Project by CNN "Convolutional neural network" – CS417 Neural networks

Face mask detection

Team members

Shahd Hesham Shawky Mohamed – 2227329

Alaa Nagah Abdo Mostafa - 2227121

Mariam Mahmoud Hefny Mahmoud - 2227352

Nada Sameh Mahmoud Ahmed - 2227562

Amira Hagag Mohamed Mohamed – 2227093

Introduction/Background

- This project presents the design and implementation of an automated Face Mask Detection system using deep learning techniques. The objective is to classify facial images into two categories: with mask and without mask, using Convolutional Neural Networks (CNNs) and a transfer learning approach.
- The system is implemented using TensorFlow and Keras and is evaluated on the kaggle face mask detection dataset. A complete data pipeline is developed to handle dataset organization, preprocessing, and augmentation. Images are resized to a fixed resolution of 128 × 128, and different preprocessing strategies are applied depending on the selected model architecture.
- Three different models are implemented and compared: a baseline CNN, an improved CNN with deeper architecture and regularization techniques, and a transfer learning model based on EfficientNetB0. All models are trained using a unified training framework with the Adam optimizer, binary cross-entropy loss, and callback mechanisms such as early stopping and model checkpointing to ensure stable and efficient training.
- The best-performing model is selected based on validation accuracy and saved for further evaluation. This project demonstrates how custom CNN architectures and transfer learning can be applied and compared within a consistent experimental setup for face mask detection tasks.

Significance and Applications

This project addresses the problem of face mask detection using deep learning techniques that provide a practical solution that can be applied in real-world scenarios. The significance of the proposed approach and its potential applications can be summarized in the following points :

- Automates the process of face mask detection, reducing the need for manual monitoring in crowded and public environments.
- Supports public health and safety by enabling consistent monitoring of mask-wearing compliance.
- Utilizes deep learning–based image classification to accurately distinguish between masked and unmasked faces under varying lighting conditions and viewing angles.
- Provides a scalable solution that can be integrated with existing CCTV surveillance systems.
- Can be deployed on edge devices such as smart cameras for real-time monitoring.

- Applicable in multiple real-world settings, including airports, public transportation, shopping malls, educational institutions, and workplaces.
- Can be used in access control systems to restrict entry based on mask-wearing compliance.
- Demonstrates the practical application of CNNs and transfer learning techniques in real-world computer vision tasks.

Dataset Documentation

1. Dataset Source

The dataset used in this project is the **Kaggle Face Mask Detection dataset** and this is its link :

https://www.kaggle.com/datasets/andrewmvd/face-mask-detection

which contains facial images labeled into two categories: **with mask** and **without mask**. The dataset includes images collected from multiple sources, providing variations in lighting conditions, facial poses, and backgrounds.

2. Dataset Organization

The dataset is organized into a directory structure compatible with Keras data generators. Images are arranged into class-specific folders and split into training, validation, and test sets. This structure allows automatic label assignment based on folder names and enables efficient batch loading during training.

3. Classes Description

- **with_mask:** Images of individuals wearing a face mask that covers the nose and mouth.






- **without_mask:** Images of individuals not wearing a face mask.

This data split allows the model to learn from a large portion of the data while being validated and tested on previously unseen samples.

3. Data Preprocessing

All images are resized to a fixed resolution of 128 × 128 pixels to ensure consistent input dimensions across all models. For the baseline and improved CNN models, pixel values are normalized using rescaling to the range [0, 1]. For the EfficientNetB0 model, the official preprocessing function is applied to match the normalization used during ImageNet pre-training.

4. Data Augmentation

To improve model generalization and reduce overfitting, data augmentation is applied to the training set. Augmentation techniques include random rotations, width and height shifts, zooming, shearing, horizontal flipping, and brightness adjustments. No augmentation is applied to the validation and test sets to ensure unbiased performance evaluation.

5. Data Split

The dataset is divided into three subsets:

- **Training set:** 70%
- **Validation set:** 15%
- **Test set:** 15%

This split ensures effective learning while allowing reliable validation and evaluation on unseen data.

Methodology

1. Data Preprocessing

- Before training, the input images undergo a structured preprocessing pipeline to ensure consistency and improve learning performance. All images are resized to a fixed resolution of 128 × 128 × 3, which standardizes the input dimensions across all models.

