Face mask detection

Vaishwi Patel
B00914336
Master of Applied Computer Science
Dalhousie University
Halifax, Canada
vs439755@dal.ca

Purvesh Rathod
B00903204
Master of Applied Computer Science
Dalhousie University
Halifax, Canada
purvesh.r@dal.ca

Ketul Patel
B00900957
Master of Applied Computer Science
Dalhousie University
Halifax, Canada
kt484025@dal.ca

Abstract—**Despite the fact that the corona virus has been out for two years, wearing a mask is still required in some places, such as airports and hospitals. To enforce the rule against wearing masks, one must monitor whether or not individuals are following the rule, necessitating the requirement for a system that monitors whether or not everyone is obeying the rule. This study intends to contribute to the solution of this problem by implementing a face mask detection system that can monitor whether individuals are wearing masks, wearing masks improperly, or not wearing masks at all. To build such a system, we used pre-trained models as a feature extraction with our own classification layer. This study has taken three pre-trained model including MobileNetV2, ResNet50 and InceptionV3 trained on a massive ImageNet dataset and explored the performances of the same using measures such as precision, testing loss, and training time. It was discovered that models using MobileNetV2 and InceptionV3 had similar accuracy while training time for MobileNetV2 was exceptionally low with minimum training loss.**

Index Terms—**CNN, COVID-19, face mask, MobileNetV2, ResNet50, InceptionV3**

I. Introduction

Even though the COVID-19 term was identified two years ago, still many organization mandates wearing a mask on their premises to prevent people from getting infected with any virus. Organizations like airports and hospitals require visitors to wear masks when entering the building. To make sure everyone wears masks, an automated system would be a preferable solution rather than hiring a person to do the job. Introducing an automated system with Artificial Intelligence capability would benefit an organization to keep a safe environment with a minimal financial cost. Furthermore, when the mask rule is enforced, the system can detect with greater accuracy than a person and can also identify someone wearing an inappropriate mask.

Machine learning and computer vision techniques can be used to build such a system. The initial stage of implementation is to identify a person's face using live-capture equipment. Currently, machine learning methods including HaarCascade, Yolo, RetinaFace, and MTCNN have been utilized to identify faces. Based on the placement of the nose, eyes, ears, and mouth as well as the space between them, these techniques identify faces. However, when a person is wearing a mask, facial features like the mouth and nose are hidden. It makes it challenging for methods to identify faces. The dataset used to train these models had not included any masked faces. As a result of that, the performance of models was reduced, and asked the question to provide an efficient solution for the problem of face detection with masks.

Global researchers used a variety of techniques to identify an effective answer. In the following section, the article will shed some light on the many approaches that have been tried, along with their benefits and drawbacks. Our primary goal was to come up with a distinct and superior approach, we tested various model designs. section III explains the specifics of those. Model performance must be assessed using the right evaluation metrics, which are provided by the evaluation process. As a result, before coming to our final conclusion, we used statistical analysis to evaluate the model, compare its performance, and provide step-by-step information about it in section IV.

II. Related Work

There are a myriad number of research already conducted to solve the problem of facial mask detection. Most of the studies used transfer learning where different versions of the pre-trained model such as MobileNetV2, ResNet50, InceptionV3, Logistic Regression, and VGG-16 were taken and changed the architecture according to the requirement. When developing our machine learning model into practice, upcoming implementations were given priority.

A. *MobileNetV2*

Wearing a mask is advised by the World Health Organization (WHO) to prevent the spread of the coronavirus. To find the face mask, the authors compared the three deep-learning architectures. 1340 total photos were divided into two categories for this classification: those with masks and those without. These images came from three separate sources [1]. Images from a camera and a Mobile device are both included in the dataset. Additionally, data from simulated and real-world faces with masks were gathered to strengthen and generalize the model. Images of various sizes were downsized to 256x256 [1].

