Object Detection - Covid face mask

Venkata Sai Mahendra Somineni
vsominen

Chandini Kondinolu
Ckondino

Vaishnavi Koyyada
vkoyyada

Abstract

In light of the widespread Covid-19 outbreak, it has become imperative to accurately detect individuals wearing facial masks as a means to control its transmission. The purpose of this report is to delve into several object detection and classification algorithms, with the specific goal of identifying people who are wearing masks versus those who are not. To achieve this, a carefully curated dataset has been utilized for the investigation. The main focus of this study revolves around assessing and contrasting the performance of various solutions, aiming to identify the most optimal algorithm for facial mask detection.

1 Dataset

1.1 Description of the dataset

The "Face Mask Detection" Dataset sourced from Kaggle comprises 853 images, distributed across three distinct classes:

- With mask
- Without mask
- Mask worn incorrectly

For the purpose of object detection models, the dataset is structured with the images as the input and corresponding bounding boxes and labels as ground truth. The images are provided in the .png file format, and the annotations are stored in XML files. These XML files contain crucial information such as bounding box coordinates and object labels, facilitating accurate object detection.

On the other hand, for classification models, a specialized dataset is created by utilizing the images and extracting bounding boxes around the faces. By leveraging the information from the XML files, the face images are cropped appropriately. The label information from the XML files is then assigned to each cropped face image, turning it into a classification task. The classification dataset consists of a total of 4072 face images, classified as follows:

- With mask: 3232 images
- Mask worn incorrectly: 123 images
- Without mask: 717 images

1.2 Data engineering

The dataset is preprocessed for object detection by using images and bounding box annotations.

For classification models:

Faces are cropped using bounding box info. Labels are assigned for classification. These steps ensure a well-organized dataset suitable for both detection and classification tasks, enabling the study of facial mask detection with diverse algorithms.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

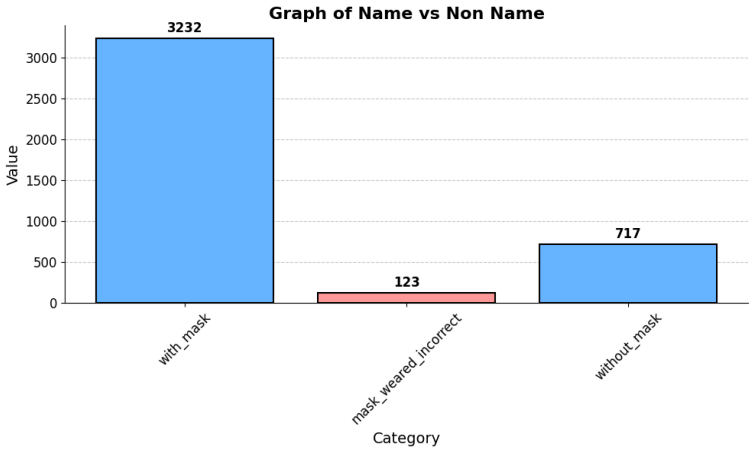


Figure 1: Data description

- Faces are cropped using bounding box information.
- Labels are assigned for classification.

2 Model Description

We conducted experiments using both Classification and Object Detection models to address two distinct problems.

2.1 VGG 16 - Classification

We utilized the VGG 16 model for classifying faces. It is a deep neural network with 16 layers, composed of sequentially connected CNN blocks with a substantial number of filters in each layer. The model employs 3x3 filters and ReLU activation function. Max pooling is applied after each CNN layer to reduce spatial dimensions while preserving essential information. The final layers consist of fully connected layers to perform the classification.

2.2 ResNet 32 - Classification

For face classification, we employed ResNet 32, a deep neural network leveraging residual learning. This concept incorporates skip connections or shortcuts that enable the network to learn residual mappings, making it easier to train deeper networks effectively. Each residual block in ResNet 32 comprises two or three convolutional layers with batch normalization and ReLU activations. The skip connections facilitate handling deeper architectures without significantly increasing computational complexity.

2.3 Efficient Net - Classification

In our classification tasks, we have incorporated the EfficientNet model. This architecture optimizes neural network scaling by systematically balancing depth, width, and resolution to achieve superior performance with fewer parameters. EfficientNet's compound scaling and efficient building blocks enable it to achieve remarkable accuracy while efficiently utilizing computational resources. By integrating EfficientNet into our framework, we enhance the effectiveness of our classification tasks, particularly in scenarios where resource efficiency is a priority.

