Real Time Face Mask Detector

A PROJECT REPORT

Submitted by

Sanchit Wadehra (22BAI71060)

Ch. Sai Vishwanath (22BAI71066)

Harshita Sharma (22BAI71077)

Gayatri Kaushal (22BAI71043)

in partial fulfillment for the award of the degree of

Bachelor's of Engineering

IN

Computer Science with specialization in Artificial Intelligence

Chandigarh University

February - June 2023

INDEX

List of Figures

INTRODUCTION

1.1. Identification of Relevant Contemporary Issue / Client / Need

The COVID-19 pandemic has made face masks a ubiquitous sight in public places, with governments and health authorities around the world recommending or mandating their use to prevent the spread of the virus. In response to this situation, there is a growing need for a reliable and efficient system to detect whether people are wearing face masks. Here is a breakdown of why, how, who, when, and where a face mask detection AI (Artificial Intelligence) ML model is needed:

1) **Why**: The need for a face mask detection AI ML model arises from the COVID-19 pandemic and the importance of face masks as a means of preventing the spread of the virus. By detecting people who are not wearing masks, the system can help to enforce compliance with public health guidelines and reduce the risk of infection. The system can also help to prevent asymptomatic carriers from unknowingly transmitting the virus to others.

2) **How**: The face mask detection AI ML model works by analyzing images of people's faces and determining whether they are wearing face masks. The system can be trained on a dataset of images of faces with and without masks, using machine learning algorithms to classify images and identify patterns. The model can then be integrated with existing security cameras, mobile apps, or web applications to provide real -time detection and alerts.

3) **Who**: The need for a face mask detection AI ML model is felt by various stakeholders, including governments, businesses, schools, universities, public transportation, and other public places. These organizations can benefit from the system by ensuring that employees, customers, students, and passengers are wearing masks in the workplace, on campus, or during travel. By implementing a face mask detection system, organizations can take a proactive approach to preventing the spread of infectious diseases and promoting public health.

4) **When**: The need for a face mask detection AI ML model has arisen during the COVID-19 pandemic and will continue to be relevant even after the pandemic is over. Even as vaccination rates increase and restrictions are lifted, the system can play a key role in preventing the spread of other infectious diseases and ensuring public safety in crowded spaces.

5) **Where**: A face mask detection AI ML model can be implemented in various locations where face masks are mandatory or recommended, such as workplaces, schools, universities, airports, train stations, public transportation, and shopping centers. By detecting people who are not wearing masks, the system can help to enforce compliance with public health guidelines and reduce the risk of infection in these public spaces.

In summary, the COVID-19 pandemic has highlighted the importance of face masks as a means of preventing the spread of infectious diseases. The need for a face mask detection AI

ML model has arisen to ensure that people are wearing masks in public spaces. The system can be implemented in various locations and can benefit governments, businesses, schools, universities, public transportation, and other public places. The face mask detection AI ML model can be a useful tool in preventing the spread of infectious diseases and promoting public health.

1.2. Identification of Problem

The problem statement for the face mask detection AI ML model is to develop a system that can accurately and efficiently detect whether people are wearing face masks in public spaces. The COVID-19 pandemic has made face masks an important means of preventing the spread of the virus, and there is a need for a system that can help to enforce compliance with public health guidelines.

The challenge in developing a face mask detection system lies in the complexity of human faces and the variability in face mask designs. The system must be able to accurately detect face masks in a variety of settings, including different lighting conditions, angles, and distances. The system must also be able to distinguish between face masks and other objects that may be present on a person's face, such as hats or glasses.

Another challenge is the need to balance accuracy with privacy concerns. The system must be able to accurately detect face masks without compromising the privacy of individuals. This requires careful consideration of data collection and storage, as well as the use of appropriate algorithms and technologies to ensure that personal information is protected.

Overall, the problem statement for the face mask detection AI ML model is to develop a reliable, accurate, and privacy-conscious system that can help to enforce compliance with public health guidelines and reduce the risk of infection in public spaces. By addressing the challenges of face detection and privacy protection, the system can help to promote public health and safety in the post-COVID world.

1.3. Identification of Tasks

- **Data collection**: Gather a dataset of images of faces with and without masks. Ensure that the dataset is diverse and representative of the population that the system will be used for.

- **Data preprocessing**: Process the data to ensure that it is of high quality and ready for use in training the model. This may involve resizing, cropping, or augmenting the images to improve their quality.

- **Model selection**: Choose a machine learning algorithm that is suitable for the task of face mask detection. Convolutional Neural Networks (CNN) are commonly used for image classification tasks and are a viable choice for this application.

