ABSTRACT

During the COVID-19 pandemic, wearing a proper mask became crucial for public health. In view of this automated systems were created in order to monitor whether or not people were following the mask protocols. The traditional approaches in this field relied on complex handcrafted features or simple models which did not perform very well. This study tries to tackle this problem of object detection of a mask using well established and powerful convolutional neural network architectures; MobileNetV2 and VGG16 and utilizes the two models to conduct a systematic review of the trade-off between accuracy and computational resources.

The models were fine-tuned on a Kaggle dataset of over 8900 images labeled for the three mask conditions. Data augmentation techniques were employed to increase the size of training data. Transfer learning from models pretrained on ImageNet allowed leveraging their powerful feature representations while fine-tuning their classification layers for the task at hand of mask detection.

The MobileNetV2 is a highly efficient architecture designed for use in mobile devices, due to its good performance and low computational requirement whereas the VGG16 is a high-performance model capable of very high accuracy given sufficient data. Extensive experiments have compared their performance on classification tasks in terms of accuracy, inference speed, and model size.

The VGG16 model achieved state-of-the-art accuracy of 99.05% on the test set, while the more compact MobileNetV2 traded some accuracy for much faster inference speed, hitting 98.05% accuracy. This enables flexible deployment based on use case requirements - high accuracy or mobile deployment. Potential applications include contactless mask compliance monitoring in public places like airports, malls, and workplaces to aid pandemic control efforts

1. INTRODUCTION

In the year of 2020, COVID-19 pandemic triggered an unprecedented global health crisis, putting immense strain on medical systems worldwide. In order to control the spread of the highly contagious virus, masks were made mandatory across many countries as a basic preventative measure. However, the task of ensuring that every citizen was following the regulations and wearing a mask in public places proved to be a challenging task. Manual monitoring was usually done but it was extremely labor-intensive, a short term and unreliable solution. This created an urgent need for systems capable of automating this process and checking whether a mask is being worn properly or not.

Traditional computer vision approaches relied on algorithms like Viola-Jones for face detection, hand-crafted features and simple classifiers like SVMs to detect mask presence [1]. However, such methods lacked the robustness and complexity required to handle the wide variety of real-world scenarios which are often difficult to analyze on these models. With the progression of deep learning, Convolutional Neural Networks (CNNs) emerged as powerful architectures capable of learning highly discriminative hierarchical features directly from data in an end-to-end manner [2]. This enabled state-of-the-art performance on many visual perception tasks like image classification, object detection and face recognition. The study done by Gu et al. [3] provides comprehensive review of the advances made in CNN's and why they are a great choice

for tasks such as image classification, object detection, object tracking and many more.

In this study, I build upon the promising results for CNN-based mask detection by exploring the trade-off between model capability and computing efficiency. As discussed earlier, I fine-tune and comprehensively evaluate two state of the art CNN architectures - the highly efficient MobileNetV2 [4] described by Howard et. al. which is tailored for mobile deployment, and the high-performance VGG16 [5] created by Simonyan et. al. - on a public three-class masked/improper/unmasked face dataset from Kaggle [6]. I also make use of transfer learning on these models in order to obtain the best possible performance from these architectures.

The primary objective of this study is to provide an in-depth analysis comparing the two models' which are made from MobileNet and VGG16 on their accuracy, inference speed and model footprint. Which will allow people to make an informed decision when choosing a specific architecture based on the accuracy needed and the computational resources available, which is often times an important trade-off to consider when deploying machine learning models such as convolutional neural nets which require high computational resources in real world scenarios.

2. LITERATURE REVIEW

The research conducted by Loey et al. [7] involved creating a deep learning model which would classify images into three categories: wearing face masks, not wearing face masks, and wearing masks incorrectly. They used transfer learning techniques, using the pretrained model ResNet [8] and achieved an impressive accuracy of 99.64% on their best model.
Das et al. [9] proposed a custom convolutional neural network (CNN) model for identifying mask-wearing individuals in images, achieving an accuracy of 95.77%. It is important to note that they did not apply transfer learning which may have improved its performance even further.