2.4 MTCNN - Object Detection

We used MTCNN as a face detection model to identify faces, followed by applying the VGG 16 or ResNet 32 models for classifying faces with and without masks. MTCNN enables us to integrate the classification models effectively for this specific object detection task.

2.5 Faster R-CNN with ResNet50-FPN - Object Detection

This object detection model extends the original Faster R-CNN architecture by incorporating a Region Proposal Network (RPN) with a Fast R-CNN detector. The "ResNet50" backbone network is based on the ResNet-50 architecture, known for its depth and image recognition performance. The model is equipped with the Feature Pyramid Network (FPN) structure, enhancing its ability to detect objects of various sizes and maintain spatial context within an image. This combination allows the model to efficiently detect objects at different scales, making it highly suitable for real-world object detection challenges.

3 Loss Function

In our classification models, namely VGG 16 and ResNet 32, we utilize the Cross Entropy loss function for handling the classification task. Cross Entropy loss measures the dissimilarity between predicted and actual class labels, acting as a log likelihood loss.

For the object detection model, we employ various loss functions to ensure effective bounding box generation and accurate classification:

3.1 Region Proposal Network (RPN) Loss

The RPN is responsible for generating bounding boxes around the objects of interest. Its loss function comprises two key components:

Classification Loss (Cross Entropy Loss): This component predicts whether a region corresponds to an object or background, aiding in accurate object proposal generation.

Regression Loss: The regression loss refines the predicted bounding box coordinates relative to the ground truth boxes, improving the accuracy of the bounding box predictions.

3.2 Fast R-CNN Loss

Applied to the region proposals generated by the RPN, the Fast R-CNN loss consists of two main components:

Classification Loss (Cross Entropy Loss): This component assigns the correct class label to each region proposal, ensuring accurate classification of objects.

Regression Loss: The regression loss further refines the predicted bounding box coordinates, fine-tuning the bounding box predictions for greater precision. These specialized loss functions play a crucial role in training our object detection model, enabling it to accurately detect objects and refine bounding boxes for real-world object detection challenges, including face detection in the context of our study.

4 Optimization Algorithm

In our experiments, we used two different optimization algorithms to train various models:

4.1 Adam Optimizer:

Adam stands for "Adaptive Moment Estimation." It is an adaptive learning rate optimization algorithm that combines the benefits of both AdaGrad and RMSProp. Adam optimizes the learning process by adjusting the learning rate for each parameter individually, allowing faster convergence and improved performance, especially for large-scale deep learning tasks.

4.2 SGD Optimizer:

SGD refers to "Stochastic Gradient Descent." It is a classical optimization algorithm used for training neural networks. In each iteration, SGD updates the model's parameters by computing the gradients based on a randomly selected mini-batch of training data. While SGD is computationally efficient, it may converge slowly and require fine-tuning of the learning rate to achieve the best results.

