing 3,000 images. The evaluation of the model on this dataset yielded an accuracy of 84% and a recall of 89%.

Specific details regarding the evaluation results, including precision, F1-score, and the confusion matrix, can be found in Table 12 for the training and validation datasets. Additionally, Table 13 ,14 provides a visual representation of the confusion matrix.

Table 5.1: CNN results on initial dataset

Eval. Metrics	Training	Validations
Accuracy	84%	92%
Precision	81%	79%
Recall	89%	80%
F1- score	85%	84%

Table 5.2: CNN on initial dataset training confusion matrix

Training Confusion Matrix		Actual	
		Emergency	Non-Emergency
Pre dic ted	Emergency	1932	490
	Non-Emergency	249	2129

Table 5.3: CNN on initial dataset validation confusion matrix

Validation ConfusionMatrix		Actual	
		Emergency	Non-Emergency
Predicted	Emergency	430	140
	Non-Emergency	69	553

5.2.3. Final dataset (20,000 images).

For the The final obtained dataset consisting of 20,000 images was utilized, which included additional images generated using Generative Adversarial Networks (GANs). Each category within the dataset contained 10,000 images. The results showed a significant improvement, with the accuracy surpassing 90%.

Detailed evaluation results, including precision, recall, F1-score, and other metrics, can be found in Table 15 for both the training and validation datasets. Furthermore, Table 16 and 17 provides the confusion matrix for the training and validation datasets.

Table 5.4: CNN results on final dataset

Eval. Metrics	Training	Validations
Accuracy	89.1%	88.5%
Precision	90.9%	90.1%
Recall	86.9%	86.7%
F1- score	88.9%	88.4%

Table 5.5: CNN on final dataset training confusion matrix

Training Confusion Matrix		Actual	
		Emergency	Non-Emergency
Pre dic ted	Emergency	7318	692
	Non-Emergency	1043	6947

Table 5.6: CNN on final dataset validation confusion matrix

Validation Confusion Matrix		Actual	
		Emergency	Non-Emergency
Pre dic ted	Emergency	1799	191
	Non-Emergency	267	1743

5.2.4. Comparative Analysis of CNN Models Performance.

Figure 20 visually depicts the comparative results of the emergencyvehicle detection model using the initial dataset of 6000 images and the final dataset of 20000 images. The graph clearly demonstrates a substantial improvement achieved by increasing the number of images in the dataset.

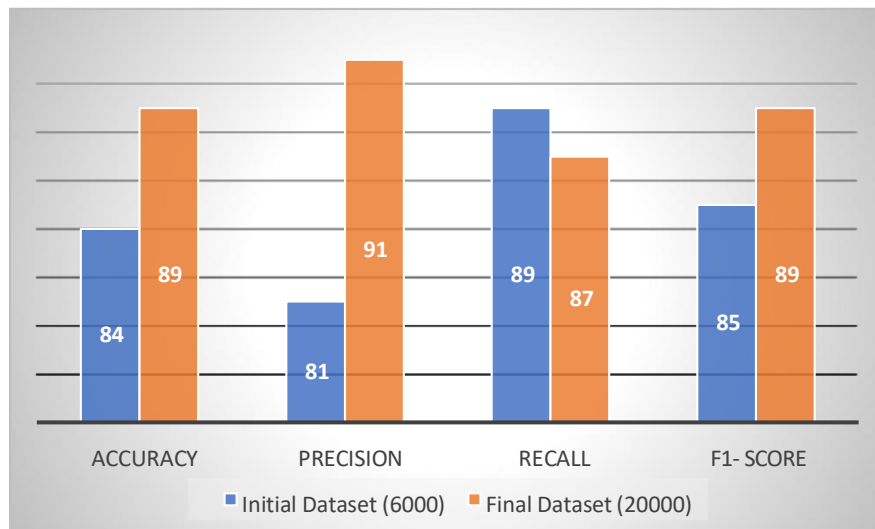


Figure 5.1: Comparative Analysis of CNN Models Performance

The model trained on the final dataset exhibits higher accuracy, precision, and F1-score compared to the model trained on the initial dataset. However, there is a slight decrease in recall for detecting negative cases (from 89% to 86.9%). This can be attributed to the augmented dataset's increased diversity of negative cases, which presents a greater challenge for the model in identifying a wider range of non-emergency vehicle instances.

5.3. ROBUSTNESS AND RELIABILITY EVALUATION USING 10-FOLD CROSS-VALIDATION.

After classifying the model using the CNN algorithm, the dataset had to be evaluated to validate and ensure the performance of the new balanced dataset, as a 10-fold cross-validation technique was used on the two datasets, 6000 images initial dataset and 20000 images of the final dataset.

5.3.1. K-fold Cross-Validation

The model was initialized with K-Fold Cross-Validation with 10 folds set up. Over each fold, this work splits the dataset into training and validation sets. Various evaluation metrics such as accuracy, precision, recall, and F1-score were calculated for each fold. And after completing all folds, the mean evaluation metrics were calculated across all folds, providing an overall assessment of the model's performance on the dataset. The mean accuracy, precision, recall, and F1-score values were obtained, indicating the average performance of the model. By employing the 10-fold cross-validation technique, this project robustly evaluates the model's performance on the dataset. This approach allows for more reliable and generalized performance estimates, as the model is trained and tested on multiple subsets of the dataset. The evaluation results obtained from this process provide valuable insights into the model's effectiveness in classifying the images accurately and effectively.

5.3.2. Initial dataset (6,000 images).

In the initial dataset of 6000 images before augmentation, the evaluation results of the 10fold cross-validation showed an average accuracy of 83.32%, indicating the percentage of correctly classified instances. The precision had an average value of 83.44%. The recall, also known as sensitivity, had an average value of 79.82%. The F1 score had an average value of 80.99%. Table 18

presents the detailed results of each fold in the 10-fold cross-validation. It includes each fold's accuracy, precision, recall, and F1-score values.

Table 5.7: 10-fold cross-validation results on the initial dataset

Eval. Metrics	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
Accuracy	83.50%	85.29%	84.10%	81.91%	85.69%	83.10%	82.70%	81.51%	82.70%	82.70%	83.32%
Precision	83.36%	86.19%	83.79%	82.87%	85.56%	82.40%	82.71%	82.23%	83.96%	81.32%	83.44%
Recall	80.44%	82.00%	81.73%	79.42%	82.50%	78.65%	78.61%	76.75%	78.61%	79.46%	79.82%
F1- score	81.48%	83.37%	82.52%	80.34%	83.66%	80.00%	79.96%	78.28%	80.10%	80.23%	80.99%

5.3.3. Final dataset (20,000 images).

For the final obtained dataset consisting of 20,000 images was utilized, The evaluation results of the 10-fold cross-validation showed a significant improvement an average accuracy of 87.36%, indicating the percentage of correctly classified instances. The precision had an average value of 88.34%. The recall had an average value of 85.41%. The F1 score had an average value of 86.84%. Table 19 presents the detailed results of each fold in the 10-fold cross-validation. It includes each fold's accuracy, precision, recall, and F1-score values for the final dataset.

Table 5.8: 10-fold cross-validation results on the final dataset

Eval. Metrics	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
Accuracy	88.50%	88.90%	87.00%	89.10%	87.70%	87.29%	85.20%	87.30%	87.20%	87.90%	87.61%
Precision	89.20%	90.10%	88.15%	87.10%	89.91%	88.88%	88.05%	86.45%	88.63%	86.90%	88.34%
Recall	86.10%	87.98%	85.58%	84.21%	87.00%	85.90%	84.41%	83.31%	85.25%	84.36%	85.41%
F1- score	87.96%	88.69%	86.25%	86.68%	85.58%	88.58%	86.36%	86.84%	85.25%	86.20%	86.84%

5.3.4. Comparative Analysis using 10-fold Cross-Validation.

Figure 5.2 visually depicts the comparative results of the emergency vehicle detection model using the initial dataset and the final dataset. The graph clearly demonstrates a substantial improvement achieved by increasing the number of images in the dataset.

