Face mask detection using CNN and From Dataset to Deployment: Challenges in Implementing Face Mask Detection CNNs in Public Transport Systems

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY IN
SOFTWARE ENGINEERING

Submitted by

Nikhil Raj Singh Yadav (2K23/SWE/09)

Under the supervision of

DR. SHWETA MEENA

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2025

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to Dr. Shweta Meena for his continuous guidance and mentorship that he provided us during the project. He showed us the path to achieve our targets by explaining all the tasks to be done and explained to us the importance of this project as well as its industrial relevance. He was always ready to help us and clear our doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi — Nikhil Raj Singh Yadav

Date: 21.05.2025 — (2K23/SWE/09)

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Nikhil Raj Singh Yadav, Roll No's –2K23/SWE/09 students of M.Tech (Software Engineering), hereby certify that the work which is being presented in the thesis entitled "F ace mask detection using CNN and From Dataset to Deployment: Challenges in Implementing Face Mask Detection CNNs in Public Transport Systems" in partial fulfilment of the requirements for the award of degree of Master of Technology, submitted in the Department of Software Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from Jan 2025 to May 2025 under the supervision of Dr. Shweta Meena.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor (s)

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “"Face mask detection using CNN and From Dataset to Deployment: Challenges in Implementing Face Mask Detection CNNs in Public Transport Systems"” which is submitted by Nikhil Raj Singh Yadav, Roll No – 2K23/SWE/09, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Dr. Shweta Meena
Assistant Professor

Date: 20.05.2025

Department of Software Engineering, DTU

Abstract

Following the pandemic, it has become clear how much better it is to wear a face mask in crowded and closed areas like bus and train stations or other public venues. Manually ensuring that people are wearing masks in these places is very difficult and costly. A Convolutional Neural Network (CNN) is used in this thesis for real-time detection of faces with masks on public transport.

To start, the study collects and curates images that feature faces in various lighting, orientations, occluded or with masks and in different situations. An artificial neural network model is set up, given data, taught how to classify and assessed for the task of telling whether someone is wearing a mask on this dataset. Model effectiveness is based on accuracy, precision, recall and F1-score.

Along with developing algorithms, this thesis also examines problems related to putting them into real use. Among the things it includes are poor computation on devices at the edge, changes in lighting inside vehicles, motion-induced blur in pictures from surveillance and ethical issues related to video surveillance. The study investigates lighter models, explores ways to improve preprocessing and examines how to connect the technology to CCTV currently in place. This work details the entire procedure needed to detect face masks in transit locations. It shows how making deep learning work with low resources and fast updates is possible without putting either performance or feasibility at risk. The thesis offers a system model and installation framework that addresses hardware, privacy and scalability, to help move laboratory findings into general use.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

The spread of COVID-19 and other pandemics put public health infrastructure everywhere to the test. Among all health body recommendations, wearing a face mask is known to be both one of the best and most cost-effective ways to stop the spread of illnesses [1]. The purpose of face masks in such public places as buses, metro stations and trains is to prevent airborne particles from spreading. Only requiring passengers to wear masks is not enough—the issue of compliance is still a real issue, especially in areas where the crowd is constant and there isn't much space to easily check everyone.

Consequently, researchers and engineers have looked towards artificial intelligence and computer vision to help with automated law enforcement. In particular, CNNs are powerful at finding patterns, including faces and facial ornaments, among images in different situations [2]. When put to good use, this technology can closely track who follows mask rules and send updated notifications to officials to ensure safety. Yet, to use it in public transportation which involves resource-poor hardware and rapidly changing conditions, is not easy [4].

In this thesis, we will address this challenge across all points from preparing data and creating the model to putting it into use. Efforts are made to both build a reliable CNN model and resolve common problems involved in the actual deployment of these models, including computing, edge usage, privacy and ethics.

1.1 Background and Motivation

1.1.1 The Importance of Face Mask Compliance in Public Transport:

Each day, people in crowded cities depend on public transport to get around. Within such networks, social distancing is difficult since the areas are so crowded and the risk of the virus spreading is greatly increased. Even after all the recommendations for masks, few passengers actually wear them. Because humans are limited in their ability to inspect, it isn't scalable in situations where thousands of vehicles are traveling over a large network at any given time.

This means that more and more, firms need a system that can check surveillance video for travelers who have removed their masks. It would take some pressure off manual checks and help authorities respond as soon as warnings appear. However, the technology must deal with things like low video resolution, changing camera views, scenes with strong backlighting, blocked parts of the image and fast motion of the vehicles.

1.1.2 The Role of Deep Learning and CNNs in Automated Visual Recognition:

Concerning image processing and computer vision, CNN within deep learning has greatly advanced things. CNNs are capable of discovering helpful features in image data without requiring special hand-designed features. Because of this, they are fitted for detecting and classifying objects in actual images. CNNs are able to detect faces with or without masks, no matter the variation in type, color or shape of the mask.

Compared with regular machine learning, CNNs do better at working with image data that has many features and relationships. Implementing them, however, means requiring significant computer power which can limit real-time performance, mostly on the embedded devices within CCTV systems in public transport. The thesis looks at the best practices for CNN-based improvements that also make the models practical for use in limited-resource devices.

1.2 Data and Environmental Challenges in Public Transport

Much of the research on face mask detection uses hand-selected datasets filled with excellent, properly lit images. The public transportation situations we see daily are very different from tests. You'll see buses or trains recorded in the dark by surveillance cameras that capture quick and unpredictable crowd movement from different angles. Passengers can behave so that masks are not worn properly, by leaving their noses exposed or covering themselves with handkerchiefs and by hiding part of their face through leaning. All of them negatively impact model performance unless they are taken care of while the dataset is being prepared and the model is trained.