An interesting study conducted by Ge et al. [10] proposed using Local Linear Embedding (LLE) as a pre-processing step before sending the images as input data to the CNN. Their experimental results using this approach showed an improvement of 15.6% over 6 stat-of-the-art methods.

Similar works in face mask detection using MobileNetV2 architecture were done by Jiang et al. [11], Kontellis et al. [12], Mercaldo et al. [13] all of these studies provide great insights and information and were able to achieve impressive accuracy on their respective datasets however, their approach did not consider the important case of incorrectly worn masks. Another notable contribution was made by Qin and Li [14], the authors proposed a hybrid model combining the MobileNetV2 architecture and a custom classification head for face mask detection. Their model achieved an accuracy of 98.7% on a publicly available dataset and demonstrated the effectiveness of lightweight architectures like MobileNet for real-time applications.

The Inception Network introduced by Szegedy et al [15] was also considered to be included in this study. As the main goal of Inception Network is to improve the utilization of the computational resources available to the network while maintaining high accuracy. However, due to time and resource constraints, the decision was made to focus on the MobileNetV2 and VGG16 for the study.

Leveraging the learnings and valuable insights of all the above-mentioned research papers, this study aims to further explore the effectiveness of

fine-tuning pretrained models like MobileNetV2 and VGG16 for accurate mask detection, contributing to the efforts towards automating mask compliance monitoring in public settings.

3. METHODOLOGY

As mentioned earlier the two convolutional neural network architectures used in order to tackle this task were MobileNetV2 and VGG16. Both were initialized from weights pretrained on the ImageNet dataset which was introduced by Deng et al [16] to leverage robust visual feature representations learned from large-scale data. They were also further fine-tuned in order to improve their performance and increase their accuracy for the classification problem at hand. The models were constantly monitored while being trained order to identify any errors or problems that may occur such as overfitting. The monitoring also ensured that the modes weren't suffering from high bias or low accuracy. After training and evaluating the model changes in their architecture were made in order to see the changes in the models and understand how it affected their performance. The work done by Tan et al. [17] helped in updating and scaling the model as their research tries to answer the question, "Given the computation budget, what a good choice for the resolution of images, depth and width of a ConvNet". It provides a detailed insight into model scaling and identifying a good tradeoff for accuracy and computational power. Also, to ensure a fair comparison the models were trained using the same hyperparameters and similar training regimen.

The entire model training process was conducted using the TensorFlow framework with Keras as the high-level API. Before delving into the architecture details, let's briefly examine the dataset used for assessing the models.

3.1 Dataset

The Face Mask Detection dataset [6] contained 8982 images belonging to three classes which are masked, improperly masked and unmasked as shown in figure 1. The labeled images were equally distributed among the three classes therefore, each class had a total of 2994 images. The dataset was further divided into training and test set, 80% of the data was used as the training set, and the remaining 20% was kept as test set in order to evaluate the model's performance after the model had been trained successfully. The dataset used had already been cleaned, and data augmentation had been applied to it.