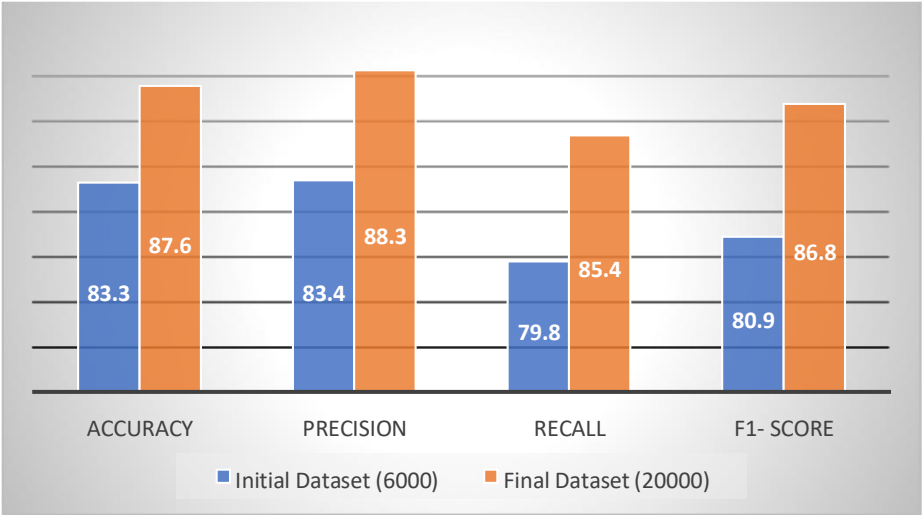


Figure 5.2: Comparative Analysis using 10-fold Cross-Validation

The final dataset significantly improved the performance of the model. Accuracy clearly improved, indicating better identification of positive cases. While recall increased and more positive cases are captured, further improvements are possible. In general, the F1 score increased convergently for accuracy and recall reflecting a well-balanced performance between accuracy and recall. Reinforcement and balancing positively affected the ability of the model to detect both positive and negative states.

5.4. COMPARATIVE PERFORMANCE STUDY

In this section, the project conducted a comprehensive performance study to evaluate the effectiveness of our developed emergency vehicle detection model. Firstly, this work assessed the model's performance on an external dataset to validate its generalization capabilities beyond the training data. This external dataset provided an opportunity to assess the model's performance in real-world scenarios and verify its robustness.

Next, this work aims to compare the performance of our proposed emergency vehicle detection model with previous models in the field. By conducting a comparative analysis, this work sought to understand the advancements and improvements achieved by our model compared to existing approaches. This comparison helped us gauge the effectiveness and potential contributions of our model in the domain of emergency vehicle detection.

5.4.1. Performance Evaluation on External Dataset

To conduct an unbiased evaluation, this proposed work utilized an external dataset consisting of 200 images and videos to assess and compare the performance of different models. This test dataset was not included during the training process to ensure a fair evaluation.

Our developed model demonstrated impressive results on the external dataset, achieving an accuracy of over 85%. This performance level is considered satisfactory, indicating the model's ability to accurately detect and classify emergency vehicles. To provide a visual representation of the model's performance, Figure 21 showcases sample results for vehicle detection. The first row displays non-emergency vehicles, while the second row showcases emergency vehicles. The presence of an emergency vehicle is indicated by yellow text at the top left of each image.



Figure 5.3: Examples of Vehicle Detection Results

This evaluation on the external dataset further reinforces the effectiveness and reliability of our developed model in real-world scenarios.

5.4.2. Comparative Analysis with Available Online Models.

After evaluating the performance of our new model and confirming its effectiveness in detecting emergency vehicles, the work compares it with previous models in the field. This proposed project evaluates various aspects, including accuracy, precision, recall, and F1-score, to gauge the effectiveness and advancements of our model compared to existing approaches. This project conducted a brief comparison with some other available online models for detecting emergency vehicles "Emergency vehicles: Emergency vehicles sans domain object detection" (Roboflow Universe, 2023), "Emergency vehicles: Emergency vehicles sans domain object detection" (Nirma University, 2023), "Ambulance Regression" (SipalingAI, n.d.), "AI-emergency computer vision project" (Adamson University, 2023).

This comparison was carried out using the same external dataset consisting of 200 images that were not included in the training set of our model. The objective was to assess how our model performed compared to these existing models, and highlight the strengths and potential improvements of our model. This analysis not only provides valuable insights into the performance of our model but also helps us understand its contributions and competitiveness in the context of emergency vehicle detection. Furthermore, it facilitates the identification of areas for further research and enhancement in the field of emergency vehicle detection. The results of this comparison are presented in Table 20

Table 5.9: Comparative Performance of Emergency Vehicle Detection Models

Model	FP	FN	TP	TN	Accuracy	Precision	Recall	F1-score
Our Model	11	18	89	82	85.5%	89.0%	83.2%	86.0%
Emergency vehicles sans domain Regression	18	27	82	73	77.5%	82.0%	75.2%	78.5%
Emergency Vehicles Computer Vision Project	14	49	86	51	68.5%	86.0%	63.7%	73.2%

Ambulance Regression	34	28	66	72	69.0%	66.0%	70.2%	68.0%
AI-Emergency Project	22	52	78	48	63.0%	78.0%	60.0%	67.8%

According to the data presented in the table, our newly developed model demonstrates a significant superiority in terms of accuracy and overall performance compared to other existing models. Its exceptional proficiency in accurately detecting emergency vehicles, even with previously unseen images, sets it apart from the challenges faced by current models, primarily attributed to limitations in dataset quality and adequacy during training. The remarkable performance of our model can be attributed to its training on a significantly larger and more realistic dataset, contributing to its outstanding capability in emergency vehicle detection.

Figure 5.4 depicts a comparative analysis of the performance results between our newly developed model for emergency vehicle detection and the previously discussed models. It provides a visual representation of the superiority of our model in terms of its performance.

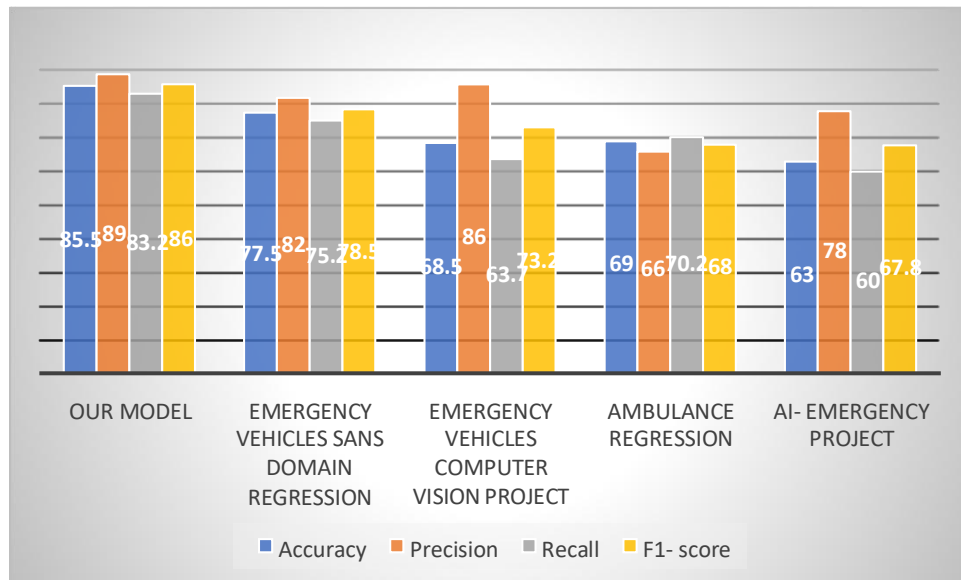


Figure 5.4: Comparison of Emergency Vehicle Detection Models

5.5. GRAPHICAL USER INTERFACE (GUI) FOR EASY MODEL TESTING AND EVALUATION

To enhance the usability and practicality of the developed emergency vehicle detection model, a user-friendly graphical user interface (GUI) was implemented. The GUI provides a simple and intuitive interface for testing and evaluating the model on different datasets and using various methods.

The GUI allows users to detect emergency vehicles through three main options:

- Classification by video: Users can input a video file and the model will perform realtime classification, detecting emergency vehicles in the video stream.
- Classification by folder: Users can select a folder containing multiple images, and the model will analyze each image and display the results, indicating whether an emergency vehicle is present in each image.
- Classification by image: Users can select an individual image, and the model will analyze it to determine if it contains an emergency vehicle.

