Face Mask Detection

Presenting the incredible team:
Mohammed Ashraf,
Mohammed Atef,
Mohammed Wageh, and
Yassen Abd El-Mohsen

Problem Statement

Our project focuses on tackling the challenge of face mask detection by employing Convolutional Neural Networks (CNNs). The goal is to automatically determine whether a person in an image is wearing a face mask. This issue has become increasingly important during global health crises, as manually monitoring mask compliance can be difficult, inefficient, and susceptible to errors, particularly in crowded environments.

Motivation & Importance

Manual supervision requires human effort, time, and continuous attention, which is not scalable.

An automated computer vision system can provide faster, more consistent, and more reliable results.

Such systems can support public health monitoring, reduce human workload, and improve safety in high-traffic areas.

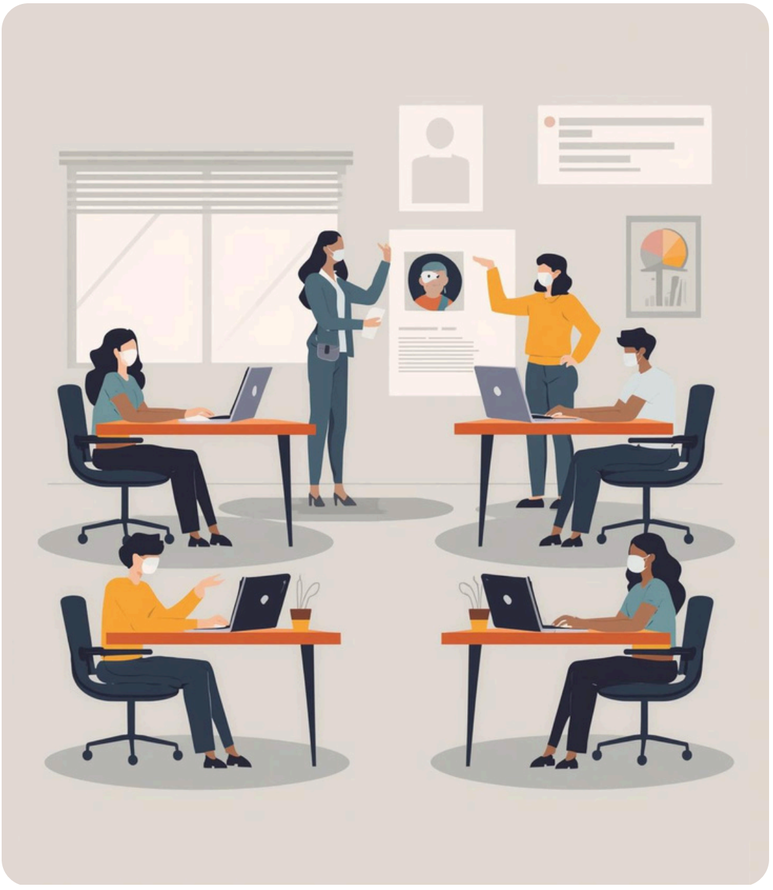
Applications

The proposed system can be applied in many real-world scenarios, such as:

- Airports and transportation hubs
- Schools and universities
- Shopping malls and workplaces
- Surveillance and access control systems

It can also be integrated with existing CCTV cameras or edge devices.

Real-World Applications of Face Mask Detection



Dataset Overview ,Data Split and Preprocessing

“I was responsible for loading and preparing a publicly available face mask dataset with two classes: With Mask and Without Mask. The dataset includes images taken under various conditions and angles.

Data Split:

- 70% for training
- 15% for validation
- 15% for testing

This ensures effective model training while retaining unseen data for evaluation.

Preprocessing:

Images resized to 128×128 pixels for consistency and reduced computational cost.
Pixel values normalized to the range of 0 to 1 to enhance training stability and convergence.

Core Techniques in Face Mask Detection

Data Augmentation

We enhanced the diversity of our dataset by implementing techniques such as rotation, zooming, flipping, and adjustments to brightness. These modifications enable the model to learn with greater reliability and robustness.

Model Architecture, Regularization, Output Layer

CNN Model Design

A straightforward model featuring three layers that employs ReLU and MaxPooling to identify essential features.

