Journal of Computer Science

Original Research Paper

Evaluation of Transfer Learning for Mask Detection

Zahin Akram, Arifuzzaman Arman, Mohammad Rakib Imtiaz and Syed Athar Bin Amir

Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh

Article history
Received: 16-10-2021
Revised: 24-01-2022
Accepted: 02-02-2022

Corresponding Author:
Zahin Akram
Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
Email: zahin.akram@northsouth.edu

Abstract: The use of masks has become crucial in combating the Coronavirus pandemic. Unfortunately, the regulation of wearing a mask is not being upheld by many citizens which is contributing to the spread of the disease. To aid the efforts of regulations and to maintain safety in public areas, both large like parks or small like public transport, Artificial Intelligent systems can play a vital role. In this article, we explore the use of transfer learning across 5 models (Mobile Net V2, InceptionV3, Resnet50V2, VGG16 and DenseNet121) and measure their effectiveness in mask detection. Due to the lack of a large, diverse and annotated dataset, we explore the use of transfer learning using supervised methods and present the results of the experiments upon the Keras open-sourced models. We find an average of 99% accuracy for all 5 models. However, when we use K-Fold Cross Validation to account for bias, we find significant differences in results with the highest accuracy being achieved by VGG16 at 98.6%. With the mixture of the standard method of training and testing alongside K-Fold Cross Validation, we present our findings for the use of transfer learning for mask detection.

Keywords: Transfer Learning, Mask-Detection, Computer Vision, Object Classification, Covid-19

Introduction

In 2020 the arrival of the Coronavirus disease caused by the SARS-CoV-2 virus bought out a wave of quarantine and regulations meant to be observed by the public. If someone had to go out in public, wearing a mask was prescribed by doctors and scientists alike (Feng *et al*., 2020). The use of masks to prevent transmission of this dangerous virus has proven widely successful (Cheng *et al*., 2020b). In many countries, the use of masks was enforced by the government. Unfortunately, many people ignored the regulations of safety and science and opted to not wear masks which contributed greatly to the spreading of the disease. To curb the rising number of infections the French government initiated the use of AI to identify passengers not wearing a mask (Paris Tests Face-Mask, 20221).

The use of Machine Learning and Deep Learning can serve in many ways in the fight against COVID-19 (Agarwal *et al*., 2020). The purpose of this article is to explore the effectiveness of transfer learning in many different models and to provide an optimal model for mask-detection. Transfer learning has proven to be one of the most efficient ways of training models (Pan and Yang, 2009). It has proven its effectiveness even when working with limited data. This is of significance because there is a limitation of labeled or annotated data of masked individuals. There never has been much of a need for a dataset comprised of people wearing masks. The scarcity of such data has impeded the emergence of models capable of detecting masks with high accuracy. Transfer learning helps mitigate some of these problems. With the use of pre-trained object detection models, that are already capable of boundary and shape detection, a few final layers can be added that are trained specifically for detecting masks. This article explores the use of transfer learning in five different models and evaluates their effectiveness in real-world scenarios. It still remains difficult since the masks obscure many features of the face like noses and lips and render much of the face a blob. Therefore, we explore the use of transfer learning in VGG 16, Mobile Net V2, Inception V3, Res Net 50V2 and in Dense Net models.

The contribution of this article is to evaluate the effectiveness of the models highlighted in their ability to detect masks. With images of masked individuals that could be used to train a model being scarce, this article aims to inform the best model and architecture to use for this purpose. On the other hand, often models built on limited data suffer from overfitting or bias. In this article, a technique is also highlighted that checks for overfitting or bias. Through the use of K-Fold Cross Validation we further present data that compares with the standard method of training and provide further evidence for the evaluation of these models. The

Science Publications

methods highlighted in this study can also serve in the detection of any object for which one has limited data.

In this article we explore some of the other recent works done for mask detection, discuss the models, the data and methodology used and the results that we obtained for each individual model.

Related Works

Transfer learning was bound to be an effective approach for mask detection especially considering the limitations of data. In this section, we look at other published works that have showcased the use of transfer learning for mask detection.

Cheng *et al*. (2020a) has shown that mask detection via transfer learning is viable through the use of YoloV3 Tiny. The You Only Look Once (YOLO) (Redmon *et al*., 2016) works a bit differently than the standard FRCNN (Ren *et al*., 2015). YOLO is a convolutional neural network that applies its algorithm in a single neural network to the full image and then divides the image into regions and predicts bounding boxes and probabilities for each region. Cheng *et al*. (2020a) trained the latest iteration of the YOLO architecture on the RMFD dataset (RMF) and have shown their precision to be around 80%. However, as we will discuss further the RMFD dataset is not very diverse in terms of the faces represented. Furthermore, the methodology of Cheng *et al*. (2020a) is limited to standard training and testing. With a limited dataset, the sort that is used, it may lead to overfitting. Their testing and training were also restricted to surgical or single-colored masks due to the limitation of the RMFD dataset.

Loey *et al*. (2021) uses a ResNet architecture for detection. However, they take it a step further by removing the last layer of the model and replacing it with three traditional machine learning classifiers: Support Vector Machine (SVM), decision tree and ensemble. Using a combination of RMFD (RMF), Simulated Masked Face Dataset (SMFD) (SMF) and the Labeled Faces in the wild (Learned-Miller *et al*., 2016), they showcased a variety of metrics. The metrics consisted of accuracy upon the individual datasets as well as a mixture of RMFD and SMFD. They showed that the highest accuracy was obtained using SVM in the last layer of their ResNet architecture which is 98%.