- For the baseline and improved CNN models, pixel values are normalized by rescaling them to the range [0, 1]. In contrast, the EfficientNetB0 model uses its official preprocessing function to match the normalization applied during ImageNet pre-training. This model-specific preprocessing ensures compatibility with the underlying architecture.

- To enhance generalization and reduce overfitting, data augmentation is applied to the training set. Augmentation techniques include random rotations, width and height shifts, zooming, shearing, horizontal flipping, and brightness adjustments. Validation and test sets are not augmented to allow fair and unbiased evaluation.

2. Model Architectures

Three different deep learning models are implemented to address the face mask detection task :

1) The baseline CNN model:

is designed to establish an initial performance benchmark. It consists of multiple convolutional layers with increasing filter sizes, followed by max-pooling layers for spatial downsampling. Instead of using a traditional flattening layer, global average pooling is applied to reduce the number of trainable parameters and improve computational efficiency.

2) The improved CNN model:

This model extends the baseline architecture by increasing depth and incorporating batch normalization layers after convolutional operations to stabilize training. Multiple dropout layers are included to reduce overfitting and improve model robustness. This architecture enables the model to learn more complex and discriminative facial features.

3) **A transfer learning model based on EfficientNetB0:**

This model is implemented where the pre-trained EfficientNetB0 convolutional base is used as a feature extractor with its weights frozen during training. Custom fully connected layers are added on top to adapt the model to the face mask detection task.

3. Training Strategy

- All models are trained using :

  1) **Adam optimizer, and**
  2) **binary cross-entropy loss**

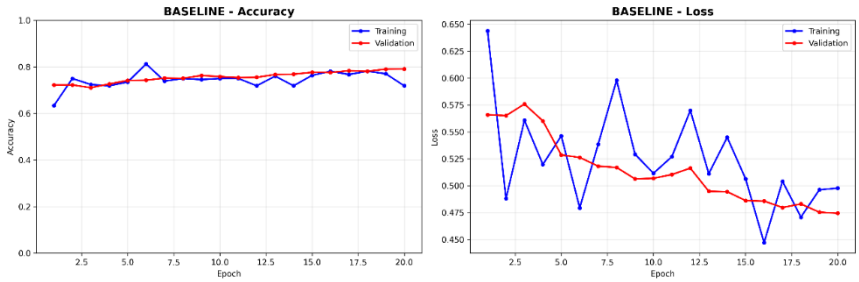
- which is suitable for binary classification problems. Different learning rates are applied depending on the model type to ensure stable convergence.

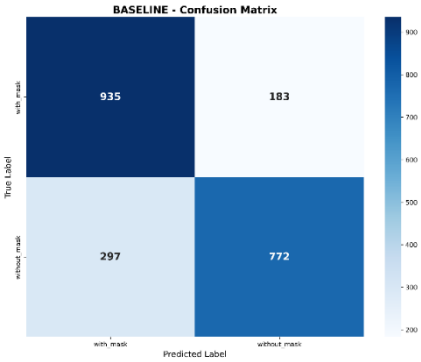
- Training is monitored using validation performance, and several callback mechanisms are employed to improve training efficiency and prevent overfitting. These include early stopping, which halts training when validation loss stops improving, model checkpointing, which saves the best-performing model based on validation accuracy, and learning rate reduction on plateau, which adjusts the learning rate when performance stagnates.
- This unified training framework allows fair comparison between different model architectures under consistent experimental conditions.

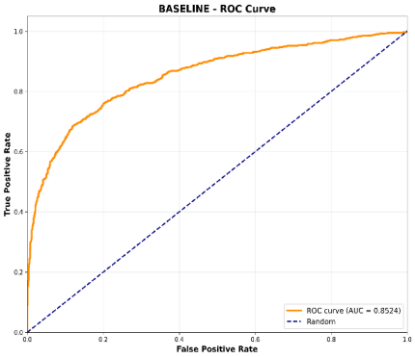
Evaluation Results

The performance of the proposed face mask detection system is evaluated using three different models: **Baseline CNN**, **Improved CNN**, and **EfficientNetB0**. The evaluation is conducted on a held-out test set and includes accuracy, confusion matrices, ROC curves with AUC scores, training behavior analysis, and qualitative inspection of sample predictions.