Regularization

To mitigate overfitting, we incorporated Dropout layers before the fully connected layers. This approach helps ensure that the model does not become overly dependent on specific neurons during training.

Output Layer

The final layer consists of a Softmax layer with two outputs, representing the two classes:

- With Mask
- Without Mask

This structure enables the model to generate class probabilities.

Training Process,Loss Function,Early Stopping

Implementation and Model Details

Training Pipeline

I implemented the training pipeline in `train.py`. The model was trained using the Adam optimizer, known for its efficiency and popularity in deep learning tasks.

Loss Function

We utilized Sparse Categorical Crossentropy as the loss function, as we are conducting binary classification with integer labels.

Hyperparameters

“The primary hyperparameters included:

- Batch size 32
- Maximum epochs 25

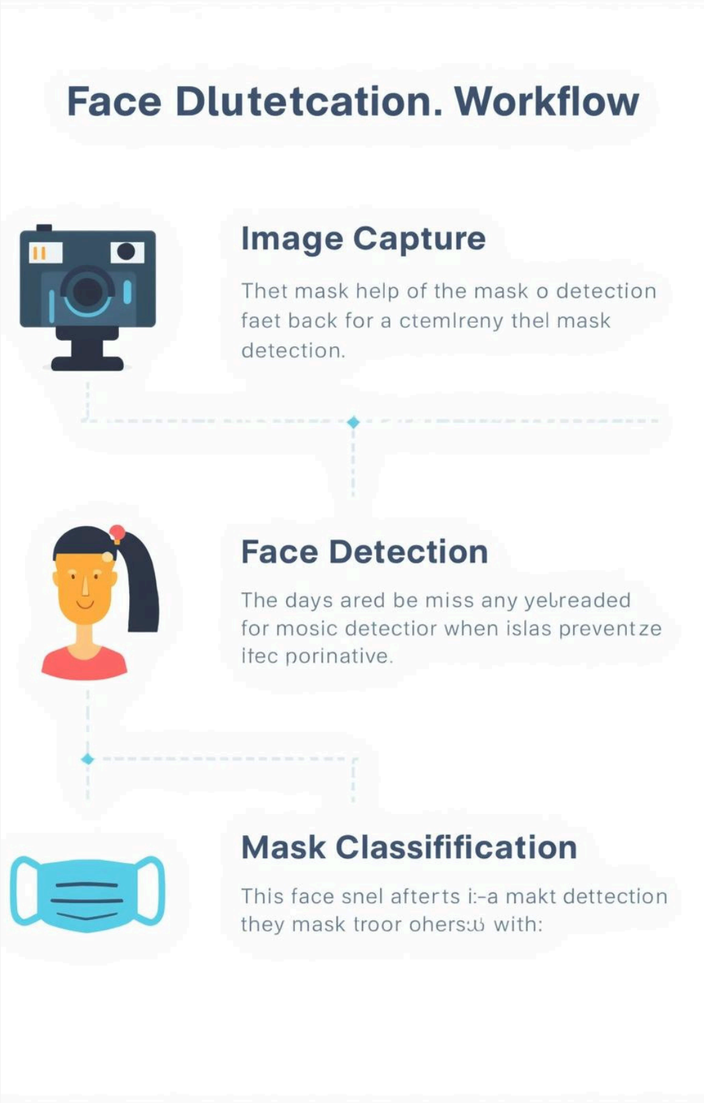
These settings strike a balance between training duration and performance.”

Early Stopping

To prevent overfitting, we incorporated Early Stopping. Training automatically halts when the validation loss ceases to improve, and the optimal model weights are restored.

Model Saving

The best-performing model is automatically saved using ModelCheckpoint in the `saved_model` directory as `best_model.h5`.