Chowdary *et al*. (2020) also uses transfer learning via the InceptionV3 using the RMFD dataset (RMF) and the Simulated Masked Face Dataset (SMFD) (SMF) for training. They showcased an incredible accuracy of 100% but that was because their testing was performed upon the SMFD dataset only. However, no steps are taken to account for bias in their proposed system. The SFMD also consists of faces with masks cropped on them, reducing its ability to replicate images for the real world. The characteristic of the images ensures that the faces and the masks are visible and in the center. Therefore, the testing accuracy should not be considered a true metric for its evaluation.

Datasets are often similar when it comes to mask detection because of their scarcity. Above, we saw the use of the RMFD dataset which is limited by its diversity. We also saw the use of the SMFD dataset, however, it is not truly suited for testing because of its pristine condition. Furthermore, both Cheng *et al.* (2020a) and Chowdary *et al.* (2020) present their accuracy with nothing to account for overfitting in their experimentation. We also saw the use of a hybrid detection system by Loey *et al*. (2021) which gives a comprehensive set of metrics for their different usage of different classifiers.

Models

In this section, we briefly explore the different models used in our experiment. The following sections cover the basics of VGG16, MobileNetV2, InceptionV3, DenseNet and Res Net V2.

VGG16

VGG16 (Simonyan and Zisserman, 2014) is a convolutional neural network that has stacks of 3x3 convolution layers with a stride of 1 followed by a 2x2 max pooling with a stride of 2. After 13 such convolution layers, there are 3 dense layers. They are 3 Fully Connected (FC) layers followed by a softmax for output. All hidden layers use ReLU as the activation function. There are a total of 16 layers in this network, as can be seen in Fig. 1, hence its name is VGG16. It was trained on ILSVRC-2012 dataset (a subset of imagenet (Deng *et al*., 2009) dataset) for image classification.

InceptionV3

Inception-v3 (Szegedy *et al*., 2016) is a 48 layers deep convolutional neural network. This model improves the inception architecture introduced here (Szegedy *et al*., 2015).

This network uses multiple-sized filters on the same level. Input is passed to different sized filters and max pooled. An extra 1x1 convolution is done before passing on to filters and after max pooling to reduce the number of input channels. All outputs are concatenated and sent to the next layer. The model was made wider instead of deeper to remove the representational bottleneck. Convolution with a large filter size is factorized into two convolutions with smaller filter sizes. A convolution of size n x n is factorized to a 1x *n* and *n* x

Convolution

These factorizations decrease the number of parameters. Auxiliary Classifier and Label Smoothing Regularization

(LSR) is used to prevent overfitting. All the above techniques are consolidated into the final architecture. A module of InceptionV3 can be seen in Fig. 2.