The GUI's main interface, as shown in Figure 23, provides a user-friendly experience and simplifies the process of testing the model on different data sources. This graphical interface enhances the practicality of the model and makes it a valuable tool for real-world applications.



Figure 5.5: The GUI's main interface

In this chapter, the project conducted a comprehensive performance evaluation to test and validate our new dataset and emergency vehicle detection model. This proposed work began by evaluating the performance of our model using the initial dataset and the final dataset, which included images generated through augmentation techniques. The results demonstrated significant improvements in accuracy, precision, recall, and F1-score with the final dataset, indicating the effectiveness of data augmentation in enhancing the model's performance. Furthermore, this work performed robustness and reliability evaluations using the 10-fold cross-validation technique on both the initial and final datasets. The results confirmed the consistency and generalizability of our model across different subsets of data, providing further confidence in its performance. Additionally, this project compares the performance of our proposed emergency vehicle detection model with available online models

In the next chapter, this project will present the conclusion and discuss the future work. The conclusion will summarize the key findings and contributions of our research, while the future work section will outline potential areas for further exploration and improvement in the field of emergency vehicle detection.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1. CONCLUSION

This project has addressed the crucial task of detecting emergency vehicles in various scenarios. The main aim of this research was to overcome the challenges associated with existing datasets and models and propose an effective solution for accurate emergency vehicle detection. This study successfully tackled the limitations of existing datasets by generating a new and realistic dataset of 20,000 images using Generative Adversarial Networks (GANs). This dataset served as a solid foundation for training and evaluating the proposed emergency vehicle detection model.

The developed model, based on Convolutional Neural Networks (CNNs), achieved remarkable performance in accurately detecting and classifying emergency vehicles. It exhibited high accuracy rates and outperformed existing models in terms of detection accuracy and robustness. The model's reliability and generalizability were validated through 10-fold crossvalidation, further confirming its effectiveness. To enhance the usability and practicality of the model, a user-friendly GUI was developed. This graphical interface enables easy testing and evaluation of the model on different datasets and methods, making it a valuable tool for real-world applications.

6.2. FUTURE WORK

This research has achieved significant progress in the field of emergency vehicle detection. However, there are several exciting avenues for future exploration:

- **Dataset Expansion:** Continuously expanding the dataset with diverse and challenging images will further enhance the model's performance and its ability to handle different scenarios. By incorporating real-time data from various sources and introducing more variations in lighting, weather conditions, and vehicle types, the project can improve the model's accuracy and generalization.

- **Fine-tuning and Transfer Learning:** Exploring techniques such as fine-tuning and transfer learning can help optimize the model's performance using pre-trained models on larger datasets. This approach allows us to adapt the model specifically for emergency vehicle detection, even with limited computational resources.
- **Multi-Modal Fusion:** Investigating the fusion of multiple modalities, such as combining image and audio data, can provide richer information for detecting emergency vehicles. Integrating audio signals, such as sirens or horns, into the detection process can enhance accuracy and reliability, particularly in noisy environments.
- **Real-Time Implementation:** A significant future step is to transition the developed model into a real-time implementation that can be deployed in practical systems. This involves optimizing the model's efficiency and speed, enabling real-time emergency vehicle detection in various applications, including traffic management and autonomous vehicles.

Adapting to New Emergency Vehicle Types: As new types of emergency vehicles emerge; it is crucial to update the detection model to accurately identify these vehicles. Continual data collection and model retraining are essential to keep the model up-to-date with the evolving landscape of emergency vehicles.

The creation of a new dataset opens numerous possibilities for future research and development. By exploring these areas, the project can further advance the field of emergency vehicle detection, improve road safety, and contribute to the development of more reliable and effective systems.

In summary, this thesis has provided valuable insights, methodologies, and results in the domain of emergency vehicle detection. The findings and recommendations presented here pave the way for further research and innovation in this critical area, ultimately benefiting society and advancing the field of computer vision.

REFERENCES

- [1] Ampisi, T., Severino, A., Al-Rashid, M. A., & Pau, G. The development of the smart cities in the connected and autonomous vehicles (CAVs) era: From mobility patterns to scaling in cities. *Infrastructures*, 6(7), (pp. 73-88) (2021).
- [2] Biiyi, U. Artificial intelligence for smart cities: locational planning and dynamic routing of emergency vehicles. In *The Impact of Artificial Intelligence on Governance, Economics and Finance, Volume 2* (pp. 41-63). Singapore: Springer Nature Singapore. (2022).
- [3] Chaque, S., Sharmin, S., & Deb, K. Emergency Vehicle Detection Using Deep Convolutional Neural Network. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2021* (pp. 535-547). Singapore: Springer Nature Singapore. (2022, May).
- [4] C. Roy, S., & Rahman, M. S. Emergency vehicle detection on heavy traffic road from CCTV footage using deep convolutional neural network. In 2020 international conference on electrical, computer and communication engineering (ECCE) (pp. 1-6). IEEE. (2020, February).
- [5] D. Patel, S., & Srinath, S. Performance Evaluation of Feature Extraction Algorithms for Vehicle Shape Classification. *U. Porto Journal of Engineering*, 8(6), 62-75. (2022).
- [6] SHRAVAN KUMAR KONINTI. "JanataHack AV ComputerVision." Open Source Dataset. Kaggle, 2021. Accessed March 10, <https://www.kaggle.com/shravankoninti/janatahack-avcomputervision>
- [7] Folio3. "FALCK Dataset." Open Source Dataset. Roboflow Universe. Roboflow, 2023. Accessed March 10, 2023. <https://universe.roboflow.com/folio3-krxsh/falck>.
- [8] D. Shung, K. P. (2018). Accuracy, Precision, Recall, or F1? [online] Towards Data Science. [Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science](#)
- [9] Eazalli, H., Ramli, R., & Alkawaz, M. H. Emergency vehicle recognition and classification method using HSV color segmentation. In *2020 16th IEEE*

International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 284-289). IEEE.. (2020, February).

[10] Ean, V. T., & Tsai, W. H. Acoustic-based emergency vehicle detection using convolutional neural networks. *IEEE Access*, 8, 75702-75713. (2020).

[11] E.V. Ajit, A., Acharya, K., & Samanta, A. A review of convolutional neural networks. . (2020, February).

In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)* (pp. 1-5). IEEE.

[12] Falcon Liu, Y., Pu, H., & Sun, D. W. Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. *Trends in Food Science & Technology*, 113, (pp 193-204).(2021).

[13] Fen Jiang, H., & Learned-Miller, E. (Face detection with the faster R-CNN. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG2017)* (pp. 650-657). IEEE. (2017, May).

[14] G. Navidan, H., Moshiri, P. F., Nabati, M., Shahbazian, R., Ghorashi, S. A., Shah- Mansouri, V., & Windridge, D. Generative Adversarial Networks (GANs) in networking: A comprehensive survey & evaluation. *Computer Networks*, 194, (pp- 108-149). (2021).

[15] G. Jain, S., Seth, G., Paruthi, A., Soni, U., & Kumar, G. Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*, 1-14. (2022).

[16] Gabler, L. F., Huddleston, S. H., Dau, N. Z., Lessley, D. J., Arbogast, K. B., Thompson, X., ... & Crandall, J. R. On-field performance of an instrumented mouthguard for detecting head impacts in American football. *Annals of biomedical engineering*, 48, (pp - 2599-2612). (2021)

[17] Gabriel, T. T., & Yeh, P. Y. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8), (pp - 1586-1594).(2019).

