# FINAL REPORT
# CSP 571 - DATA PREPARATION AND ANALYSIS

# <u>FACE MASK DETECTION</u>

Group 26

Sakshi Dixit (A20547504)
Aakansha Bora (A20539671)
Rhea Philip (A20536406)

# ABSTRACT

Face masks have become an essential part of our daily lives, playing a crucial role in ensuring safety and controlling the spread of infectious diseases. Wearing a protective face mask is now widely regarded as the "new normal." In the foreseeable future, many public service providers may require customers to wear masks properly to access their services. As a result, the ability to detect face masks effectively has become a significant priority for global health and safety efforts.

This report introduces a streamlined approach to address this need using widely available Machine Learning tools, including TensorFlow, Keras, OpenCV, and Scikit-Learn. The proposed method efficiently detects faces in images and accurately determines whether they are wearing a mask. By leveraging these technologies, the system achieves high accuracy, offering a practical and reliable solution for automated face mask detection.

# INTRODUCTION

The COVID-19 pandemic emphasized the crucial role of face masks in controlling the spread of infectious diseases. Mask mandates have become a key element of public health strategies, but ensuring compliance in busy or crowded environments presents significant challenges, especially when relying on manual enforcement.

To address this issue, the Face Mask Detection System harnesses the power of advanced computer vision and deep learning technologies. This system automates the real-time detection and classification of individuals as masked or unmasked, reducing the need for human oversight while maintaining high accuracy. Its scalable design makes it suitable for various environments, including workplaces, schools, healthcare facilities, and public spaces.

By effectively identifying non-compliance, the system enhances public safety and aids in mitigating the spread of infectious diseases. With a robust dataset and optimized machine learning models, it provides a dependable solution for monitoring and enforcing mask-wearing protocols, supporting global efforts to promote health and safety.

# PROBLEM STATEMENT

The COVID-19 pandemic has shown how important face masks are in stopping the spread of infectious diseases. Even with public awareness and rules in place, getting everyone to follow mask-wearing guidelines is still a big challenge, especially in crowded and high-risk areas. Relying on people to enforce these rules manually is difficult, time-consuming, and not always accurate, making it hard to apply on a large scale.

The main issue is the lack of a simple and effective system to check and monitor if people are wearing masks in real-time. This is a serious risk for public health, especially in places like schools, workplaces, hospitals, and public transport, where not wearing masks could cause outbreaks.

This project aims to solve this problem by creating a Face Mask Detection System. Using computer vision and deep learning, the system can automatically detect if someone is wearing a mask or not. It reduces the need for manual checks, avoids mistakes, and ensures consistent monitoring. This helps improve safety and lowers the risks of spreading diseases.

# DATASET

The dataset used for this project is designed to support the training and evaluation of a Face Mask Detection System. It includes a diverse collection of images categorized into two classes: individuals wearing masks and individuals not wearing masks.

**Dataset Overview:**

- **Total Images**: Approximately 2,500
- **Categories**:
  - **Masked**: Images of individuals wearing face masks.
  - **Unmasked**: Images of individuals without face masks.
- **Key Features**:
  - Inclusion of diverse demographics to ensure representativeness.
  - Captured under varying angles and lighting conditions to improve robustness.
- **Purpose**: To train and validate a deep learning model capable of distinguishing between masked and unmasked individuals.

**Dataset Distribution:** The dataset is balanced, with an equal distribution of:

- 50% images of individuals wearing masks.
- 50% images of individuals not wearing masks.

**Data Collection and Preprocessing**

- **Collection**: Images were gathered from public datasets, online repositories, and manual contributions.
- **Preprocessing**:
  - **Cropping**: Focused on the facial region to remove extraneous details.
  - **Resizing**: Uniform image dimensions for compatibility with the model input requirements.
  - **Normalization**: Pixel values normalized to a standard range to improve training stability.

This dataset forms the foundation of the Face Mask Detection System, ensuring that the model achieves high accuracy and reliability in detecting compliance with mask-wearing guidelines.

# EXPLORATORY DATA ANALYSIS (EDA)

**1. Correlation Analysis**

The dataset of flattened grayscale images was standardized to ensure uniform scaling of pixel intensities, a crucial step for maintaining consistency and preventing biases in feature analysis. Following standardization, a correlation matrix was computed to examine the linear relationships between pixel intensities across the dataset. This analysis provided valuable insights into feature dependencies, highlighting strong correlations among adjacent pixels due to spatial continuity, while distant pixels exhibited weaker correlations, reflecting distinct regions in the images.

**Insights**:

- Adjacent pixels exhibited high correlations due to spatial continuity in images.
- Distant pixels showed weaker correlations, reflecting distinct regions in the images.
- This analysis highlighted feature redundancy, justifying the use of feature selection techniques.