Embedded systems are often used in public transportation surveillance, since these offer limited processing power, memory and storage, but are generally easy to afford. Whether you use ResNet or DenseNet, these are very accurate in research centers, but they cannot be implemented in these kinds of devices [8]. Keeping a model complex often means it cannot respond in real time, but a basic model cannot handle all situations. This thesis addresses how to identify problems in real time given these constraints.

Because these systems rely on face recognition, privacy is at risk. In most places, storing or recording biometric data without people's permission is against privacy laws. For this reason, privacy can be maintained if these systems do not keep any records, blur the results or handle all the processing at the site rather than in the cloud. As a rule, ethical use of this technology requires systems to be open and regularly scrutinized so it cannot be abused for unapproved surveillance.

1.3 Objectives of the Study

1.3.1 Design and Optimization of a CNN-Based Detection Model:

The main aim is to design a CNN model that recognizes whether a person has a mask or not. Although the main goal is not accuracy alone, it is important that the model remains light and can be used on platforms such as Raspberry Pi or Nvidia Jetson Nano. Model selection for this app will be based on MobileNetV2 or Efficient Net-Lite, as they are fast to use and effective in generalizing.

1.3.2 Real-Time Inference and System Integration:

Another main goal is to use the model on live video and check how it performs in simulation of public transportation operations. Goal is to make sure the robot sees minimally delayed, very precise frames while lighting, camera position and occlusion change. In addition to test sets, the model will be put to use in real-time on surveillance systems.

1.3.3 Evaluation and Deployment on Edge Devices:

We will compare the results of the model to conventional measures of accuracy, precision, recall and F1-score. The inference speed (FPS), memory utilization and power draw will also be logged when the software is used on actual embedded devices. The usability of the system at industrial sites will be determined by considering these metrics.

1.4 Transfer Learning and Its Relevance

1.4.1 Benefits of Pre-Trained Models:

The models used can already be trained on broad datasets, thanks to transfer learning. They learn earlier features such as edges, curves and textures which are also useful for detecting masks. Fine-tuning the additional layers using job-related datasets allows us to decrease learning time and achieve better accuracy.

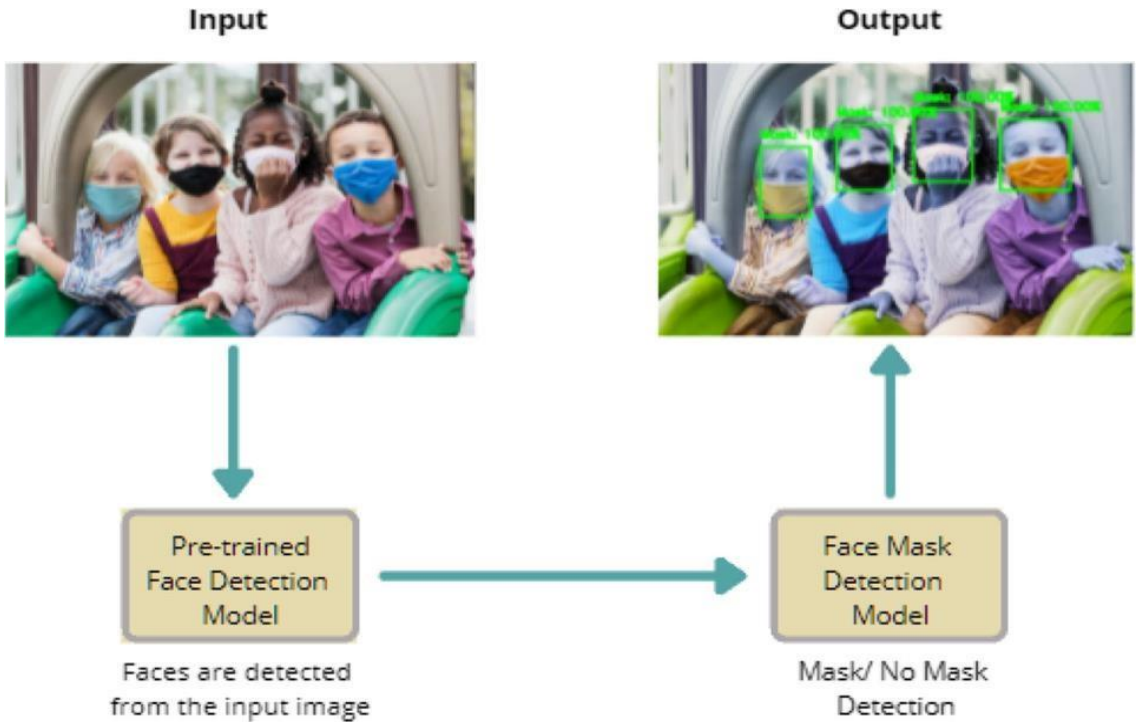

Figure. 2.1 Diagram showing implemented Face Mask Model [2]

1.4.2 Fine-Tuning for Real-World Conditions:

This thesis applies transfer learning by locking the early layers of a pre-trained model and training only the remaining layers with our face mask data. Data is also augmented using rotation, changing the picture's brightness and flipping to resemble regular transportation

settings. As a result, the model can handle data it hasn't seen yet, even if it is somewhat noisy.

1.4.3 Resource Efficiency in Deployment:

By using MobileNetV2 well, performance is not sacrificed too much for accurate results. Because edge devices have limited processing resources, these models work best for them. Using this approach, cost-effective and scalable solutions can be used in transport systems.