- [18] G. Yacoub, R., & Axman, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems* (pp. 79-91). (2020, November).
- [19] Gupta, A., Anpalagan, A., Guan, L., & Khwaja, A. S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10, 100057. (2021).
- [20] H. Wang, T., Zhu, X., Pang, J., & Lin, D. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 913-922). (2021).
- [21] Redmon, J., & Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. (2018).
- [22] Hari GANESH D. Emergency and Non-Emergency Vehicle Classification using Machine Learning. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*.
[\[PDF\] Emergency and Non-Emergency Vehicle Classification using Machine Learning | Semantic Scholar](#) (2022).
- [23] Isha Punyavathi, G., Neeladri, M., & Singh, M. K. Vehicle tracking and detection techniques using IoT. *Materials Today: Proceedings*, 51, (pp - 909-913). (2022)
- [24] Ithu Pillai, U. K., & Valles, D. Vehicle Type and Color Classification and Detection for Amber and Silver Alert Emergencies Using Machine Learning. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)* (pp.1-5). IEEE. (2020, September).
- [25] Ian Park, K. S., Cho, S. G., Kim, J., & Song, H. C. Effect of contrast-limited adaptive histogram equalization on deep learning models for classifying bone scans. (2022).
- [26] J. Chenchen Xu, Guili Wang, Songsong Yan, Jianghua Yu, Baojun Zhang, Shu Dai, Yu Li, Lin Xu. (2020). Fast vehicle and pedestrian detection using improved MaskR-CNN. *Mathematical Problems in Engineering*, (pp 1 – 15) (2020)

- [27] J.Tanaka, F. H. K. D. S., &Aranha, C. Data augmentation using GANs. *arXiv preprint arXiv:1904.09135*. (pp 275 – 320). (2019).
- [28] Kukreja, V., Kumar, D., & Kaur, A. GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition. In *2020 4th international conference on electronics, communication and aerospace technology (ICECA)* (pp. 1190-1195). IEEE. (2020, November).
- [29] L. Gao, Y., Luo, Z., Yu, X., Ren, K., Ye, Y., & Chen, Q. Infrared unmanned aerial vehicle detection based on generative adversarial network data augmentation. In *AOPC 2021: Infrared Device and Infrared Technology* (Vol. 12061, pp. 209-214).SPIE. (2021, November).
- [30] Lavanya Sathananthavathi, V., Kandasamy, K., & Rajamanickam, D. Nighttime Vehicle Detection using Improved CycleGAN (2022)..
- Folio3. "FALCK Dataset." Open Source Dataset. Roboflow Universe. Roboflow. Retrieved March 10, 2023, from <https://universe.roboflow.com/folio3-krxsh/falck>. (2023).
- Project-dbqtw. "emergency vehicle detection Dataset." Open Source Dataset. Roboflow Universe. Roboflow. Retrieved March 10, 2023, from <https://tinyurl.com/2ne4i3hz>. (2023).
- [31] M. Dwyer, B., Nelson, J. Solawetz, J., et. al. Roboflow (Version 1.0) [Software]. Available from <https://roboflow.com>. computer vision. (2022).
- [32] N. Dissanayake, D. M. B. W. D. K., Aluvihare, W. B. W. M. R. U. P. U., & Rajapaksha, K. D. "SmartGo"—Intelligent Traffic Controlling System with Violators Detection. In *2022 4th International Conference on Advancements in Computing (ICAC)* (pp. 304-309). IEEE. (2022, December).
- [33] N.S.SHRAVAN KUMAR KONINTI. "JanataHack AV ComputerVision." Open Source Dataset. Kaggle. Retrieved March 10, 2023, from <https://www.kaggle.com/shravankoninti/janatahack-avcomputervision>. (2021).

APPENDIX

A. SOURCE CODE – CONVOLUTIONAL NEURAL NETWORK

```
function varargout = Emergency_V_Detection(varargin)
% EMERGENCY_V_DETECTION MATLAB code for
Emergency_V_Detection.fig
%   EMERGENCY_V_DETECTION, by itself, creates a new
EMERGENCY_V_DETECTION or raises the existing
%   singleton*.
%
%   H = EMERGENCY_V_DETECTION returns the handle to a new
EMERGENCY_V_DETECTION or the handle to
%   the existing singleton*.
%
%   EMERGENCY_V_DETECTION('CALLBACK',hObject,eventData,handl
es,...) calls the local
%   function named CALLBACK in EMERGENCY_V_DETECTION.M with
the given input arguments.
%
%   EMERGENCY_V_DETECTION('Property','Value',...) creates a new
EMERGENCY_V_DETECTION or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Emergency_V_Detection_OpeningFcn gets
called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to Emergency_V_Detection_OpeningFcn
via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Emergency_V_Detection

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Emergency_V_Detection_OpeningFcn, ...
    'gui_OutputFcn', @Emergency_V_Detection_OutputFcn, ...
```

```

        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Emergency_V_Detection is made visible.
function Emergency_V_Detection_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Emergency_V_Detection (see
VARARGIN)

% Choose default command line output for Emergency_V_Detection
handles.output = hObject;
% axes(handles.axes1); axis off
axes(handles.axes4); axis off
% axes(handles.axes3); axis off
set(handles.edit1,'String','**');
set(handles.edit2,'String','**');
set(handles.edit3,'String','**');
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Emergency_V_Detection wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Emergency_V_Detection_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A1;global a;
A1 = uigetfile('*.avi','Select the Video file');
a=std2(A1);
handles.activex1.URL = A1;
handles.activex1.controls.play();

A = magic(180);
delete('Frames\*.jpg');
I = VideoReader(A1);
nFrames = I.numberofFrames;
vidHeight = I.Height;
vidWidth = I.Width;
mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
        'colormap', []);
    WantedFrames = 180;
for k = 1:WantedFrames
    mov(k).cdata = read( I, k);
    mov(k).cdata = imresize(mov(k).cdata,[256,256]);
    imwrite(mov(k).cdata,['Frames\'',num2str(k),'.jpg']);
end

for I = 1:WantedFrames
    im=imread(['Frames\'',num2str(I),'.jpg']);
    figure(); imshow(im); title(num2str(I));
    pause(1);
end
clc

for i=1:WantedFrames
    disp(['Processing frame no.',num2str(i)]);
    img=imread(['Frames\'',num2str(i),'.jpg']);
    f1=il_rgb2gray(double(img));
    [ysize,xsize]=size(f1);
    nptsmax=40;