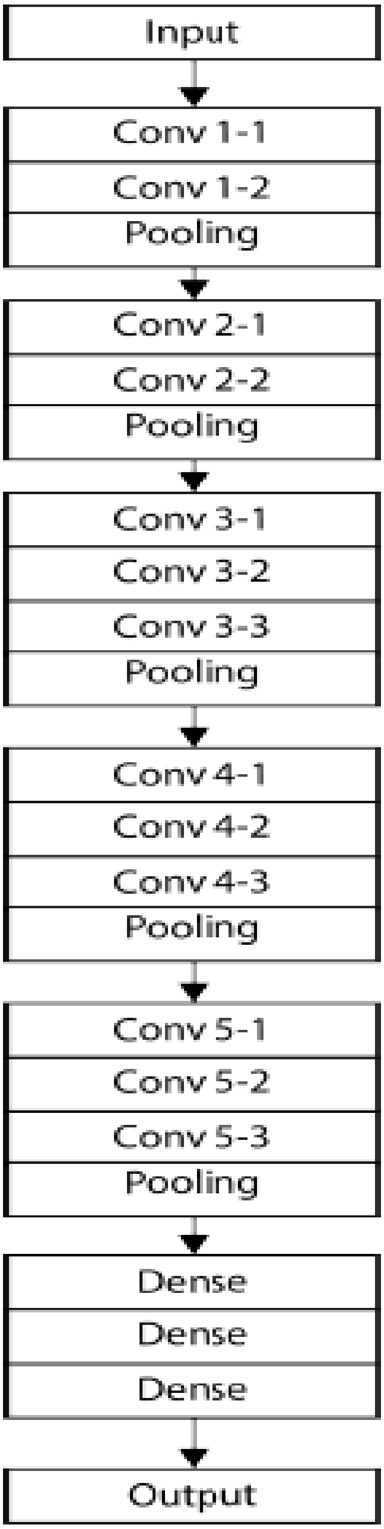

Fig. 1: VGG16 architecture

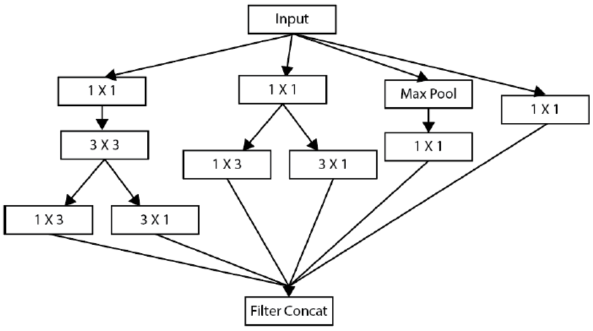

Fig. 2: InceptionV3 module

MobileNetV2

MobileNetV 2 (Sandler *et al*., 2018) is a convolutional neural network that is optimized for mobile devices. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. Each residual bottleneck layer takes a low dimensional compressed representation as input. The input is expanded into higher dimension using 1x1 convolution. Features then go through a lightweight depth wise convolution. Afterward, another 1x1 convolution layer shrinks the features back to a lower dimension. This layer is linear to stop non-linearities from destroying too much information. There is a skip connection here with the original input. ReLU6 is used instead of ReLU as the activation function of this network. MobileNetV2 convolution block can be seen in Fig. 3. Batch normalization is used at the end of every convolution. the model size varies between 1.7 M and 6.9 M parameters. MobileNetV2 is very small in size and takes a small time to train while still giving good performance.

DenseNet

A DenseNet (Huang *et al*., 2017) is a convolutional neural network where every layer is connected to every other network, as can be seen in Fig. 4. In normal CNN models the deeper the network is, the worse the vanishing gradient problem becomes. Dense Net addresses this problem by feature reuse. This model has dense blocks and a transition layer. Every layer of a dense block receives the outputs of all previous layers as input and its output is used as input for all subsequent layers. This concatenation is possible because the feature map remains the same in a dense block. The feature map is reduced in transient layers. A transition layer takes dense block output and performs batch normalization, 1x1 convolution and, 2x2 average pooling. Since each layer's output is concatenated with all previous layers' output, the input can get quite large. So, a bottleneck layer is introduced to reduce input size. CIFAR, SVHN and ImageNet are the datasets of this model.

ResNet50V2

ResNetV2 (He *et al*., 2016a) is an improvement of the original ResNet architecture (He *et al*., 2016b). ResNet models consist of residual units than can be seen in Fig. 5. In each residual unit, there is a shortcut connection from the input to the output of the unit that acts like an identity function. So, the main path can focus on learning the residual mapping. ResNetv2 applies batch normalization and activation function before convolution. ResNet architecture can be used to build 1000 layers deep neural networks well optimized.

Dataset

The dataset consists a total of 4554 images. Of these, training dataset consisted of 2606 images of people wearing masks and 1948 images of people with no masks. Images of varying sizes were used throughout the training.

The average height and width was 300 pixels. However, the shape of the images ranged from 140x102 to 4608x3456 (height and width respectively). Some of the images contained multiple individuals with or without masks. The training dataset was split to an 80:20 ratio for training and validation respectively. The dataset was comprised of the Real-World Masked Face Dataset (RMFD) (RMF) and the Simulated Masked Face Dataset (SMFD) (SMF). We used approximately 1900 images from the RMFD dataset and about 700 images from the SMFD dataset, examples of both can be seen in Fig. 6. We found that this consisted of the optimum balance between training data and accuracy. The combination of these datasets were to ensure the diversity of the people in the images as well as the variety of the masks shown. The RMFD dataset consisted of many different colors of masks but was limited to people with pale skins. The SMFD dataset was consisted of solely surgical types of masks but consisted of people with diverse ethnicity.

To address the concern of bias or overfitting we also utilized the use of K-Fold cross validation alongside standard training and validation to ensure that our predictions produced the best results while restricting bias as much as possible.

Our dataset contains many images like in Fig. 6, where the people are of many different races, gender and ethnicity. They are also wearing masks of different colors and shapes to further replicate the real-world environment.

Furthermore, the combination of the dataset ensured different angles and closeups of the masked and nonmasked individuals. The dataset did not contain any labeling of region of interest and was merely labeled as ”mask” or ”without mask”.