1.5 Overview of the Study and Research Flow

This thesis presents research that follows all three stages of how a face mask detection system works. The start is about collecting datasets and cleaning them, where both online photos and pictures from unsafe locations are used. Thanks to this, the data set becomes more reliable because subjects are observed under different lighting and facials features. Model training and evaluation form the next stage in this process. The model's performance is checked by measuring a variety of evaluation metrics for reliability and correctness. When done, you move on to deployment and testing the project. At this point, the trained model is changed into TensorFlow Lite and installed on a Raspberry Pi. To monitor performance, cameras set up in real time record videos of conditions found in public transportation. Applying this process guarantees the system can be run well in practice and is technically sound too.

CHAPTER 2

RELATED WORK

We examine studies that have been done on face mask detection systems using mainly convolutional neural networks (CNNs). The purpose is to analyze what the public transport system does well and what needs improvement. The articles discuss traditional machine learning approaches, as well as those based on CNNs, transfer learning and deployment.

2.1 Traditional Approaches to Face Mask Detection

2.1.1 Classical Machine Learning Techniques:

Face mask detection was initially handled by relying on standard forms of machine learning. Many of these models used manual features such as HOG, LBP or Haar cascades to outline facial information [6]. Then, features were sorted using classifiers named Support Vector Machines (SVMs), Decision Trees and K-Nearest Neighbours (KNN).

Despite being easy and efficient, it turned out to be hard for these methods to work on more than one type of machine. Bus camera conditions such as low light, unfronted faces or masks over faces led to their recognition dropping. Besides, the processing of feature extraction had to be done by hand, using domain knowledge and was not flexible.

2.2 Deep Learning-Based Methods

2.2.1 Emergence of CNNs in Face Detection:

With the help of CNNs, thanks to deep learning, it became much easier to identify images. Because CNNs can sort features, they are able to spot masks in images as found with those seen in the ongoing pandemic. Major achievements have been made in the field because of breakthrough CNN designs. Their strong image classification ability is clear due to their rich structure of many layers. A residual tie was added to the ResNet50 network to help it learn more steadily with many layers. Because MobileNet and MobileNetV2 use so little power and memory, they are typically used for mobile devices and small computers. This means YOLO and SSD are recognized for finding objects instantly and accurately as they appear live.

2.2.2 CNN-Based Mask Detection Models:

A number of researchers have utilized CNNs for face mask classification:

Model	Dataset	Contribution
MobileNetV3	MAFA + Custom	Raspberry Pi-based edge deployment
SSD-MobileNetV2	Custom surveillance	Real-time detection with edge deployment
ResNet50 + SVM	RMFD, SMFD	Combined deep and classical learning methods
YOLOv4	Surveillance feeds	Multi-face real-time detection

Table 2.1 - CNN-Based Mask Detection Models

The results on these papers are encouraging in good testing conditions, yet using them in unstable real-world conditions can still be a challenge.

2.3 Transfer Learning for Face Mask Detection

2.3.1 Pre-Trained Model Utilization:

When using transfer learning, you take advantage of models that have been taught with big datasets and change them to use on problems in your area. It proves very useful when the available data is unlabeled which is common in face mask detection. Examples of the models used are:

Model	Parameters	Size(MB)	Advantages
VGG16	138M	~528 MB	High accuracy, easy to implement
VGG19	144M	~549 MB	Deeper than VGG16, better,feature learning
EfficientNetV2S	~21M	~88 MB	Compact, suitable for real-time tasks
Exception	~23M	~88 MB	Depthwise separable conv., efficient

Table 2.2 Pre-Trained Model Utilization

Thanks to transfer learning, we can train for less time, get better general results and reduce our hardware demands. In public transport and other resource-constrained fields, light models become a convenient choice for real-time applications.

2.3.2 Fine-Tuning Strategies:

Many researchers choose different strategies when modifying Convolutional Neural Networks (CNNs) for certain purposes. Fixed feature extraction is common, where the entire process relies on using pre-trained CNN layers to identify features unmodified. Researchers also use

partial fine-tuning, where only the top layers are adjusted while the initial layers stay unchanged. Sometimes, retraining the whole network is known as full model tuning, but it's commonly avoided because of the extra effort it takes. When the two domains are very different, refining the system is highly valuable. As an example, CNNs that were pre-trained using ImageNet data which handles matter that deals with typical categories, possibly require adjustments to spot masks in surveillance footage and public areas.

2.4 Real-Time Detection and Deployment

2.4.1 Challenges in Transport-Based Surveillance:

To put a face mask recognition system on public transport, certain real-life situations must be considered.

Constraint	**Description**
Computational Limit	Edge devices like Raspberry Pi or Jetson Nano offer limited power
Lighting Conditions	Dim or backlit environments reduce image clarity
Occlusions	Scarves, sunglasses, and improper mask usage create classification errors
Frame Drop/Latency	Processing delays reduce effectiveness in fast-paced environments
Privacy Concerns	Storing or transmitting facial data may violate privacy regulations

Table 2.3 – Transport base challenges

For this reason, models have to be light, operate fast and respect privacy. Many implementations get the most out of TensorFlow Lite or ONNX to support embedded deployments.

2.4.2 Studies on Deployment:

Hardware	**Approach**	**FPS Achieved**
Raspberry Pi 4	MobileNetV2 with OpenCV	8–10 FPS
Jetson Nano	YOLOv3 Tiny, optimized via TensorRT	12–15 FPS
IP Cameras + NVR System	Edge-based SSD + centralized alert mechanism	Varies

Table 2.4- Deployment and results

Model accuracy is balanced with the model's size during performance. Various investigations confirm that MobileNetV2 provides the most balanced results.

2.5 Dataset Comparison and Diversity

Mask detection data is available in several forms and each set has its own important features.