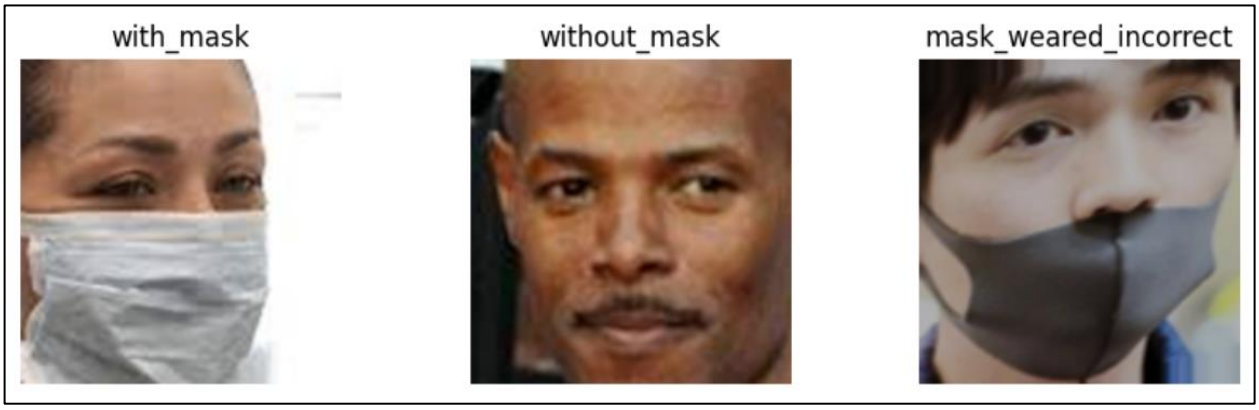

Figure 1: Sample images from the dataset showcasing the three classes: 'Masked', 'Improperly Masked', and 'Unmasked'

3.2 MobileNetV2

MobileNetV2 is an efficient CNN designed for mobile and embedded vision applications where computational constraints are critical. Unlike the traditional convolutional approach which is computationally expensive It uses depth-wise separable convolutions which drastically reduce the computational resources and time required to run the model. Depth-wise separable convolutions have two working parts: depthwise convolution and pointwise convolution.

In depthwise convolution, instead of applying a filter to all the channels in the input and getting an input for the entire image, we have N number of filters where N is equal to the number of channels in the input. One filter is applied to a single channel therefore each filter is responsible for a single channel only. After depthwise convolution we apply pointwise convolution in which we use a 1x1 convolution to shrink the input volume i.e., we reduce the channel depth of the input. These two steps together are referred to as depthwise separable convolution and are responsible for bringing down the computational cost of the MobileNet model [4].

MobileNetV2 also contains skip connections seen in ResNets [8] and addition of an expansion layer which alongside the other layers in the network is collectively referenced as the bottleneck block. This block allows us to increase the volume of the input data, extract features from it and then reduce the volume again before sending it to the next layer.

The MobileNetV2 model pre-trained on ImageNet was used as the base model, all the layers were frozen meaning their weights could not be updated. In order to fit it for the classification task at hand, the top layer which is responsible for classification was removed and replaced with Average Pooling followed by a Dense layer with three neurons and a softmax activation. Adam optimizer [18] was chosen as the optimizer function over Gradient Descent [19] and categorical cross entropy was set as the loss function during compile time. The model was trained using the training data set for 30 epochs. At the end of the training, the model had an accuracy of 99.22% on the training data set and an accuracy of 97.56% on the validation set. When tested on the test set it yielded an accuracy of 97.55%.

In order to leverage the true potential of transfer learning, the model was further fine-tuned, the first revision involved addition of three new layers before the output layer. A new fully connected layer with 12 units and activation function as ReLu [20], and in order to prevent