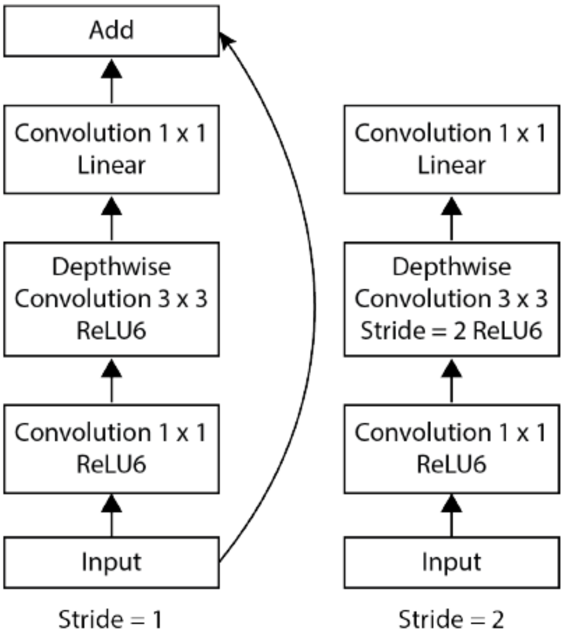

Fig. 3: MobileNetV2 Convolution Block

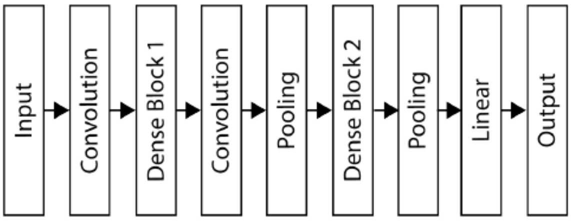

Fig. 4: DenseNet Architecture

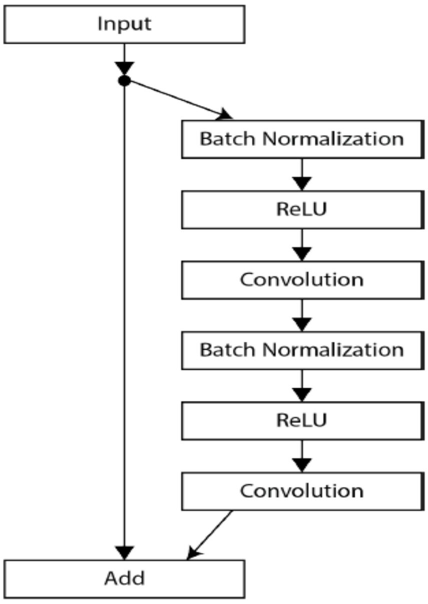

Fig. 5: ResNetV2 residual unit

Fig. 6: Example of masked image

Methodology

The article aims to find the best use of Transfer Learning across the models mentioned above. As such, the data utilized for training, validation and testing were kept the same across all 5 models. Other parameters like resources for training as well as the number of epochs for all the models were kept the same. Facial features were extracted using the FaceNet model (Schroff *et al*., 2015) that detects the eyes and processes the region around them. These features were then fed to the base model that was pre-trained. Further evaluation was also performed using the K-Fold cross validation to ensure that there was no issue of data bias or overfitting and obtain optimal results.

We start off by loading the base Keras ”ImageNet” models. These pre-trained models are capable of detecting a wide variety of objects. The utilization of these models ensures the models are capable of processing different object boundaries that we now don’t have to train. We then freeze the base model. We add 2 new layers for the purpose of mask detection. Later, we configure the input shape of the final layers to the output shape of the base model. We then perform the training and validation. The base model provides the necessary details regarding the Region-Of-Interest (ROI) provided by the FaceNet model so that the ROI only consists of the necessary portion of the image. We then test upon our prediction.

Finally, we also do all the above steps for the K-Fold cross-validation by diving the dataset into 5 sets. In this method, the dataset is divided into 5 parts. 4 of the parts are used for training via transfer learning and the one that is left is used for validation. The method of training is the same as the standard training. The number of iterations of training and other parameters are also kept constant. Once complete, the process is repeated with another set being used for validation. This ensures that at no point during the validation step does the model encounter any images that was used during training. This eliminates bias or overfitting. This is done 5 times so that each set is chosen once for validation. Out of the 5 iterations of training, we save the weights that produce the highest accuracy. This steps are repeated for all 5 models. We then perform a final testing for the best weights per model to perform a thorough comparison.

All the steps for both the methods are kept consistent across all 5 models to ensure proper evaluation.

Results and Discussion

In this section, we present the results that we obtained for each model using standard training and the K-Fold training. Among the 5 training sets for the K-Fold cross validation we will only be looking at the best version for each model. The dataset provided for the standard training and the data used for validation was the same across all the models. The same is true for the K-Fold cross validation. We will first present the results for the standard training/validation and then the results obtained by K-Fold cross validation.

MobileNetV2

MobileNetV2 performed exceptionally when it came to detection. It achieved 100% across both training and validation for the standard training. It also reached its optimum results soon into its training and maintained its results as can be seen in Fig. 7.

For the K-Fold cross validation, the best version was achieved for the 4th iteration of 5-fold cross validation. MobileNetV2 again showed very good results when compared to the other models. It was the second most accurate during the testing for cross validation. For MobileNetV2 the best result for cross validation training and validation were 92 and 93% respectively. The results are presented below in Fig. 8.

InceptionV3

InceptionV3 obtained the lowest value in terms of accuracy for training and validation among all the models used. It reached 96 and 97% for training and validation during the standard method of training respectively. Even though it displayed very high accuracy, it was still outperformed by the other models. We show the accuracy for InceptionV3 base training and validation in Fig. 9.

InceptionV3 also ranked at the bottom when it came to K-Fold cross validation. At merely 78 and 84% for training and validation, InceptionV3 is not the way to go when it comes to mask detection through transfer learning. It seems that the variation in data revealed that the model suffered from overfitting during the standard method of training and validation and the biases in its results were exposed during cross validation. Figure 10 shows the graph of its performance.

ResNet50V2

For the standard training, ResNet50V2 obtained the second best results tied with DenseNet121. ResNet50V2 showed 99% accuracy for both training and validation and showed consistent accuracy as seen in Fig. 11.

However, it performed rather poorly during cross validation. ResNet50V2 only performed slightly better than InceptionV3 and was exceeded by all the other models. Coming at only 79 and 88%, as seen in Fig. 12, for training and validation, cross validation exposes that the model might have been overfitted during just training and validation.

VGG16

VGG16 outperformed InceptionV3 during the standard training and validation. At 98 and 97% accuracy for training and validation, it ranked at the middle of the pack in terms of its accuracy. Figure 13 shows the performance of VGG16.