Evaluation & Results

Evaluation Metrics

I handled the evaluation phase in evaluate.py.
We evaluated the model using:

- Accuracy
- Precision, Recall, and F1-score
- Confusion Matrix

Learning Curves

We also plotted:

- Training and validation accuracy curves
- Training and validation loss curves

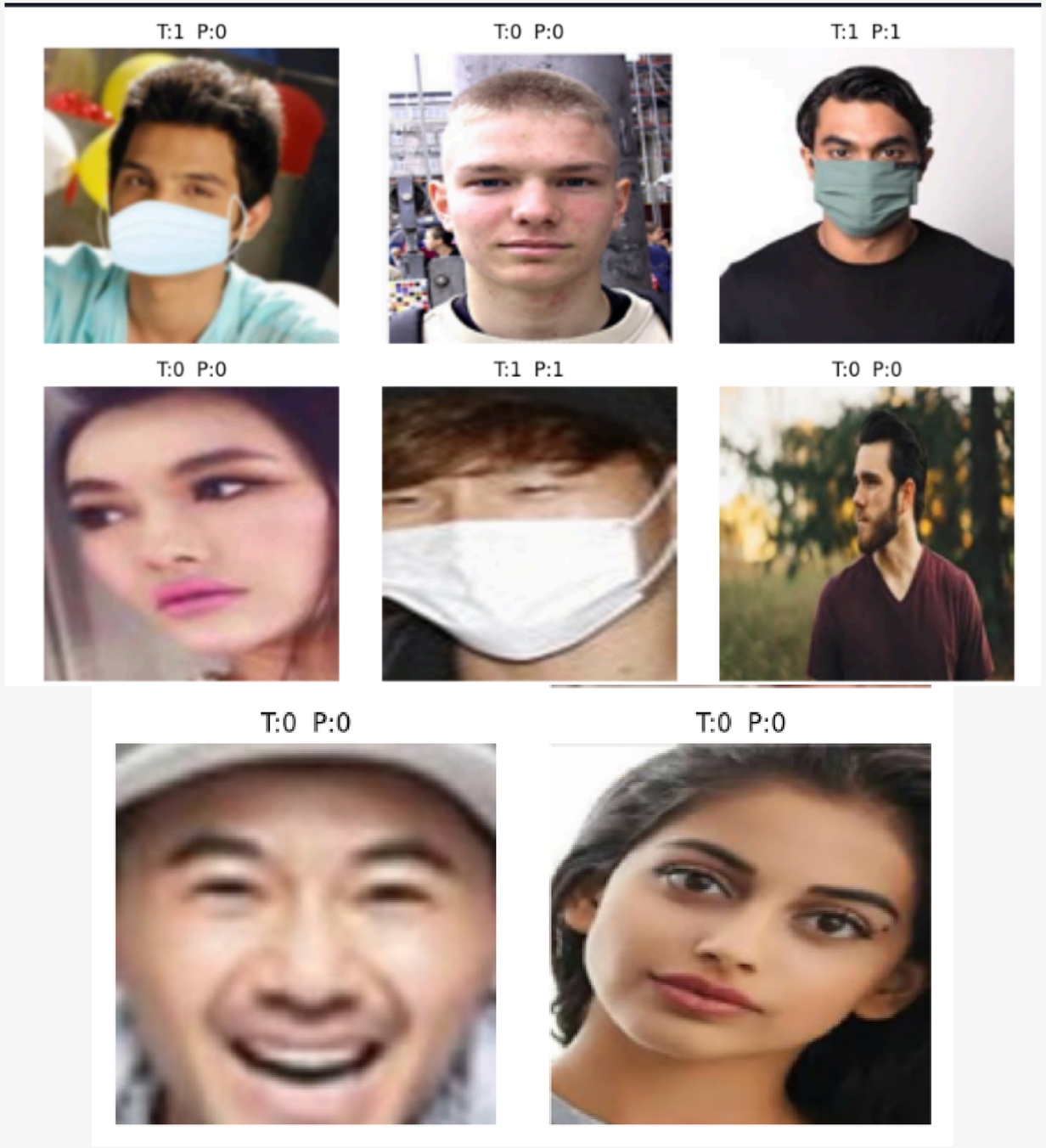
These plots help us analyze the learning behavior of the model and detect overfitting or underfitting.

Confusion Matrix

The confusion matrix shows how well the model distinguishes between masked and unmasked faces, and highlights misclassifications

Sample Predictions

We generated sample predictions on previously unseen images to visually validate the accuracy of the model’s outputs.