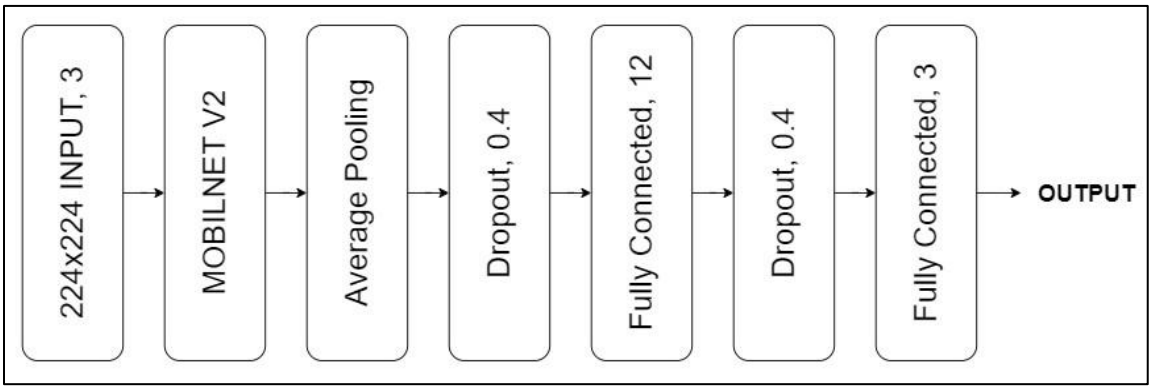

Figure 2: Our Customized MobileNet Architecture

overfitting two Dropout [21] layers were introduced before and after the fully connected layer with dropout rate of 0.4. The dense layer was introduced to enhance the model's capacity for learning complex patterns and the dropout layers were added to reduce overfitting in the model. The model was trained for 20 epochs, it's accuracy on the test set was 96.77% which is a bit less compared to the previous performance.

In the second and final revision Block 15 and Block 16 which are the last two blocks of the MobileNet model were unfrozen meaning, their weights were allowed to be updated during back propagation. This version of the model was trained for 30 epochs and achieved an accuracy of 98.65% and 99.10% on the training and validation set respectively, and on the test set it was able to achieve an accuracy of 98.05%, which is the best performance out of all the three versions.

3.3 VGG16

The VGG16, unlike MobileNet has a high computing cost but in return offers high accuracy given sufficient data, it achieved success in the ImageNet challenge for large-scale image recognition by securing the first position for object detection and the second position for classification. The 16 in its name refers to the number of trainable layers it has which include 13 convolutional layers and 3 fully connected layers organized into 5 blocks.

Its strength lies in its simple and uniform architecture. In order to achieve this uniformity, it uses a small 3x3 filter for every convolution layer with stride as one and padding as same, the number of filters used in every block are also doubled. The convolution layers in each block are also followed by a max pooling layer which uses a filter of size 2x2 and stride of 2. The first two fully connected layers have 4096 units each and the third layer which is the output layer, contains 1000 units in order to perform classification for 1000 classes. ReLu [20] is used as the activation function for all the hidden layers [5].

This simple yet elegant architecture allows VGG16 to reduce the number of hyperparameters while increasing its ability to understand complex data input and achieve high accuracies at image classification tasks. Similar to the MobileNet approach, the VGG16 implementation followed the strategy of freezing the ImageNet-pretrained base model and training a custom classification head for the three mask scenario classes. An average pooling layer along with a new output layer with three units and activation function as Softmax was also added for classification of the three classes. During compilation, the model used Adam [18] over

