CS230

Facial mask recognition Using Residual Network Backbones

Tiancheng Zhang
Dingchen Sha
Stanford University
tz11@stanford.edu
shadc@stanford.edu

1 Introduction

During the pandemic of Covid-19, there has been a protocol of wearing facial mask in the public area. Out of this consideration, the facial recognition technology can be developed to ensure the public health safety, and deep learning is considered as a good candidate to perform such task. In this project, convolutional neural networks (CNN) is implemented to detect face images where people wear no masks or wear masks improperly.

2 Related work

The fast revolution of CNN has enabled a plethora of standard face recognition systems. However, Alzu'bi et al. stated that due to the increased complexity of facial variation, the traditional face recognition models such as AlexNet, VGG16 and VGG19 should be improved or substituted [1]. More layers are needed to overcome the variation and residual network (ResNet) is usually implemented to alleviate the degradation of training accuracy.

By doing a further literature review, we found that there is some research that contributes to detecting face masks. A project proposed a model to detect masks using the yolo3 framework, in which the researcher applied the yolo3 algorithm to extract the face prediction area, and calculate the skin exposure rate of the mouth and nose of the face using the gray image, so as to judge whether a person is wearing a mask properly or not [2]. This model achieved a recognition rate of 86.6%.

On the basis of the SSD algorithm (Single-Shot Multi-box Detection), another project developed the SSD-Mask model, which introduced a channel attention mechanism to improve the performance in expressing salient features, and the model demonstrated good detection results in the detection of mask-wearing with AUC of 0.941 [3]. However, the results also show that this model is less accurate in detecting smaller objects, and it demands a lot of computational resources if the model is gigantic.

We import the ResNet-50 neural network from this repository.[4] The ResNet-50 model is considered as a baseline model, and is mentioned by Zheng Zhu et al. that using ResNet-50 as backbone of the model can ensure the recognition times be within 2000ms, which shows the practical in the real world scenario [5].

3 Dataset and Features

We referred the dataset called "Real-World Masked Face Dataset" (RMFD) [6]. This dataset uses web crawling to capture the images from real people, and has been organized and cleaned. The dataset contained 2500 images of the masked face of 525 people, and 80000 images of the same

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

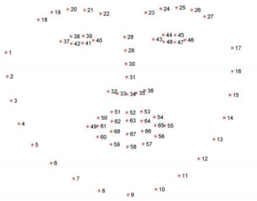


Figure 1: Dlib's 68 Face Features

subjects without masks. Since the images are organized into separate folders, the images are labelled accordingly during training and validation.

The aim of this project also includes the recognition of improper facemask-wearing as well as the warning of those people, where the RMFD dataset has not provided such fashion. Therefore, more images that mask covering the face improperly are required. Since it is hard to capture enough images that people wear masks improperly (such as not covering their nose) and properly, we decided to use a synthesis tool that can fit masks to face images in the wild by utilizing the 3D morphable model as part of our training set and validating set [7]. In this tool, we input images of people not wearing a mask, and generate images of people wearing a mask in a proper way or improper way. To realize this functionality, the tool applied the Dlib 68 points Face landmark Detection with OpenCV and Python [8], in which we can access the face features like eyes, eyebrows, and nose so that we can put the simulated masks on the face accurately by setting features points as Figure 1 shown above. In this process, we run the algorithm to randomly simulate different styles of masks, such as color, shape, pattern, and texture. To expand our synthesis strategy, we selected points 3, 9, 15, 52, 58 to generate an improper wearing image as the Figure 7 in the appendix. To generate images of people wearing masks properly, we chose points 3, 14, 27, 30, 57 as Figure 8 in the appendix. To begin with, we made all without-mask images that have directory names starting with "z" to wear masks improperly, and the synthesized fashion is shown in appendix. We synthesized 535 images of people wearing masks improperly and put them into the without-mask category. In addition, we made all without-mask images that have directory names starting with "y" to wear masks properly and move those images into the masked face category. The masks styles are enriched, so that the model should perform well. Around 6000 images are used for synthesis, but due to the poor quality of some images, only around 3400 masked images are generated.