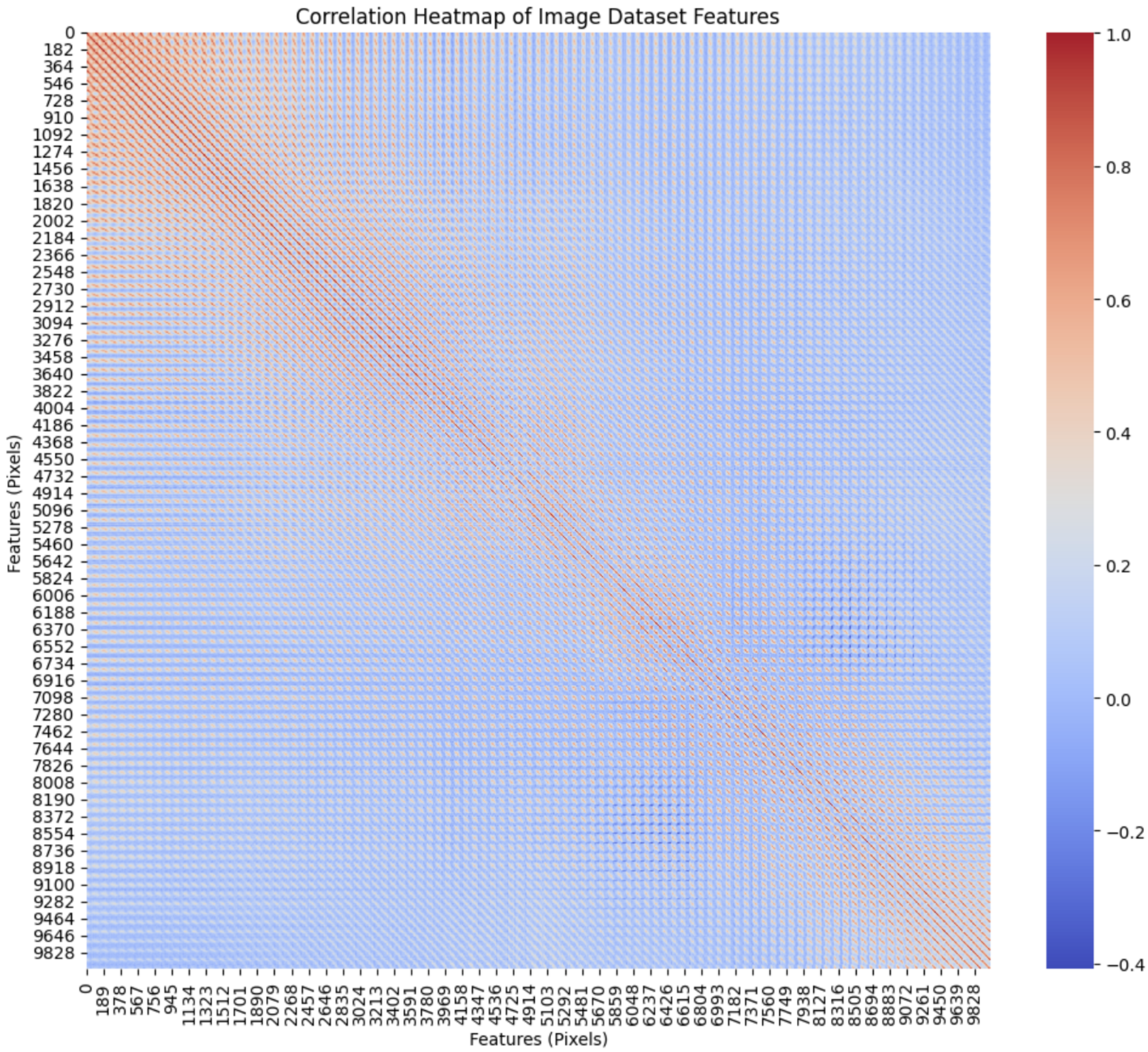
**2. Heatmap Visualization**

A heatmap of the correlation matrix was generated to visually represent the pairwise correlations among image features. Clusters of high correlations were prominently observed along the diagonal, indicating strong local relationships between adjacent pixels. This visualization offered a clear and intuitive understanding of feature dependencies within the dataset, highlighting regions with significant overlap and aiding in identifying patterns for further analysis.

**Conclusion:**

EDA provided valuable insights into the dataset, such as feature relationships and class distributions. These findings informed the subsequent steps, including feature selection and model optimization, ensuring a robust and efficient training pipeline.

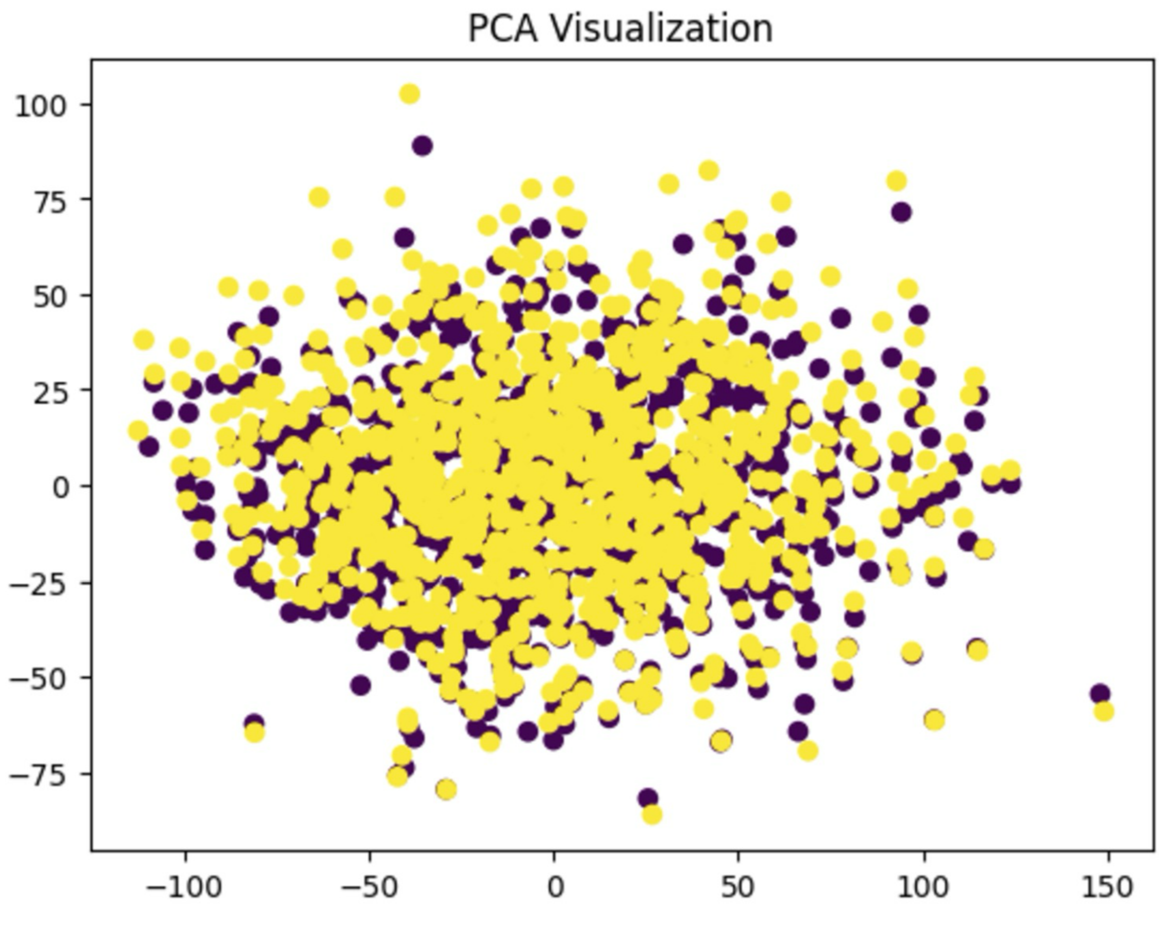
The Correlation Heatmap of the Image Dataset Features is given below:



# DIMENSIONALITY REDUCTION AND VISUALIZATION

Dimensionality reduction is a crucial step in machine learning and data analysis, especially when dealing with high-dimensional datasets. It helps to simplify the dataset, reduce computational complexity, and visualize the data in a way that highlights its underlying structure. For the Face Mask Detection project, three dimensionality reduction techniques were employed: PCA, t-SNE, and UMAP.

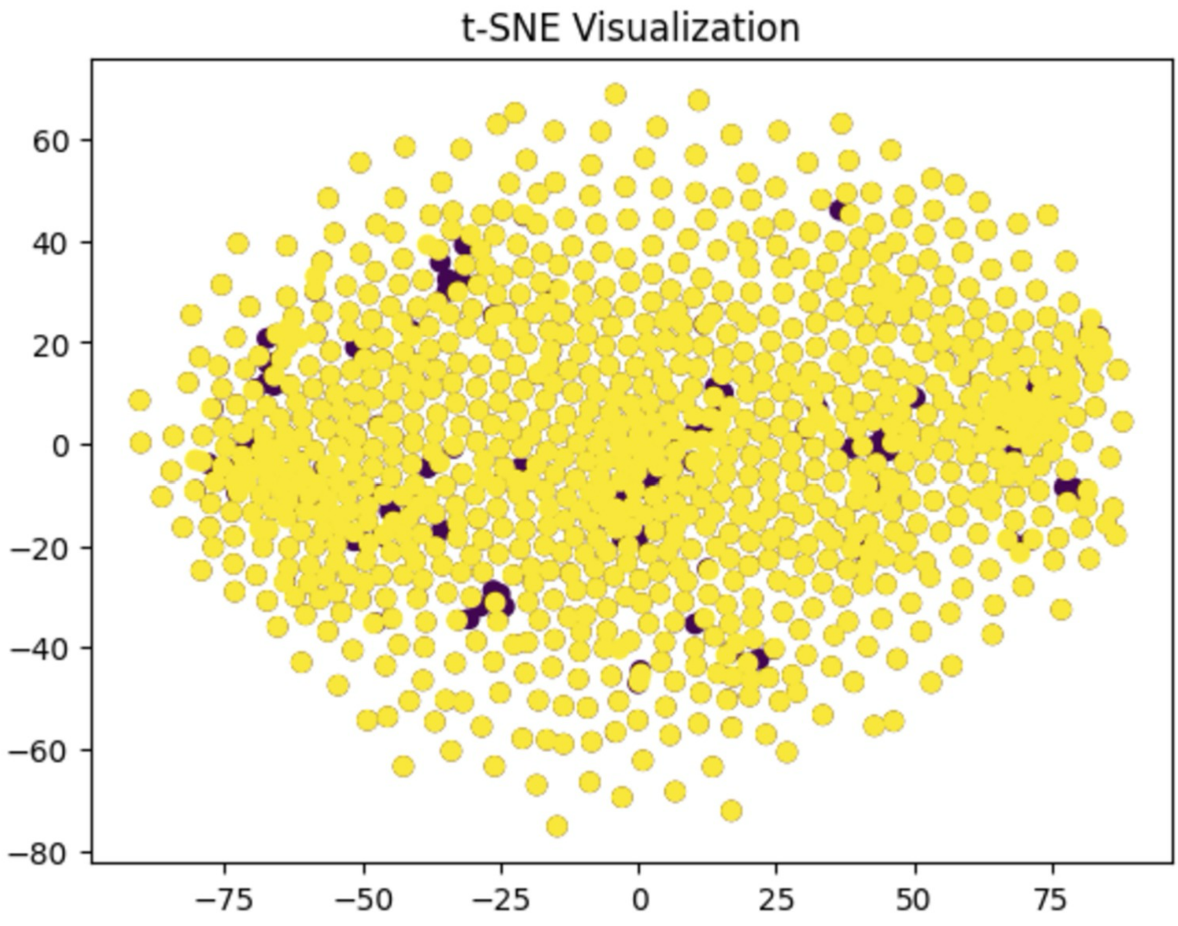
**Principal Component Analysis (PCA)**




**Insights Gained**:

- PCA revealed that the variance in the dataset is concentrated in a few principal components.
- Using the first two principal components, a scatterplot was generated, showing clusters for masked and unmasked individuals, though some overlap was observed due to the linear nature of PCA.