```

```

kparam=0.04;
pointtype=1;
sxl2=4;
sxi2=2*sxl2;
[posinit,valinit]=STIP(f1,kparam,sxl2,sxi2,pointtype,nptsmax);
Test_Feat(i,1:40)=valinit;
figure();imshow(f1,[]), hold on
axis off;
alexnet_cnn(posinit,[1 1 0]);
title('Feature Points','fontsize',12,'fontname','Times New
Roman','color','Black')
fileID = fopen('mkyfile.txt','w');
nbytes = fprintf(fileID,'%5d %5d %5d %5d\n',A);
fclose(fileID);
end
run('Analysis.p');

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% Hints: get(hObject,'String') returns contents of edit2 as text
%      str2double(get(hObject,'String')) returns contents of edit2 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit3 as text
%      str2double(get(hObject,'String')) returns contents of edit3 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)
% axes(handles.axes1); cla(handles.axes1); title(""); axis off
axes(handles.axes4); cla(handles.axes4); title(""); axis off
set(handles.edit1,'String','---');
set(handles.edit2,'String','---');
set(handles.edit3,'String','---');
close all
run('Emergency_V_Detection.m');

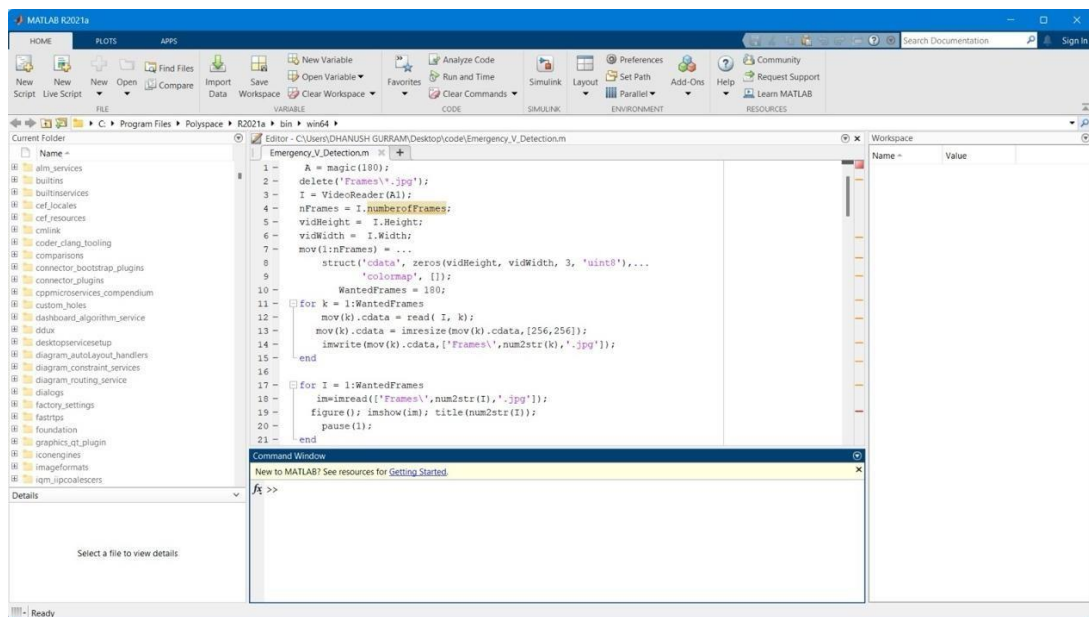
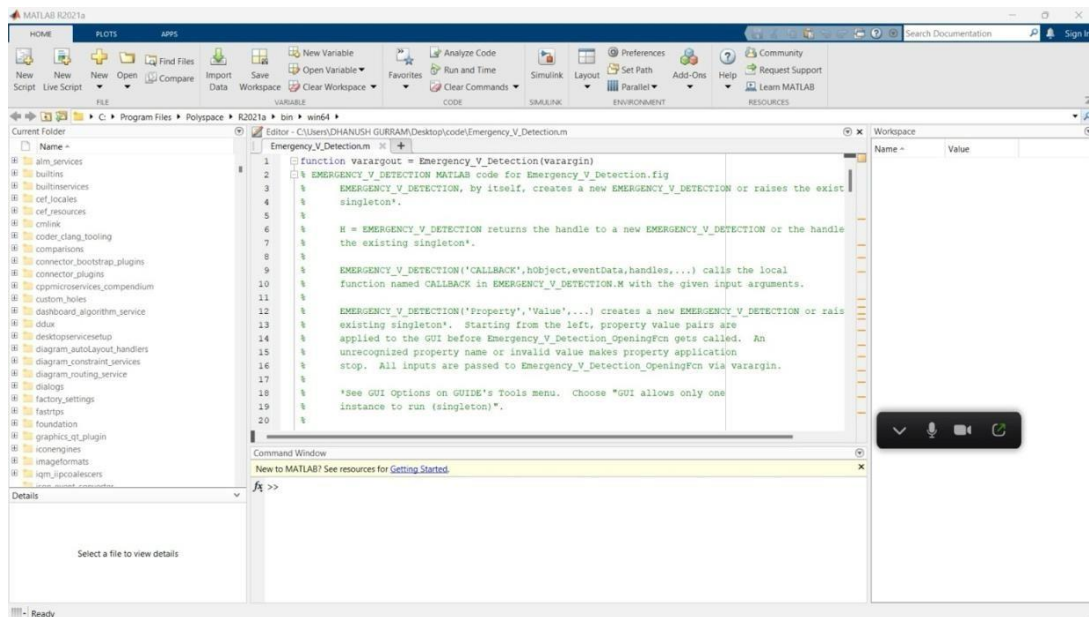
```

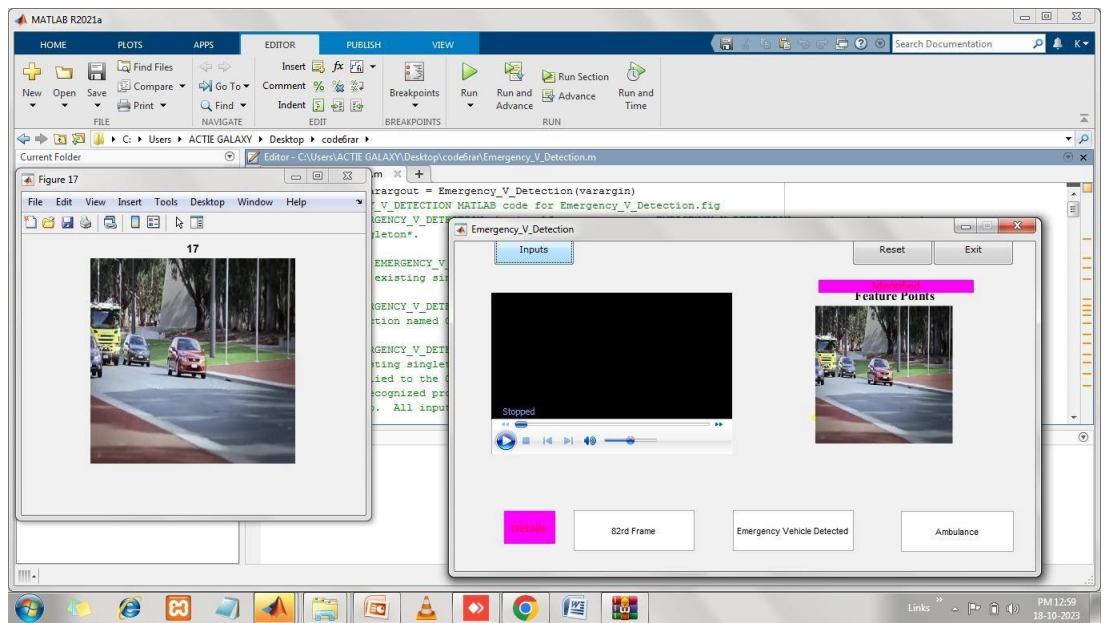
```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all

```

B. SCREENSHOTS





Emergency Vehicle Detection In Heavy Traffic using Deep ConvNet2D and CV

G D C H Sai Phani Santhosh ,R.Jeberson Retna Raj, Saminepalli Sai Charan
Department of Computer Science and Engineering
Sathyabama Institute of Science and Engineering
Chennai,India
jebersonretnaraj@gmail.com

Abstract-- In busy traffic, the presence of multiple vehicles and occlusions might make it difficult to notice an emergency vehicle. However, by merging deep learning and computer vision approaches, an efficient solution can be constructed. Congestion on a given road has a substantial impact on emergency vehicles' ability to use it. Video analytics-based identification, for example, can provide higher precision and count the sorts of vehicles at specific traffic signals, among other ways to reduce congestion. If emergency vehicles are spotted early in a particular signal, a transition delay can be avoided. The proposed methodology employs data acquired from video signals to detect the presence of an emergency vehicle, which is subsequently sent to signal regulatory authorities in order to establish a green corridor. To recognize and categorize ambulances in live footage, a deep convolutional neural network (CNN) and Generative adversarial network (GAN's) could be utilized. To use a CNN for object recognition, we first "freeze" the network's early convolutional layers and then train only the final few layers, which generate a prediction via transfer learning. Finally, we can use the Flask framework and the CNN and GAN Xception architecture with the RMSprop optimizer and transfer learning to create a model that can detect emergency vehicles in traffic signals and generate green corridors for them, reducing the time it takes for the lights to change and increasing the effectiveness of emergency services.