Hand engineering will be used for the testing of our model, since the quality of images in RMFD data set slightly deviates from the complex real world images. Real world images will be collected by hand engineering for both the masked face, unmasked face, and improper masked face.

4 Methods

Given the discussion above in the related work, we decided to choose pre-trained ResNet-50 architecture for the model training. There are several reasons we choose ResNet-50 over other models. To begin with, since the limit increase of AWS machine is not approved, the computational power of the local machine cannot support a super large model. Specifically, VGG-19 has 143M trainable parameters, whereas ResNet-50 has 25M trainable parameters and MobileNet-V2 has 3.5M trainable parameters. Considering the relative small size of the RMFD data set, even though data augmentation is applied, more trainable parameters are needed to ensure the complexity of our model. To sum up, ResNet-50 is the most appropriate model given the constraint of accuracy, computational power, as well as data set size.

The ResNet-50 is a 50-layer deep residual neural network that is modified from ResNet-34 architecture. [9] The reason of choosing residual network is trying to eliminate the degradation problem caused by stacking significant layers in a plain neural network. The degradation problem is due to the saturation of accuracy as well as the difficulties of solvers in approximating the identity mappings of multiple nonlinear layers. The idea of using deep residual learning is to let layers fit a residual mapping. The illustration of the building block is shown below. The proof of residual learning is shown. If $\mathcal{H}(x)$ is an underlying mapping to be fit by a few stack layers and it is hypothesized that multiple non-linear layers can asymptotically approximate the underlying complex function, then this is equivalent to

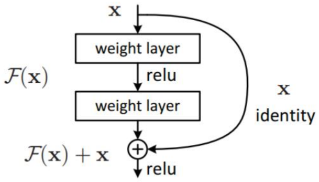

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2: The basic demonstration of residual network unit.

hypothesize that those layers can also approximate the underlying mapping function subtracted by $\mathbf{x}$, and therefore: $\mathcal{F}(x) := \mathcal{H}(x) - \mathbf{x}$. Therefore, the original function becomes: $\mathcal{F}(x) + \mathbf{x}$, and the new function is passed by the shortcut notation shown in Figure 2 during the forward propagation process. The advantage of residual learning is that no extra parameters are added but $\mathbf{x}$.

The illustration of residual network architecture (ResNet-34) is shown below in Figure 2. The shortcut connections are inserted to turn the network into its counterpart residual version. It can also be observed that the input and output are of same dimensions, so that the identity shortcuts can be directly used. The difference between ResNet-34 and ResNet-50 is that the latter uses a stride of 3 rather than 2 of the shortcut connection, which in turn increases the number of layers of the architecture to 50.

The main workflow of our project can be broke into three parts: firstly, we train our mini model with a small amount of data to ensure the viability of our model. once the model is feasible, we train the model with the whole RMFD data set and test the model with our test_mask script. The test_mask script firstly performs the blobFromImage function, which pre-processes the image by mean subtraction and scaling, then uses the *res10_300x300_ssd_iter_140000.caffemodel* to performs the face detection on the image [10]. Once the face is recognized by this imported model, the previously trained model will be deployed to give a classification on the faces. A rectangular box will be drawn on the recognized face and the confidence level will be shown. The test result image is shown in the appendix. To test efficiently, we reformatted the code into a new script called test_masks, which can test multiple images at one time, but only show the comparison of ground truth and prediction result as a data frame. Thirdly, considering the performance of the test data from both the RMFD data and the hand-collected data, modification of the training data set will be made to check if the model can be further improved.

5 Experiments/Results

Prior to train the whole dataset on AWS machine, a small subset of 200 images is ran on the local machine. The batch size is adjusted to 16 to fit the number of training examples. ResNet-50 performs really well on this size of the data, and subsequently the more images is used for the training process. The learning rate is 10^{-5} because transfer learning is used, the number of epochs is 5, and the batch size is shrunk to 4 in order to fit the large data set. Another trial is made BS with 8 and number of epoch to be 20, which is the maximum capability of our local machine. Overall the test result reaches a really high precision for the test set.

