Masked Face Recognition based on Residual Networks

Fei Gao, Shiwen Tang, Shiyu Hu
{gaof, shiwent, Shiyuhu}@bu.edu

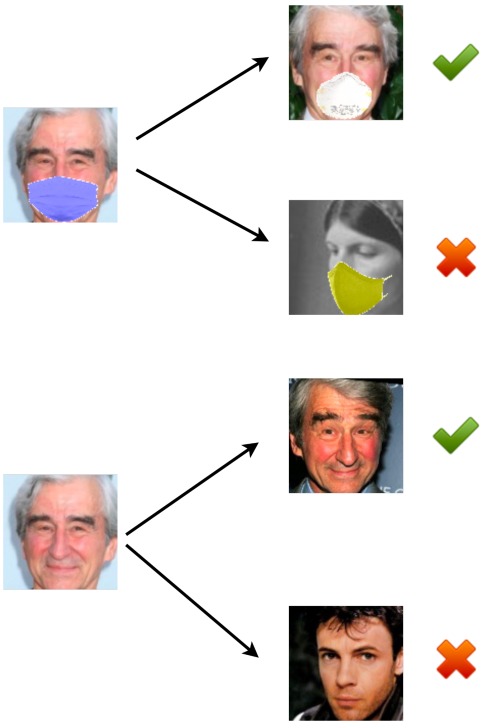

Figure 1. The example scenario of testing data

1. Task

With the development of deep learning, face recognition has been widely used in accessing control and paying systems. However, during the pandemic, wearing masks makes it much more difficult for face recognition. In this way, it is necessary to improve the performance of face recognition for users. In this project, our task is to do a multi-class classification by using a feature extraction backbone called Residual Networks. The input is face images of individuals and the model predicts which individuals (or classes) these images belong to. In general, it would be difficult to capture discriminative features for face recognition. So we mainly re-implement two powerful models named IResNet [6] and the ResSaNet [3]. Besides, there are two datasets we used to perform, the MS-Celeb-1M dataset(unmasked dataset) and the masked dataset. The masked dataset is created by ourselves using the add virtual masks toolkit which is called MaskTheFace [5]. At last, an additive Angular Margin Loss (ArcFace) was used to calculate the loss, and accuracy was used to evaluate the performance.

2. Related Work

Before implementation, we used several papers to help us understand the process.

Jiankang et al. organized Masked Face Recognition (MFR) challenge. The challenge has two tracks: the InsightFace track and the WebFace260M track. The InsightFace Track Report collects three different test sets and develops an online face recognition model testing system that provides a comprehensive evaluation of face recognition models [2]. The major flaw in the model is that the dataset it used was racially inclined. The challenge employs two existing datasets as training data: the MS1M dataset (The MS1M training dataset is cleaned from the MS-Celeb-1M dataset.) and the Glint360K dataset. For our project, we used the MS-Celeb-1M dataset. MS-Celeb-1M is a large-scale data set for face recognition. This dataset is rich in information that helps to improve recognition accuracy. Besides, it can also be applied to real-world applications such as image captioning and news video analysis [4].

Residual Network is a backbone that combines CNN and Self-attention modules into the same network for face recognition. It has a good result in simultaneously obtaining local and global information about the face region [3]. Inout et al. proposed an improved version of Residual Network for image and video recognition, which is called IResNet. All three main components of a ResNet are addressed by this improvement: the information flow through the layers, the residual building block, and the projection shortcut. With this improved architecture can learn extremely deep networks and there is no optimization difficulty when depth increases [6]. Another improved version of ResNet is ResSaNet, which is proposed by Weiyi et al. ResSaNet integrates CNN and the Self-attention module into the same network. It captures the local and global information of the face area at the same time, achieving greater results on both masked and unmasked datasets than IResNet [3].

For the loss, Jiankang et al. propose an additive Angular Margin Loss (ArcFace) for highly discriminative face recognition. By training with large-scale data, this method of loss could outperform the models trained by softmax loss [1].

3. Approach

We utilized the MaskTheFace toolkit proposed in [5] to add virtual masks on face images in the modified MS1M-RetinaFace dataset [4]. The model we re-implement is IResNet proposed in [6] and ResSaNet proposed in [3]. Besides, we optimize the ArcFace Loss [1] when training our model. Stochastic gradient descent (SGD) is chosen as the optimizer during training.