1. Baseline CNN Results

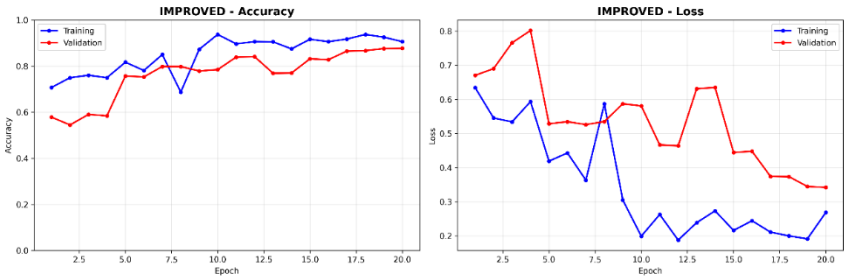


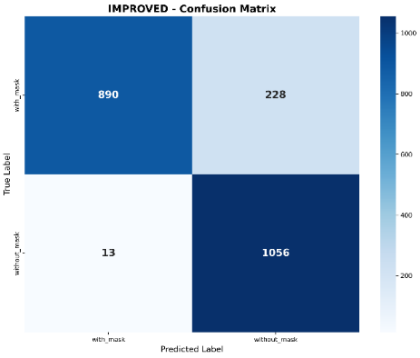


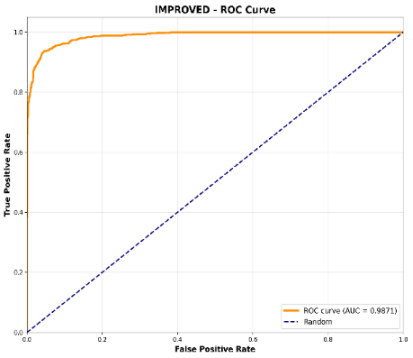


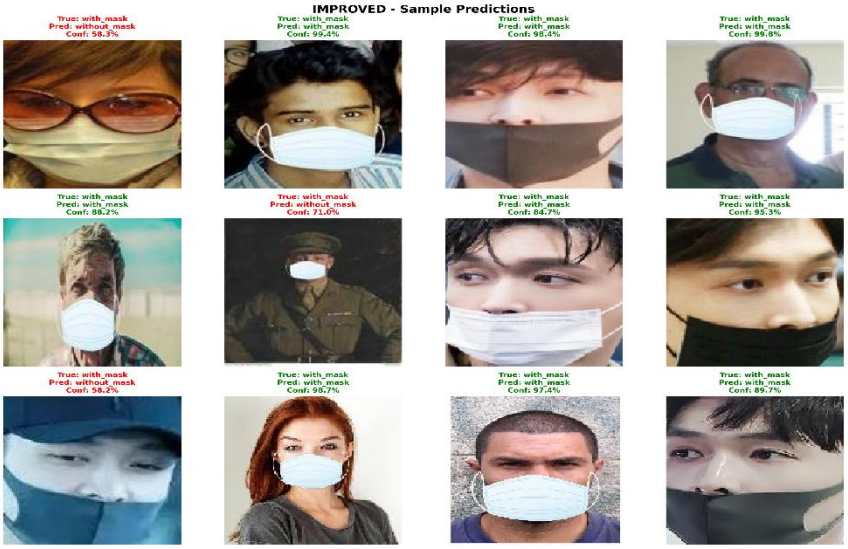
The baseline CNN achieves a **test accuracy of 78.05%** with an **AUC score of 0.8524**. Training and validation curves show gradual improvement; however, fluctuations in loss indicate limited capability in learning complex visual patterns. The confusion matrix reveals a notable number of misclassifications, particularly under challenging conditions such as partial face coverage and lighting variations. The ROC curve reflects moderate class separability, highlighting the limitations of the simple architecture.