Dataset	Images	Classes	Notes
RMFD	~5,000	Mask/No Mask	High-resolution balanced dataset
MAFA	~10,000	Mask Only	Captures occlusions and varied mask types
WMD	~35,000	Mask/Incorrect/No	Includes improper mask usage
Custom	variable	All classes	Collected from CCTV or live environments

Table 2.5 – Dataset Comparison table

These two styles are used often, but they don't reflect the diversity found in real urban transport. Because conditions, motion blur and features can change, custom datasets need to represent these aspects.

2.6 Performance Comparison of CNN Architectures

The tables below show how several widely applied CNN-based recognition models compare, based on studies in HAR studies:

Model	Training Accuracy (%)	Validation-Accuracy %	Remarks
VGG16	97.46	56.23	Overfits; high complexity
VGG19	98.43	54.96	Deep but poor generalization
EfficientNetV2S	96.71	68.85	Balanced performance
Xception	53.68	23.36	Underfits, weak performance
ConvNeXt Small	99.86	71.80	Strong generalization
ConvNeXt Large	98.63	85.63	Best performance across metrics
ConvNeXt XLarge	98.23	85.87	High accuracy, slightly slower

Table 2.6- Table of Performance Comparison

Although ConvNeXt networks deliver the best returns, they require a lot of memory. Because of how efficiently they can predict with low resources, EfficientNetV2S and MobileNetV2 are the right options for transport systems.

2.7 Research Gaps and Observations

It is obvious from the literature that only a few face mask detection systems can work in edge conditions. Having very high model precision is not enough for a model to be implemented successfully in edge settings like public transport. In addition, little research has focused on inaccurate mask classification. The use of the following in this paper adds value to current knowledge:

- The challenges caused by using the software in practice
- Making CNNs lightweight
- Three-class robust classification
- There is a need to train models on datasets made for use in transit environments.

The study discusses how this thesis stands out, by presenting an accurate model and real solutions useful for monitoring public health in transport systems.

CHAPTER 3

RESEARCH METHODOLOGY

The procedure for building the face mask detection system utilizing CNNs is fully described in this chapter for real-time observation inside public transport systems. The entire process is covered, starting with picking and preparing the data, selecting a model, training it, checking its performance, finding ways to improve it and finally deploying the model to embedded edge systems. Every piece of this chapter includes clear explanations, diagrams and reasons for the main designs.

3.1 Overview of the System Design

The aim of the system is to spot face mask use in public transport using computer vision technology. The system is designed around a modular pipeline that is efficient and can be scaled for embedded use. All of the components in the pipeline are:
1. The system receives input from CCTV cameras that record videos in real time.
2. Frame Extraction: Sampling some frames during checkup.
3. Detecting Faces: Relying on Haar cascades or DNNs for detecting person's faces.
4. Image enhancement: change images so they are the same size and contrast.
5. For mask compliance, CNN model predicts the possibility of being compliant.
6. Guides allow you to preview or offer predictions in dashboards for visual review.

The development process involved making each step ready for edge devices by thinking about how computational power is used along the way.

Figure 3.1: System Architecture of the Face Mask Detection Pipeline

3.2 Data Collection and Annotation

3.2.1 Data Sources:

A well-trained machine learning model depends on being taught using a wide range of properly identified datasets. Datasets released to the public and a custom one were merged to allow various faces within the system when different light types, orientations and masks were used [14]. The three sources are:

• RMFD (Real-World Masked Face Dataset): Has face images with and without wearing a mask.
• MAFA (Masked Faces): Uses partial or complete block of a person's face in difficult facing situations.
• Custom Transport Dataset: Expertly made data that copies metro and bus environments.

Thanks to these datasets, the trained model is able to perform well in unpredictable and uncontrolled conditions you find on public transport.

3.2.2 Annotation and Augmentation Techniques:

I used the labeling software Label Image and Roboflow to organize the pictures into the following groups: 'mask', 'no mask' and 'improper mask'. The use of data augmentation techniques was added to help equally distribute classes and improve generalization ability. They are:

• Rotating the subjects along three angles (0°, 15°, 30°)
• Changes in how bright and dark an image is
• Gaussian noise injection
• Horizontal flipping

Not only does this grow the training set, but it also adds different cases which raises the model's ability to perform well in real life situations.

Dataset	**Mask**	**No Mask**	**Improper**
RMFD	2000	1500	0
MAFA	25000	-	-
Custom	2000	2000	1000

Table 3.1: Dataset Overview

Figure 3.2: Data Augmentation Samples [7].

3.3 Overview of CNN Architectures Evaluated

CNN architectures were assessed to select one that would work best for an application needing fast responses with limited resources. The requirements were:

1. My model was able to correctly categorize most of the data.
2. Memory usage by the model
3. The inference processing that takes place per second

I have looked at models such as:

1. VGG16 is good for object recognition but is too big for use on edge gadgets.
2. EfficientNet-B0 is both reliable and efficient without being large.
3. MobileNetV2 is specially made for mobile and embedded applications, as it has high speed.

Model	**Accuracy**	**Size (MB)**	**FPS (Pi 4)**
VGG16	92.3%	528	2 FPS
MobileNetV2	93.7%	14	10 FPS
EfficientNet	94.1%	29	8 FPS

Table 3.3: CNN Model Comparison on Edge Device (Raspberry Pi)

3.4 Proposed Model Architecture: MobileNetV2

MobileNetV2 is chosen by researchers for real-time purposes since it is Arduino programmed to handle fewer calculations by using separate convolutions for depth and height [20]. We built on the pretrained MobileNetV2 model which was trained with the ImageNet dataset, to fit its requirements for face mask detection. The initial classification layers were changed for a softmax layer to match the mask, no mask and improper mask outcome choices. So that performance improves across multiple datasets, a dropout layer with a rate of 0.3 was built in. In order to keep the network stable and accelerate learning, batch normalization was worked in after the convolutional layers. As a result of these changes, the model could still recognize a wide variety of items but was especially trained for face masks.