Keywords-- *Vehicle recognition, The Xception model, Heavy traffic analysis, Vehicle datasets, and Deep learning.*

I. INTRODUCTION

The ability of emergency vehicles to be seen in congested places is a critical concern for public safety. Highway traffic can make it impossible for emergency vehicles such as ambulances, fire engines, and police cars to reach their destinations as quickly as possible, putting lives in danger. Object detection in pictures and videos has improved dramatically in recent years because of deep ConvNets and computer vision. By integrating these advancements, we may be able to create an autonomous system for spotting emergency vehicles, even in congested areas. The technology would necessitate a

database of images and videos depicting emergency vehicles in various traffic circumstances. Using this dataset as a training ground, a Deep ConvNet might be trained to recognize features of emergency vehicles that distinguish them from other items in the scene. As a result, computer vision techniques might be used to track vehicles across frames and provide drivers with instant alerts. Rapid detection of emergency vehicles combining deep ConvNets and computer vision could significantly improve response times and reduce the likelihood of accidents in densely populated urban areas . Deep learning techniques such as convolutional neural networks (ConvNets) have revolutionized object detection in computer vision. ConvNets are trained to recognize objects by evaluating them at increasingly abstract levels, starting with the image's borders and textures and on to its detailed forms and patterns. ConvNets may be trained on a large dataset of photographs and videos of emergency vehicles in various traffic circumstances, making them perfect for detecting such vehicles. The ConvNet might be trained to recognize emergency vehicles based on distinguishing factors such as size, color, and flashing lights. Following training, ConvNet can recognize emergency vehicles in streaming video. The technology can examine each frame of footage using bounding boxes to establish the precise location of responding emergency vehicles. Vehicles may be tracked from frame to frame using computer vision algorithms, and drivers can be warned promptly. Overall, the application of deep learning-based emergency vehicle identification systems might considerably improve response times and reduce the risk of accidents. In this situation, there are numerous things moving in various directions and at diverse speeds, as well as ranging in size. It can be difficult to distinguish emergency vehicles from other objects, such as cars, pedestrians, and cyclists. Emergency vehicle detection systems are unaffected by low-visibility conditions such as fog, rain, or darkness. High-tech sensors and algorithms built for severe environments are required for this work.

A. Purpose

This project intends to recognize emergency vehicles such as ambulances, firetrucks, and police cars in real-time for the purpose of public safety and to allow the rapid arrival of emergency personnel. One effective method is to use computer vision techniques and deep learning algorithms. An emergency vehicle detection system based on deep ConvNets and computer vision could significantly improve emergency vehicle response times and reduce the risk of accidents in heavy traffic scenarios. Deep learning techniques such as convolutional neural networks (ConvNets) have revolutionized object detection in computer vision. ConvNets are trained to recognize objects by evaluating them at increasingly abstract levels, starting with the image's borders and textures and on to its detailed forms and patterns.

Following training, ConvNet can recognize emergency vehicles in streaming video. The technology can examine each frame of footage using bounding boxes to establish the precise location of responding emergency vehicles. Vehicles may be tracked from frame to frame using computer vision algorithms, and drivers can be warned promptly. The system must be real-time and low-latency in order for individuals to respond swiftly in an emergency.

Overall, the application of deep learning-based emergency vehicle identification systems might considerably improve response times and reduce the risk of accidents. The main goal is to enable autonomous learning, which will allow computers to adjust their behavior without the assistance of humans.

B. Inference From The Survey

The proposed technology takes photos of a certain area at predetermined intervals. To ensure driver and pedestrian safety, the captured image is segmented so that all objects in a specific frame are extracted, and the extracted image is then fed into the deep learning model, which recognizes the object and returns the label of the thing being detected. The present deep learning models may be able to recognize and identify emergency vehicles in video footage, according to evidence. CNNs and GANs in this model can learn information from images and videos that allow them to distinguish emergency vehicles from other types of vehicles. These algorithms can be trained on massive, annotated datasets of photographs and videos of rescue vehicles to increase their accuracy. This existing model works only on visual representation (detecting the emergency vehicle) but this model does not work from audio acoustic signals.

II. LITERATURE SURVEY

When an accident occurs these days, people are either too terrified to call for help or they make such a big mess that by the time help arrives, it's too late. A. Kunjumon et al. presented a system which has been developed to actively detect all types of road accidents and

alert the appropriate personnel; in the case of an accident, this includes the nearest police station, hospitals, general ambulances, and the registrant of the vehicle in the accident and their emergency contacts, whereas in the case of a hit-and-run, the police can be given the vehicle number of the vehicle responsible.

V. Chava et al. [7] proposes a breakthrough smart traffic system that uses average vehicle area and emergency vehicle recognition in real time to improve traffic flow and emergency response times. This system employs a pre-trained model based on the YOLOv4 and MobileNet V2 convolutional neural networks to correctly determine the number of automobiles on the road, the average vehicle area, and to recognize emergency vehicles in real time. Because of this system's ability to automatically adjust traffic signals and reroute vehicles, congestion can be lessened, and emergency vehicles can get through faster. Based on trial results, this system could be a viable option for 21st-century traffic management and emergency services, significantly reducing average travel times and emergency response times.

This [8] configuration was presented by P.C. Sanjai et al. which is frequently used to segregate ambulances, fire engines, police cruisers, and other sorts of emergency vehicles. This study categorizes ambulances using a variety of machine learning algorithms. Ambulances must be correctly classified as emergency vehicles in order to respond quickly and effectively in times of crisis.

[9] Because satellite communications are usually employed during times of crisis, Y. Zhao et al. presents a study of full analysis of the issues involved in encrypting satellite communications and proposes an elliptic curve-based mixed encryption system as a solution. A case study of portable station sites during large-scale emergencies is done to validate the model's efficacy and scientific validity. In addition, the paper's emergency finding model for satellite communication portable stations is a useful resource for the future.

K. Shetty et al. [10] presents an automated method for spotting emergency vehicles in surveillance footage using deep convolutional neural networks (CNN). The idea is to apply cutting-edge object identification technologies in real time to reliably distinguish and categorize emergency vehicles. The proposed approach detected and classified emergency vehicles more accurately (91.73 percent) and with less loss (0.2120) than earlier optimizers.

The [11] H.V Kumar et al. presented system may prove to be a feasible means of enhancing safety standards by focusing on distance warning, autonomous emergency braking, radar-based blind spot recognition, and cloud-based data storage. Radar can assist the driver because it works in poor vision and inclement weather.

This [12] will cover the many low-power LoRa (long-range) designs that are presently available. Ambulances can be swiftly deployed from the scene of an accident to the nearest hospital, where critical care can be

delivered to the injured, using cloud computing. The same is true for emergency data, which can be transmitted directly from the cloud, along with the system's response to alerting the environment and calling the necessary clinic. The proposed model incorporates GPS and impact sensors to develop a complex vehicle framework in order to swiftly detect the scene of an accident and respond properly.

In this model S.Bapodara et al. talk about an adaptable and effective traffic signal control system for mixed (less-disciplined) traffic that can be easily added to static traffic lights when resources are limited. We create a sound-sensor-based system that can recognize each type of siren in order to properly detect and categorize emergency vehicles. Data from traffic cameras is evaluated in real time to establish the number of PCUs present at each junction approach and to locate silent emergency vehicles. The experimental results show a 100% success rate for detecting emergency vehicles, a 95% success rate for classifying emergency vehicles, and a 65% success rate for classifying vehicles and counting PCUs. We also offer a queuing theory-based cost function that takes into account the current traffic situation as well as the existence of priority vehicles at a junction. We may use the cost function to optimize the green phase of the various approaches to the junction, resulting in smoother traffic flow and a shorter response time for emergency vehicles.

The [14] of this project is to address the issue with the current system. The model uses an image processing system to scan the scene for signs of an emergency vehicle as well as the number of automobiles in each lane before sending an ON/OFF signal to the traffic lights to reduce the number of cars stuck in traffic. The system presented by S.J Arul et al. employs visual processing to estimate the vehicle density in the region and identify the emergency vehicle, providing the lane with an all-clear signal. This system was built using OpenCV and the Yolo (you only look once) method from the field of image processing. The suggested smart traffic system has been tested, and the findings show that it works. To ensure correctness and validity, the suggested system has been programmed and tested numerous times.

The development of assisted and autonomous driving systems is primarily reliant on road detection, which is a component of environmental perception. High-precision road detection technology can provide critical environmental information for autonomous vehicle planning and decision-making. The road detection mechanism is introduced first, along with the technical foundations on which it is built. The algorithm for detecting roads based on visual input is then demonstrated. Finally, a collection of linked data sets in the field of road detection is created, providing researchers with new insights and paths for inquiry [15].

III. PROPOSED ARCHITECTURE

A. System architecture

We investigate the feasibility of detecting the noises emitted by emergency vehicles such as ambulances, fire engines, and police cars in this study. Given that drivers may be inattentive to emergency vehicle siren alerts, particularly when in-car audio systems are used, we propose developing an automatic detection system that can detect and notify other drivers if there are siren noises from emergency vehicles in the area.