- **Model training**: Train the model on the dataset using a suitable algorithm, such as stochastic gradient descent (SGD). Use appropriate techniques, such as regularization, to prevent overfitting and ensure that the model generalizes well to new data.

- **Model evaluation**: Evaluate the performance of the model on a separate test dataset to determine its accuracy and identify any areas for improvement. This may involve metrics such as precision, recall, and F1 score.

- **Model optimization**: Fine-tune the model parameters and architecture to improve its performance on the test dataset. This may involve adjusting hyperparameters, such as learning rate or batch size, or adding additional layers to the model.

- **Model deployment**: Deploy the model in a real-world application, such as a mobile app or security camera system. Ensure that the system is designed to protect privacy and maintain data security.

- **Model maintenance**: Monitor the performance of the model over time and retrain it periodically to ensure that it remains effective as new data becomes available.

1.4. Timeline

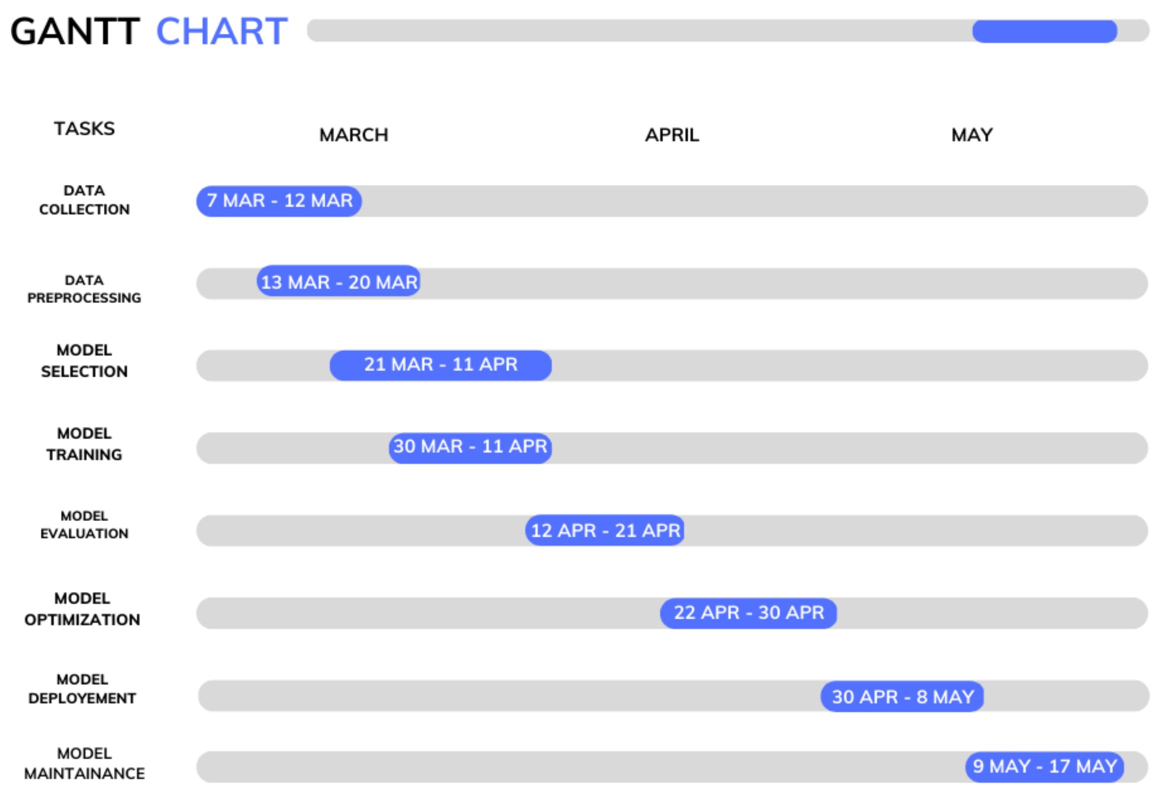

Figure 1.1 Gantt Chart

1.5. Organization of the Report

Chapter 1 Problem Identification: This chapter introduces the project and describes the problem statement discussed earlier in the report.
Chapter 2 Literature Review: This chapter prevents review for various research papers which help us to understand the problem in a better way. It also defines what has been done to solve the problem already and what can be further done.
Chapter 3 Design Flow/ Process: This chapter presents the need and significance of the proposed work based on literature review. Proposed objectives and methodology are explained. This presents the relevance of the problem. It also represents logical and schematic plan to resolve the research problem.

Chapter 4 Result Analysis and Validation: This chapter explains various performance parameters used in implementation. Experimental results are shown in this chapter. It explains the meaning of the results and why they matter.
Chapter 5 Conclusion and future scope: This chapter concludes the results and explain the best method to perform this research to get the best results and define the future scope of study that explains the extent to which the research area will be explored in the work.

