The Efficient Implementation of Face Mask Detection Using MobileNet

Yuan Zhou1*
1 Beijing Normal University - Hong Kong Baptist University United international college, Zhuhai, Guangdong, China
*Corresponding author email: n830026170@mail.uic.edu.cn

Abstract. Under the condition of the global pandemic, Wearing the mask in the public area is the direct and ordered action to prevent the spreading of the virus. Therefore, developing the machine learning model to detect whether the mask is appropriately wearing is practical and meaningful. This paper unfolded the mask recognition based on the MobileNet model, which belongs to one of the branches of Convolutional Neural Network, CNN for short, in the deep learning field. With the background and tremendous needs of face mask detection, in the beginning, we go through the review of the MobileNet’s history and the practicability of it on demanding functions. After that, we explained in a more detailed way to help gain a better understanding of MobileNet’s structure and how those differences make it outstanding. During the experiment, approximately 9000 images were used as inputs by the model for training and optimizing. The result turns to the less weighted model presenting an acceptable accuracy of 87.96% and 93.5% for testing whether a person wears a mask and whether the mask is covered correctly correspondingly. We also discovered several shortcomings and limitations during the testing process. Those problems and some of the possible further directions of MobileNet will be discussed. As a result, the MobileNet shows the huge probability for further improvement to be equipped into the small-scale but sophisticated utensil.

1. Introduction
Since the coronavirus outbreak in 2019, the virus has spread rapidly and widely, coming from countless grievous news, sharp unemployment increases, and ruined families. Up to date, the threatening pandemic has already caused more than 4 million deaths worldwide, which is comparable to the total number of war deaths in the past 40 years. It is necessary to wear a face mask that prevents the virus from entering our respiratory system when we breathe. Since the coronavirus is highly communicable and is transferred through droplets and breathing, the mask, which reaches more than 95% of filtering efficiency, becomes our first protective armor to prevent contact with the virus effectively.

Recently, the good news is that more people take it for granted to wear masks when going out. According to the wearing specification, the correct usage of the face mask should cover the user’s mouth and nose, two accesses of our respiratory system. However, some people fail to follow the wearing instructions. As a result, some supervisors are always needed. The point at this situation, technologists may encounter several difficulties, especially how to check whether individuals wear the
mask correctly or not, even without human supervisors’ monitoring in public. One solution is to introduce machines with intelligent face mask detection algorithms. After embedding relevant codes, those iron supervisors can help managers monitor whether customers wear the mask correctly or not when entering the buildings.

2. Related Works
Since one branch of CNN, the AlexNet, won the Large Scale Visual Recognition Challenge held by ImageNet in 2012 with an unbelievable Top 5 error rate of 15.4% compared to 26.2% for the second place, the performance of CNN shocked the whole CV field [5]. It can be seen that to pursue classification accuracy in the competitions, the model became deeper and deeper, along with the model complexity getting higher and higher. For example, the number of layers of the ResNet even reached 152 at ILSVRC2015. However, in some practical situations, such as mobile or embedded devices, such a large and complex model is difficult to be applied.

Researchers realize that it is essential to study the small and efficient CNN model under feasible consideration. Currently, researches can be divided into two directions: one is to compress the trained complex model; the second is to design and train small models directly, MobileNet belongs to the latter.

In 2017, a group of scientists, Andrew et al. from Google, launched the architecture of MobileNet by publishing a thesis focus on lightweight CNN networks in mobile terminals or embedded devices [1]. After a year, Mark, Andrew, and their colleagues introduced another masterpiece of Google, MobileNet V2 [2]. Based on the original version, the updated structure replaced ReLU with Linear Bottlenecks for correction and newly increased the applications of Inverted Residuals. Only a few months later, the latest version, V3, not only combined the depthwise separable convolution from V1 with Inverted Residuals, Linear Bottleneck, and SE module mainly maintained in V2 but also additionally manipulated Neural Architecture Search for optimization.[3] After the explosion and spread of the covid-19 at the end of the same year of MobileNet V3 is out, engineers embarked on designing the face mark detection algorithm. Multiple existing CNN nets were already applied throughout the programming process, for example, the DenseNet, AlexNet, and sequential CNN… It seems most of the researchers used a complicated and demanding algorithm to obtain better effects. However, when considering the limitation of capacity, that ideal effect may compromise the function under the realistic situation.