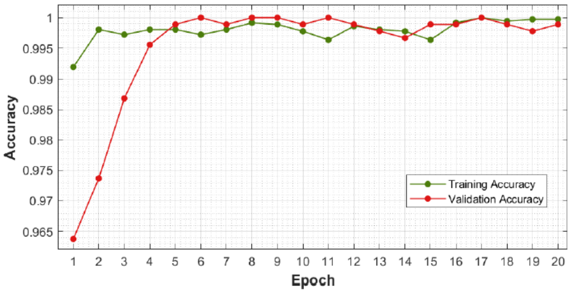

Fig. 7: MobileNetV2 standard training and validation

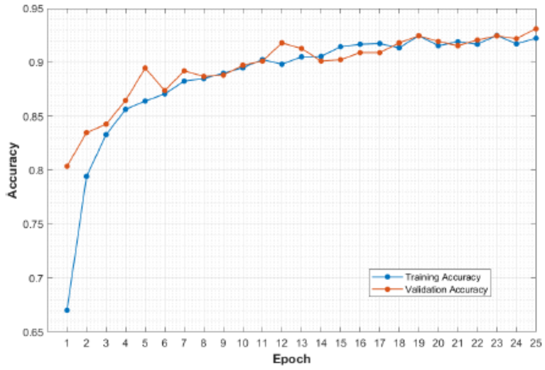

Fig. 8: MobileNetV2 training and validation accuracy

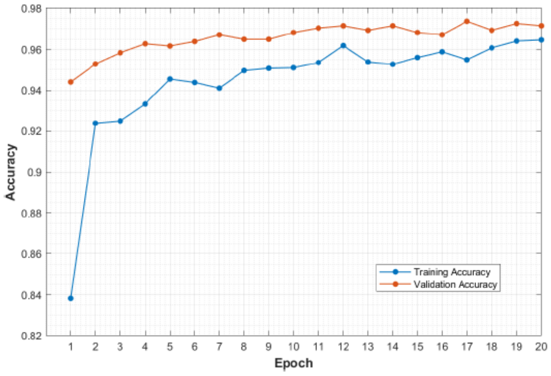

Fig. 9: InceptionV3 standard training and validation

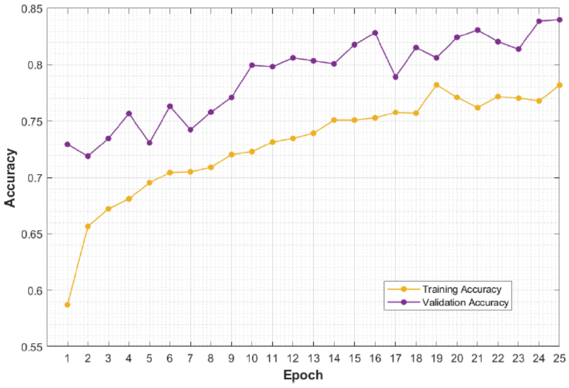

Fig. 10: InceptionV3 training and validation accuracy

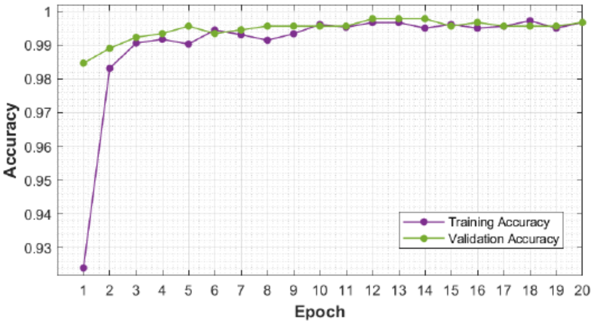

Fig. 11: ResNet50V2 standard training and validation

Surprisingly, however, VGG16 performed the best during cross validation. It achieved a staggering 98 and 99% for training and validation respectively. Outstripping MobileNetV2 who attained 92 and 93% for training and validation respectively. It seems that the architecture of VGG16 is optimally suited for mask detection via transfer learning. The performance can be seen in in Fig. 14.

DenseNet121

Tying the score with ResNet50V2 for the standard method of training, the accuracy for training and validation was 99% for both categories represented in Fig. 15.

However, it may too have been suffering from some overfitting as it finished with an accuracy of 88 and 91% for training and validation respectively. This may not have been as bad as it was for InceptionV3 and ResNet50V2, so DenseNet121 is in the middle of the pack for cross validation. Figure 16 shows the performance of DenseNet121 for K-Fold cross validation.

Loss Comparison

The values for Loss comparison can be seen in Fig. 17 and 18. They are for the K-Fold cross validation for training and validation respectively.

Discussion

For easier comparison, we can take a look at Table 1. Here, we can see that all 5 models performed excellently during standard training. With the results so close together it is rather difficult to choose just one to utilize.

However, Table 2 shows a different set of results. From this table, for K-Fold cross validation, we can see that VGG16 is the best for mask detection via transfer learning. With a rather extreme departure from the standard training and testing method, both InceptionV3 and ResNet50V2 show that they were suffering from overfitting or bias. While DenseNet121 did not suffer to such a severe degree, it is still a prominent departure from where it was during the standard training. The use of cross validation mitigated much of the issues of overfitting. An important thing to note is that many published results do not take this overfitting into account.

For further evaluation we tested the time taken for detection. The results based on time can be seen in Table 3. The model time comparison was run for 3000 images.