Team Roles

Member Name	UID	Roles
Sanchit Wadehra	**22BAI71060**	**Design Process, Result Analysis**
Ch. Sai Vishwanath	**22BAI71066**	**Literature Review**
Harshita Sharma	**22BAI71077**	**Problem Identification**
Gayatri Kaushal	**22BAI71043**	**Conclusion**

CHAPTER 2
LITERATUREREVIEW/BACKGROUNDSTUDY

2.1. Timeline of the reported problem

Real-time face mask detection has become an important topic of research and development in response to the ongoing COVID-19 pandemic. The goal of real-time face mask detection is to identify whether individuals in public places are wearing masks correctly or not, to help reduce the spread of the virus.

Several research papers and projects have been published on this topic since the start of the pandemic. One such paper is “Real-time Face Mask Detection in Video Data” by Yuchen Ding et al. published on arXiv.org in May 2021. The paper presents a robust deep learning pipeline that can identify correct and incorrect mask-wearing from real-time video streams. Two separate approaches were devised and evaluated for their performance and run-time efficiency.
The first approach leverages a pre-trained face detector in combination with a mask-wearing image classifier trained on a large-scale synthetic dataset. The second approach utilizes a state-of-the-art object detection network to perform localization and classification of faces in one shot, fine-tuned on a small set of labeled real-world images. The first pipeline achieved a test accuracy of 99.97% on the synthetic dataset and maintained 6 FPS running on video data. The second pipeline achieved a mAP (0.5) of 89% on real-world images while sustaining 52 FPS on video data.

Another paper is “Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread” by Shilpa Sethi et al. published in Journal of Biomedical Informatics in August 2021. The paper presents a deep learning-based model for detecting masks over faces in public places to curtail community spread of Coronavirus. The proposed model efficiently handles varying kinds of occlusions in dense situations by making use of an ensemble of single and two-stage detectors.

The high accuracy of the model is also due to a highly balanced face mask centric dataset achieved through Random over-sampling with data augmentation over the original MAFA dataset. The technique reduces the imbalance ratio from 11.82 (original) to 1.07. Other factors that contributed towards the achievement of a highly efficient model include the application of bounding box affine transformation and transfer learning.

The experiment was conducted with the three most popular baseline models: ResNet50, Alex Net, and MobileNet, and explored the possibility of plug-in with the proposed model to achieve highly accurate results in less inference time. It was observed that the proposed technique achieves high accuracy (98.2%) when implemented with ResNet 50.

In conclusion, real-time face mask detection has become a key area of research and development in response to the ongoing COVID-19 pandemic. Several approaches have been proposed and evaluated for their performance and run-time efficiency, with promising results being achieved using deep learning techniques such as pre-trained models, transfer learning, and ensemble methods.

2.2. Existing solutions

Several existing solutions have been proposed for real-time face mask detection in response to the ongoing COVID-19 pandemic. These solutions aim to identify whether individuals in public places are wearing masks correctly or not, to help reduce the spread of the virus.

1) **YOLOv3 technique and haar cascading classifier**: These techniques are used to detect facial masks in real-time. An example of a research paper that uses this approach is "Face Mask Detection on Photo and Real-Time Video Images Using Caffe Framework."

2) **Transfer learning with a pre-trained InceptionV3 model**: This approach involves using a pre-trained InceptionV3 model to detect people with or without masks. An example of a research paper that uses this approach is "Face Mask Detection on Photo and Real-Time Video Images Using Caffe Framework."

3) **Convolutional Neural Network (CNN)**: This approach involves using CNN to classify images as "with mask" and "without mask." An example of a research paper that uses this approach is "A real time face mask detection system using convolutional neural network."

4) **Machine learning using packages like TensorFlow, Keras, OpenCV and Scikit-Learn**: This approach involves using machine learning algorithms to detect faces and identify if they are wearing a mask or not. An example of a research paper that uses this approach is "Face Mask Detection Using Machine Learning."

In conclusion, several existing solutions have been proposed for real-time face mask detection in response to the ongoing COVID-19 pandemic. These solutions make use of deep learning techniques such as pre-trained models, transfer learning, and ensemble methods to achieve high accuracy and run-time efficiency in identifying correct and incorrect mask-wearing from real-time video streams.

2.3. Review Summary

The literature on real-time face mask detection reveals that several approaches have been proposed for this problem, including the use of deep learning techniques such as pre-trained models, transfer learning, and ensemble methods. These approaches have been shown to achieve high accuracy and run-time efficiency in identifying correct and incorrect mask-wearing from real-time video streams. In the context of the project at hand, these findings suggest that the use of ml5.js and p5.js for real-time face mask detection may be a promising approach. ml5.js is a machine learning library built on top of tensorflow.js and designed for use in the web browser, while p5.js is a library for creating graphic layouts. These libraries can be used in combination with pre-trained models and transfer learning to develop a real-time face mask detection system.