3. The Outline
Although there are already many detection algorithms, a faster and more accurate one is still needed for further implementation. In order to control the complexity of the detection algorithm for a build-in chip with a small capacity, MobileNet, a light but still efficient neural network, becomes the first choice. Firstly, we apply the MobileNet on the dataset of about 9,000 face pictures, which distributes one-third for each with and without the mask, the rest for incorrect wearing. Then, we divide the whole project into conduct two tasks. One aims at the accuracy of MobileNet for checking whether the mask is worn when the other focuses on the result of inspecting the mask are correctly worn or not. After that, by observing the features of the neural network and adjusting hyperparameters, the test accuracies of these two tasks could surprisingly reach above 90%, which is acceptable. Under the model training situation, the MobileNet demonstrates a splendid performance, which implies a tremendous potential for its development for practice situations.

The rest of this paper is organized as follows. Section 2 gives an overview of the proposed methods, including MobileNet, a widely used lightweight neural network. Section 3 shows the experimental process and the testing results of formerly mentioned tasks and the stages of optimizing the method by adjusting the parameters and getting the outcomes. The limitations to be improved and prospects are discussed in Section 4.
4. Methods Used
The testing dataset of this project is download from Kaggle, which is subdivided into 3 folders, with-mask, without-mask, and with-mask-incorrectly. Then, we chose the first version of the MobileNet from the series to deploy and train the oriented image set since this generation demonstrates the first valid definition of depthwise separable convolution and the overall net structure, which are dedicated to reducing the amount of calculation and parameters. Once ensuring the algorithm is available, the experiment around the model training begins. As below, we cover an introduction to MobileNet V1.

4.1. MobileNet V1
In this part, we first show the model's architecture, including different types, amounts, and combinations of layers used, followed by the innovative resource-saving principles of MobileNet, and finally, an even dwindling Thinner Models with width multiplier and resolution multiplier. Table 1 demonstrates the data structure of the Mobile Net [1]. A batch norm and ReLU nonlinearity follow all the Standard convolutional layer and Depth-wise Separable convolutions used. Finally, the result will be fed into the SoftMax layer for classification. Those pool layers are removed through the whole 28 layers of the network (AVG pool and softmax are excluded here), but stripes are frequently applied for down sampling [10]. As for the training consumption, the model puts almost all intensive operations into one × one convolution. There are 95% of training duration, and three of fourth parameters spend on those operations, and the rest for fully connections' use.

| Type / Stride | Filter Shape | Input Size       |
|---------------|--------------|------------------|
| Conv / s2     | 3 × 3 × 3 × 32 | 224 × 224 × 3    |
| Conv dw / s1  | 3 × 3 × 32 dw | 112 × 112 × 32   |
| Conv / s1     | 1 × 1 × 32 × 64 | 112 × 112 × 32 |
| Conv dw / s2  | 3 × 3 × 64 dw | 112 × 112 × 64   |
| Conv / s1     | 1 × 1 × 64 × 128 | 56 × 56 × 64  |
| Conv dw / s1  | 3 × 3 × 128 dw | 56 × 56 × 128   |
| Conv / s1     | 1 × 1 × 128 × 128 | 56 × 56 × 128 |
| Conv dw / s2  | 3 × 3 × 128 dw | 56 × 56 × 128   |
| Conv / s1     | 1 × 1 × 128 × 256 | 28 × 28 × 128 |
| Conv dw / s1  | 3 × 3 × 256 dw | 28 × 28 × 256   |
| Conv / s1     | 1 × 1 × 256 × 256 | 28 × 28 × 256 |
| Conv dw / s2  | 3 × 3 × 256 dw | 28 × 28 × 256   |
| Conv / s1     | 1 × 1 × 256 × 512 | 14 × 14 × 256 |
| 5× Conv dw / s1 | 3 × 3 × 512 dw | 14 × 14 × 512   |
| 5× Conv / s1  | 1 × 1 × 512 × 512 | 14 × 14 × 512 |
| Conv dw / s2  | 3 × 3 × 512 dw | 14 × 14 × 512   |
| Conv / s1     | 1 × 1 × 512 × 1024 | 7 × 7 × 512  |
| Conv dw / s2  | 3 × 3 × 1024 dw | 7 × 7 × 1024   |
| Conv / s1     | 1 × 1 × 1024 × 1024 | 7 × 7 × 1024 |
| Avg Pool / s1 | Pool 7 × 7   | 7 × 7 × 1024    |
| FC / s1       | 1024 × 1000 | 1 × 1 × 1024    |
| Softmax / s1  | Classifier | 1 × 1 × 1000    |
Comparing to those networks commonly approaching to deeper and more complicated, the model aims to build a small but efficient architecture, and this intention perfectly satisfied the requirements. It is a challenge to not only maintain the stability but also reduce the complexity of the model. For MobileNet, the standard convolution filters are divided into depth-wise convolution filters and pointwise convolutions to form depth-wise separable convolutions to cope with the issue [6].