Upon the milestone, the limit increase of AWS machine is still not approved, so that the model is still ran on the local machine. Due to the limit that the video RAM of our graphics card is only 8GB, there is a limit of our training dataset. Another model with ResNet-50 is run with Batch Size = 8, learning rate = 10^{-5}, and image size = $64 \times 64 \times 3$. The number of training epochs is chosen to be 20, and it is experimentally verified that this amount of epochs will not cause over fitting in the validation step. Dropout is applied for layers at a rate of 0.5 to reduce over-fitting. The size of the data set is around 90000 images, and considering the size, we use a 80%-20% split of the training set and validation set. We also applied data augmentation to enhance the training size. The same setting is applied for all the remaining training processes. Upon training, the validation result is shown as Figure 3. To further validate our model, we propose to use some of the real life images serve as another metric of analyzing our model. According to the result in the Appendix, the accuracy is 80.9%, so that the model can be improved. To be specific, the model can recognize the person who wears mask properly, but three faces with nose uncovered are recognized as masked. The drop of accuracy from the validation set to the testing set casts doubts on the robustness of this model. There are two potential reasons causing such problem. On one hand, our model might suffer from over-fitting. However, from the training loss plot in Figure 4, it can be observed that the training loss drops in the first epoch and then converges over the remaining epochs. The convergence of training loss and validation loss shows that over-fitting should not be a problem. On the other hand, the RMFD data set can be tuned. To begin with, this data set is not general in the norms that people wear masks. There is no images of the improper face-mask wearing. Secondly, there is only around 2500 masked face images, whereas the number of unmasked face images is around 80000. Last but not least, there is one miss classification that the person who wears mask properly is recognized as not wearing mask.

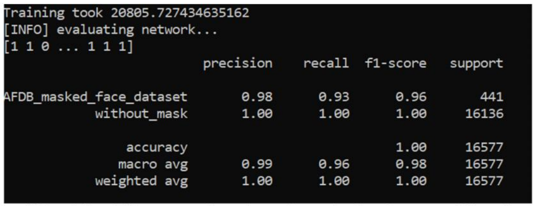
To further improve our model, we utilized the synthesized data to rearrange our data set. As all images are synthesized and moved to the designated category, the model is trained again with the same hyper parameters. The validation results are shown in Figure 5 and Figure 6. It can be observed that even though the accuracy of masked face is dropping in the validation part, both the recall for the masked face data set and the macro average recall of the data set is improved. The macro average means the average of unweighted mean per label. Consequently, it can be concluded that synthesizing the data from unmasked face to masked face is helpful for the data set to recognize the masked face correctly rather than directly recognize them as unmasked. The model is tested for the hand-collected data, and compared with the original model, its accuracy increased to 95.2%. Specifically, it becomes more robust at recognizing the improper masked face. The detailed comparison images are listed in the Appendix.

6 Discussion/Future Work

For this project, the face mask recognition model is built based on the ResNet-50 backbone architecture. The state-of-art technologies of face recognition are explored, and the reason of choosing ResNet-50 is explained. Specifically, the computational power, the amount of available data, the recognition time determine the choice of this model. The advantage of residual network in the context of deep neural network is explained, and the workflow of the project is introduced. The model is firstly trained with the original RMFD data set, which contains 2500 masked faces and 80000 unmasked faces. The validation result shows a 100% accuracy, but the performance of this model on the test set, which contains all the real-world images, drops to the accuracy of 80.9%. The error analysis is performed, and finally the main reason is due to the data set. There are two main problems of the data set. To begin with, the distribution of the category is highly skewed with a ratio of around 30:1. The lack of masked face images results in the over-strict classification of masked face, which might not efficient in the public auditing circumstances due to the huge amount of false negatives. Secondly, the image quality of the RMFD data set is not very desirable. This is found during the data synthesis part. Some of faces cannot be detected by the mask-rendering script, so that the yield rate of the data synthesis is much lower than expected.