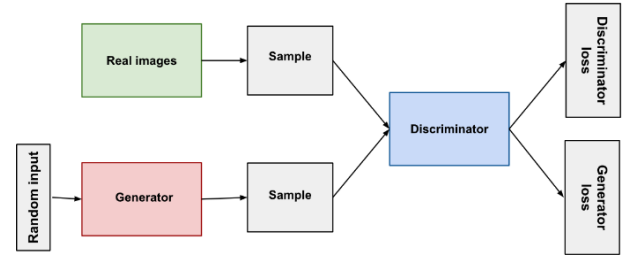


Fig. 1. System Architecture

a) Modules

The system consists of four modules which are described below.

- Gathering Data
- Data preprocessing
- Model Design
- Analyze and Predict

b) Gathering Data:

The actual process of building the machine learning model and accumulating data begins at this point. The quantity and quality of data that can be collected will determine how effectively the model works, and this stage is critical. Techniques used for data acquisition include manual interventions, web scraping and others. The traffic dataset used for analysis was downloaded from Kaggle.

c) Data preprocessing:

Preprocessing involves cleaning and preparing data for training, including removing duplicates, correcting errors, handling missing values, normalizing, and changing data types to remove the influence of data structure or collection order. Randomizing the data is also recommended as a useful technique. Data visualization is recommended to identify significant relationships between variables or class inequalities, but potential bias should be considered. Exploratory analysis can be conducted in various ways, and breaking down data into smaller units can serve educational and evaluative purposes.

d) Convolution neural networks:

CNNs are powerful models used for detecting emergency vehicles in images or video streams. They are designed to mimic how the human

brain processes visual information, CNN networks work to detect vehicles in emergency as follows (Ajit Arohan et al., 2020)

e) R-CNN:

R-CNN stands for Region-based Convolutional Neural Network. It is an extension of CNN that focuses on object detection. R-CNN proposes regions of interest in an image and applies a CNN to each region to classify and localize objects. R-CNN starts by dividing the input image into multiple regions or subregions. These regions are referred to as "region proposals" or "region candidates." The region proposal step is responsible for generating a set of potential regions in the image that are likely to contain objects. R-CNN does not generate these proposals itself; instead, it relies on external methods like Selective Search or EdgeBoxes to generate region proposals.

f) Fast R-CNN:

Fast R-CNN is an improved version of R-CNN that addresses some of its limitations. It introduces a shared feature extraction network and performs region-wise pooling, enabling faster and more efficient object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9x faster than R-CNN, is 213x faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3x faster, tests 10x faster, and is more accurate.

g) GANs:

Generating new emergency vehicle images using GANs involves leveraging Generative Adversarial Networks (GAN's) which are a type of generative model in machine learning. GANs consist of a generator network and a discriminator network. The generator network creates new samples, such as images, while the discriminator network attempts to differentiate between real samples from a training dataset and generated samples from the generator (Navidan et al., 2021). Specifically, Deep Convolutional GANs (DCGANs) are a variant of GANs that utilize deep convolutional neural networks. DCGANs enhance the traditional GANs framework by replacing fully connected layers with convolutional layers in both the generator and discriminator networks. This modification enables DCGANs to capture complex patterns and spatial dependencies in the data, resulting in the generation of more sophisticated and realistic images (Jain et al., 2022).

h) K- fold cross validation:

To validate the performance of our system, we

utilize k-fold cross-validation. This technique involves dividing the dataset into k equal-sized subsets, or folds. The system is then trained and evaluated k times, with each fold serving as the validation set once, while the remaining folds are used for training. This process helps to provide a more robust and reliable estimate of the system's performance by utilizing different subsets of the data for training and validation (Wong et al., 2019).

The convolutional neural network (CNN)-based ensemble model (SirenNet) was developed to classify the sounds of the traffic soundscape into siren sounds, vehicle horns, and noise using two network streams, one of which (WaveNet) processes the raw waveform and the other (MLNet) works with a combined feature formed by MFCC (mel-frequency cepstral coefficients) and log-mel spectrogram. Experiments on a large dataset show that raw data may be utilized to enhance the MFCC and log-mel features, yielding a highly encouraging accuracy of 98.24%.

Convolutional neural networks (CNNs) are mostly used for audio recognition in this study. When it comes to data processing, there are two main schools of thought when it comes to audio recognition methods: the first uses audio feature engineering techniques to extract useful features in the time-domain and/or frequency-domain before performing the recognition task, and the second extensively uses deep neural networks to construct an end-to-end recognition system that learns features directly from raw waveforms.

The purpose of this research is to combine both methodologies in order to improve recognition accuracy. We specifically study whether or not deep neural network characteristics can supplement handcrafted features in dealing with a recognition task.

IV. EXPERIMENTAL RESULT

First, a dataset of pictures of emergency vehicles such as ambulances, police cars, and fire trucks must be compiled in the proposed system. This data set must be comprehensive, encompassing a wide range of use scenarios and lighting settings. As part of the preprocessing phase, images are shrunk to the same dimensions, pixel values are standardized, and the dataset is separated into a training set and a test set. Data augmentation methods like rotation, scaling, and flipping may be employed to reduce overfitting and make the training data more diversified. As a pre-trained model for emergency vehicle detection, the Xception model, which is a deep convolutional neural network designed for image classification, can be used. The process of improving the parameters of a previously trained model for use on a different task is known as transfer learning. In this case, the Xception model can be optimized for emergency vehicle identification by replacing the final classification layer with one that generates probabilities rather than labels.

After training, the model can be assessed using the secret testing data. The capacity of the model to generalize to previously unseen data is dependent on the addition of unique pictures to the testing set. Once trained and validated, the model may be used to create new pictures of rescue vehicles to provide predictions. The model is fed an image and returns the probability that each class is right. The image's suggested class has the highest likelihood. The dataset should include a diverse variety of lighting conditions, vehicle types, and traffic density. To collect data and develop the model, the CNN xception architecture is employed. The RMSprop optimizer, like the gradient descent technique with momentum, is used here. Vertical oscillations are dampened using the RMSprop optimizer. Using transfer learning, we freeze the early convolutional layers and only train the last few layers that produce a prediction for object identification with a CNN. This allows us to boost our learning rate and our algorithm's ability to take larger horizontal steps, resulting in faster convergence. Finally, we can use the Flask framework and the CNN Xception architecture with the RMSprop optimizer and transfer learning to create a model that can detect emergency vehicles in traffic signals and generate green corridors for them, reducing the time it takes for the lights to change and increasing the effectiveness of emergency services.

TABLE I. PERFORMANCE COMPARISON OF ALL ALGORITHMS

Algorithm	Precision	Recall	F1 Score	Accuracy
GAN	99.34	99.67	99.59	99.19%
CNN	98.68	99.32	99.17	98.37%

The model performs well in training but significantly worse in validation, according to the accuracy plot. This indicates not only the possibility of overfitting but also the model's capacity to generalize effectively to new data. The model performs well in training but significantly worse in validation, according to the accuracy plot. This indicates not only the possibility of overfitting but also the model's capacity to generalize effectively to new data.

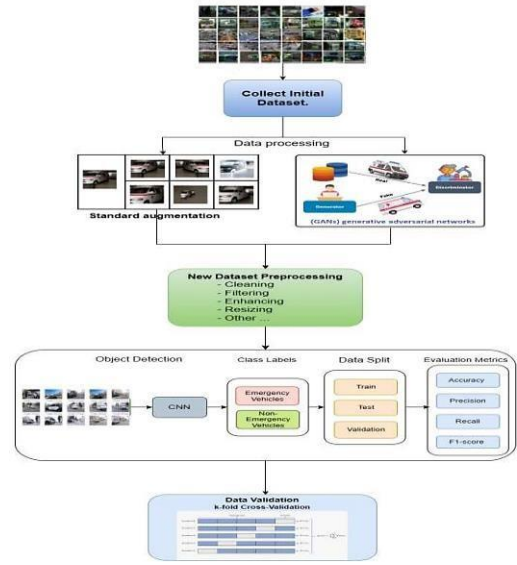


Fig 1: Proposed Architecture

The following figure shows that the training loss is very low at 0.3, while the validation loss is fairly high at 1.5%. This indicates that the model has a low training loss but a significant validation loss, indicating that it is overfitting to some extent. The plot can be used to monitor how the model is progressing during training and to determine whether to make modifications to the model's architecture or training parameters to improve the model's performance on new data.