3.1 ResNet

ResNet is a residual learning framework that can easily train substantially deeper networks. It is made up of four types of residual blocks shown in figure 2.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2. Structure of ResNet

The main concept of the residual block is the skip connection. It refers to the output of the block being added to its input which skips layers in the block. Compared with the traditional convolutional neural network, ResNet has a good performance when the layer goes deeper. In our experiment, ResNet-50 is trained for comparison.

3.2 IResNet

IResNet was inspired by the structure of ResNet. As shown in Figure 3, IResNet uses a bottleneck from 1x1 convolution to 3x3 convolution. By using this architecture, the 3x3 will have the biggest number of channels and higher detection of patterns.

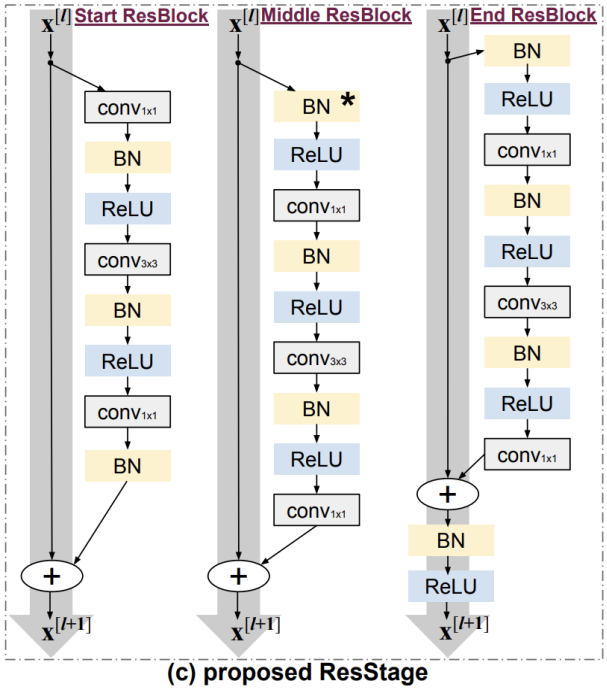

Figure 3. ResBlock (IBasic Block) architecture [6].

IResNet is composed of four different stages. Its ResBlock is built in each stage, also called IBasic block. In addition, the number of blocks could be changed. For example, the IResNet-50 has [3, 4, 14, 3] blocks in four different stages and IResNet-100 has [3, 13, 30, 3]. In this project, we use both IResNet-100 and IResNet-50 models.

stage	output (size, #channel)	IResNet-50	ResSaNet-50	IResNet-100	ResSaNet-100
conv1	$112\times112, 64$	Conv 3×3	Conv 3×3	Conv 3×3	Conv 3×3
stage1	$56\times56, 64$	IBasic $\times3$	IBasic $\times3$	IBasic $\times3$	IBasic $\times3$
stage2	$28\times28, 128$	IBasic $\times4$	SE-IBasic_F $\times4$	IBasic $\times13$	SE-IBasic_F $\times13$
stage3	$14\times14, 256$	IBasic $\times14$	SE-IBasic_F $\times14$	IBasic $\times30$	SE-IBasic_F $\times30$
stage4	$7\times7, 512$	IBasic $\times3$	IBT $\times3$	IBasic $\times3$	IBT$\times3$
FC	$1\times1, 512$				
#Params		43.57×10^6	44.13×10^6	65.16×10^6	66.35×10^6
#FLOPs		6.31×10^9	6.29×10^9	12.12×10^9	12.11×10^9
Inference Time (ms)		4.51	5.18	7.03	9.69

Figure 4. Structure of IResNet and ResSaNet [2].

3.3 ResSaNet

ResSaNet was modified from the IResNet. Different from the IBasic block in IResNet, the IBasic block in ResSaNet replaces ReLU with parametric ReLU as the activation function. SE-IBasic_F block and IBT block are introduced in the ResSaNet as shown in Figure 4.

3.3.1 SE-IBasic_F block

In the SE-IBasic_F Block depicted in Figure 5, SE is the abbreviation of “squeeze and excitation”. Squeeze is implemented by the global average pooling layer and excitation is implemented by the convolutional layer. By using the SE block, useful features are emphasized and useless features are suppressed. F

stands for the flexible ReLU, which contains the depth-wise convolution layer. It is able to expand the pixel-wise modeling capacity.