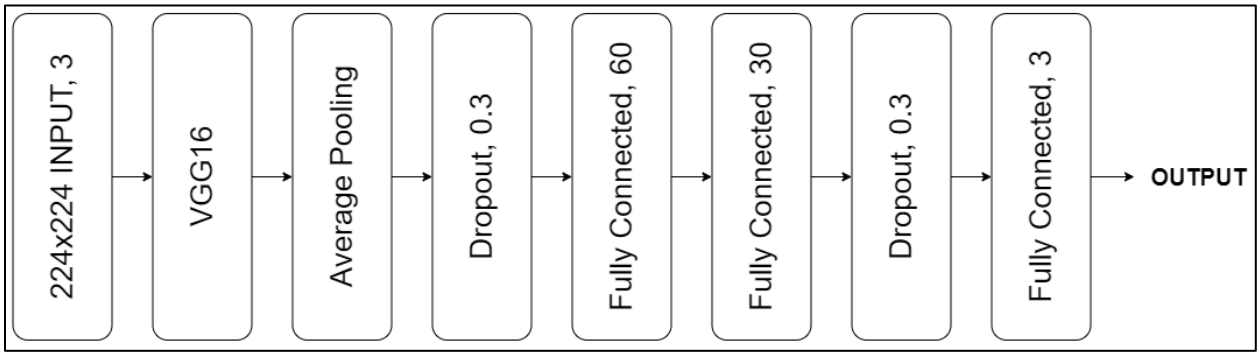

Figure 3: Our Customized VGG16 Architecture

Gradient Descent [19] and categorical cross entropy as the optimizer and loss function, respectively. The model was trained for 80 epochs since over the last 10 epochs its accuracy did not improve a lot. It achieved an accuracy of 93.83% on the training data and 93.04% on the validation data. Its performance on the test set was 93.6%.

In order to improve the performance slight changes in the architecture were implemented, the last block i.e., block five of the model was unfrozen and its weights were allowed to be updated so that the model could adapt more effectively to the specific features of the dataset. New layers were added to the model in order to better fit the problem. After the average pooling layer, a dropout [21] layer with a rate of 0.3, two fully connected layers with 60 and 30 units with relu activation, and a final dropout layer with a rate of 0.3 were added before the output layer of three units. This model was trained for 16 epochs after which it was able to achieve an accuracy of 98.76% on the training set and 99.10% on the validation set. When the model's performance was measured using the test set, it achieved an accuracy of 99.05%. These adjustments significantly improved the model's performance, enhancing its ability to accurately classify images in the dataset.

3. RESULTS

For both the MobileNet and VGG16 architecture, only two model variants are considered for inferring the results, the base model which utilized the pre-trained weights from ImageNet and an output classification layer, and the final fine-tuned version, where the models were the base models were modified and updated by unfreezing the model layers, adding more custom layers in order to better fit the specific dataset. From now on the base model will be referred using notation (B) and the fine-tuned version will be referred using (F).

The training and validation accuracy/loss curves, as shown in figure 4, provide valuable insights into the model's learning behavior during the fine-tuning process. These graphs can help identify potential issues such as overfitting or

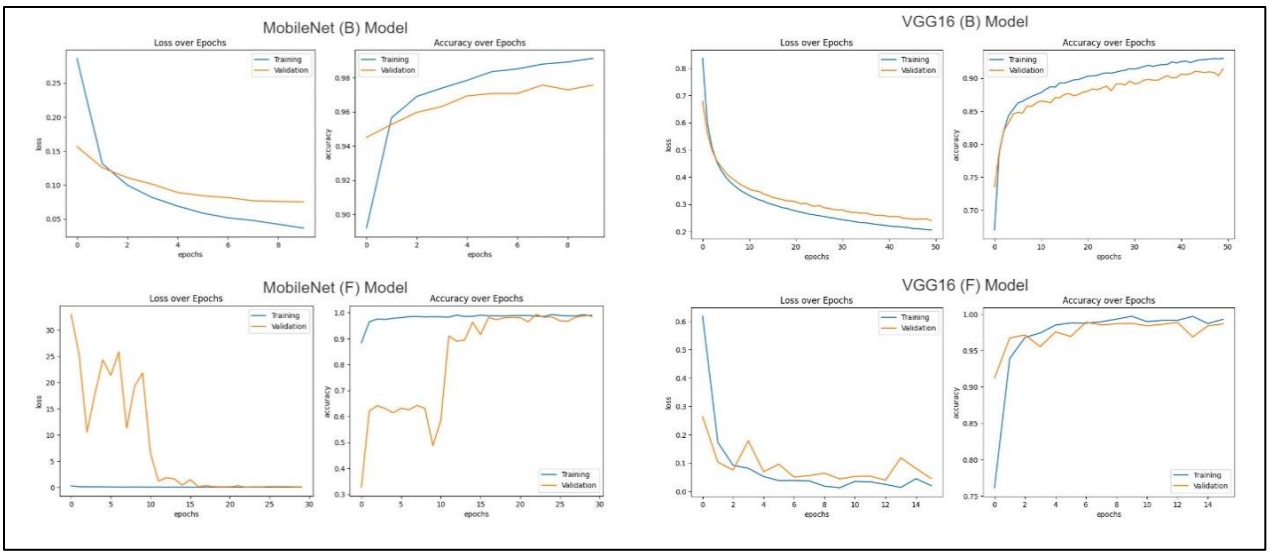

Figure 4: Loss and Accuracy Plots for both the models

underfitting and aid in analyzing the model's performance at different stages of training.