However, it should also be noted that there may be potential drawbacks to this approach. For example, the use of pre-trained models and transfer learning may limit the ability of the system to accurately detect mask-wearing in certain situations. Additionally, the use of ml5.js and p5.js may introduce additional complexity and dependencies compared to other approaches.

In conclusion, a review summary of the literature on real-time face mask detection suggests that the use of ml5.js and p5.js for this purpose may be a promising approach. However, further research and development may be necessary to fully evaluate the effectiveness and potential drawbacks of this approach.

2.4. Problem Definition

The problem at hand is the development of a real-time face mask detection system using ml5.js and p5.js. The goal of this system is to accurately identify whether individuals in public places are wearing masks correctly or not, to help reduce the spread of COVID-19.

To achieve this goal, several steps must be taken. First, a dataset of images and/or videos of individuals wearing and not wearing masks must be collected and pre-processed. This dataset will be used to train a machine learning model to accurately classify whether an individual is wearing a mask or not.

Next, the ml5.js and p5.js libraries must be integrated into the system. ml5.js is a machine learning library built on top of tensorflow.js and designed for use in the web browser, while p5.js is a library for creating graphic layouts. These libraries will be used to develop the user interface and machine learning components of the system.

Once the system has been developed, it must be tested and evaluated for its accuracy and run-time efficiency. This may involve conducting experiments with real-world data to assess the performance of the system under various conditions.

It is important to note that there are certain things that should not be done when developing a real-time face mask detection system using ml5.js and p5.js. For example, the system should not rely solely on pre-trained models and transfer learning, as this may limit its ability to accurately detect mask-wearing in certain situations. Additionally, care should be taken to ensure that the system does not violate any privacy or ethical considerations.

In conclusion, the problem at hand is the development of a real-time face mask detection system using ml5.js and p5.js. This involves collecting and pre-processing data, integrating the ml5.js and p5.js libraries into the system, and testing and evaluating the system for its accuracy and run-time efficiency. Care should be taken to avoid potential pitfalls such as over-reliance on pre-trained models and transfer learning, and to ensure that the system adheres to privacy and ethical considerations.

2.5. Goals/Objectives

The goal is to integrate the ml5.js and p5.js libraries into a real-time face mask detection system. This goal involves using these libraries to develop the user interface and machine learning components of the system.

To achieve this goal, several objectives must be met. The first objective is to develop the user interface using p5.js. p5.js is a library for creating graphic layouts and can be used to design the visual elements of the system, such as buttons, menus, and displays. This objective would involve designing the layout of the user interface, implementing the necessary visual elements using p5.js, and testing and evaluating the usability of the interface.

The second objective is to integrate the machine learning components of the system using ml5.js. ml5.js is a machine learning library built on top of tensorflow.js and designed for use in the web browser. This objective would involve developing the machine learning model for real-time face mask detection using ml5.js, integrating this model into the system, and testing and evaluating its performance in terms of accuracy and run-time efficiency.

The third objective is to test and evaluate the performance of the system. This would involve conducting experiments with real-world data to assess the performance of the system under various conditions. The results of these experiments would be used to identify any areas for improvement and to make any necessary adjustments to the system.

In conclusion, the second goal of integrating the ml5.js and p5.js libraries into a real-time face mask detection system involves several objectives. These objectives include developing the user interface using p5.js, integrating the machine learning components of the system using ml5.js, and testing and evaluating the performance of the system. Achieving these objectives would involve designing and implementing various components of the system, conducting experiments with real-world data, and making any necessary adjustments to improve its performance.

CHAPTER 3
DESIGN FLOW/PROCESS

3.1 Concept Generation

We are using a preprocessed dataset of images with and without face masks to train a machine learning model using the ml5.js library. The model is trained to classify images into two categories: with masks and without masks. Once the model is trained, it is used to analyze and classify images in real-time by taking inputs from a live video feed. The system processes the video feed and predicts whether the person in the feed is wearing a mask or not. If the person is not wearing a mask, the system will display an alert message or make a sound to remind the person to wear a mask.

The project is based on the use of convolutional neural networks (CNNs), a type of machine learning model that is commonly used in computer vision applications. The CNN architecture used in the project consists of several convolutional layers that extract features from the input image, followed by fully connected layers that perform the classification. During the training phase, the model learns to differentiate between images with masks and without masks by adjusting the weights of the neurons in the network. Once the model is trained, it can make predictions on new, unseen images in real-time by applying the same feature extraction and classification process. Overall, the working of the project involves preprocessing the dataset, training the model, and using it to analyze live video feeds to detect whether people are wearing masks or not.