![Figure 1. (left) depth-wise separable Convolution, (right) pointwise convolution.](image1)

For calculating resources saving, the depth-wise separable convolutions can reach \( \frac{1}{N} + \frac{1}{Dk} \) of one with standard convolution filters, here \( N \) and \( D_k \times D_k \) stands for output channels the kernel size respectively. Its goal is to reduce the model size while maintaining the model performance and improve the model speed. For instance, when \( N \) is relatively large, if a 3x3 convolution kernel is used, the calculation amount of depthwise separable convolution can be reduced by about 9 times compared with standard convolution.

![Figure 2. (left) standard Convolution (right) depth-wise separable Convolution [1].](image2)

Figure 2 shows the basic structure of one depthwise separable revolution compared to a standard convolutional layer. We can easily find that more ReLU layers are used, which enhances the model's generalization ability. Even though those functions used already significantly reduced the scale of the model, engineers always encounter the demand of an even smaller model. The width multiplier and resolution multiplier may help. The width multiplier mainly reduces the number of channels in proportion.
In contrast, the resolution multiplier is mainly used to reduce the size of the feature graph in proportion. Introducing these two parameters will undoubtedly degrade the performance of MobileNet. The decision is to make a compromise between accuracy and computation and between accuracy and model size.

5. Experiment Results and Analysis
The experiment develops with testing MobileNet on the dataset with 9,000 images into three categories (with-mask, without-mask, and with-mask-incorrectly). Specifically, the dataset has already been cleaned and equally distributed across each class. The experimental dataset in this section is obtained from Kaggle, which Vijay Kumar shares. All the experimental results are obtained by adjusting different hyperparameters. There are two tasks, one aims to check whether the mask is worn, and the other focuses on inspecting the mask is correctly or wrongly worn. It is ideal for training the model with the whole dataset with 6,000 images for each task. Training, validation, and testing directions divide the whole dataset into three parts with the proportion 8:1:1 of all those images with 128*128. The experimental results show that the proposed method performs well in the data set. For task 1, the with-without check, the highest accuracy reaches 87.96% when the learning rate range from 0.01 to 0.001 and epoch is 50; as for task 2, the correct-incorrect check, the trained model shows a 93.5% correctness for best performance when the learning rate range from 0.01 to 0.001 and the epoch runs 30 times.

5.1. Task 1: Whether the person wear a mask?
For the first attempt, when the learning rate through 0.001 to 0.0001, and the Epoch gets 25, the result turns out to be 52% of test accuracy, which shows a relatively lower accuracy. Then, we try to set the learning rate range from (0.0001, 0.001) to (0.001, 0.01), and meanwhile, change the epochs as 25, 30, 50 successively to get as higher accuracy as possible. As the table shows below, the first column shows a regular increase as the value of Epoch grows. After we get ten times the initial learning rate, the second column displays that the model is stable and reliable with the given attributes. Not only does the learning rate start from 0.001 to 0.01 for the second line performs, in general, better than the former, when we set the Epoch as 50, but we also obtain the best result from so far of 87.96% of test accuracy.

| Learning rate range \ Epoch | 25  | 30  | 50  |
|-----------------------------|-----|-----|-----|
| (0.0001, 0.0001)            | 52.67% | 56.5% | 67.8% |
| (0.01, 0.001)               | 83.33% | 78.94% | 87.96% |