Based on the error analysis, the model is trained again with modified data. To be specific, some of the unmasked faces are synthesized into masked face or improper masked face. Even though the validation accuracy drops from 100% to 99%, the recall on the masked face dataset is improved from 93% to 98%. Furthermore, the accuracy of this model on the test set increases to 95.2%. Specifically, this model is better at recognizing the improper masked face, which is one of the main model objective.

To summarize, the model can be further enhanced in the following ways. Firstly, it is observed that both models, no matter what kind of the training data set is, fails to correctly classify one masked image. As this image is inspected, it is found that the face is not only covered by the mask but also covered by the hat. This can be more challenging for the model to perform classification since our data set is lack of such kind of images. Out of this consideration, the generalizability of the data set can be improved in terms of dressing code, age, ethnicity and many other aspects. That is to say, more data and various data synthesis technique needs to be equipped to improve the model performance. Secondly, if the computational power can be increased, the model should be improved. Due to the limit of computation power, the training images are confined to have size of 64×64 pixel, whereas the imported source code utilizes a 224×224 pixel training size. Images with higher resolution can definitely provides more useful features to the model, so that the performance can be improved under the same data set.



```
Training took 20805.727434635162
[INFO] evaluating network...
[1 1 0 ... 1 1 1]
                         precision    recall  f1-score   support

AFDB_masked_face_dataset       0.98      0.93      0.96       441
            without_mask       1.00      1.00      1.00     16136

                accuracy                           1.00     16577
               macro avg       0.99      0.96      0.98     16577
            weighted avg       1.00      1.00      1.00     16577
```

Figure 3: The training result of ResNet-50 network using the whole dataset. Number of training epoch is 20, and batch size is 8. Overall the precision is pretty high, but the recall could be further improved, which is considered as the future work direction.

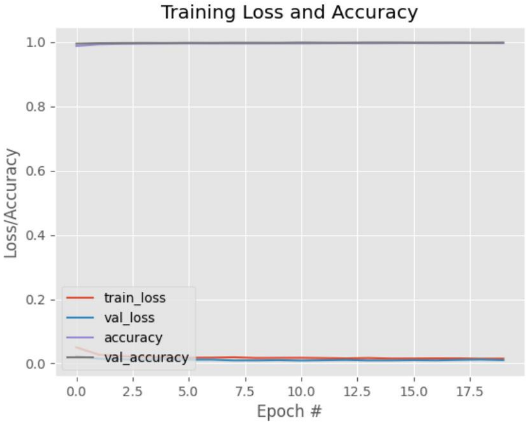
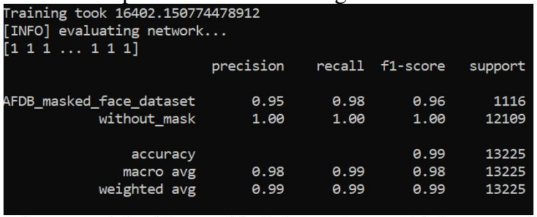


Figure 4: The evolution of training loss and accuracy over the number of epochs. It can be observed that the training loss drops during the first two epochs and then converges over the remaining epochs.



```
Training took 16402.150774478912
[INFO] evaluating network...
[1 1 1 ... 1 1 1]
                         precision    recall  f1-score   support

AFDB_masked_face_dataset       0.95      0.98      0.96      1116
            without_mask       1.00      1.00      1.00     12109

                accuracy                           0.99     13225
               macro avg       0.98      0.99      0.98     13225
            weighted avg       0.99      0.99      0.99     13225
```

Figure 5: The performance of our model with modified data set, on which we synthesized some of the data. It can be observed that even though the precision drops 3%, there is a 5% increment of the recall of this model, which alleviates our previous problem that the masked face is directly recognized as unmasked faces.