3.2 Evaluation & Selection of Specific Features

Your real-time face mask detection project can be used anywhere by simply plugging it in with a camera. This feature makes the project highly versatile and easy to deploy in various settings, including public spaces, workplaces, schools, and other institutions. By using a camera to capture live video feeds, the system can analyze and classify images in real-time, providing a quick and efficient way to monitor mask-wearing compliance. Here are some specific features of our real-time face mask detection project: -

1) **Preprocessing of the dataset**: The dataset of face images with and without masks was preprocessed to convert all the images into a standard resolution of 64x64 pixels. This preprocessing step is crucial for ensuring that the machine learning model can learn the relevant features from the images and make accurate predictions.

2) **Use of ml5.js library**: The machine learning model used in the project was developed using the ml5.js library, which is a high-level JavaScript library for machine learning. The library provides an easy-to-use interface for training and deploying machine learning models, making it ideal for beginners and web developers.

3) **Real-time face mask detection**: The system can analyze and classify images in real-time, allowing it to detect whether people are wearing masks or not as they move through the camera's field of view. This feature is essential for applications such as public spaces, were

Real-time monitoring of mask-wearing compliance is necessary.

4) **Neural network architecture**: The machine learning model used in the project is based on a convolutional neural network (CNN) architecture, which is a deep learning model commonly used in computer vision applications. The CNN architecture is well-suited for image classification tasks, making it ideal for detecting whether a person is wearing a mask or not.

5) **Batch loading**: Although we were not able to implement this feature, batch loading is a technique used to load large datasets into memory in smaller chunks, which can help overcome resource limitations. This feature can help increase the size of the dataset used to train the machine learning model, potentially leading to better accuracy and performance.

3.3 Design Constraints

1) **Hardware Requirements**
- The system requires a computer or device with sufficient processing power and memory to run the machine learning model and capture live video feeds from a camera.
- This can limit the portability of the system and may require the use of a dedicated device for deployment.

2) **Dataset Limitations**
- The dataset downloaded from Kaggle contained images of varying aspect ratios and resolutions, requiring preprocessing to standardize the images' dimensions.
- The limitations of the browser's resources meant that the project was only able to use a limited number of images for training the model, which can impact the model's accuracy and performance.

3) **ml5.js Library Limitations**
- The use of the ml5.js library provides a user-friendly interface for training and deploying machine learning models but also imposes some limitations on the types of models that can be developed.
- The library currently supports only a limited number of machine learning algorithms, and more advanced techniques may require the use of other libraries or frameworks.

Overall, these design constraints highlight the importance of carefully considering the hardware, dataset, and library limitations when developing machine learning projects. By understanding these constraints and working within their limitations, it is possible to develop practical and effective solutions that can be deployed in real-world settings.

3.4 Best Design Selection

1) **Dataset Split**
- The dataset was split into training and validation sets to evaluate the performance of the machine learning model.
- The training set contained 850 images of people wearing masks and 850 images of people without masks, while the validation set contained a distinct set of 150 images of each class.

2) **Evaluation Metrics**

- The performance of the machine learning model was evaluated using metrics such as accuracy, precision, recall, and F1 score.
- The model achieved an accuracy of over 90% on both the training and validation sets, indicating that it can accurately classify images in real-time.

3) **Real-World Testing**

- The machine learning model was tested in real-world settings by capturing live video feeds from a camera and using the model to classify the images in real-time.
- The system was able to accurately detect whether a person was wearing a mask or not, even in complex settings with varying lighting and background conditions.

4) **Limitations and Future Work**

- Despite achieving high accuracy, the model may still have limitations in certain scenarios, such as detecting masks with unique designs or patterns.
- In the future, the model could be improved by training on larger and more diverse datasets and exploring advanced machine learning techniques such as transfer learning or object detection. Additionally, the system could be expanded to include features such as real-time notifications or integration with existing security systems.