The MobileNetV2 architecture, which consists of three 2D Convolution Neural Network (CNN) layers with Max pooling and Average pooling with ReLU activation function, has been suggested by the authors above the other two architectures. The face is recognized using the Caffe model in OpenCV. The optimizer employed Adam stochastic gradient descent methods. The author trained the model using 80% of the whole dataset [1]. The weights of the model trained on the ImageNet dataset were used by the authors to train MobileNetV2 and the initial layers were frozen. To prevent overfitting, a drop-out layer has been placed at the top [1].

MobileNetV2, trained by transfer learning, demonstrated promising accuracy on the 13th epoch with a validation accuracy of 99.82% [1]. As the model was trained on a sizable ImageNet dataset, transfer learning has the advantage of reliably extracting characteristics from the image. Additionally, MobileNetV2 architecture is portable and simple to set up on edge/IoT devices. Training time is also shortened by this approach [1]. Although accuracy should be as high as possible, the loss is also considerable. Instead of comparing the architecture with just the convolution layers, it is essential to compare the model with other models of the same level of complexity.

B. *InceptionV3 and Logistic Regression*

The study aims to develop an architecture for developing a face mask detection model by combining Transfer Learning (TL) and Deep Learning approaches. The InceptionV3 model is used to implement transfer learning in order to extract features from images. The InceptionV3 model is a 48-layer deep pre-trained CNN trained on over a million images from the ImageNet collection [2].

Deep learning algorithms are used to classify people's faces with, without, and with partial masks. To classify people's faces, several classification methods such as Random Forest, Logistic regression, CNN, and SVM are used [2]. They built their own dataset of 771 photos to train the model, which includes faces with complete markings, partial masks (mouth-chin/only chin), and regular faces without marks. The experiments were carried out using InceptionV3 in conjunction with various algorithms, and they reached a maximum accuracy of 96.3% utilizing InceptionV3 with the Logistic regression model [2].

The study offers a novel solution for developing a classification model using transfer learning that not only reduces training time but also performs well with a limited dataset [2]. Furthermore, the architecture enables solutions that are not just associated with mask detection but can also be utilized to broaden the use case, such as constructing video surveillance systems or biometrics. However, while designing such an architecture, it is important to remember that transfer learning might employ a negative transfer issue, which may result in a model with worse accuracy.

C. *Convolution Neural Network*

For the most part, picture categorization, detection, and identification issues have been resolved using Convolution Neural Networks. A study was undertaken by Rizki Purnama Sidik to develop a CNN method that can classify mask usage into ”masked” and ”non-masked” categories. [3].

4423 photos of faces at various camera angles that were 224x224 in size and covered just one face were taken into consideration for the model's training and testing. On the other hand, 250 photographs taken in public areas under uncontrolled circumstances at different angles with the subject's face toward the camera were used to assess the model's performance [3]. The training dataset was used to hold weights for CNN. However, masked face detection employed a 640 x 480-pixel CCTV image with numerous faces that were either layered or not [3].

Convolution neural networks had a two-part architecture, the first of which was the responsibility of feature extraction using convolution and pooling. While the second component was utilized to determine the image's class, such as whether it was masked or not [3]. The feature map created from the convolution part of the first component was passed to the fully connected layer of CNN to convert the feature metrics into vectors. The flattened findings were then fed into a multi-layer perceptron algorithm that has three layers: input, hiding, and output [3]. The softmax function received the output weights and returned the likelihood of masking and not masking images. Finally, the Cross-Entropy loss function was used to gauge how well the model performed [3].

The model provided nearly 80% accuracy when trying to detect the masked/non-masked faces in images captured by CCTV [3]. The study revealed that the accuracy of the Convolution model was dependent on the face detection model. Considering the accuracy provided by face-detection models, it can be said that to overcome the issue of real-time mask detection in public spaces, additional performance enhancements are still required.

III. Methodology

We chose the top-down strategy while keeping the problem description in mind. We needed to start by locating an image dataset comprising pictures of people who were properly, incorrectly, and not masking their faces. For that, there were a ton of materials online. Before selecting it, we had to confirm that the dataset we had gathered was appropriate for our field. To be sure of that, we made the decision to visualize the dataset in order to learn more about it. Then, using pre-processing techniques, we made the data ready to ensure that it was evenly distributed and that the learning process would not be impacted by the dataset's order [4]. The pre-processed dataset was then divided into three sets: training, validation, and testing. The model was trained using the training set. The validation data, which is intended to provide an estimate of model skill while adjusting the model's hyper-parameters, were withheld from training your model [5]. While the testing set was chosen to evaluate the model's efficiency following training In the end, we tried three alternative models. The project's high-level process flow diagram can be seen in the Figure 1.