2. Improved CNN



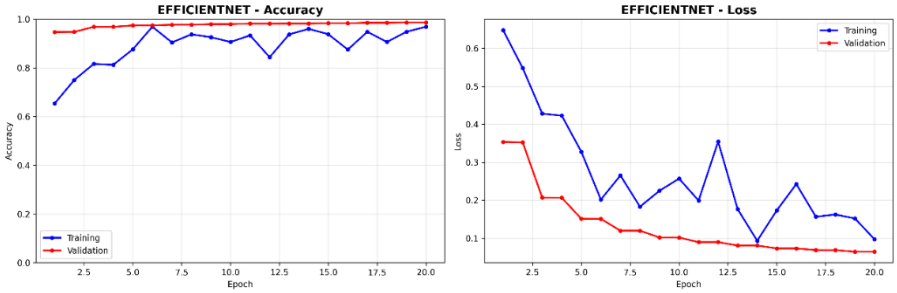


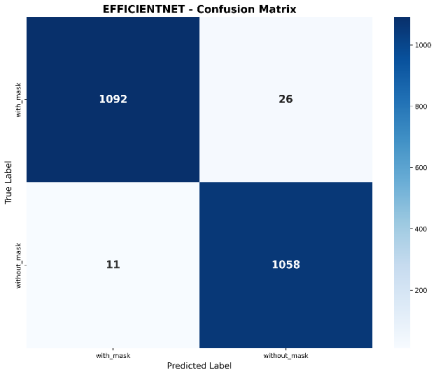


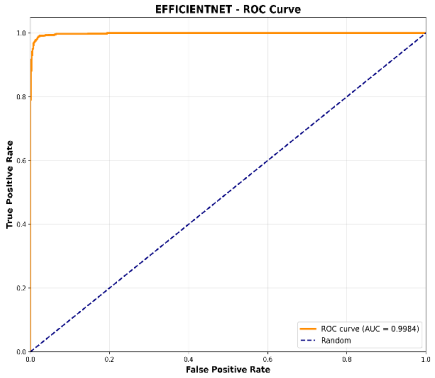


The improved CNN shows clear performance gains, achieving a **test accuracy of 88.98%** and an **AUC score of 0.9871**. Training behavior is more stable, and loss curves indicate improved optimization. The confusion matrix demonstrates fewer misclassifications and more balanced precision and recall across both classes. The ROC curve confirms strong class separability, and sample predictions show higher confidence and robustness compared to the baseline model.

3. EfficientNetB0



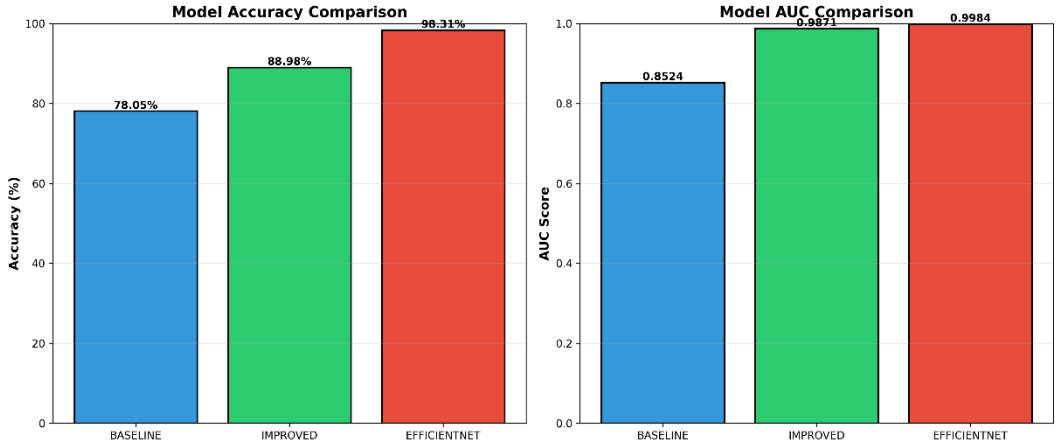




EfficientNetB0 achieves the best performance with a **test accuracy of 98.31%** and an **AUC score of 0.9984**. Training converges rapidly with consistently low loss values. The confusion matrix shows minimal misclassifications, and precision, recall, and F1-scores are consistently high. The ROC curve is close to ideal, indicating excellent class separability. Sample predictions further confirm strong generalization across diverse conditions.

Overall, the results demonstrate a clear progression in performance from the baseline CNN to the improved CNN, with EfficientNetB0 outperforming all models due to its transfer learning capability.

Comparative Analysis



A direct comparison between the three models highlights the trade-off between model complexity and performance. While the baseline CNN offers a computationally efficient solution with reasonable accuracy, its performance is limited. The improved CNN strikes a better balance between accuracy and complexity, showing substantial gains through architectural enhancements and regularization.

EfficientNetB0 clearly outperforms the other models by leveraging transfer learning and pre-trained ImageNet features. Based on the overall evaluation results, **EfficientNetB0 is selected as the best-performing model** for the face mask detection task.