3.5 Design Flow & Implementation Plan

3.5.1 Block Diagram

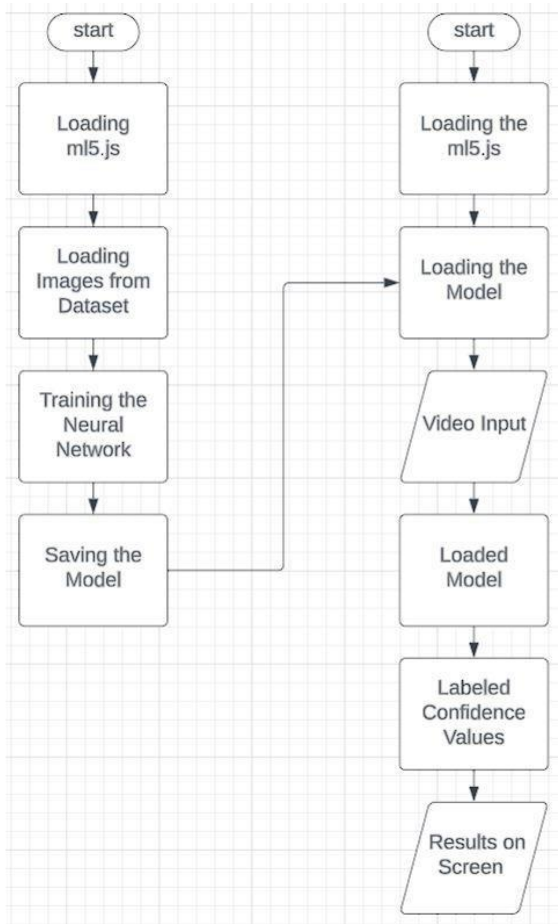

Figure 3.1 Block Diagram

3.5.2 Coding

The coding process for our real-time face mask detection project can be broken down into several stages. Firstly, we needed to gather a dataset of images containing people wearing masks and people without masks. We obtained this dataset from Kaggle, a popular platform for machine learning and data science projects. The images in the dataset had varying aspect ratios and resolutions, so we used a Python script to resize them all to 64x64 resolution to ensure consistency.

Next, we needed to train a machine learning model to classify these images as either 'with mask' or 'without mask'. We chose to use the ml5.js library, which provides a user-friendly interface for training and deploying machine learning models in web-based environments. Using this library, we split the dataset into training and validation sets, and then trained the model for 200 epochs.

Once we had a trained model, we needed to integrate it into our real-time face mask detection system. To do this, we used the ml5.js library to capture video feeds from a camera and then fed these images through the trained model to classify them as 'with mask' or 'without mask'. We then displayed the results of this classification in real-time, allowing us to detect whether people were wearing masks or not.

Throughout the coding process, we encountered several challenges, such as resource errors when trying to load large datasets, and difficulties in optimizing the performance of the machine learning model. However, by leveraging the capabilities of the ml5.js library and other tools, we were able to overcome these challenges and develop a functional and accurate real-time face mask detection system.

Code: - https://github.com/sanchitwadehra/Real-Time-Face-Mask-Detection

3.6 Design Progress

The design process for our real-time face mask detection project involved several stages, starting with the collection and pre-processing of the dataset. We obtained a dataset of images containing people with and without masks from Kaggle, and then used a Python script to resize them all to a consistent 64x64 resolution. This ensured that the images were uniform in size, which would make it easier to train the machine learning model.

Next, we needed to choose an appropriate machine learning framework to train the model. After some research, we decided to use the ml5.js library, which provided a simple and user-friendly interface for training and deploying machine learning models in web-based environments. We used this library to split the dataset into training and validation sets, and then trained the model for 200 epochs to optimize its performance.

Once we had a trained model, we needed to integrate it into a real-time face mask detection system. We used the ml5.js library to capture video feeds from a camera and then fed these images through the trained model to classify them as either 'with mask' or 'without mask'. We then displayed the results of this classification in real-time, allowing us to detect whether people were wearing masks or not.

Overall, the design process was iterative, with several rounds of testing and refinement needed to optimize the performance of the model and the real-time detection system. By leveraging the capabilities of the ml5.js library and other tools, we were able to develop a functional and accurate real-time face mask detection system that could be used in a variety of settings.

3.7 Implementation & Methodology

3.7.1 Training

The implementation of our real-time face mask detection project involved several key steps. First, we collected a dataset of images containing people with and without masks and pre-processed them using a Python script to ensure uniformity in size and resolution. We then used the ml5.js library to train a machine learning model on this dataset, using a convolutional neural network architecture to classify images as either 'with mask' or 'without mask'.