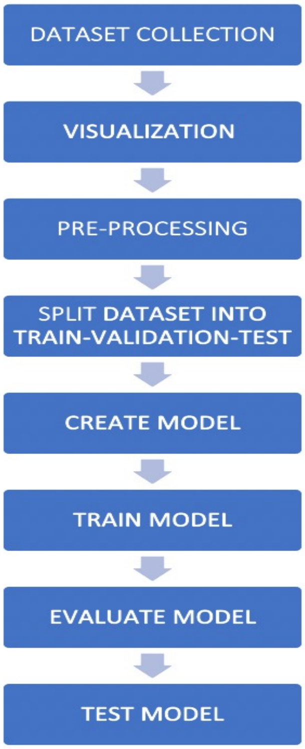


Fig. 1. Flow chart of methodology

A. *Dataset*

We created our model by using the Kaggle dataset contributed by LARXEL [6] comprises 8949 pictures with three color channels. The dataset is categorized into three folders, each labeled with the class to which it belongs namely: ”proper” ”improper” and ”no mask” as illustrated in Figure 2. The classes ”proper” and ”no mask” contains an equal number of images while ”improper” contains less number of images shown in Figure 3. The dataset contains images of people of all ages taken from various viewpoints and distances, allowing the model to train using diverse angles. Images are input into the model in RGB format, which provides more information than a grayscale image. Figure 4 illustrates the image's characteristics in each layer of the image.

TABLE I
NUMBER OF IMAGES IN EACH DATASET.

Type of Dataset	Number of images
Train	5727
Validation	1432
Test	1790

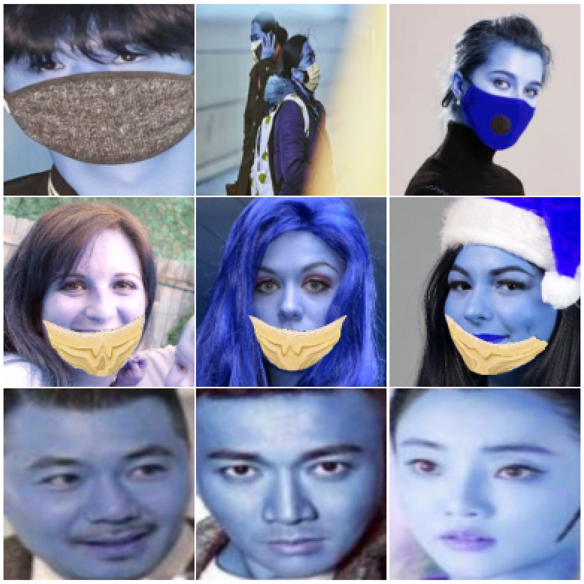


Fig. 2. Image visualization of 3 classes.

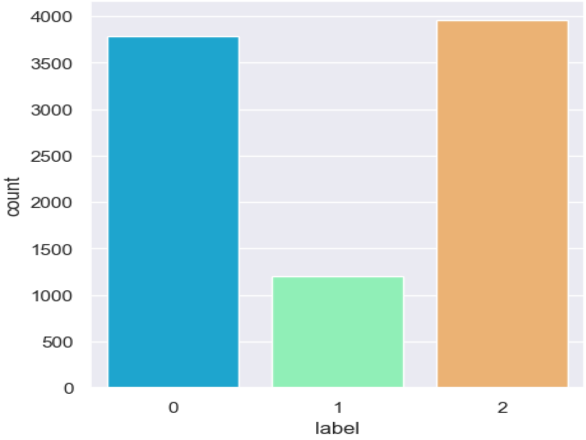


Fig. 3. Image distribution count based on the class

B. *Pre-Processing*

Preprocessing data is an important step for data modeling. It is performed by removing the missing data, normalizing, and making data consistent for a more accurate outcome. The dataset is an image dataset, hence it won't have any missing data. Normalization is a scaling technique that changes the range of pixel intensity values. We've performed 3 types of normalization on our image data:

Fig. 4. RGB color channel

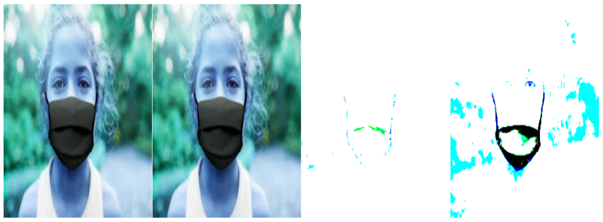


Fig. 5. Normalization

- Min-Max normalization with 0 to 1 pixel value:

$$img = \frac{img}{255} \tag{1}$$

- Normalize the image based on the image range:

$$img = \frac{min(img)}{max(img)} - min(img) \tag{2}$$

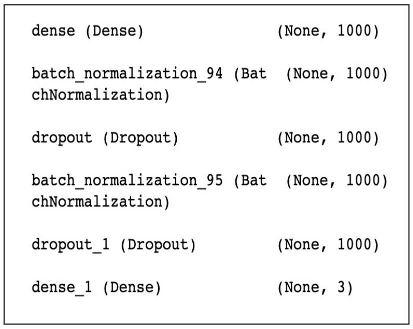
- Normalize by percentile:

$$img = \frac{5^{th}_{percentile}(img)}{95^{th}_{percentile}(img)} - 5^{th}_{percentile}(img) \tag{3}$$

From Figure 4 we can see that Equation 1 normalization technique scales the image while preserving the features. Hence, we have used this normalizing technique for the dataset.

C. *Models*

Based on our research for image classification models, we decided to use transfer learning with precautions of overfitting. In research, there is no comparison of a standard architecture for face mask detection. That's why we decided to experiment with a different standard pre-trained model like MobileNetv2, InceptionV3, and ResNet50 with a few more layers on it to avoid overfitting. Layers include dense, batch normalization, and drop out. All these layers help to avoid the problem of overfitting. All these models are trained on a huge dataset ImageNet consisting of approximately 1 million images so it learned the rich feature representation which helps to extract the feature efficiently for this task as well. Other hyperparameters are kept the same for all the models as shown in the table below.



```
dense (Dense)               (None, 1000)

batch_normalization_94 (Bat  (None, 1000)
chNormalization)

dropout (Dropout)           (None, 1000)

batch_normalization_95 (Bat  (None, 1000)
chNormalization)

dropout_1 (Dropout)         (None, 1000)

dense_1 (Dense)             (None, 3)
```

Fig. 6. Classification layers

TABLE II
LIST OF HYPERPARAMETES

Hyperparameter	Value
Optimizer	Adam
Learning rate	0.000001
Batch size	10
Epoch	50
Dropout Rate	0.3
Loss function	Sparse categorical loss

1) *MobileNetV2:*

MobileNetV2 is a 53-layer deep image classifier designed for embedded applications. MobileNetV2 architecture uses depth-wise separable convolutions, which consist of a depth-wise and a point-wise convolution after one another [7]. This type of convolution makes the model faster with lesser parameters which can be easily deployed on edge devices.

Furthermore, MobileNetV2 has two different blocks than MobileNet: Inverted residual block and Linear bottleneck. In the inverted residual block, the author has introduced the concept of the skip connection so that one can go deeper into the network while preserving the information from the previous layers. This architecture uses the ReLu6 activation function to limit the output range. In Figure 8, the architecture of MobileNetv2 has been defined, in which the input 2D convolution layer is followed by 7 bottleneck blocks [7]. The last few layers are 2 conv2d layers with average pooling as shown in the Figure 8

The bottleneck block consists of 3 layers, 1x1 convolution which widens the input followed by 3x3 depth-wise convolution, and 1x1 convolution layers with skip connection which means the output of the bottleneck will be input added to the last layer of output as shown in the Figure 7 [7].