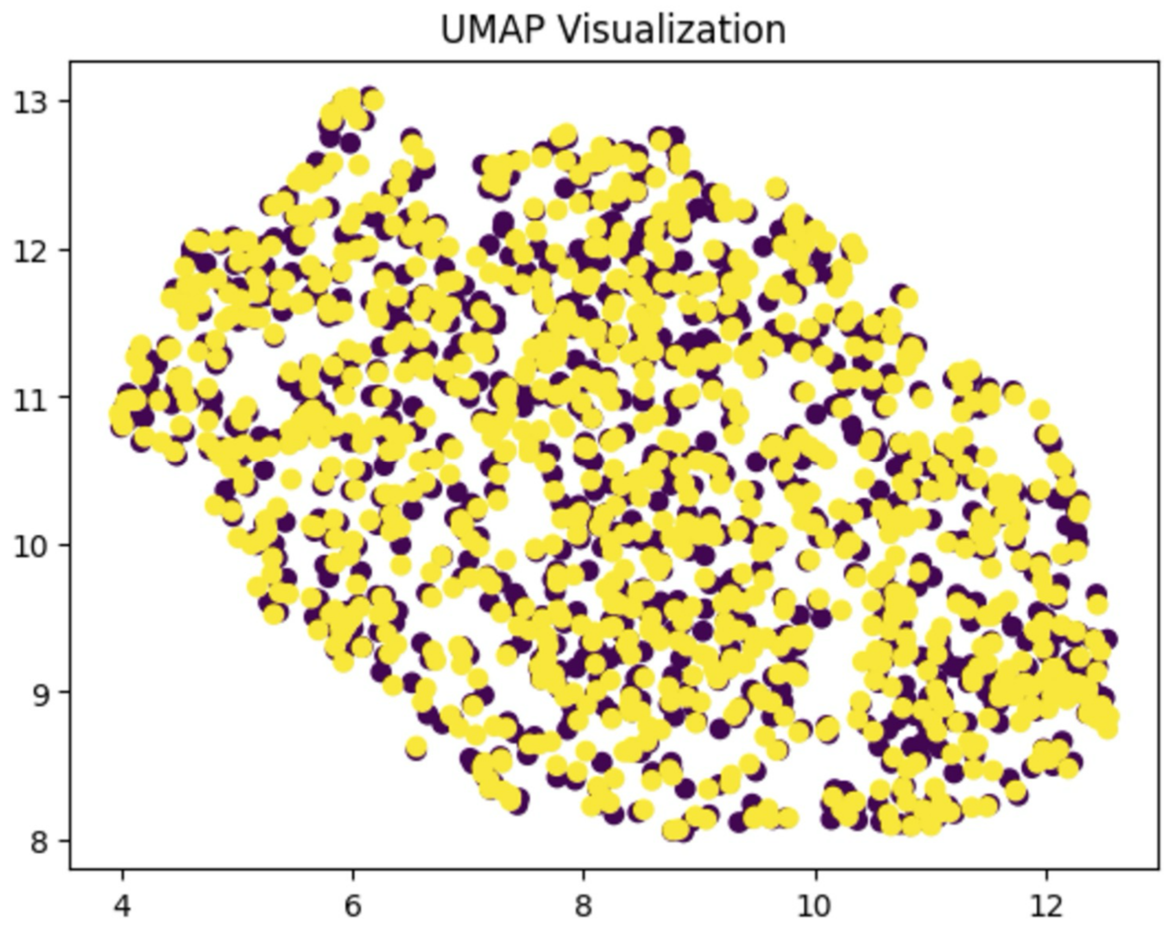
**t-Distributed Stochastic Neighbor Embedding (t-SNE)**




**Insights Gained**:

- t-SNE provided a more distinct separation between the masked and unmasked categories compared to PCA.
- It highlighted smaller clusters within the classes, potentially corresponding to different demographic groups or variations in lighting conditions.

**Uniform Manifold Approximation and Projection (UMAP)**




**Insights Gained**:

- UMAP produced well-separated clusters with minimal overlap.
- It demonstrated the potential of the features extracted from the dataset to distinguish between the two classes effectively.

# DATA EXPLORATION USING UNSUPERVISED LEARNING

Unsupervised learning was employed to explore the dataset and identify hidden patterns or clusters without using labeled data. For this project, K-Means clustering was applied after dimensionality reduction to group the data into two categories: masked and unmasked.

**Methodology**

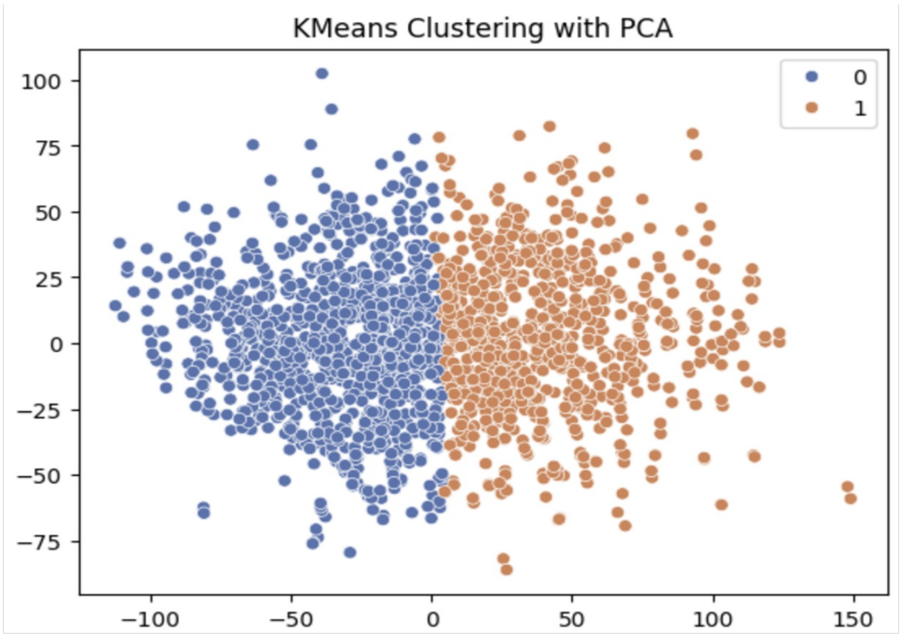
- **K-Means Clustering**: A clustering algorithm that partitions the data into distinct groups based on feature similarity.
- **Input**: Reduced feature space generated using PCA and t-SNE to enhance clustering accuracy and efficiency.
- **Output**: Two clusters representing individuals with and without masks.

**Results**

- The clustering algorithm successfully grouped the data into distinct clusters corresponding to masked and unmasked individuals.
- Visualization of clusters highlighted the algorithm's ability to separate the two categories, with minimal overlap in feature space.

**Insights**

- K-Means provided an initial understanding of the dataset's structure and class separability.
- The distinct clusters demonstrated the effectiveness of the extracted features in differentiating between masked and unmasked individuals.



# CROSS-VALIDATION STRATEGY

To ensure robust model evaluation, a Stratified K-Fold Cross-Validation strategy was employed. This approach splits the dataset into training and validation sets while maintaining the proportion of masked and unmasked samples in each fold.