Discussion & Challenges

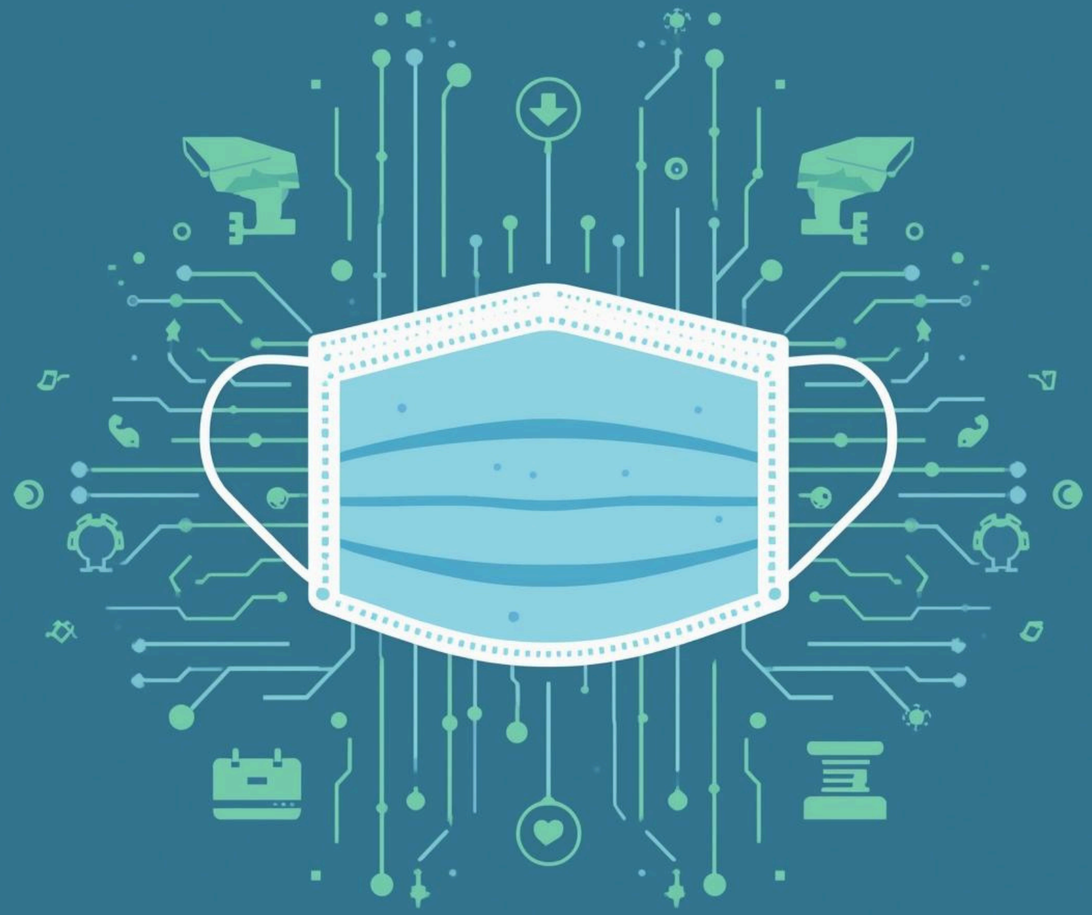
Challenges:

The model performs well on clear images, but we faced several challenges:

- Variations in lighting conditions
- Occlusions such as glasses or scarves
- Different face angles
- Limited dataset size

Limitations:

Because the dataset is relatively small, the model’s accuracy is limited compared to large-scale industrial systems.



Conclusion & Future Work

Conclusion

In conclusion, our project demonstrates that CNNs are effective for face mask detection using image data.

The system successfully classifies masked and unmasked faces and follows a clean, modular, and reproducible design.

Future Work

In the future, we can improve the system by:

- Using transfer learning models such as MobileNet or EfficientNet
- Expanding the dataset with more real-world images
- Extending the system to real-time video detection
- Supporting multi-class classification for incorrectly worn masks

Safety First

Thank you for your time.
We are ready to answer
any questions.