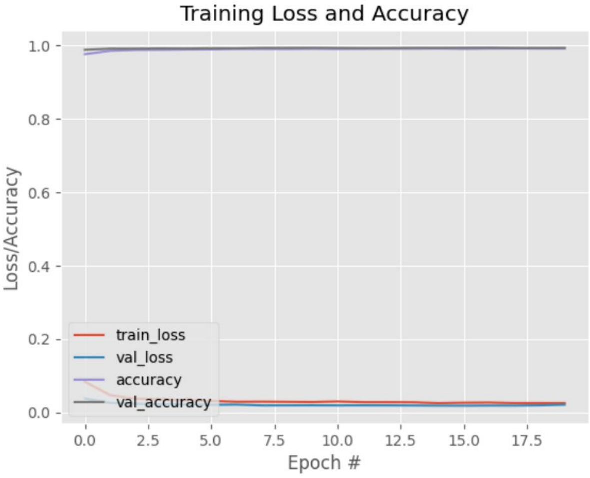


Figure 6: The evolution of training loss and accuracy over the number of epochs for the modified data set. It can be observed that the training loss drops during the first two epochs and then converges over the remaining epochs. No obvious difference is observed compared with the previous data set.

References

[1] Ahamad Alzu'bi, Firas Albalas, Tawfik AL-Hadhrami, Lojin Bani Younis, Amjas Bashayreh. Masked Face Recognition Using Deep Learning: A Review. https://doi.org/10.3390/electronics10212666

[2] J. He, "Mask detection device based on YOLOv3 framework," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2020, pp. 268-271, doi: 10.1109/ICMCCE51767.2020.00067.

[3] M. Xu, H. Wang, S. Yang and R. Li, "Mask wearing detection method based on SSD-Mask algorithm," 2020 International Conference on Computer Science and Management Technology (ICCSMT), 2020, pp. 138-143, doi: 10.1109/ICCSMT51754.2020.00034.

[4] Face Mask Detection. https://github.com/chandrikadeb7/Face-Mask-Detection Major libraries included: tensorflow, keras

[5] Zheng Zhu, Guan Huang, Jiankang Deng, et al. Masked Face Recognition Challenge: The WebFace260M Track Report. arXiv: 2108.07189v1

[6] Real-World Masked Face Dataset, RMFD.
https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset

[7] Wear-Mask 3D. https://github.com/jhh37/wearmask3d

[8] Facial landmarks with dlib, Open-CV, and Python.
https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/#:~:text=Dlib's%2068%2Dpoint%20facial%20landmark,facial%20landmark%20detection%20models%20exist

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition.https://arxiv.org/pdf/1512.03385.pdf

[10] Deep Learning: How openCV's blobFromImage works. https://pyimagesearch.com/2017/11/06/deep-learning-opencvs-blobfromimage-works/

Appendix

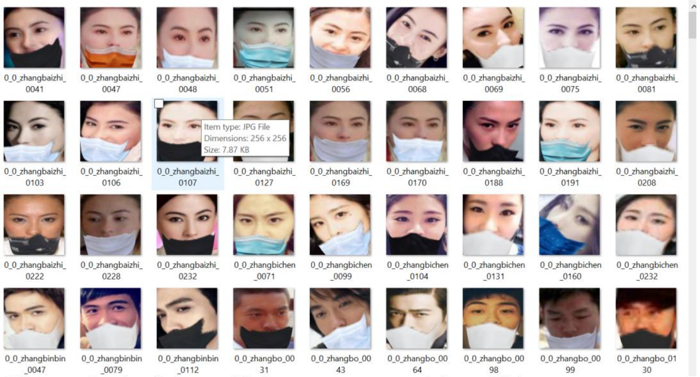


Figure 7: The synthesized data with improper covered face. This can augment our dataset.