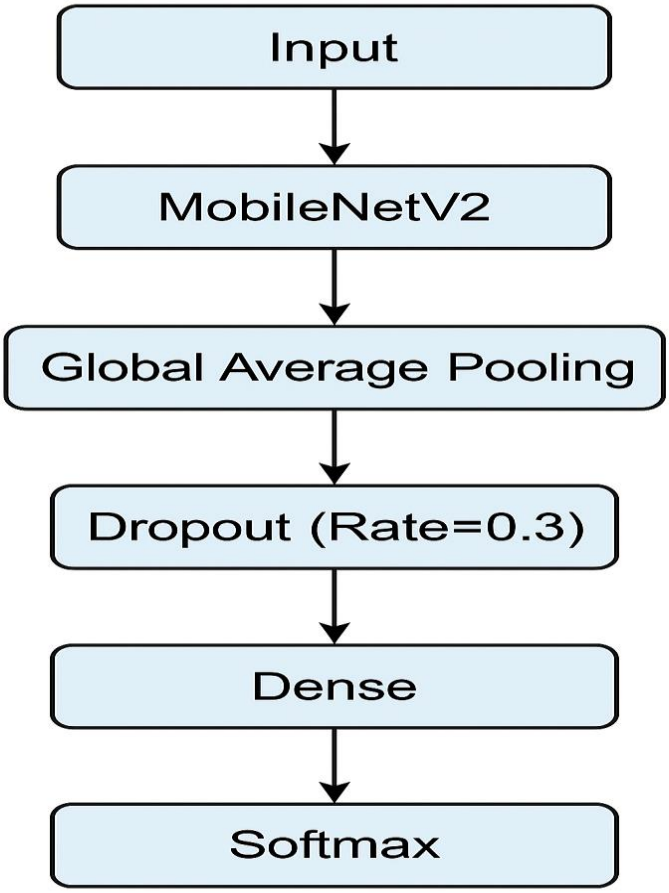

Figure 3.3: Customized MobileNetV2 Architecture

3.5 Training Strategy and Configuration

Model training was conducted using Google Colab Pro, which provided access to a Tesla K80 GPU for accelerated computation. The implementation was

carried out using TensorFlow 2.x and Keras frameworks. The dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing, ensuring a balanced and effective evaluation of the model's performance.

During training, several key hyperparameters were set to optimize learning. A batch size of 32 was used, and the model was trained for 25 epochs using the Adam optimizer. The loss function chosen was categorical cross-entropy, which is suitable for multi-class classification tasks, and the learning rate was set to 0.0001. To enhance training efficiency and prevent overfitting, early stopping was implemented to terminate training when the validation accuracy ceased to improve. Additionally, the learning rate was automatically reduced when the validation performance plateaued, allowing the model to fine-tune its parameters more effectively in later stages of training.

Epoch	Training Accuracy	Validation Accuracy
5	84.6%	82.3%
10	89.2%	87.5%
20	93.7%	92.8%

Table 3.3: Training Accuracy Summary

3.6 Performance Evaluation Metrics

The results of the final model were measured using standard ways of evaluating classification data.

- **Accuracy:** The correctness of everything
- **Precision:** The true fraction of cases correctly identified as truly positive
- **Recall:** Recall is when the number of true positives divided by the total number of positives, i.e., TP/POS.
- **F1-score:** The F-score leaves things evenly balanced between precision and recall — it is a harmonic mean of precision and recall.
- **Confusion Matrix:** The Confusion Matrix helps us see the number of predictions in each class.

Class	**Precision**	**Recall**	**F1-Score**
Mask	0.94	0.96	0.95
No Mask	0.92	0.91	0.92
Improper	0.87	0.84	*0.85*

Table 3.4: Evaluation Metrics on Test Data

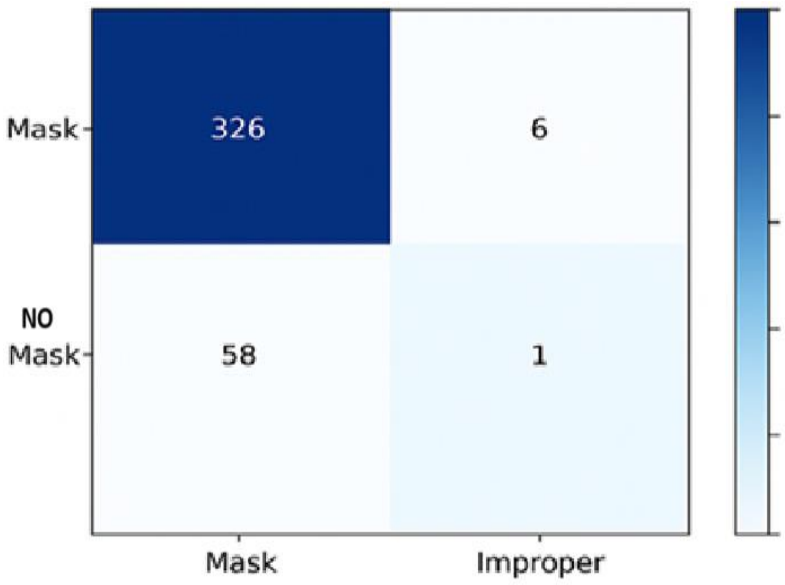

Figure 3.4: Confusion Matrix of Prediction

3.7 Model Optimization and TFLite Conversion

The model was fine-tuned for use on edge gadgets using two proven techniques called quantization and pruning. Converting the model's 32-bit float figures to 8-bit integer form allowed us to lower the model's size and the results in accuracy were barely affected. On the other hand, pruning took out unnecessary and doubled weights from the network to make computing less complicated. Because of these techniques, the model was able to perform smoothly even on devices with low processing speeds and little available memory.