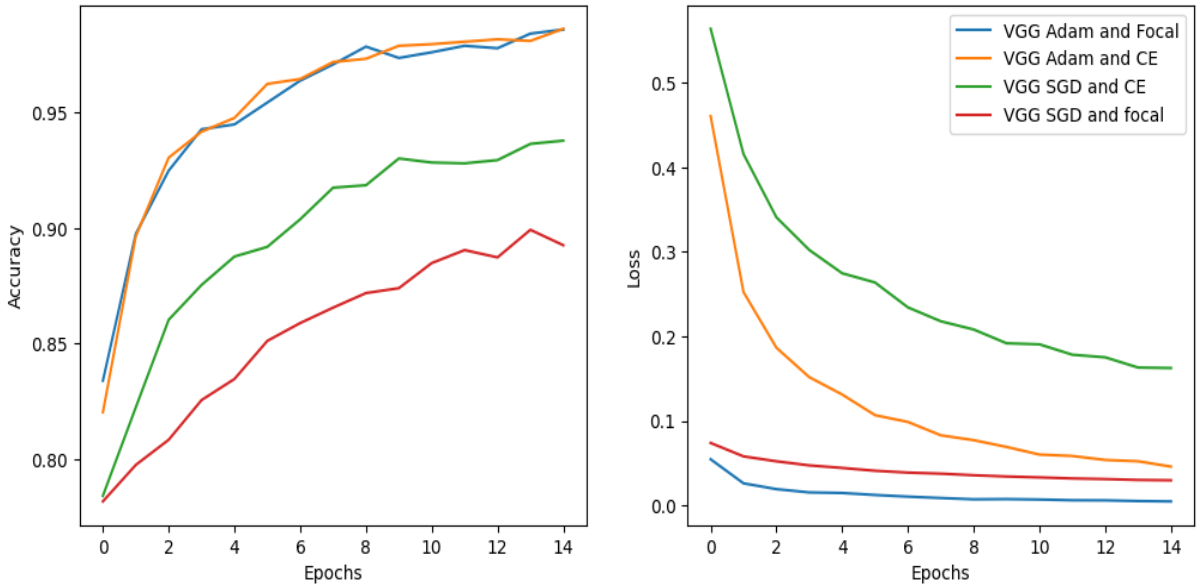


Figure 2: VGG 16 Optimization algorithms comparison

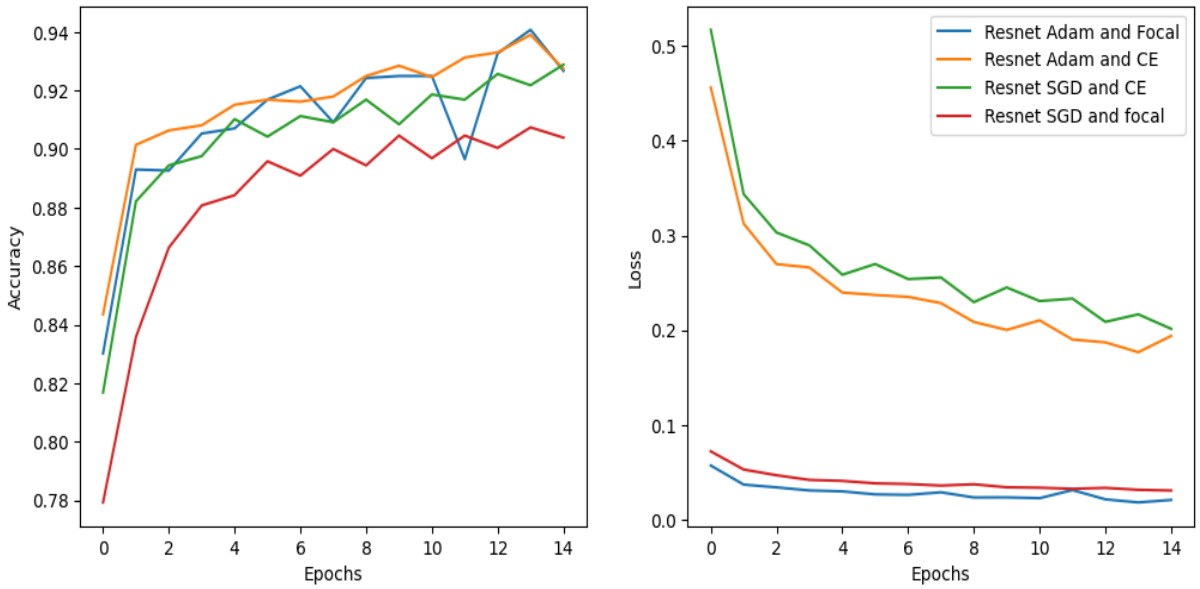


Figure 3: Resnet Optimization algorithms comparison

4.3 Innovations on Optimization Algorithm:

While we experimented with various optimization algorithms, we did not implement any specific innovations on the existing algorithms. Our primary focus was on comparing the performance of different models with standard optimization algorithms to identify the most effective approach for our facial mask detection task.

4.4 Comparisons

Figures 2 through 9 show the different comparisons between the optimizations algorithms within the same model and with other model too.