The tests were performed for all models on the same images and on the same system to ensure there was no hardware discrepancy. The pre-processing time was constant for all 5 models as expected. The time taken for the model to perform inference was the primary parameter. The tests were done only for the versions generated from K-Fold cross validation as they represent the truest form of output and accuracy. VGG16 again performs the best in terms of time taken for inference. MobileNetV 2 and Res Net 50V2 are rather close in time and are followed by InceptionV3 and DenseNet121. Even considering the amount of time taken for inference, VGG16 clearly outperforms others and remains the most optimal choice for mask detection through transfer learning.

Since VGG16 obtained the highest result in terms of accuracy during the cross validation as well as time of inference, it is safe to say that VGG16 is an optimal candidate for mask detection via transfer learning. A result of its detection among the example images of our dataset is provided in Fig. 19. Here, VGG16 confidently detects all the masked individuals. Furthermore, the methods in this article, transfer learning and K-Fold cross validation, can be used for any object detection even if there is limited data. Transfer learning can be used for any object detection and object classification and K-Fold cross validation can prevent overfitting when working with a small dataset.

The primary reason we performed cross validation was to account for the limitations of data. As we demonstrated above, models under such constraints suffer from bias or overfitting. Chowdary *et al*. (2020) showcases InceptionV3 and its rather high accuracy for mask detection. Upon repeating their process we also found a similar result. We used the same dataset and found the initial results promising. However, the high accuracy fails to take other issues into account. We showed that once steps are taken to account for bias or overfitting, there is a significant loss of accuracy. Chowdary *et al.* (2020) fails to account for these bias issues. However, our proposed model of VGG16 clearly outperforms InceptionV3 by a large margin and is also capable of making predictions faster as well. Rather, out of the 5 tested models, InceptionV3 performed the slowest.

Cheng *et al*. (2020a) utilizes YOLOV3 tiny. Their training consisted of only the RMFD dataset and they reported really high accuracy. Their testing also relies solely on the RMFD dataset. With training and testing done on such a small dataset, it can be said that they also suffer from bias or overfitting. On the other hand, even though no accounts are provided for the time of inference, we can conclude that VGG16 would still be faster than YOLOV3 Tiny. The proposed system of Cheng *et al*. (2020a) runs inference on the entire image. However, our proposed method of using FaceNet (Schroff *et al*., 2015) to detect eyes or facial boundary features narrows down the search region and considerably speeds up the inference step. Even if there are false positives for eyes detected, the region-of-interest would still be smaller than the entire input allowing our proposed model, VGG16, to perform inference faster than YOLOV3 Tiny.

It is safe to say that many proposed methods of mask detection use similar datasets but don't account for overfitting or bias. While methods like those of Loey *et al*. (2021) do include Support Vector Machine (SVM), decision tree and ensemble they ultimately reach the same accuracy as our proposed VGG16 model trained through transfer learning. With bias accounted for and through the use of FaceNet, our VGG16 is probably more accurate and as fast as any other proposed method for mask detection.