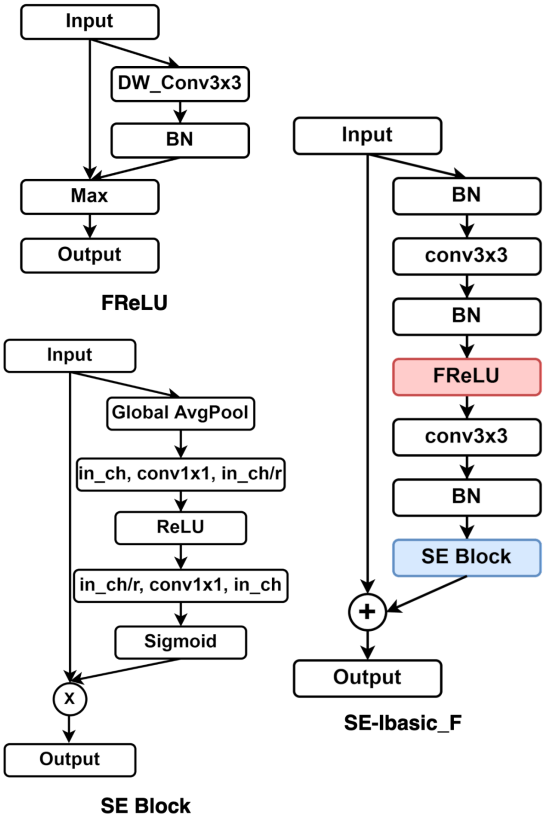

Figure 5. SE-Ibasic_F architecture

3.3.2 IBT block

IBasic Transformer (IBT) block shown in Figure 6 is constructed by an IBasic Self Attention (IBSA) block and a Mobile Inverted Bottleneck Convolution (MBConv) block. IBSA block integrates the self-attention module, substituting the convolutional layer with the Multi-head Self-attention (MHSA) block [7]. Different from the convolutional layer which focuses on the relationship between adjacent pixels, the self-attention module captures long-range dependencies within an image. The model including both convolutional layers and MHSA block could take advantage of these different features and gets better performance. In addition, the MBConv block gets low dimension inputs, expands it to high dimension, and then projects the data back to the low dimension. In this way, the performance of the model could be improved.