The graphs offer valuable insights into the training process for both MobileNet and VGG16 architectures. If we take a look at the loss curves, it's clear that in both models, for their base (B) and fine-tuned (F) variants, a steady decrease in loss over training epochs can be seen. Notably, the fine-tuned versions in both architectures seem to achieve lower loss values compared to the base models. Examining the accuracy curves reveals a clear benefit of fine-tuning. Both MobileNet and VGG16 show a consistent improvement in accuracy throughout training. However, the fine-tuned versions consistently outperform their base counterparts in terms of achieved accuracy.

The graph for MobileNet (F) model seems significantly different than all the other graphs since it is fluctuating but with a long-term upward trend compared to all the other graphs which have a smoother accuracy and loss gradient. This is because dropout layers have been added to the model in order to prevent overfitting in the model The gap in accuracy between base and fine-tuned models is particularly high for VGG16, highlighting the potential of fine-tuning to significantly improve the model's ability to learn complex features relevant to the specific classification task.

The classification accuracy scores obtained on the test dataset reveal notable differences between the base and fine-tuned models. The MobileNet (B) model achieved an accuracy of 97.55%, while its fine-tuned counterpart demonstrated improved performance, attaining an accuracy of 98.05%. Similarly, the VGG16 (B) model exhibited an accuracy of 93.60%, significantly lower than the VGG16 (F) model, which achieved an impressive accuracy of 98.94%.

Figure 5 is a plot of the confusion matrices for the models. The confusion matrix provides a class wise breakdown of the models' prediction on the test dataset. By looking at figure 5 we can infer that the MobileNet (F) model has a high number of false negatives for mask worn incorrect class (16). This suggests that the model might have difficulty distinguishing images where masks are worn incorrectly from other classes. Whereas the VGG 16 (F) model has high number of false negatives for without