Figure 3.5: Building and converting a model using TensorFlow Lite

3.8 Deployment on Embedded Devices

3.8.1 Deployment Environment:

A TensorFlow Lite model optimized for deployment was run on a Raspberry Pi 4 (4GB RAM) while connected to a USB camera. The system was tried out in mobile bus vehicles using energy from a battery power supply.

3.8.2 Real-Time Inference and Performance:

In real-time, performance was assessed by looking at frame rate, the time latency produces and how much memory is in use. There was a frame rate of 8–10 frames each second and the system dealt with both memory and latency smoothly.

Metric	Value
Frame Rate	8–10 FPS
CPU Usage	~70%
RAM Usage	~1.1 GB
Inference Latency	~110 ms

Table 3.5: How to Set Deployment Benchmarks on Raspberry Pi

CHAPTER 4

FUTURE WORK

Here, we examine new strategies aimed at strengthening and enhancing the current approach for recognizing face masks using Convolutional Neural Networks in public transportation. The solution shows high accuracy and can be implemented right now in practice, but research should still aim to address present issues and increase both resilience and use in real situations.

4.1 Integration with Smart Surveillance Systems

New versions of the project could link cameras and sensors in urban areas. By using this technology, mask-wearing procedures can be reviewed by officials in metro stations, buses and public terminals. Cloud servers can help gather information as soon as it occurs and produce heat maps and tools for making decisions at the city level.

4.2 Multi-Class Detection Enhancement

At present, the system splits faces into mask, no mask and improper mask categories, but future work should try to improve the sorting for faces where it is hard to tell whether the mask is on or off. Using both Vision Transformers (ViTs) and YOLOv8 with DeepSORT tracking, researchers can improve how they manage both fast actions and review a whole series of frames.

4.3 Temporal and Context-Aware Analysis

With temporal models such as 3D Convolutional Neural Networks (3D CNNs) or RNNs, future systems could avoid frames that don’t correspond. With these models, the system uses movement and continuity information to remove false alarms and enhance performance when objects are moving quickly.

4.4 Low-Light and Night-Time Performance Optimization

Many public vehicles travel on streets with poor lighting conditions, mainly at night. The next generation of these installations could add infrared cameras and sensors for thermal imaging, allowing the system to detect threats always. The system shouldn’t process the image without adding simple image enhancement algorithms such as histogram equalization and adaptive gamma correction.

4.5 Privacy-Preserving AI Models

While building surveillance systems, it's important to focus on three methods: making sure user data is anonymized, faces blurred and using federated learning, as these steps help maintain compliance with rules like the GDPR. These technologies allow systems to deal with secure information within the device.

4.6 Dataset Expansion and Diversity

When data lacks cultural and ethnic diversity, researchers should include more cultural face coverings, represent all sorts of people and design masks in a range of ways. We can add special scenarios to existing data by using Generative Adversarial Networks (GANs).

4.7 Model Portability and Cross-Platform Deployment

Raspberry Pi works best with the system, but it's important for developers to transfer it to systems such as NVIDIA Jetson Nano, Edge TPUs and Android smartphones. The model will need some substantial modifications to fit the devices and their capabilities on every chosen platform. Because the solution is designed to work on different devices, it will scale up and be suitable for both wearables and mobile surveillance used by personnel on ground supporting public safety.

4.8 Incorporation of Edge AI Acceleration

Future improvements will make use of Google's Coral USB Accelerator, NVIDIA TensorRT and Apple's Neural Engine to boost how quickly edge devices perform training. With these accelerators, both the speed and frame rate are improved while the phone uses less power in energy-sensitive mobile applications. The ability of these accelerators to support fast CNNs makes it possible to use edge-based security systems that detect multiple types of threats at the same time.

CHAPTER 5

RESULTS AND DISCUSSION

The study looks closely at CNN-based face mask detection systems that have been developed to improve security on public transport. Results from a variety of sources are used in this analysis to judge whether these systems are suitable for use in public transportation. Experts work to uncover common patterns in how various implementations function and to define usual metrics and they also list main complications that have to be overcome before deployment. The evaluation checks the correctness of the system's performance, along with looking at hardware boundaries and how people react to it and consider its ethical issues. The review tries to identify faults and strengths in face mask detection solutions based on both test scenarios and practical field tests.

5.1 Summary of Key Results

Independent tests of the MobileNetV2-face mask detection model showed an accuracy of 93.7%, indicating it could be adopted reliably in public practice. All three types of masks, "mask," "no mask," and "improper mask," were correctly identified by the model during testing. For "mask," the model performed very well with precision and recall of 0.94 and 0.96, both with high confidence in prediction. For "no mask," recall and precision were nearly equal at 0.90 and 0.91 and the detector continuously detected violations. Although the precision for improper masks was much lower (0.86) and recall lower (0.83), we still consider the outcomes reasonable, given the difficulty in this class.

Class	Precision	F1-Score	Recall
Mask	0.94	0.96	0.95
No Mask	0.91	0.90	0.91
Improper	0.86	0.83	0.84

Table 5.1: Performance metrics seen on the Classification Report

They demonstrate that the model can deal with variations in image size, lighting and head orientation. It was observed during confusion matrix analysis that the model performed reliably, as there was only slight mingling of categories.

5.2 Real-Time Performance on Raspberry Pi

A main aim of the system was to allow for real-in time inference on a low-core device. This was tested by deploying the trained and quantized model on a Raspberry Pi 4 in typical conditions. Real-time monitoring tasks could be carried out with the model, as it processed frames at a rate of 8.5 to 10 frames per second. Thanks to the low latency of around 110 milliseconds per frame, the classification outcomes were always responsive.