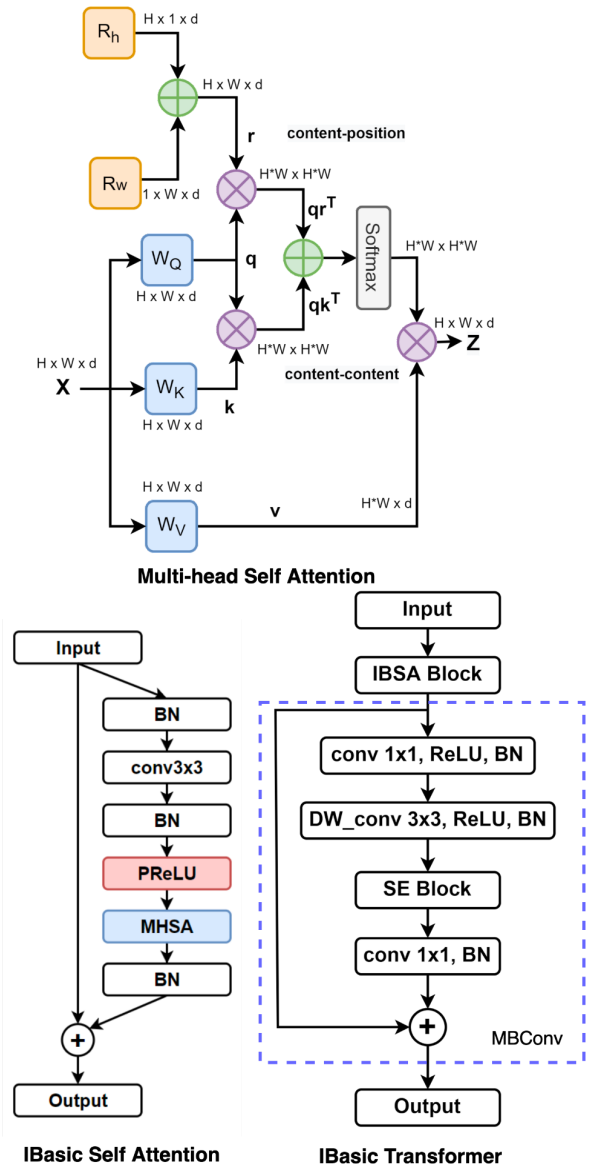

Figure 6. IBasic Transformer architecture

3.4 ArcFace Loss

The traditional loss function used in deep face recognition is softmax loss. However, there is no explicit optimization of the feature embedding in the softmax loss function for ensuring higher similarity for samples within a class and differentiation for samples between classes[1]. Then we used Additive Angular Margin (ArcFace) loss [1]. ArcFace loss computes the geodesic distance margin in the arc space. It enhances the discriminative power of the face recognition model as well as stabilizes the training process. It is presented as follows:

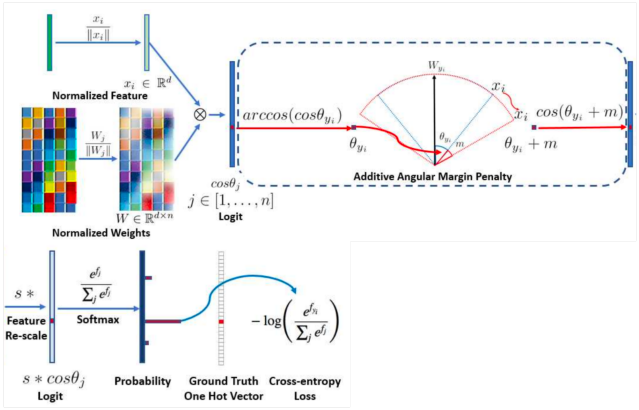

Figure 7. ArcFace loss [1].

Based on the feature x_i and weight W normalization, we can get the logit value $cos\Theta_j$ according to $W_j^T x_i = \| W_j \| \| x_i \| cos\Theta_j$. Then we calculate and get the angle between the feature x_i and the ground truth weight W_{y_j}. After that, we add an angular margin penalty m on the ground truth angle Θ_{y_i}, and calculate $cos(\Theta_{y_i} + m)$ and multiply the feature scale s with all logits. Finally, after passing through the softmax function, the logits contribute to the cross-entropy loss [1]. The equation of Arcface loss is presented as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{scos\theta_j}}$$

N and n are the batch size and the class number respectively. Θ_{y_i} is ground truth angle, m is margin penalty, and s is feature scale.

4. Datasets

In this project, the MS-Celeb-1M dataset is used, containing over 10 million images of nearly 100,000 individuals [4].

Figure 8. Preview of MS-Celeb-1M dataset

We add different types of masks to the images in MS1M-RetinaFace by using the MaskTheFace toolkit [5]. About 8% of total images are added successfully. The number of masked face images for each individual ranges from 0 to 180. To get a relatively large dataset, we keep images of the individual who has more than 80 masked face images. Besides, data augmentation is implemented by flipping the images horizontally. The cleaned dataset contains around 1,000 classes and ~200,000 images. 90% of the data is classified as training data and the rest is classified as testing data. In addition, for reference, we create an unmasked face image dataset by collecting the original images of images in the masked dataset.

5. Evaluation Metrics

In our experiment, accuracy is used as the evaluation metric. After we predicted the class label of the face image and get the ArcFace loss. The prediction was used to calculate the accuracy. It would be True when the prediction is correct and False when the prediction does not match with the ground-truth label. Ture positive (TP) stands for the times that the model correctly classifies a positive sample as positive. False-negative (FN) stands for the times that the model incorrectly classifies a positive sample as negative. False-positive (FP) stands for the times that the model incorrectly classifies a negative sample as positive. Ture Negative (TN) stands for the times that the model correctly classifies a negative sample as negative. The overall formula for accuracy is shown below:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

6. Results

We implemented the virtual mask adding, the construction of ResNet, IResNet, and ResSANet, and the model evaluation.

For all the models, we set the optimizer to SGD, the learning rate to 0.05, the momentum to 0.9, and the weight decay to 5e-4. In addition, we use accuracy to measure the performance of these three different models.