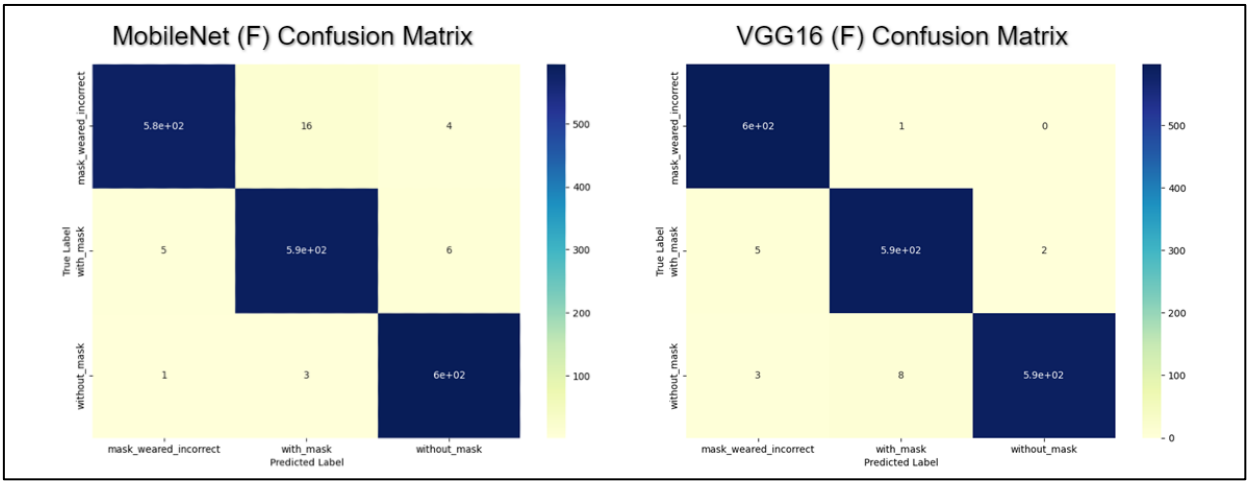

Figure 5: Confusion Matrices for MobileNet (F) and VGG16 (F) models

mask class. This could potentially lead to missed detections of individuals not wearing masks, which might be a concern in real-world applications.

In addition to accuracy, other performance metrics were evaluated. The F1-score, which combines precision and recall into a single metric, was calculated for both models, with the fine-tuned VGG16 model achieving an F1-score of 98.93% and the fine-tuned MobileNet model achieving an F1-score of 97.91%. Precision and recall values were also computed, with the fine-tuned VGG16 model exhibiting a precision of 98.96% and a recall of 98.92%, while the fine-tuned MobileNet model had a precision of 97.79% and a recall of 97.96%.

In order to understand how the models compared to other architectures built for the same task, they were compared with two works which were mentioned in the literature review section of the paper, the first being a custom CNN architecture model created by Das et al. [9] which will be referred to as Model 1, and the second one being a transfer learning model which uses ResNet for feature extraction and ensemble a model for the final classification task created by by Loey et. al. [7] which will be referred to as Model 2. Both these models worked on the similar problem of classifying whether a person in an image is wearing a mask or not.

In order to better understand and effectively compare the performance of all the models I have mentioned earlier, I have tabulated the accuracy scores for model 1, model 2 and for both the base and fine-tuned variants of my VGG16 and MobileNet models. Based on figure 6, we can confidently say that both the MobileNet (F) and VGG16 (F) models which are the fine-tuned versions of the base model outperformed Model 1 by a fairly large margin

AUTHOR	MODEL	ACCURACY
Das et al.	Model 1	94.58 %
Our Model	MobileNet(B)	97.55 %
Our Model	MobileNet(F)	98.05 %
Our Model	VGG16(B)	93.60 %
Our Model	VGG16(F)	99.05 %
Loey et al.	Model 2	99.64 %

Figure 6: Accuracy of my models compared to other similar works

but the best accuracy out of all the models is achieved by Model 2 which uses a transfer learning approach for feature extraction and a separate classifier for the classification task.

4. DISCUSSION

The results of this study demonstrate the effectiveness of fine-tuning pretrained CNN models, specifically MobileNetV2 and VGG16, for the problem of classifying whether a person is wearing a mask or not based on the input image. Both models, when fine-tuned, showed significant improvements in accuracy compared to their base pretrained versions. The study also allows us to understand the accuracy/performance tradeoff in using architectures that require varying computation resources.

The MobileNetV2 model, was able to achieve an accuracy of 98.05% after fine-tuning which is an improvement of 0.5% from its base model accuracy. It is also worth noting that the improvement in accuracy for MobileNet was not very large, this could signify the fact that even after fine-tuning there may be limitations on how well a model may perform on a dataset.

On the other hand, the VGG16 model showed a much more substantial improvement after fine-

tuning, as its accuracy was increased by 5.45%, giving the final accuracy as 99.05%. This improvement suggests that VGG16, with its deeper architecture and higher computational cost, is able to benefit more from fine-tuning compared to the more cost-effective MobileNetV2.