Discussion

The evaluation results reveal clear performance differences among the three models, emphasizing the effect of architectural complexity on face mask detection.

- **The baseline CNN** achieves reasonable performance for a simple architecture, but analysis of training curves and the confusion matrix shows limited ability to learn complex visual patterns. Performance degradation is observed in challenging cases such as partial occlusions, lighting variations, and non-standard mask appearances, which is reflected in its moderate AUC score.

- **The improved CNN** demonstrates noticeable improvement over the baseline model. The addition of deeper layers, batch normalization, and dropout leads to more stable training, fewer misclassifications, and stronger class separability, as shown by smoother learning curves and a higher AUC score. However, some ambiguity remains in cases involving occlusions or unclear mask usage.

- **The EfficientNetB0** model achieves the highest performance across all metrics. Rapid convergence, low loss values, near-perfect ROC curves, and minimal misclassifications indicate strong generalization and reliable feature extraction. These results highlight the effectiveness of transfer learning, particularly when working with a relatively limited dataset.

Overall, the results confirm that while simpler CNN models offer computational efficiency, transfer learning with EfficientNetB0 provides the most robust solution for accurate face mask detection.

Challenges and solutions

During development, we encountered a critical preprocessing mismatch that significantly impacted model performance.

Challenge: Preprocessing Inconsistency

Our baseline CNN was inadvertently trained using EfficientNet preprocessing (ImageNet normalization, pixel range ~[-1, 1]) but evaluated using standard rescaling (pixel range [0, 1]). This inconsistency caused:

- Test accuracy of only 78% despite 87% validation accuracy (unexplained 9% gap)
- Further degradation to 75% when attempting fixes without addressing the root cause
- Severe class imbalance (93% vs 57% recall between classes)

Solution

We identified that train.py was importing default generators regardless of model type.
The fix involved:

1. Refactoring to dynamically load appropriate generators: train_generator, val_generator, _ = get_generators(model_type)
2. Ensuring each model received correct preprocessing:
    - Baseline/Improved CNNs: Standard rescaling (1./255)
    - EfficientNet: ImageNet preprocessing

Results after fix

- **Baseline CNN:** 78.05% (balanced performance restored)
- **Improved CNN:** 88.98% (+10.93% improvement)
- **EfficientNet:** 98.31% (near-perfect balance, 97-99% precision/recall)

This highlighted the critical importance of preprocessing consistency across training and evaluation pipelines.

Limitations

1. Dataset Representativeness

The Kaggle dataset (~7,000 images) may not fully capture real-world complexity:

- Crowded environments with overlapping faces
- Low-resolution surveillance footage
- Extreme lighting conditions (backlighting, shadows, nighttime)
- Limited demographic and mask-type diversity

2. Binary Classification Constraint

The model cannot detect improperly worn masks (below nose, on chin), which is a significant real-world use case.

3. Deployment Challenges

- **Computational cost:** EfficientNet requires significant resources for edge devices
- **Real-time processing:** Video stream inference may need optimization (affecting accuracy)
- **Domain shift:** Performance may degrade with different camera angles, resolutions, or environments

4. Generalization

The validation-test gap observed during development indicates potential challenges when encountering data distributions different from training.

Despite these limitations, the final model's 98.31% accuracy with balanced performance demonstrates strong suitability for controlled environments and provides a solid foundation for future improvements.

Future Improvements

Several enhancements can be applied to improve the performance and robustness of the face mask detection system in future work :

1) the use of transfer learning with pre-trained models such as MobileNetV2, EfficientNet, or ResNet, which can significantly increase accuracy while reducing training time.

2) The classification task can also be extended to include an additional class for incorrectly worn masks, allowing the system to better handle real-world scenarios. Improving data diversity by collecting more real-world images from different environments and camera qualities would further enhance model generalization.

Additional improvements include incorporating a face detection and alignment step before classification, optimizing the model for real-time deployment on edge devices, and applying model compression techniques such as pruning or quantization to reduce computational cost.

References

- **Kaggle Face Mask Detection Dataset**
  https://www.kaggle.com/datasets/andrewmvd/face-mask-detection
- **TensorFlow Documentation**
  https://www.tensorflow.org
- **Keras Documentation**
  https://keras.io
- **Scikit-learn documentation**

  https://scikit-learn.org/stable/index.html