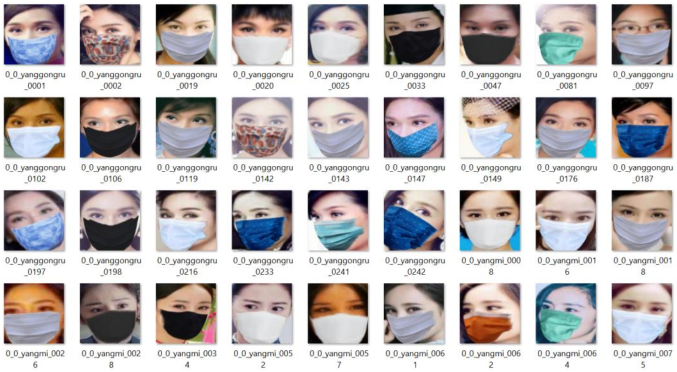


Figure 8: The synthesized data with proper covered face. This can augment our dataset.

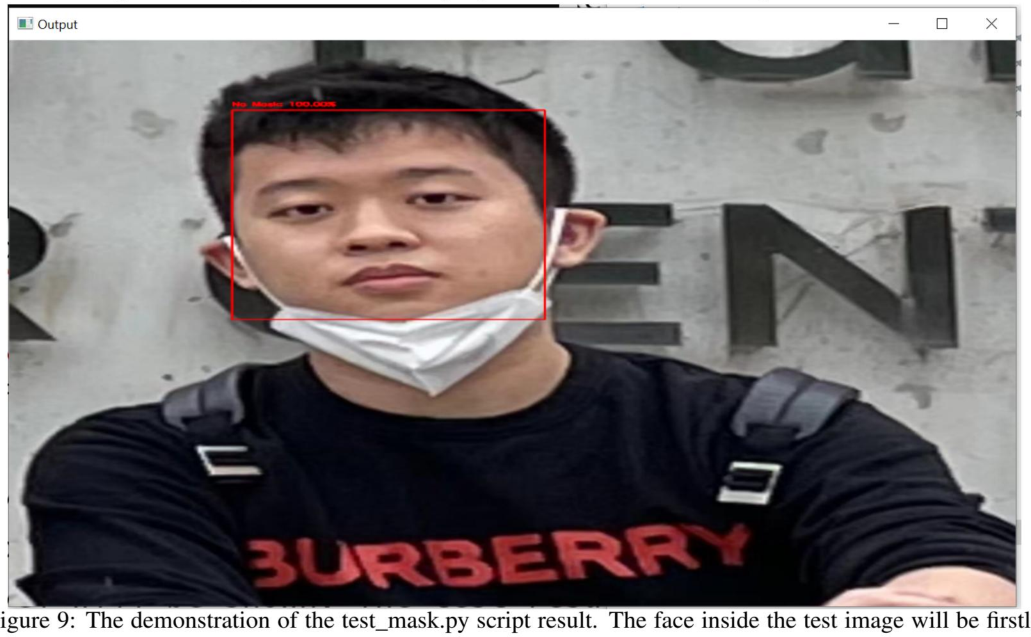


Figure 9: The demonstration of the test_mask.py script result. The face inside the test image will be firstly detected by the imported model, then the face mask classification problem will be performed by the trained model. Once the classification is finished, the rectangular box will be rendered on the face, combined with the classification result and its confidence value.