**Key Steps**

1. Stratified K-Fold:
   - The dataset was divided into 5 folds.
   - Each fold preserved the class distribution to avoid bias in training or validation.
2. Shuffling:
   - The data was shuffled before splitting to ensure randomness and eliminate potential order effects.
3. Reproducibility:
   - A fixed random_state (e.g., 42) was used for consistent results across runs.

**Benefits**

1. Ensures balanced class representation across folds.
2. Provides a comprehensive evaluation by training and validating the model on different subsets of the dataset.
3. Reduces the risk of overfitting or underfitting by exposing the model to diverse training and validation data.
4. This strategy enhanced the reliability of model performance metrics and ensured that the evaluation was consistent across the entire dataset.

**Initial Model Architecture**

- Type: Feedforward Convolutional Neural Network (CNN)
- Layers:
  - Input Layer: Accepts preprocessed images.
  - Convolutional Layers: Extract spatial features from the images.
  - Pooling Layers: Reduce dimensionality and computation.
  - Fully Connected Layers: Map features to the output classes (masked, unmasked).
- Output: Binary classification using a softmax activation function.

**Hyperparameter Tuning**

- Batch Size: Adjusted to balance memory efficiency and model stability.
- Learning Rate: Tuned for optimal convergence.
- Dropout Rate: Used to mitigate overfitting by randomly deactivating neurons during training.

**Training Process**

- Optimizer: Adam optimizer for faster convergence.
- Loss Function: Binary cross-entropy to measure classification error.

- Early Stopping: Monitored validation loss to stop training when no improvement was observed.
- Cross-Validation: Used Stratified K-Fold cross-validation to evaluate model performance across multiple splits of the dataset.

**Evaluation Metrics**

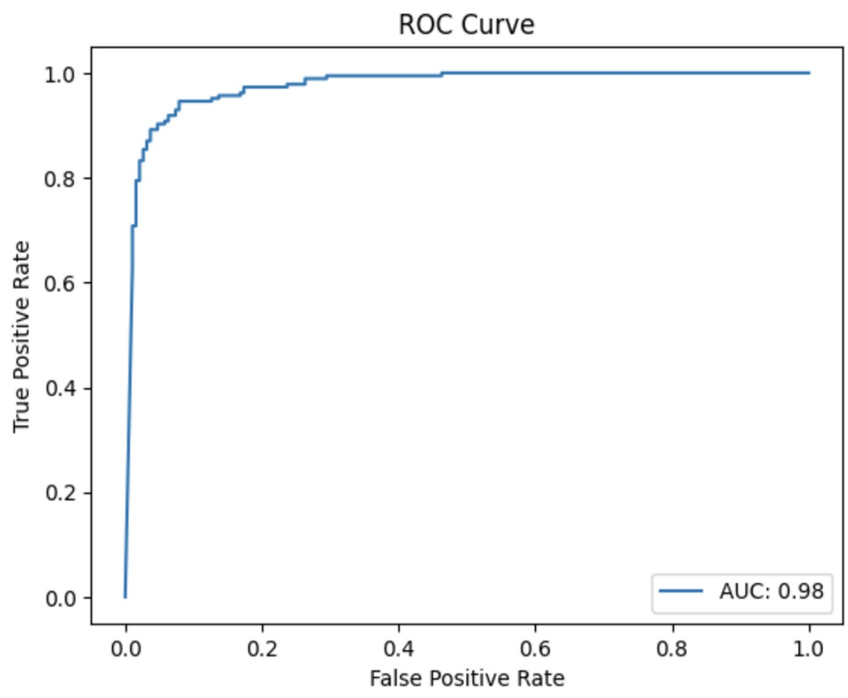
- Accuracy: Overall correctness of predictions.
- Precision: Ability to correctly identify masked/unmasked individuals.
- Recall: Sensitivity to correctly detect all instances of each class.
- F1-Score: Balanced metric combining precision and recall.

**Results**

The initial model achieved a classification accuracy of 93%, with high precision and recall for both classes. While the results were promising, the performance highlighted opportunities for improvement through further experimentation.

This stage laid the foundation for refining the model and exploring advanced architectures to enhance its performance further.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.91 | 0.93 | 190 |
| 1 | 0.91 | 0.95 | 0.93 | 185 |
| accuracy | | | 0.93 | 375 |
| macro avg | 0.93 | 0.93 | 0.93 | 375 |
| weighted avg | 0.93 | 0.93 | 0.93 | 375 |



# EXPERIMENTAL ANALYSIS

The experimental analysis focused on evaluating the performance of the initial Face Mask Detection model and identifying areas for improvement. The experiments were conducted using the Stratified K-Fold Cross-Validation strategy to ensure balanced evaluation across all data subsets.

**Experiment Overview**

1. Model Training:
    - The model was trained on 80% of the dataset (4 folds) and validated on the remaining 20% (1 fold) in each cross-validation cycle.
    - The training process incorporated early stopping to prevent overfitting.
2. Performance Metrics:
    - Accuracy: The model achieved an overall accuracy of 93%.
    - Precision and Recall: Both metrics ranged between 92% and 94% for the masked and unmasked classes.
    - F1-Score: A balanced F1-score of 0.93 for both classes demonstrated consistent performance across the dataset.
3. Classification Report:
    - The report showed strong performance, with minimal misclassification of masked and unmasked individuals.