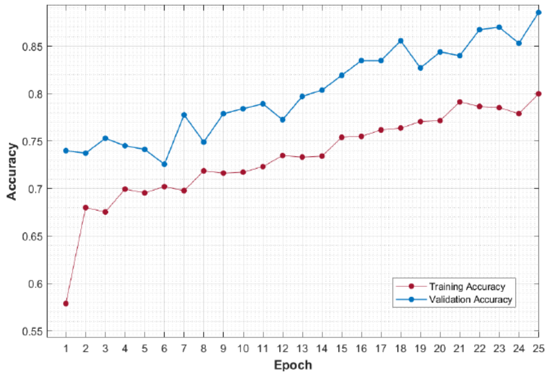

Fig. 12: ResNet50V2 training and validation accuracy

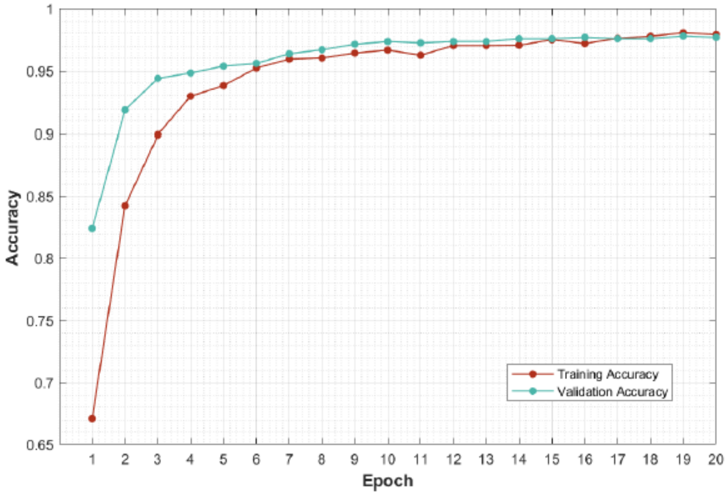

Fig. 13: VGG16 standard training and validation

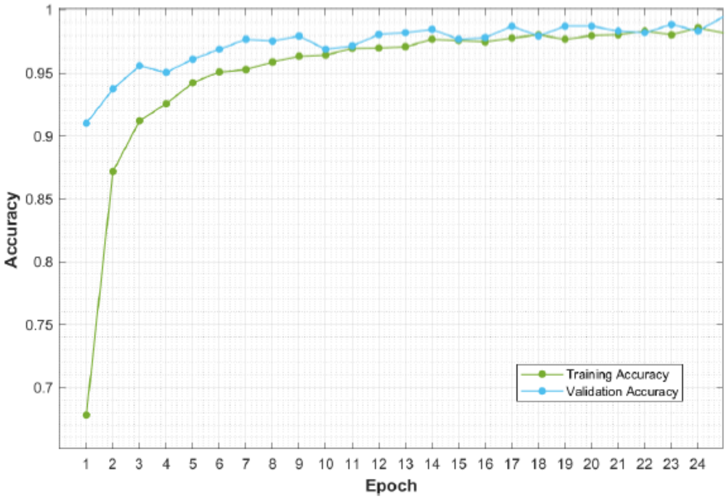

Fig. 14: VGG16 training and validation accuracy for

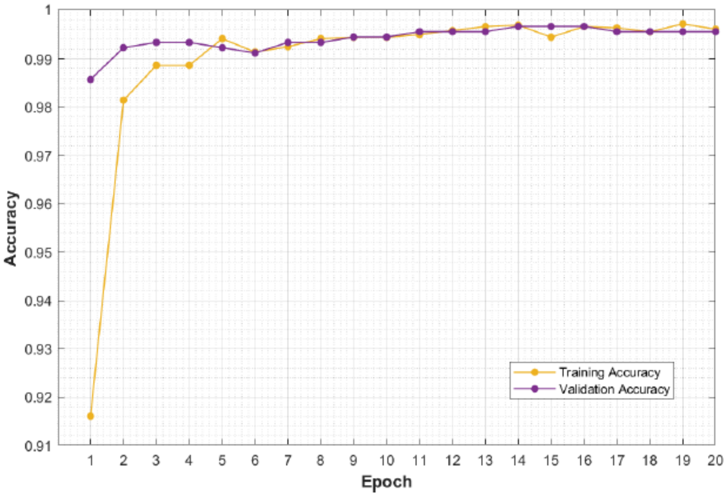

Fig. 15: DenseNet121 standard training and validation

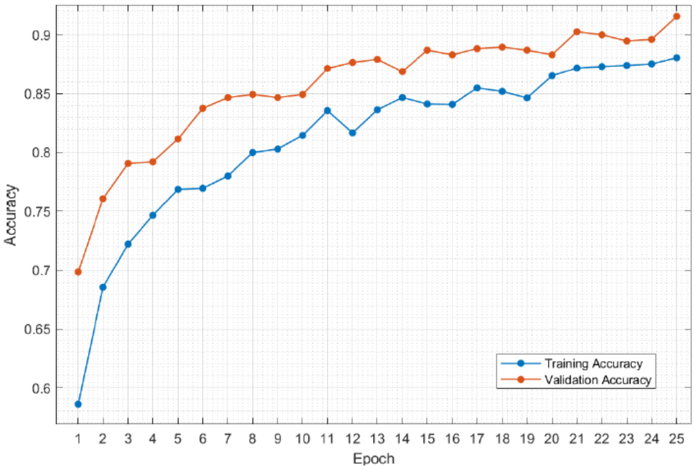

Fig. 16: DenseNet121 training and validation accuracy

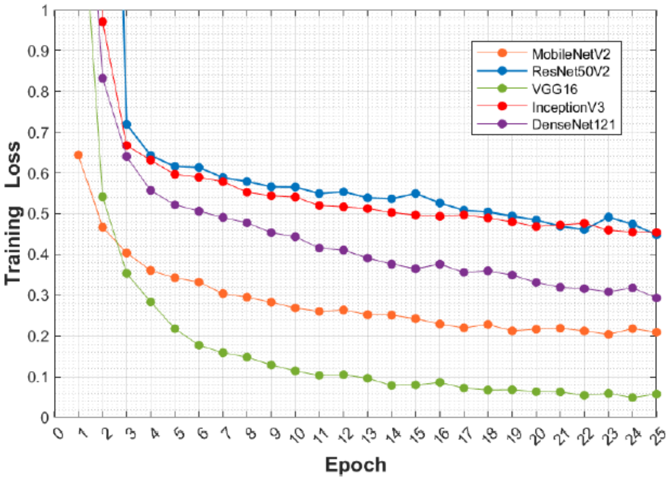

Fig. 17: Best training loss comparison

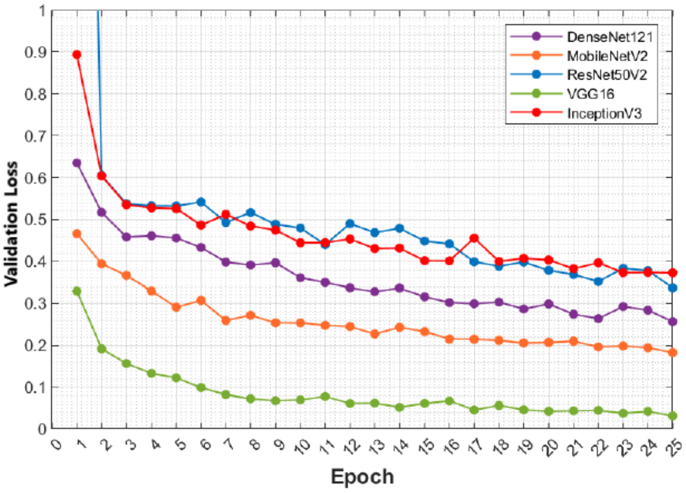

Fig. 18: Best validation loss comparison

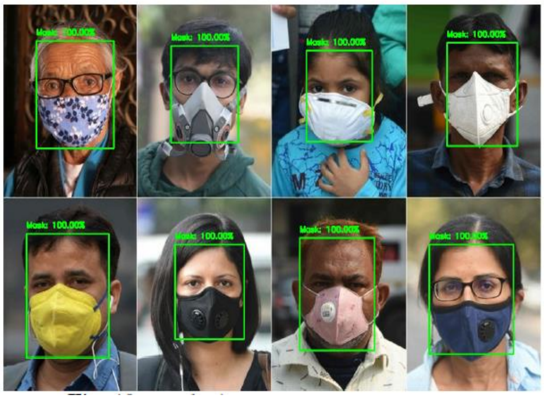

Fig. 19: Masked Image Output for VGG16

Table 1: Standard training best results

Model Name	Training accuracy (%)	Validation accuracy (%)	Training loss (%)	Validation loss (%)
MobileNetV2	100	100	0.0645	0.00739
InceptionV3	96.5	97.4	10	7.89
ResNet50 V2	99.7	99.6	0.923	1.03
VGG16	98.1	97.8	7.34	6.16
DenseNet121	99.7	99.7	1.15	1.05

Table 2: K-Fold cross validation best results

Model name	Best KFold	Training accuracy (%)	Validation accuracy (%)	Training loss (%)	Validation loss (%)
MobileNetV2	4	92.5	93.1	20.4	18.3
InceptionV3	3	78.2	84.0	45.4	37.3
ResNet50V2	3	80.0	88.6	44.8	33.7
VGG16	4	98.6	99.5	4.88	3.13
DenseNet121	3	88.0	91.5	29.3	25.6

Table 3: Model time comparison run on 3000 images

Model name	Best K-fold	Inference time (seconds)
MobileNetV2	4	500.0
InceptionV3	3	665.4
ResNet50V2	3	535.0
VGG16	4	458.8
DenseNet121	3	665.4

Future Work

We plan for even greater evaluation for the best and optimal model that could be used for mask detection. Furthermore, other methods aside from K-Fold Cross validation could be used on these models. Machine learning algorithms like SVM and decision tree could also be used for additional evaluation. Alongside this, more models could also be used to create a system that fulfills all the requirements for mask detection.

Conclusion

In this article, we explore the use of transfer learning to perform mask detection in the ongoing fight against Covid or other airborne diseases. The limited availability of usable masked images made transfer learning an optimum method for mask detection. We trained and tested 5 different models to test whether transfer learning is a viable solution to this issue. We found good results that showed potential. Furthermore, we used the K-Fold cross validation that to ensure that no model suffered from bias from the limited dataset. We discovered that certain models were indeed suffering from overfitting where others remained consistent across both methods. We conclude that transfer learning is indeed a capable solution and recommend the use of VGG16 for mask detection as it performed most optimally across both the standard training and K-Fold cross validation.