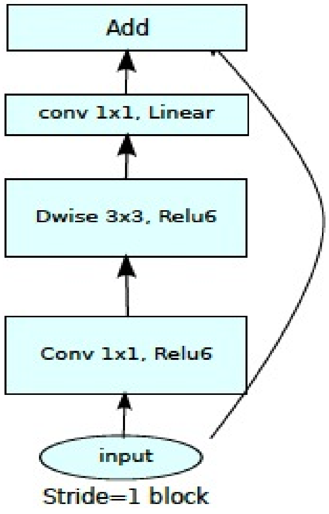
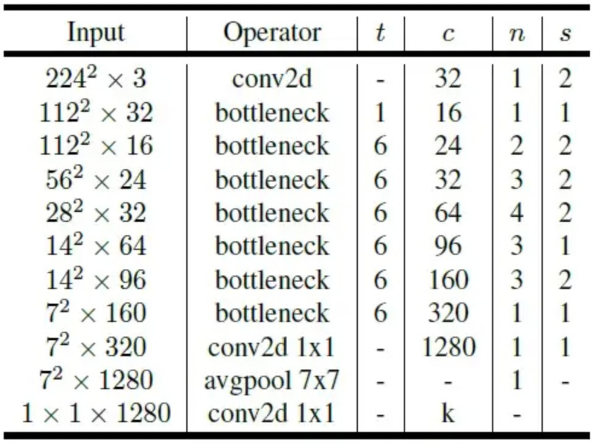


Fig. 7. Bottleneck block of MobileNetV2 Architecture [7]



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Fig. 8. MobileNetV2Architecture [7]

2) *Residual Network (ResNet50):*

Convolution neural networks are said to benefit from being deeper the better. However, some studies have shown that performance deteriorates beyond a certain depth. Therefore, the gradients from which the loss function is calculated quickly drop to zero after many chain rule applications. [8].As a result, since the weights' values are never updated, learning is not happening. Gradients from later layers to the first filters can pass right through the skip connections when using ResNets. [8]. According to the idea, the training error decreases as the layer depth increases. However, in practice, it has been noted that the training error increases after a certain depth [10].ResNet50 makes use of a technique called skip connection to get around this issue. Convolution layers are stacked on top of one another, as seen in the figure on the left. On the right, we continue to stack convolution layers in the same manner as before, but now we include the original input in the convolution block's output [9]. This is called a skip connection or shortcut. The Residual Network mainly consists of six layers. The input processing layer comes first. Then, four separate Convolution Neutral Network blocks at various levels based on the various versions, followed by the Fully Connected Layer, as shown in the below image [11].

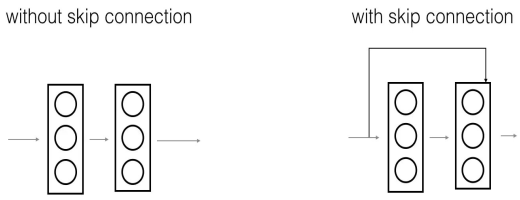


Fig. 9. Plain V/S ResNet Network [9]

3) *InceptionV3:*

InceptionV3 [12] is an enhanced network of the well-known GoogLeNet, which has achieved higher classification performance in a variety of applications applying transfer learning. Simple Convolution Neural Networks only use one convolution layer at each level, therefore when we build more complex CNN, the data becomes overfitted. The InceptionV3 V1 model employs the concept of having many filters of various sizes on the same level to prevent this from occurring. As a result, the model is wider rather than deeper in the InceptionV3 models since we have parallel layers instead of deep ones. This type of architecture lowers the number of parameters to be taught and hence the computational complexity. Figure 11 depicts the fundamental architecture of InceptionV3 [13].