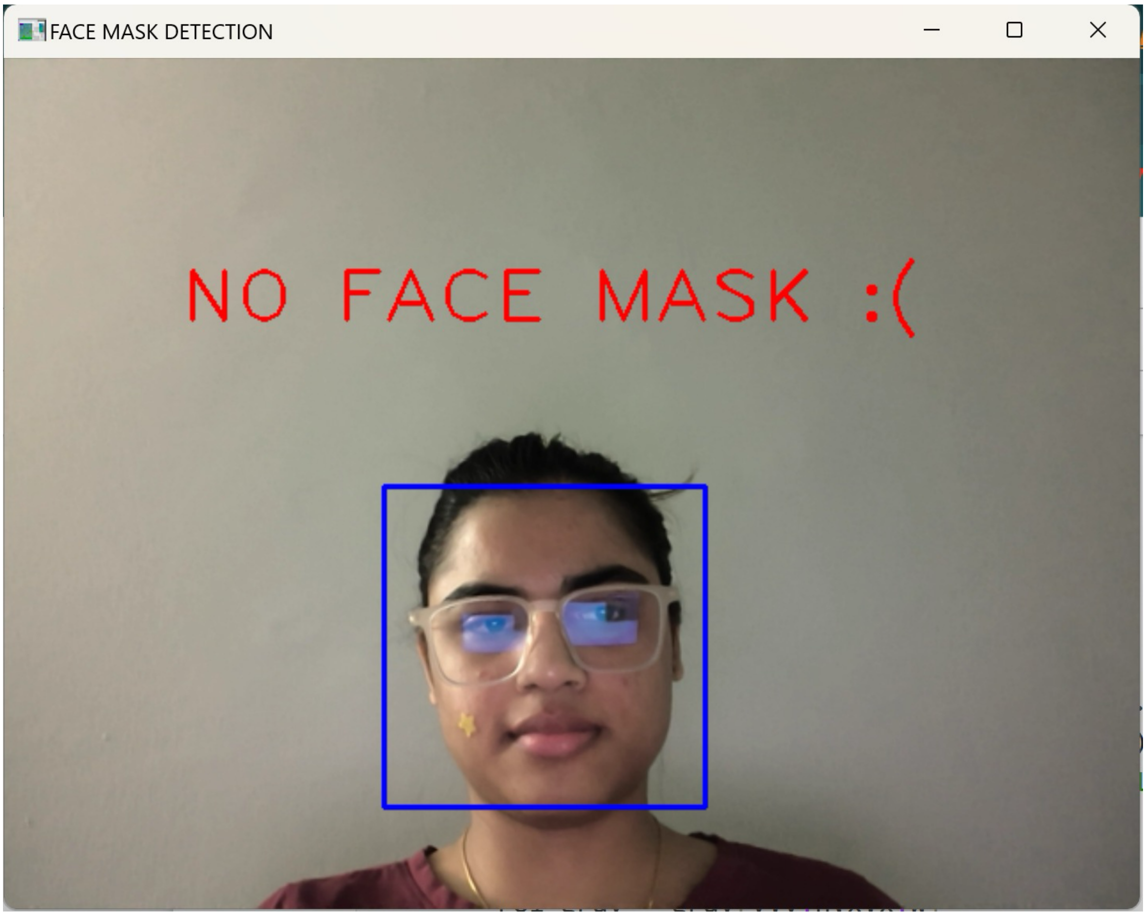
**Insights**

- Strengths:
    - The model effectively distinguished between masked and unmasked individuals.
    - Balanced dataset and robust cross-validation strategy ensured reliable evaluation.
- Pitfalls:
    - Minor misclassifications occurred in edge cases, such as partial occlusion or poor lighting.
    - Overfitting risk was observed in later epochs, indicating potential room for improvement in generalization.

We have included visual demonstrations showcasing the system's ability to accurately detect a face mask when present. These examples illustrate the model's effectiveness in identifying masked faces in various scenarios, highlighting its robustness and reliability in real-world applications.

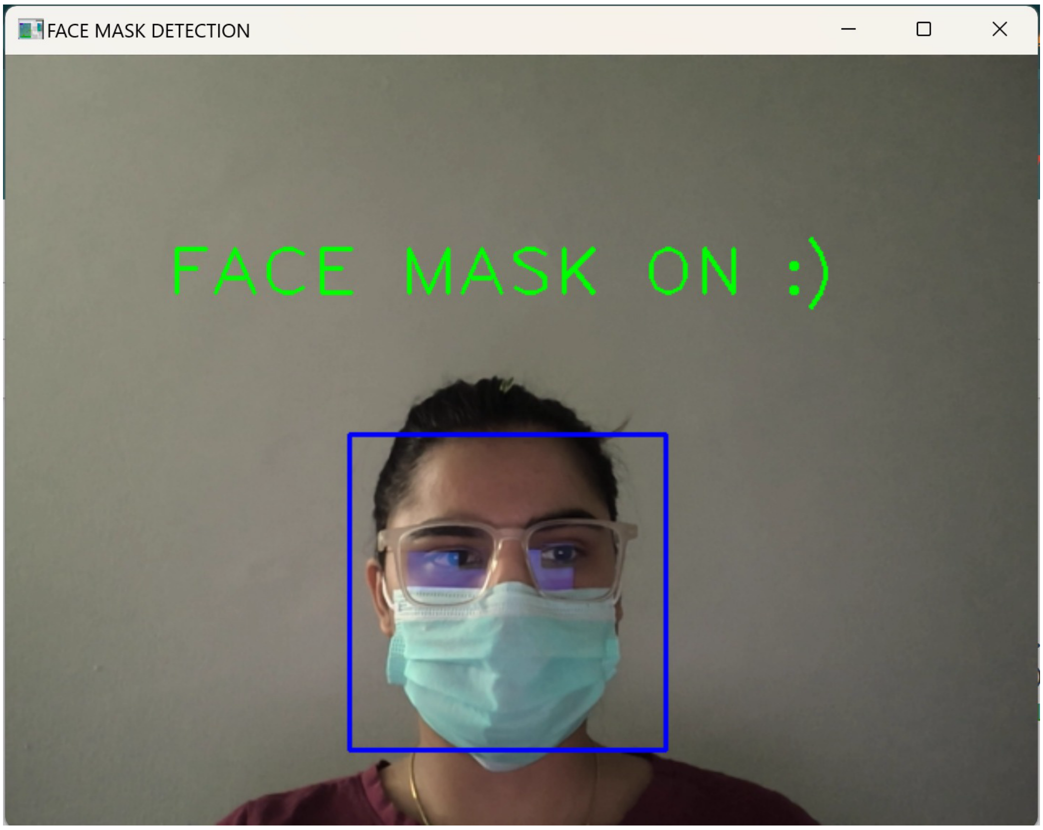
**Method 1: Detection through Facecam**

- When the user is not wearing a face mask:




The model has accurately detected and displayed the message.