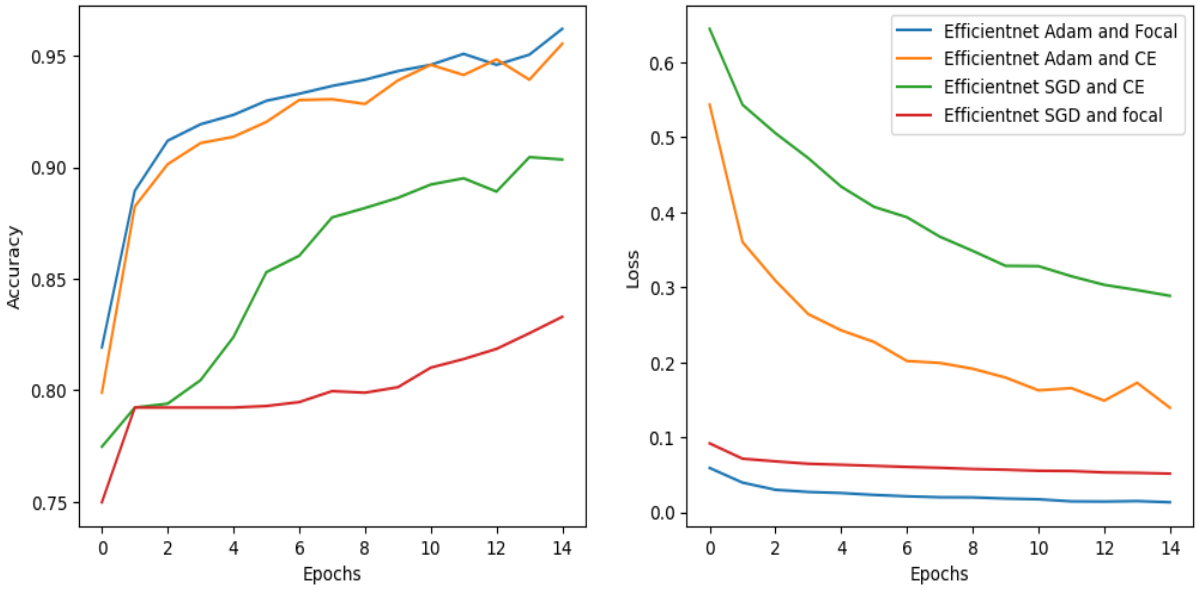


Figure 4: Efficient Net Optimization algorithms comparison

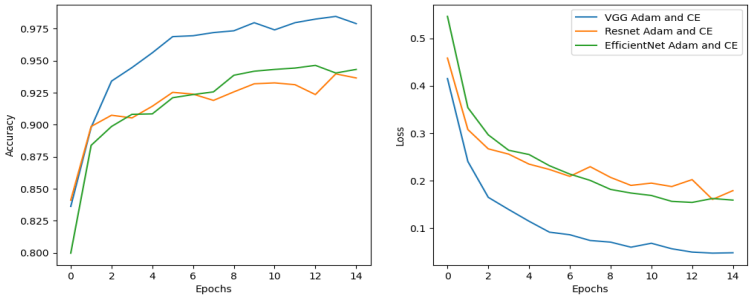


Figure 5: VGG 16 vs Resnet vs Efficientnet

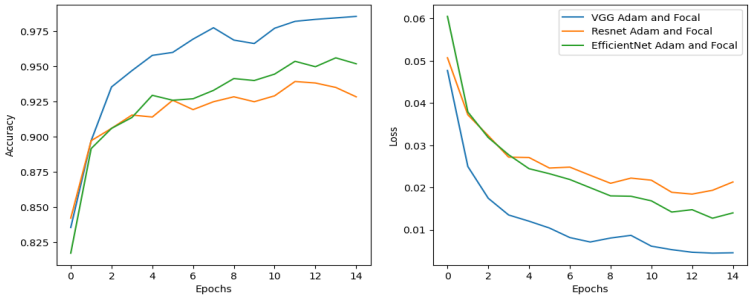


Figure 6: VGG 16 vs Resnet vs Efficientnet

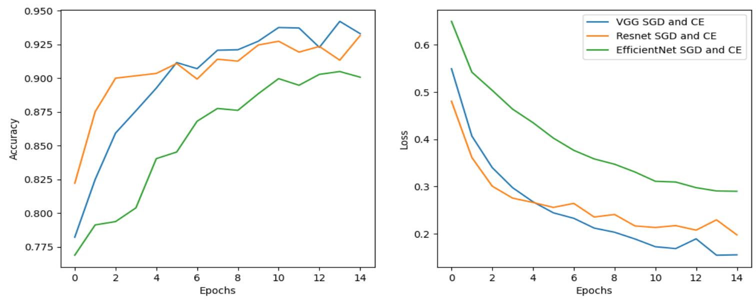


Figure 7: VGG 16 vs Resnet vs Efficientnet

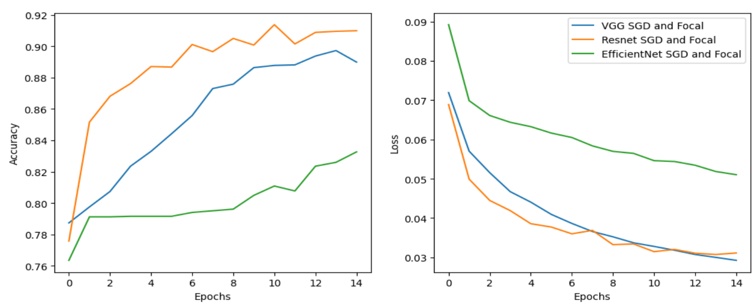


Figure 8: VGG 16 vs Resnet vs Efficientnet

5 Metrics and experimental results

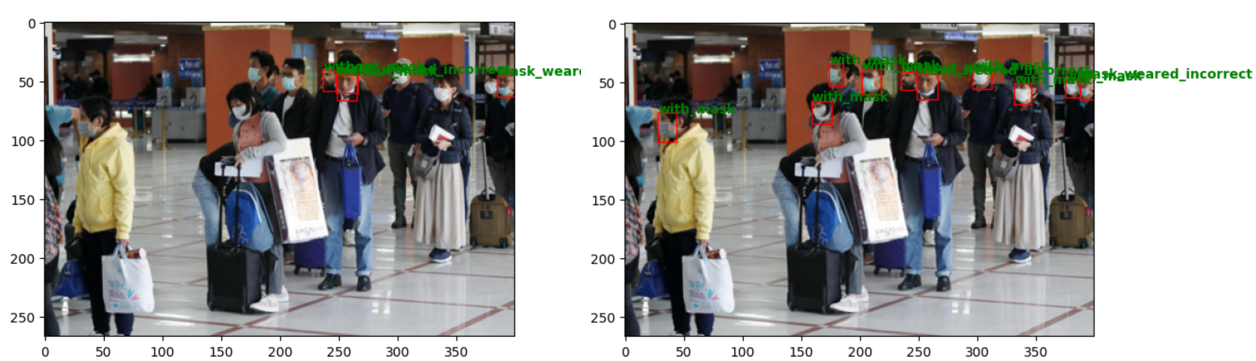
For classification models, the below image results are shown.