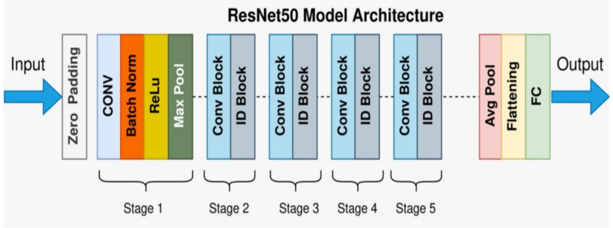


Fig. 10. General Architecture for Residual Networks [11]

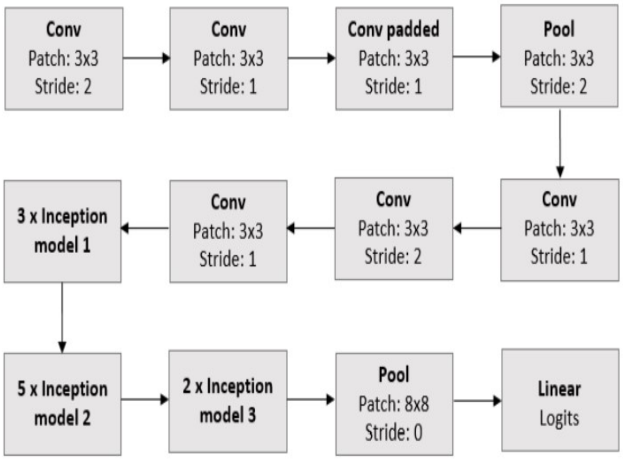

Conv Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
Conv padded Patch: 3x3 Stride: 1
Pool Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
Conv Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
3 x Inception model 1
5 x Inception model 2
2 x Inception model 3
Pool Patch: 8x8 Stride: 0
Linear Logits

Fig. 11. The basic architecture of InceptionV3 [14]

IV. Experiments

The validation loss and accuracy graphs were compared after the models had been successfully trained. Figure 12 shows that while MobileNetV2 rose and ResNet50 fluctuated after each epoch starting at 1, InceptionV3 model validation accuracy is almost identical. Figure 13 shows that after each epoch from 1 to 50, the loss values for ResNet50 varied, MobileNetV2 shrank, and InceptionV3 was nearly the same.

There are many evaluation metrics available to assess any model's performance, including precision, recall, AUC, accuracy, mean-absolute error, F1-Score, etc. As a result, choosing amongst them becomes challenging. we had to decide between

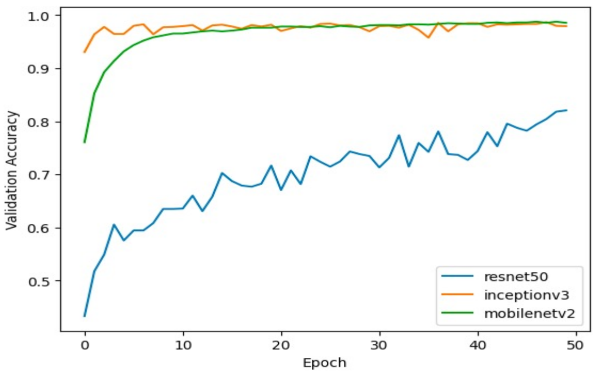


Fig. 12. Validation Accuracy comparison

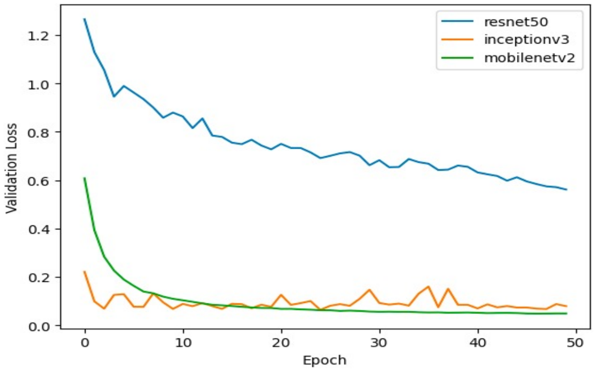


Fig. 13. Valdiation Loss comparison

Accuracy, Precision, Recall, F1-score, and AUC value because this was a multi-class classification problem. As from the Figure 2, it can be said that the dataset was imbalanced. Recall and Precision are suitable options for evaluation metrics. Between them, we made the decision to use Precision as our measurement standard. The table contains the findings for the Precision Value for each model for each class. According to the 'Table III, ResNet50's performance was the worst among the three models for every class. The precision values produced by the MobileNetV2 and InceptionV3 models were nearly identical for each class.