5.2. Task 2: Whether the person wear it correctly or not?
With the experience with task 1, we initially set the learning rate down to 0.0001 to 0.00001, which quickly results in probable overfitting, with an accuracy of 99.0%. Then, we start to adjust the model to fit the dataset even more smoothly. At those continuous experiments, when we use the rate of 0.001 to 0.0001, the results turn out to be 73.46%, 88.34%, and 84.78%, respectively, when the epoch is appointed 25, 30, and 50. Although the outcomes are already meet expectations, even better results appear when the learning rate is from 0.01 to 0.001, as the third line shows below. However, we may assume that when the epoch gets 50, the obtaining result of 99.83%, which exceeds the envisaged result much and get genuinely close to 1, a total non-error model, may be a result of overfitting. Therefore, we exclude the splendid value and choose the accuracy of 93.5% as the highest one. Overall, similar to task 1, the model performs better test accuracy whenever we multiply the altogether learning rate. As well as the epoch, another factor that directly affects the outcome, the optimized model’s performance is always one of the accompaniments to its rise.
Table 3. the accuracies obtained by Task2

| Learning rate range | 25  | 30  | 50  |
|---------------------|-----|-----|-----|
| (0.0001, 0.00001)   | 99.0% | \ | \ |
| (0.001, 0.0001)     | 73.46% | 88.34% | 84.78% |
| (0.01, 0.001)       | 89.7% | 93.5% | 99.83% |

6. Discussion

6.1. Limitations

On the one hand, we find that the application of deep separable convolution and introduced hyperparameters of MobileNet truly downsize the consumption no matter the training time and parameters used. On the other hand, we need to admit that it will also lose accuracy certainly. Although the research group proved to gradually promote the stability of the algorithm of v1 to v3, it is always a dilemma to be both faster and more accurate.

Within this experiment, the decrease of time cost is not that clear, but there are still several hidden worries of the model. Even though the experiment results surprisingly get to above 90%, which shows that the model has excellent performance, the realistic situation could be much more complex on the more extensive scale of the sample size and stricter response requirements. When considering the feasibility of the detection model into production, every factor matters from basically the algorithm itself to the detailed implementation, such as the hardware required, the maintenance needed, and the optimization always expected. All of those elements may compromise the ideal results we get. Moreover, the model needs times of training to get unchanged stability.

6.2. Future studies

Researchers make their efforts to figure out a method to improve the performance of the MobileNet even a tiny step while keeping the absolute superiority on the control of computation’s size. Additionally, MobileNet will hopefully be modified and equipped with more computer vision applications, especially within compact or portable devices.

Combined with the need for face mask detection, the MobileNet technologically owns the possibility of further implementation on all kinds of mobile or concrete machines. Those models trained by this Convolutional Neural Networks method are always looking to perform correctly and consistently. Even further, MobileNet can be used for more significant tasks, such as recognizing identities even with a face mask, which may help filter those covered suspects from the crowds. There are much more functions to achieve or improve. The general trend is always pursuing higher and higher accuracy.

7. Conclusion

After reviewing the MobileNet v1 and executing the whole experiments based on it, it is clear and sure to declare that the MobileNet gets its superior forces on the function of Face Mask Detection.

Although there is still a long way for the architecture to get into mass production, its capability is already apparent even at the current preliminary planning stage. The introduction and experiment center on MobileNet show the potential of being deployed on machines, sluice gates, and almost those devices with limited capacity.

References

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.0486v1, 2017.

[2] M. Sandler, A. G. Howard, M. Zhu, A.Zhmoginov, and L-C Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In CVPR, 2018.
[3] A. G. Howard, M. Sandler, G. Chu, L-C Chen, B. Chen, M. Tang, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for MobileNetV3. arXiv preprint arXiv:1905.02244v5, 2019.

[4] M. Wang, B. Liu, and H. Foroosh. Factorized convolutional neural networks. arXiv preprint arXiv:1608.04337, 2016.

[5] A. Krizhevsky, I. Sutskever, and H. G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In ILSVRC-2012, 2012.

[6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In CVPR, 2017.

[7] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. CoRR, abs/1512.06473, 2015.

[8] M. Courbariaux, J.-P. David, and Y. Bengio. Training deep neural networks with low precision multiplications. arXiv preprint arXiv:1412.7024, 2014.

[9] K. Simonyan, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[10] Mobilenet source code. Available from https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.