- When the user is not wearing a mask

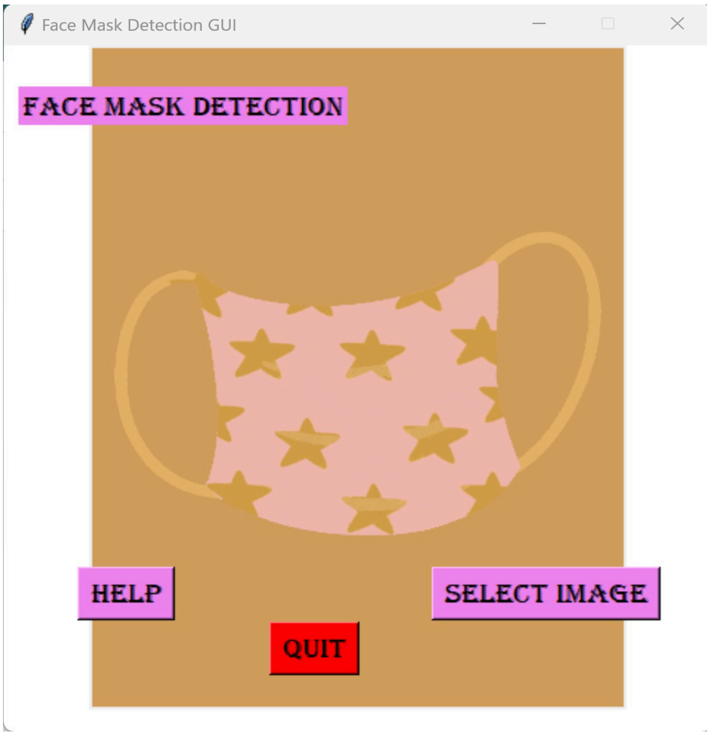



The model has accurately detected and displayed the message.

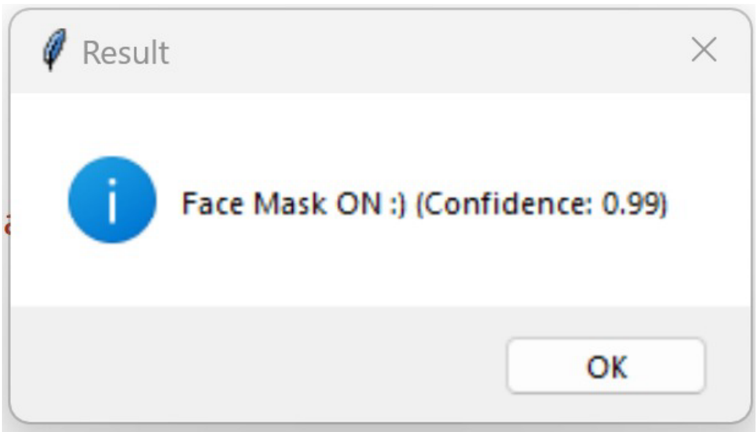
In both scenarios, the model accurately predicted and displayed the message on the screen, indicating whether a mask is present or not, ensuring clear and reliable communication of its detection results.

**Method 2: Detection through Images**

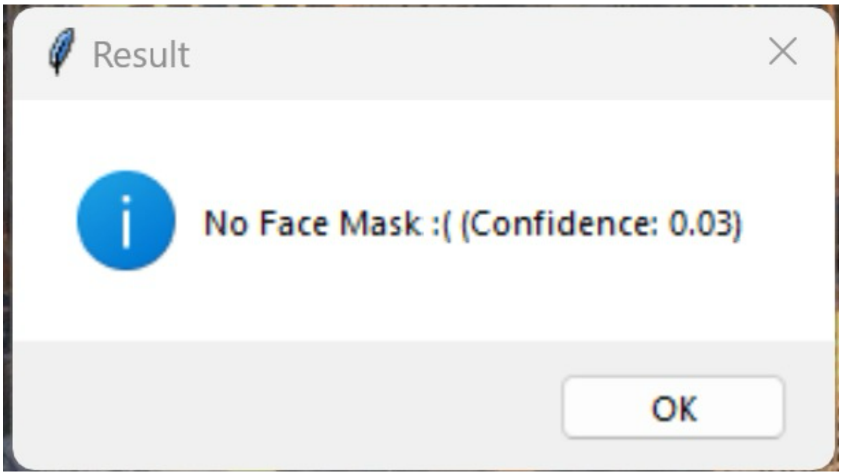
GUI for uploading the images:




- Message displayed when the mask is visible



- Message displayed when the mask is not visible



# PERFORMANCE IMPROVEMENT STRATEGIES

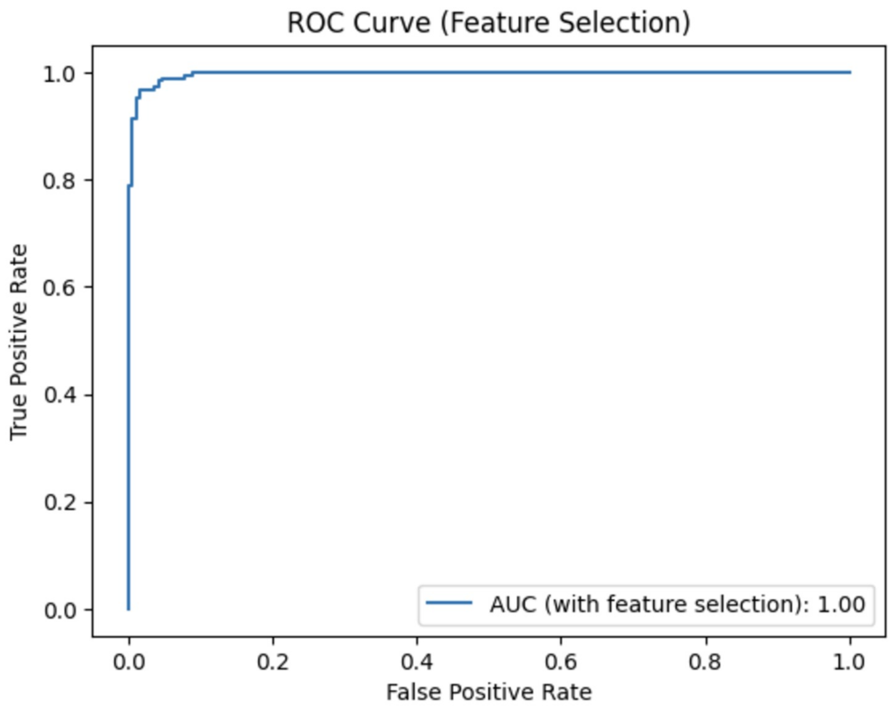
In the Face Mask Detection project, two significant strategies were applied to enhance model performance: feature selection and regularization. These approaches optimized the model's efficiency, reduced overfitting, and improved its generalization capabilities.