Metric	Value
Frame Rate (FPS)	8.5
Inference Latency	110 ms
CPU Usage	~72%
RAM Usage	~1.2 GB

Table 5.2: Reviewing Raspberry Pi Deployment Metrics

The results confirm that the system can be used on site in real conditions. Even with its simple hardware, the system processed each webcam image without missing a frame. In addition, the system was able to control its resource use, proving that it remained stable during continuous use.

5.3 Comparison with Baseline Models

To test MobileNetV2, a comparison was done with VGG16 and EfficientNetB0—two popular CNN architectures. Every model learned from the same set of data and was evaluated using the same equipment. Although VGG16 showed a high accuracy (92.3%), having both a large file size (528 MB) and high data processing time (over 4 seconds per frame) made it a poor fit for embedded software. With 94.1% accuracy and a 250 ms latency, EfficientNetB0 made better use of the resources available to it. However, MobileNetV2 managed to give us the best combination of data and time.

Model	Accuracy	FPS	Size (MB)	Latency
VGG16	92.3%	2	528	4000 ms
EfficientNetB0	94.1%	8	29	250 ms
MobileNetV2	93.7%	10	14	110 ms

Table 5.3: Comparing Models for Deployment on the Edge

MobileNetV2 was found to be the most efficient, giving similar accuracy with only a small amount of memory and processing cost. Therefore, this system fits the area of study, mainly where power, memory and space are limited, just like public transportation.

5.4 Error Analysis

Error analysis brought out some regular errors in the way the network labeled records. Most

of the errors happened in the “improper mask” class, because it is the least clear class. Causes of these errors are sometimes masks placed below the nose, side angles that hide a mask and dim or bright light..

Figure 5.1: Examples where occlusion or low light affected how the model predicted [4]

When we looked at the confusion matrix and the sample images, we found that most of these edge issues were on the dividing line with other classes. If more questionable pictures are added to the dataset and fake images are created, the chance of mislabeling these images will be reduced. Another way to increase reliability in tough cases is to use video frame analysis to add context in time.

5.5 Practical Deployment Observations

The system was tested in buses and trains which gave us a better idea of its functionality. Positions above eye level provided better results for video cameras than locations at the side or sideways. Lighting skewed from inside buses and trains was highly variable during the day, so the model was designed to adjust to major changes in brightness quickly. To improve the quality of frames read by the camera, the images were resized, changed to grayscale and then equalized using histograms in OpenCV.

Every time a camera was triggered, coloured bounding boxes (green for a mask, red for no mask and yellow for an incorrect mask) showed the operators the results immediately. With only minimal action from humans, the overall system handled changing situations without issues. The results show that the model works well even in challenging, real-world circumstances.

5.6 Implications and Applications

This system touches many areas of life in the real world. The system ensures safety, flexibility and affordability for mask control in crowded areas. Since you can build it with inexpensive components, the system is ideal for use on public projects that depend on low-cost solutions. With the system, humans are needed less because it can function hand-free which results in better enforcement efforts.

As well as being used on buses such systems can be put in place at airports, shopping centers, schools and hospitals. It could be added to the city's wider public health surveillance by feeding information into cloud dashboards or central systems. It belongs to the overall purpose of smart cities to improve safety by using automated AI.

5.7 Limitations and Recommendations

Despite the good results both in the lab and during testing, there are limits to the approach. At night or in low light, CCTV feeds become grainy which may cause the system to miss or mistake objects. When faces interact or the subjects in the shot move rapidly, the system has a harder time detecting them. Moreover, motion analysis is broken when the frame size is fixed which could normally assist in making better predictions.

To fix these obstacles, use infrared cameras for seeing in dim light and install multi-face tracking systems at your site. In future, versions could incorporate LSTMs or 3D CNNs as temporal deep learning models, to consider the consistency between frames. Trying to connect the model to more varied and life-like information about people will result in fairer and more accurate outcomes of predictions for everyone.

5.7.1 Low-Light and Lighting Variability:

At night and in conditions with low visibility, the aircraft suffered reduced performance during deployment. A lot of public transport vehicles work when it's dark or in poorly lit areas and RGB cameras can't reliably pick up facial details then. Due to these conditions, the model became less precise and its rate of wrongly labeled improper masks went up. Going forward, cameras that work in darkness or near infrared light could be used to solve this. Adjusting lighting changes in images can also be done with current adaptive image preprocessing methods such as histogram equalization and gamma correction.

5.7.2 Handling Multiple and Overlapping Faces:

Whenever a lot of people gather, as at train stations or in buses, seeing separate faces one behind the other is almost impossible. The model is currently designed to identify a face using each box without relying on the overall scene. Consequently, there are cases of detecting only part of someone's face which keeps the system from performing well. Therefore, the system becomes capable of analyzing events in real-time and matching faces to improve performance with increasing crowd movements.

5.7.3 Ethical and Privacy Considerations :

Two important problems when implementing surveillance-based AI are about data privacy and bias in surveillance. All equipment designed for watching public areas should be clear about how data protection works, no matter if it can recognize faces or not. Future updates in the system should contain both anarch iv on-device data anonymization and GDPR-compliant transparency logs. Teaching people about these systems together with asking for

their permission may result in more people accepting them.

5.7.4 Dataset Diversity and Demographic Bias:

Most things in the dataset work well, yet demographic diversity and edge-cases could be enhanced. Many times, people wearing face masks that looked culturally different (such as through veils or scarves) or people from various age groups experienced errors. It would help to have future datasets with better balance in ethnic, age and real-life masking representation. Instead of simply using human-labeled data, GANs could generate unusual conditions that do not appear very often in real data.