Funding Information

This manuscript has not received any funding.

Author's Contributions

All authors contributed equally to the contents of this article.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Cheng, G., Li, S., Zhang, Y., & Zhou, R. (2020a). A Mask Detection System Based on Yolov3-Tiny. The Frontiers of Society, Science and Technology, 2(11). doi.org/10.25236/FSST.2020.021106

Cheng, K. K., Lam, T. H., & Leung, C. C. (2020b). Wearing face masks in the community during the COVID-19 pandemic: Altruism and solidarity. The Lancet. doi.org/10.1016/S0140-6736(20)30918-1

Chowdary, J. G., Punn, N. S., Sonbhadra, S. K., & Agarwal, S. (2020, December). Face mask detection using transfer learning of inceptionv3. In International Conference on Big Data Analytics (pp. 81-90). Springer, Cham. doi.org/10.1007/978-981-13-9406-5_34

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee. doi.org/10.1109/CVPR.2009.5206848

Feng, S., Shen, C., Xia, N., Song, W., Fan, M., & Cowling, B. J. (2020). Rational use of face masks in the COVID-19 pandemic. The Lancet Respiratory Medicine, 8(5), 434-436. doi.org/10.1016/S2213-2600(20)30134-X

He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham. doi.org/10.1007/978-3-319-46493-0_38

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708). https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html

Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., & Hua, G. (2016). Labeled faces in the wild: A survey. In Advances in face detection and facial image analysis (pp. 189-248). Springer, Cham. doi.org/10.1007/978-3-319-25958-1_8

Loey, M., Manogaran, G., Taha, M. H. N., & 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, 108288. doi.org/10.1016/j.measurement.2020.108288

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359. doi.org/10.1109/TKDE.2009.191

Paris tests face-mask. (2021). Recognition software on metro riders. https://www.bloombergquint.com/politics/paristests-face-mask-recognition-software-on-metro-riders

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28. https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520). https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823). https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://arxiv.org/abs/1409.1556

Agarwal, S., Punn, N. S., Sonbhadra, S. K., Tanveer, M., Nagabhushan, P., Pandian, K. K., & Saxena, P. (2020). Unleashing the power of disruptive and emerging technologies amid COVID-19: A detailed review. arXiv preprint arXiv:2005.11507. https://arxiv.org/abs/2005.11507

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9). https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826). https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html