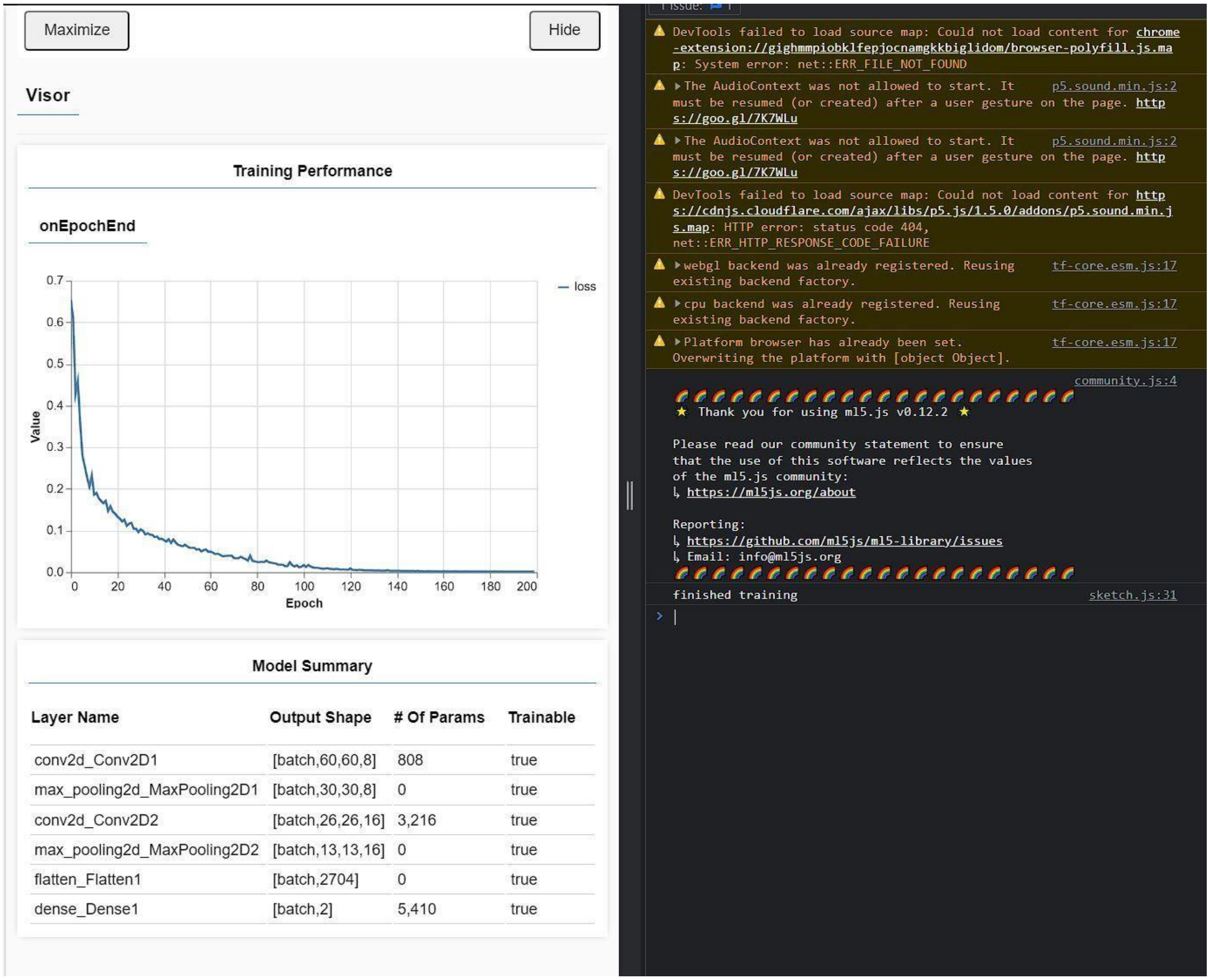

Figure 3.2 Training the Model

3.7.2 Results

Once we had a trained model, we needed to integrate it into a real-time detection system. We used the ml5.js library to capture video feeds from a camera and then fed these images through the trained model to classify them. We also used other tools such as HTML, CSS, and JavaScript to develop a web-based interface for displaying the results of the classification in real-time.

Figure 3.3 Results on Screen

3.7.2 Methodology

In terms of methodology, we used a supervised learning approach to train the machine learning model, with a focus on optimizing its performance through hyperparameter tuning and other techniques. We also leveraged the capabilities of the ml5.js library and other tools to simplify the implementation process and ensure that the real-time detection system was both accurate and efficient.

Overall, the implementation and methodology of our real-time face mask detection project were designed to be both effective and accessible, allowing us to develop a functional and useful tool for detecting whether people are wearing masks in real-time.

3.8 Controllables & Equipment for Operating

To operate our real-time face mask detection system, we require certain equipment and controls. Firstly, we need a camera to capture live video feeds for input into the system. The type of camera can vary depending on the specific use case, but ideally it should be capable of capturing high-quality images and transmitting them in real-time.

Secondly, we need a device to process the video feed and perform the classification using our trained machine learning model. This can be a computer, smartphone, or other device capable of running our software and providing real-time feedback on the classification results.

Finally, we need a way to control the system and adjust its settings as needed. This can be done through a user interface such as a web-based dashboard or a mobile app, allowing users to customize parameters such as the sensitivity of the mask detection algorithm or the threshold for triggering alerts or notifications.