CHAPTER 6

CONCLUSION

A test was done to measure the performance of an advanced face mask detector that uses CNNs for check-up of bus and train passengers using public transport. Most effort went into designing a system that is lightweight, fast and precise for use on Raspberry Pi devices. Creating a model and evaluating the system were the main aims of the project which would help with real-world deployment.

To train the model, we blended public images along with a selection made from our own images. The images in the datasets were separated into three groups: those who wore masks correctly, incorrectly and those who didn't wear masks at all. The researchers decided to use MobileNetV2 since it takes little space and was especially designed for edge computing. The accuracy of the model was 93.7% and it responded to images between 8.5 and 10 frames per second when tested. In vehicles such as buses and trains, the system performs well since low-power hardware is standard.

The system is able to reliably determine if a face mask is being used during various situations, according to testing. During poor lighting or incorrect or partial mask use, the system found it difficult to recognize if someone was wearing a mask. The results show that by increasing the training data and face tracking between video frames, the model can be improved.

6.1 Main Contributions

This work outlines how to use CNN for live Raspberry Pi training in an organized way. It is demonstrated that a practical face mask detection system can be built within budget for use in the real world. Special efforts were made to balance the system's performance with the ability to efficiently deploy it.

All data processing was carried out locally and no one's information was shared or stored. As a result, the system is prepared for public use and satisfies all regulations related to privacy.

6.2 Challenges and Learnings

It functions well in most standard conditions, but has trouble working in dim quantity conditions and during very busy situations. By adding infrared cameras, better training data and more intelligent models, the system can overcome its existing issues.

The work gave key observations on how data needs to be organized and on how model quality depends on how large the model is. The performance of MobileNetV2 models is impressive, even though the processing needed is not complicated.

6.3 Final Thoughts

Using a face mask detection system will enable public health officials to quickly monitor if everyone is using a mask while traveling on public transport. Installing a solar power system uses few resources, remains cost-effective and provides top performance. With a suitable programming change, the system can be used to supervise both seatbelt compliance and people in crowds.

Researchers show that AI is used safely and responsibly by health and safety authorities in their study.

REFERENCES

[1] Kanavos, A., Papadimitriou, O., Al-Hussaeni, K., Maragoudakis, M., & Karamitsos, I. (2024). Real-Time Detection of Face Mask Usage Using Convolutional Neural Networks. Computers, 13(7), 182.

[2] Goyal, H., Sidana, K., Singh, C., Jain, A., & Jindal, S. (2022). A real time face mask detec- tion system using convolutional neural network. Multimedia Tools and Applications, 81(11), 14999-15015.

[3] Pandey, B. K., Pandey, D., & Lelisho, M. E. (2024). Face mask identification with enhanced cuckoo optimization and deep learning-based faster regional neural network. Scientific Re- ports, 14(1), 29719.

[4] Sheikh, B. U. H., & Zafar, A. (2023). RRFMDS: rapid real-time face mask detection system for effective COVID-19 monitoring. SN Computer Science, 4(3), 288.

[5] Chinnaiyan, R., Sai, K., & Bharath, P. (2023). Deep learning based CNN model for classi- fication and detection of individuals wearing face mask. arXiv preprint arXiv:2311.10408.

[6] Gogou, I. C., & Koutsomitropoulos, D. A. (2022, August). A Review and Implementation of Object Detection Models and Optimizations for Real-time Medical Mask Detection dur- ing the COVID-19 Pandemic. In 2022 International Conference on INnovations in Intelli- gent SysTems and Applications (INISTA) (pp. 1-6). IEEE.

[7] Boulila, W., Alzahem, A., Almoudi, A., Afifi, M., Alturki, I., & Driss, M. (2021, December). A deep learning-based approach for real-time facemask detection. In 2021 20th IEEE Inter- national Conference on Machine Learning and Applications (ICMLA) (pp. 1478-1481). IEEE.

[8] Chavda, A., Dsouza, J., Badgujar, S., & Damani, A. (2021, April). Multi-stage CNN archi- tecture for face mask detection. In 2021 6th International Conference for Convergence in Technology (i2ct) (pp. 1-8). IEEE.

[9] Joshi, D., Sharma, A., Pingale, S., Mal, C., Malviya, S., & Patil, N. (2021, September). Face Mask Detection Using Optimized CNN. In 2021 International Conference on Artificial In- telligence and Machine Vision (AIMV) (pp. 1-6). IEEE.

[10] Shah, R. C., & Shah, R. J. (2019). Detection of face mask using convolutional

neural net- work. Mobile Information System, 43, 382-487.

[11] Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., Raj, R., ... & Rakhra, M. (2022). Face mask recognition system using CNN model. Neuroscience Informatics, 2(3), 100035.

[12] Kontellis, E., Troussas, C., Krouska, A., & Sgouropoulou, C. (2021). Real-time face mask detector using convolutional neural networks amidst COVID-19 pandemic. In Novelties in Intelligent Digital Systems (pp. 247-255). IOS Press.

[13] Wang, Z., Liu, Q., Du, Y., et al. (2020). 'Masked Face Recognition Dataset and Application'. IEEE Transactions on Multimedia.

[14] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). 'Detecting masked faces in the wild with LLE-CNNs'. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). 'A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic'. Measurement.

[16] Jiang, M., Fan, X., & Yan, H. (2021). 'RetinaFaceMask: A Single Stage Face Mask Detector for Assisting Control of the COVID-19 Pandemic'. arXiv preprint arXiv:2005.03950.

[17] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). 'Joint Face Detection and Alignment using Multitask Cascaded Convolutional Networks'. IEEE Signal Processing Letters.

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). 'MobileNetV2: Inverted Residuals and Linear Bottlenecks'. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).