The comparison with other works, specifically Model 1 and Model 2, further highlights the effectiveness of the proposed approach. Both MobileNetV2 and VGG16 outperformed Model 1 by a significant margin, demonstrating the effectiveness of transfer learning models over custom convolution architectures. However, Model 2, which also used a transfer learning approach, achieved the highest accuracy of 99.64%. It's use of pre-trained ResNet architecture for feature extraction and a separate ensemble model for the purpose of classification may be the reason behind this. It is important to note that this model will take up more computational resources due to its increase in complexity and therefore is line with the aim of our research paper to better understand accuracy/resources tradeoff. The model also tells us that our models are still not perfect and with further fine-tuning we may be able to achieve even better accuracy.

The difference in accuracy of 1% between the VGG16 and MobileNet models may not look a lot but it is very significant when deploying a model in real world scenarios. The results of the confusion matrix showed us that the MobileNet model had difficulty in predicting mask worn incorrect images and misclassified them as masked images, however this was not a significant problem in VGG16 model. Differentiating between mask worn incorrect and mask worn classes is the most difficult classification task in this problem due to the similarity of both images. This offers us an insight about the loss in performance we have to endure for needing less computational resources, since the MobileNet model is not able to understand features to the degree that VGG16 is able to and hence the drop in accuracy.

Therefore, a thorough evaluation should be conducted before approaching any problem about the complexity of the problem and the degree of fault tolerable in the system. Only after this evaluation is conducted should a user choose which model is appropriate for their scenario, if they cannot afford the high computational costs and can handle slight inaccuracies then the MobileNet architecture is well suited for their needs but in need of high accuracies then VGG16 is a clear winner.

5. CONCLUSION

In this paper we conducted an in depth and systematic review of the accuracy and computational resources tradeoff by exploring the problem of Face Mask Classification using the VGG16 and MobileNet architectures. After through evaluation we came to the conclusion that choosing a model architecture needs to be done after careful and thorough evaluation of the needs of the problem, resources available and margin of error that is tolerable. As these conditions are different for all applications and situations. We can consider MobileNet as a cost-effective solution with reasonable accuracy, and the VGG16 as a powerhouse which offers great performance but requires high computational resources.

6. REFERENCES

[1] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE

Computer Society Conference on Computer Vision and Pattern Recognition, pp. I-511-I-518 vol.1.

[2] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25.

[3] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, Tsuhan Chen (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377.

[4] Sandler, Mark & Howard, Andrew & Zhu, Menglong & Zhmoginov, Andrey & Chen, Liang-Chieh. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 4510-4520. 10.1109/CVPR.2018.00474.

[5] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[6] https://www.kaggle.com/datasets/vijaykumar1799/face-mask-detection

[7] Loey, M., Manogaran, G., Taha, M.H.N. and Khalifa, N.E.M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement, 167, p.108288.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90

[9] A. Das, M. Wasif Ansari and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342585.

[10] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with lle-cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2682-2690)

[11] Jiang, M., Fan, X., & Yan, H. (2020). RetinaMask: A Face Mask detector. arXiv preprint arXiv:2005.03950.

[12] Kontellis, Efstratios & Troussas, Christos & Krouska, Akrivi & Sgouropoulou, Cleo. (2021). Real-Time Face Mask Detector Using Convolutional Neural Networks Amidst COVID-19 Pandemic. 10.3233/FAIA210102.

[13] Mercaldo F, Santone A. Transfer learning for mobile real-time face mask detection and localization. J Am Med Inform Assoc. 2021 Jul 14;28(7):1548-1554. doi: 10.1093/jamia/ocab052. PMID: 33713140; PMCID: PMC7989332.

[14] Qin, B., & Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors, 20(18), 5236.

[15] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[16] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE

Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255

[17] Tan, M. and Le, Q.V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 June 2019, 6105-6114. http://proceedings.mlr.press/v97/tan19a.html

[18] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[19] Bottou, L. (2014). Stochastic gradient descent. In Cumulative Research in Neural Networks: Representations and Deep Learning (pp. 161-252). Springer, Berlin, Heidelberg.

[20] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 807-814).

[21] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1921-1958.