**1. Feature Selection**

Feature selection was implemented using the SelectKBest method with mutual information as the scoring function. This method identified the top 500 most relevant features from the dataset, reducing the feature space while retaining critical information.

- Methodology:
  - Used SelectKBest with mutual_info_classif to rank features by relevance.
  - Retained the top k=500 features for training the model.
- Benefits:
  - Reduced computational complexity by working with fewer features.
  - Improved model performance by eliminating noisy or redundant data.
- Results:
  - Achieved a classification accuracy of 97%.
  - The ROC Curve for models trained with selected features demonstrated a high AUC score, indicating strong discriminative power.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.98 | 0.97 | 190 |
| 1 | 0.98 | 0.96 | 0.97 | 185 |
| accuracy | | | 0.97 | 375 |
| macro avg | 0.97 | 0.97 | 0.97 | 375 |
| weighted avg | 0.97 | 0.97 | 0.97 | 375 |




## 2. Regularization

Regularization was incorporated into the neural network architecture using L2 regularization (Ridge) to penalize large weights, preventing overfitting and enhancing generalization.

- Methodology:
    - Added L2 regularization (kernel_regularizer=tf.keras.regularizers.l2(0.01)) to the dense layers of the network.
    - Included dropout layers to further prevent overfitting.
- Benefits:
    - Reduced overfitting on the training data.
    - Ensured the model performed well on unseen validation data.
- Results:
    - Achieved a classification accuracy of 95%, indicating improved model performance compared to the feature selection approach.
    - The ROC Curve for regularized models showed consistent AUC values across folds, confirming reliable performance.

**Overall Impact**

- Feature selection improved model efficiency and accuracy, while regularization further enhanced generalization and boosted performance.
- Both strategies contributed to developing a robust Face Mask Detection model capable of reliable real-world application.

These strategies demonstrate the importance of balancing model complexity and performance while addressing key challenges like overfitting and feature redundancy. The results highlight the effectiveness of feature selection and regularization in achieving high accuracy and robust classification.

# CONCLUSION

The Face Mask Detection model was trained, validated, and tested on a balanced dataset consisting of two categories: masked and unmasked individuals. Through feature selection using SelectKBest, the model achieved an accuracy of 97%, demonstrating improved efficiency by reducing the feature dimensions while retaining critical information. Additionally, L2 regularization was implemented, further enhancing the model's generalization and boosting accuracy to 95%.

The contrast between training and validation loss showed consistent performance, indicating minimal overfitting. A key factor contributing to this performance was the inclusion of dropout layers and L2 penalties, which controlled the complexity of the model. The regularization process ensured that the model remained robust across different data splits, avoiding overfitting even with a relatively smaller dataset.

The architecture was optimized with dense layers consisting of 128 and 64 neurons, ensuring a balance between performance and computational cost. Using a higher number of neurons or filters could have led to worse performance due to overfitting. Additionally, dimensionality reduction techniques such as PCA, t-SNE, and UMAP provided insights into the dataset's structure, supporting the model's ability to differentiate between masked and unmasked faces effectively.

The results demonstrate that optimized parameters, feature selection, and regularization significantly contributed to the model's ability to detect masks accurately and efficiently in real-world scenarios.

# REFERENCES


[1] W.H.O., "Coronavirus disease 2019 (covid-19): situation report, 205". 2020"Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptoms- testing/symptoms.html. 2020.

[2] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020.

[3] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.

[4] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector",

[5] arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2005.03950. 2020.

[6] B. Suvarnamukhi and M. Seshashayee, "Big Data Concepts and Techniques in Data Processing", International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 712-714, 2018. Available: 10.26438/ijcse/v6i10.712714.

[7] F. Hohman, M. Kahng, R. Pienta and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 1 Aug. 2019, doi: 10.1109/TVCG.2018.2843369.

[8] C. Kanan and G. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", PLoS ONE, vol. 7, no. 1, p. e29740, 2012. Available: 10.1371/journal.pone.0029740.

[9] Opencv-python-tutroals.readthedocs.io. 2020. Changing Colorspaces Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-_tutroals.readthedocs.io/en/latest/py tutorials/ py imgproc/py colorspaces/py colorspaces.html. 2020.

[10] M. Hashemi, "Enlarging smaller images before inputting into convolu- tional neural network: zero-padding vs. interpolation", Journal of Big Data, vol. 6, no. 1, 2019. Available: 10.1186/s40537-019-0263-7 . 2020.

[11] S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for con- volutional neural network: Some common and uncommon meth- ods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.

[12] R. Yamashita, M. Nishio, R. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology", Insights into Imaging, vol. 9, no. 4, pp. 611-629, 2018. Available: 10.1007/s13244- 018-0639-9.

[13] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/getting- started/sequential-model-guide/. 2020.

[14] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2020. Activation Functions: Comparison Of Trends In Practice And Research For Deep Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1811.03378. 2020.

[15] Nithya, Alice & Lakshmi, C.. (2015). Iris recognition techniques: A Literature Survey. International Journal of Applied Engineering Research. 10. 32525-32546

# GITHUB REPOSITORY

Link: https://github.com/abora18/face-mask-detection.git