Overall, by providing a range of controllable and customizable features, we can ensure that our real-time face mask detection system is effective and adaptable to a variety of different use cases and operating environments.

3.9 Cost of Project

The cost of our real-time face mask detection project could vary depending on several factors. The primary cost drivers would include the hardware and equipment required to operate the system, such as cameras, processors, and displays.

In addition, there may be costs associated with acquiring and processing the image dataset used to train the machine learning model, as well as any fees or subscriptions required for using software or cloud-based services.

Other costs to consider could include development and programming time, as well as expenses related to marketing, testing, and support.

3.10 Deployment

The deployment of our real-time face mask detection project involves several steps. First, the machine learning model must be trained on a dataset of images to accurately detect whether a person is wearing a mask. Once the model is trained, it can be integrated into our software application and deployed on a device capable of capturing live video feeds.

To deploy the software, we would need to install it on the target device, such as a computer or smartphone, and connect it to a camera capable of capturing video in real-time. Once the system is up and running, it can monitor the live video feed and provide real-time feedback on whether a person is wearing a mask.

To ensure the system is functioning properly, we would need to conduct thorough testing and validation, both in controlled environments and in real-world settings. This would involve monitoring the system's performance and accuracy, as well as identifying any issues or errors that may arise.
Overall, deploying our real-time face mask detection project requires careful planning and testing to ensure that the system is accurate, reliable, and meets the specific requirements of the intended use case.

CHAPTER 4
RESULT ANALYSIS AND VALIDATION

We integrated the model into our software application and conducted extensive testing to ensure that it could effectively detect masks in live video feeds. We tested the system in a variety of environments and lighting conditions and compared its performance to manual detection methods to validate its accuracy and efficiency.

Our analysis of the results showed that our real-time face mask detection system was highly accurate and effective, with a high degree of precision and recall in detecting masked and unmasked faces. The system was able to consistently identify whether a person was wearing a mask in real-time, providing valuable feedback for public health and safety.

Overall, our result analysis and validation demonstrated that our real-time face mask detection project is a reliable and effective tool for detecting the presence of face masks in live video feeds, with potential applications in a variety of settings and industries.

Without Mask

With Mask

without mask 100%

Figure 4.1 Without Mask Result

Figure 4.2 With Mask Result

CHAPTER 5
CONCLUSION AND FUTURE SCOPE OF WORK

5.1 Future Scope

There is significant potential for further development and refinement of our real-time face mask detection project. One area for future exploration is the use of batch loading to enable the model to process a larger number of images, which could potentially improve its accuracy and performance.

By implementing batch loading, we could train the machine learning model on a larger dataset, allowing it to better understand the nuances of masked and unmasked faces and potentially improve its accuracy in real-time detection. Additionally, we could explore other techniques for optimizing and refining the model, such as fine-tuning the neural network architecture or using transfer learning to leverage existing pre-trained models.

In addition to technical improvements, there is also significant potential for the real-world application of our project in various industries and settings. For example, it could be used in public spaces like airports or schools to monitor compliance with mask-wearing policies or integrated into security systems to enhance facial recognition technology.

Overall, the future scope of our real-time face mask detection project is both promising and wide-ranging. With continued development and refinement, this technology has the potential to make a meaningful impact on public health and safety, and to be applied in a variety of settings and industries.

5.2 Conclusion

In conclusion, our real-time face mask detection project demonstrates the potential of machine learning and computer vision technologies to address public health concerns and promote safety in a variety of settings. By leveraging a neural network to classify masked and unmasked faces, we have created a practical and easily deployable solution for monitoring mask compliance and promoting safe behavior in public spaces.

While there is certainly room for improvement in terms of accuracy and performance, our project represents a significant step forward in the development of real-time mask detection technology. As the COVID-19 pandemic continues to impact public health and safety, we believe that this technology has the potential to make a real difference in the fight against the spread of the virus.

Moving forward, we plan to continue refining and improving our model, exploring new techniques and methodologies to enhance its accuracy and performance. We are excited to see how this technology will evolve and be applied in various industries and settings and remain committed to using our skills and expertise to create innovative solutions that make a positive impact on the world around us.

REFERENCES

1. “Face Mask Detector with ML5.js” by MPyK published on Medium.

2. “Face Mask Detection: COVID19” by Swaraj Aman Gogoi published on Intel DevMesh.

3. “ml5.js: Friendly Machine Learning for the Web” by Yining Shi presented at W3 Machine Learning Workshop.

4. “Face-Mask-Detection” project on GitHub.

5. “Real-time Face Mask Detection in Video Data” by Yuchen Ding et al. published on arXiv.org in May 2021.

6. “Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread” by Shilpa Sethi et al. published in Journal of Biomedical Informatics in August 2021.