TABLE III
PRECISION VALUES FOR EACH MODEL FOR EACH CLASS.

Model-1	Model-2	P-Value
MobileNetV2	Mask	0.99
	Improper-mask	0.98
	Non-mask	0.99
ResNet50	Mask	0.80
	Improper-mask	0.71
	Non-mask	0.78
InceptionV3	Mask	0.99
	Improper-mask	0.99
	Non-mask	0.97

Even though there are ways to assess the effectiveness of the model, such as the k-fold, occasionally that method can lead us astray because it is difficult to tell whether the difference between the mean skill scores is a true difference or a statistical fluke. As a result, we chose to compare the performance of the model using a statistical significance test called the t-test. The'Table IV presents the findings. P-Value comparisons between models show that the ResNet50 model's performance was not statistically different from the other two models. There was no effect between MobileNetV2 and InceptionV3 because their P-Values were higher than 0.05.

TABLE IV
P-VALUE AFTER APPLYING T-TEST BETWEEN MODELS.

Model type	Class	P value
MobileNetV2	InceptionV3	0.067
ResNet50	MobileNetV2	2.74 x e^{38}
InceptionV3	ResNet50	3.72 x e^{43}

V. CONCLUSION AND FUTURE WORK

In this paper, we studied the topic of identifying face masks. We conducted experiments to determine the optimum face detection model capable of recognizing faces in various situations. For face mask recognition, we tested three pre-trained models as feature extraction layers paired with our classification layers. According to our observations, ResNet50 required a long time to train while providing low accuracy when compared to other models. Furthermore, MobileNetV2 and InceptionV3 performed similarly, although MobileNetV2 required less time to train while showing low training loss.

For future improvements, as the acquired dataset was imbalanced with very few images for the improper masks in comparison to others, we can work on balancing the dataset. Moreover, it is possible to add or remove layers from the model for future improvements because the performance of the model depends on the architecture's layers. One limitation of our research was that we tested different models with the same hyperparameters. However, different hyperparameters may be evaluated with several model layers and versions to improve model performance.

REFERENCES

[1] F. J. M. Shamrat, S. Chakraborty, M. M. Billah, M. Al Jubair, M. S. Islam, and R. Ranjan, "Face mask detection using convolutional neural network (cnn) to reduce the spread of covid-19," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2021, pp. 1231–1237.

[2] S. Reddy, S. Goel, and R. Nijhawan, "Real-time face mask detection using machine learning/deep feature-based classifiers for face mask recognition," in *2021 IEEE Bombay Section Signature Conference (IBSSC)*. IEEE, 2021, pp. 1–6.

[3] R. P. Sidik and E. Contessa Djamal, "Face mask detection using convolutional neural network," in *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*. IEEE, 2021, p. 85–89.

[4] Y. G, "The 7 steps of machine learning." Towards Data Science, Sep 2017. [Online]. Available: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e

[5] J. Brownlee, "What is the difference between test and validation datasets?" Aug 2020. [Online]. Available: https://machinelearningmastery.com/difference-test-validation-datasets/

[6] Larxel, "Face mask detection," May 2020. [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/face-mask-detection

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[8] P. Ruiz, "Understanding and visualizing resnets," Apr 2019. [Online]. Available: https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

[9] P. Dwivedi, "Understanding and coding a resnet in keras," Mar 2019. [Online]. Available: https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

[10] A. Ng, "C4w2l03 resnets," Nov 2017. [Online]. Available: https://www.youtube.com/watch?v=ZILIbUvp5lk

[11] S. Mukherjee, "The annotated resnet-50," Aug 2022. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] A. N. T, "Inception v3 model architecture." OpenGenus IQ: Computing Expertise amp; Legacy, Oct 2021. [Online]. Available: https://iq.opengenus.org/inception-v3-model-architecture/

[14] L. Nguyen, D. Lin, Z. Lin, and J. Cao, "Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation," 05 2018, pp. 1–5.