Test set Accuracy: This can be observed in the figures 2, 3 and 4 for VGG 16, Resnet 32 and Efficientnet respectively.

Experimental results: Figures x through x show the different results.

6 Member Contribution

The team member contribution is mentioned in table 1.

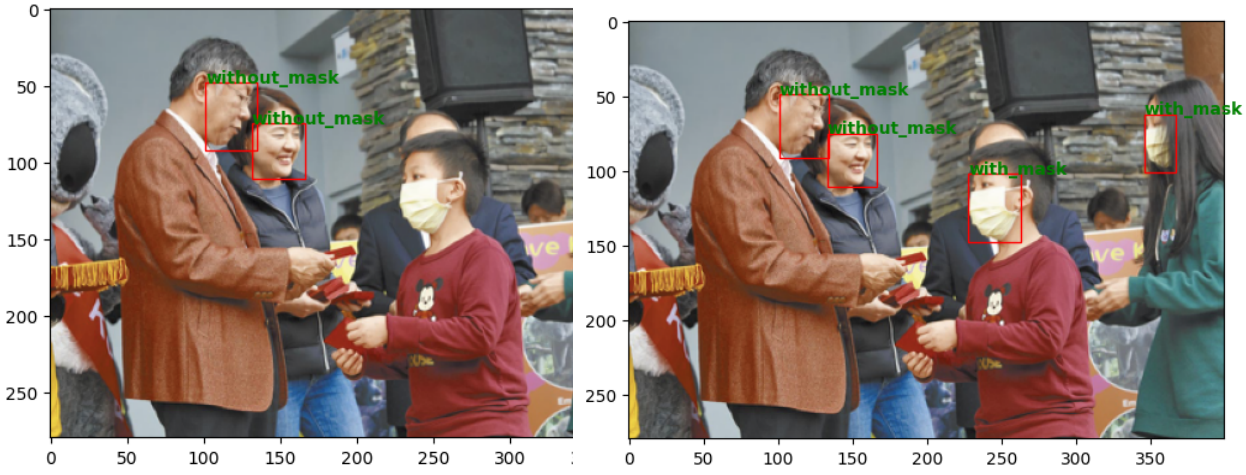


(a) Prediction

(b) Target

Figure 9

Figure 10: Prediction and Target



(a) Prediction

(b) Target

Figure 11

UBIT ID	Percentage
vsominen	33.33
ckondino	33.33
vkoyyada	33.33

Table 1: Team Contribution

References

[1] Tan, M., Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[2] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).