The existing model has a limitation in that it can only recognize a single object at a time, whereas the upgraded model with image segmentation can detect many things at once. Both the detection rate and the number of probable identifications grow considerably.



Fig 2: Examples of Vehicle Detection Results

The output prediction figure, seen in Figure 2 is a visual representation of the model's predictions. As seen in the figure, a user submitted a camera-captured image for prediction. According to the system, the image features an emergency vehicle. The label "Emergency Vehicle" is shown over the detected object to denote its type.

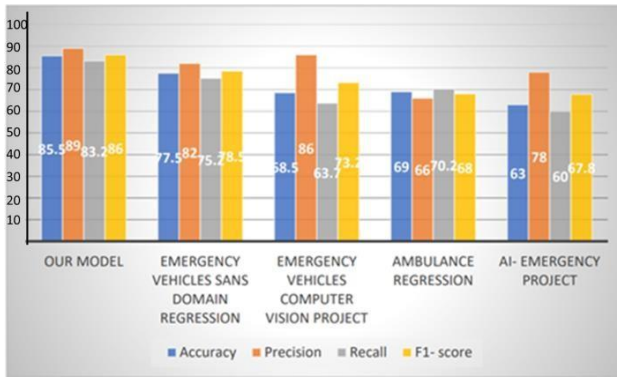


Fig 3: Comparison of Emergency Vehicle Detection Models

Figure 3 is the output prediction figure, which provides a visual representation of the model's predictions. As seen in the figure, a user submitted a camera-captured image for prediction. According to the system, the image features an emergency vehicle. In addition, the label "Non-Emergency Vehicle" displays at the top to further describe the purpose of the discovered object.

Finally, combining the Xception model with the Flask framework for recognizing emergency vehicles from photographs results in a solid and approachable method for doing so quickly and reliably. The Xception model is a cutting-edge deep learning system that has been successfully applied to the task of object detection in pictures, including emergency vehicle identification. Using the Xception model and the Flask framework, we can create a web app that gives users near-instant feedback on whether or not a submitted image contains an emergency vehicle. This can be invaluable in times of crisis, when every second counts. Flask is a lightweight, adaptable, and simple-to-install online application framework. Using Flask, we can create a user-friendly, responsive, and scalable online application. This allows for rapid expansion of the system over multiple sites and huge amounts of demand.

V. CONCLUSION

The application uses the design that was created in order to separate the emergency vehicles from most other vehicles. Two different algorithms in computer vision, object detection and instance segmentation, have been described for emergency vehicle detection and localization. Both algorithms can be used for applications in intelligent transportation systems, such as the deployment of smart traffic signals and autonomous cars, as they are effective at distinguishing an emergency vehicle from a group of vehicles. By implementing both algorithms the emergency vehicle is detected and GAN's algorithm shows much better accuracy than cnn algorithm. GAN's algorithm shows 98.6% whereas the cnn algorithm shows 97.3% accuracy. Two different algorithms in computer vision, object detection and instance segmentation, have been described for emergency vehicle detection and localization. Both algorithms can be used for applications in intelligent

transportation systems, such as the deployment of smart traffic signals and autonomous cars, as they are effective at distinguishing an emergency vehicle from a group of vehicles. The application uses the design that was created to separate the emergency vehicles from most other vehicles. When the computer determined the detained automobile was an emergency vehicle and "a notification made to the traffic controller," we were able to successfully convince the computer to change one traffic light to green while other traffic lights were set to red.

REFERENCES

- [1]. J. Lee, K. Oh, Y. Yoon, T. Song, T. Lee and K. Yi, "Adaptive Fault Detection and Emergency Control of Autonomous Vehicles for Fail-Safe Systems Using a Sliding Mode Approach," in IEEE Access, vol. 10, pp. 27863-27880, 2022, doi: 10.1109/ACCESS.2022.3155738.
- [2]. V. -T. Tran and W. -H. Tsai, "Audio-Vision Emergency Vehicle Detection," in IEEE Sensors Journal, vol. 21, no. 24, pp. 27905-27917, 15 Dec.15, 2021, doi: 10.1109/JSEN.2021.3127893.
- [3]. J. Lee et al., "Emergency Pull-Over Algorithm for Level 4 Autonomous Vehicles Based on Model-Free Adaptive Feedback Control With Sensitivity and Learning Approaches," in IEEE Access, vol. 10, pp. 27014-27030, 2022, doi: 10.1109/ACCESS.2022.3156275.
- [4]. F. Khelladi, M. Boudali, R. Orjuela, M. Cassaro, M. Basset and C. Roos, "An Emergency Hierarchical Guidance Control Strategy for Autonomous Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pp. 4319-4330, May 2022, doi: 10.1109/TITS.2020.3043485.
- [5]. N. Kumar, D. Acharya and D. Lohani, "An IoT-Based Vehicle Accident Detection and Classification System Using Sensor Fusion," in IEEE Internet of Things Journal, vol. 8, no. 2, pp. 869-880, 15 Jan.15, 2021, doi: 10.1109/JIOT.2020.3008896.
- [6]. J. M, R. P, A. Kunjumon, M. Thamban and A. Roy, "Convolutional Neural Networks (CNN)-based Vehicle Crash Detection and Alert System," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 161-164, doi: 10.1109/IDCIoT56793.2023.10053471.
- [7]. V. Chava, S. S. Nalluri, S. H. Vinay Kommuri and A. Vishnubhatla, "Smart Traffic Management System using YOLOv4 and MobileNetV2 Convolutional Neural Network Architecture," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 41-47, doi: 10.1109/ICAAIC56838.2023.10141268.
- [8]. P. C. Sanjai, J. J. Sam, S. Iyengar and K. Kalimuthu, "Ambulance Detection in Traffic Signals Using Image Processing," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2023, pp. 927-931, doi: 10.1109/ICIRCA57980.2023.10220909.
- [9]. J. -J. Liu, S. -F. Hai, Z. -R. Wang, Y. Zhao and X. Shi, "Construction of Meteorological Emergency Communication System Based on Portable Satellite Emergency Communication Station," 2023 12th International Conference of Information and Communication Technology (ICTech), Wuhan, China, 2023, pp. 509-514, doi: 10.1109/ICTech58362.2023.00100.
- [10]. M. S, K. Shetty, A. Kodipalli, T. Rao and K. S, "Emergency vs Non-Emergency Vehicle Classification: Enhancing Intelligent Traffic Management Systems," 2023 International Conference on Network, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2023, pp. 1-6, doi: 10.1109/NMITCON58196.2023.10276208.
- [11]. B. S P, H. V M, K. L. Reddy, L. M and H. V. Kumar, "Effective Method for Vehicle-Automatic Emergency Braking System and Blind Spot Detection Using Radar and IoT," 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 2023, pp. 1-5, doi: 10.1109/ICSSES58299.2023.10200594.

- [12]. K. Jain and H. K. Saini, "An Emergency Rescue Framework through Smart IoT LPWAN," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 668-672, doi: 10.1109/InCACCT57535.2023.10141728.
- [13]. S. Bapodara, S. Mesvani, M. Chaturvedi and P. Rajput, "Traffic Congestion and Emergency Vehicle Responsive Traffic Signal Control in Resource Constrained Environment," 2023 11th International Symposium on Electronic Systems Devices and Computing (ESDC), Sri City, India, 2023, pp. 1-6, doi: 10.1109/ESDC56251.2023.10149873.
- [14]. S. J. Arul, M. B. S, S. L, Sufiyan, G. Kaliyaperumal and J. K. K A, "Modelling and Simulation of Smart Traffic Light System for Emergency Vehicle using Image Processing Techniques," 2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM), Uttar Pradesh, India, 2023, pp. 1-4, doi: 10.1109/ICIPTM57143.2023.10117651
- [15]. .C. Liu, X. Li, Q. Liu, F. Yang, Z. Li and M. Li, "A review of vision-based road detection technology for unmanned vehicles," 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-8, doi: 10.1109/IV55152.2023.10186761.

CERTIFICATE