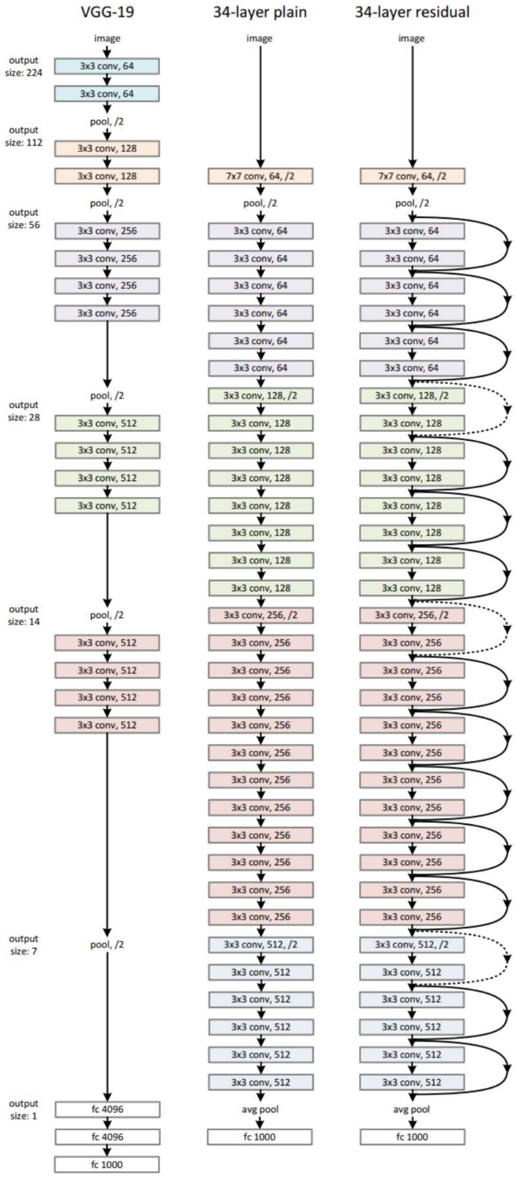
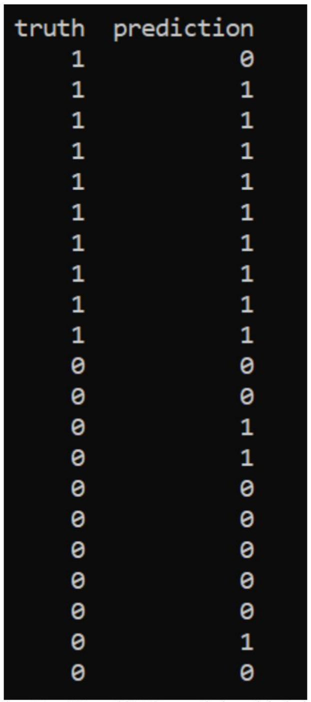
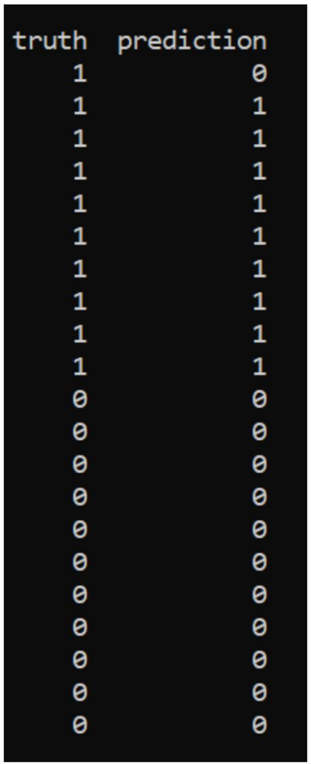


Figure 10: The illustration of the VGG-19 neural network, the expanded 34-layer plain neural network, and a 34-layer residual neural network. The dimensions of the plain neural network are equal among each layer to ensure the mapping requirement. Different from ResNet-34, ResNet-50 has a stride of 3 shortcut connection, which increases the number of layer to 50.



```
truth  prediction
    1           0
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    0           0
    0           0
    0           1
    0           1
    0           0
    0           0
    0           0
    0           0
    0           0
    0           1
    0           0
```

Figure 11: The test-set performance of the ResNet-50V2 model ,which is trained by the original RMFD data set. The accuracy of this model is 80.9%.



```
truth  prediction
    1           0
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    1           1
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
    0           0
```

Figure 12: The test-set performance of the ResNet-50V2 model ,which is trained by the modified data set. Synthesized data is added. The accuracy of this model is 95.2%. The performance of the model is improved in term of recognizing improper masked face.