6.1 Models with Different Batch Sizes

As shown in Figure 9, IResNet-50 reaches the best performance with batch size = 512.

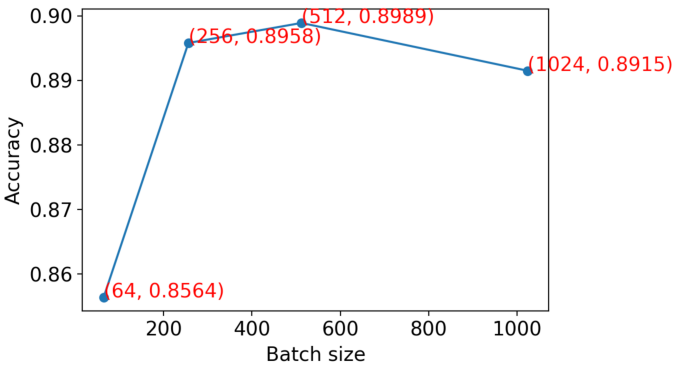

Figure 9. Accuracy on IResNet50 with different batch sizes

6.2 Evaluated on Masked vs. Unmasked Dataset

By training the baseline model IResNet-50, we got the accuracy as follows:

Batch = 512, IResNet-50	
Type	Accuracy
Masked	89.89%
Unmasked	94.94%

Table 1. Accuracy of IResNet-50 on different datasets

It is obvious that IResNet-50 would get higher accuracy on the unmasked datasets, which is 94.94%. However, when testing on masked datasets, the IResNet-50 will only have an accuracy of 89.89%. The reason is that it would be hard for a model to extract features when adding the masks on the face images.

6.3 Different Models

By training the different Residual models with the same number of batches, 512, we get the accuracy as follows:

Batch = 512, Masked	
Model	Accuracy
ResNet-50	89.62%
IResNet-50	89.89%
IResNet-100	89.07%
ResSaNet-50	90.17%
ResSaNet-100 (Batch = 256)	92.08%

Table 2. Accuracy of the masked dataset on different models

The ResSaNet model gets higher accuracy on the masked face testing datasets than the ResNet model and IResNet model. With the same 512 batches, ResSaNet-50 gets the highest accuracy. However, when we tried to test the ResSaNet-100 of 512 batches on the masked face datasets, the GPU would be out of memory. So finally, we used a smaller batch size, 256, to get the accuracy of ResSaNet-100. It turned out that ResSaNet-100 got the best performance, which is 92.08% accuracy.

7. Conclusion

In this project, we implemented IResNet and ResSaNet for the evaluation of masked face recognition as well as non-masked face recognition. By training on the masked-face datasets, ResSaNet-100 has an obvious excellent performance for face recognition with respect to the baseline model IResNet. Besides, by training on the same IResNet-50 model, unmasked datasets have higher accuracy than masked datasets.

For future work, since we cannot get the result of model ResSaNet-100 with batch size 512, we would like to train ResSaNet-100 in larger datasets with higher memory GPUs to evaluate the performance. In addition, we would like to design and apply different loss functions for masked-face datasets. At the same time, more metrics are expected to be found to measure the performance of different models.

Appendix A. Detailed Roles

Table 3 is shown below.

Appendix B. Code repository

https://github.com/shiyuhu1933/ResSaNet

References

1) J. Deng, J. Guo, N. Xue, S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4690-4699), 2019.
2) J. Deng, J. Guo, X. An, Z. Zhu, S. Zafeiriou. Masked face recognition challenge: The insightface track report. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1437-1444), 2021
3) W. Y. Chang, M. Y. Tsai, and S. C. Lo. ResSaNet: A Hybrid Backbone of Residual Block and Self-Attention Module for Masked Face Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1468-1476), 2021.
4) Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision* (pp. 87-102). Springer, Cham, 2016.
5) A. Anwar, A. Raychowdhury. Masked Face Recognition for Secure Authentication, arXiv.org, 2020.
6) I. C. Duta, L. Liu, F. Zhu, L. Shao. Improved residual networks for image and video recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 9415-9422), 2021.
7) A, Srinivas., T, Lin., N, Parmar., J, Shlens., P, Abbeel., A, Vaswani. (2021). Bottleneck Transformers for Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16514-16524), 2021.

Table 3. Team member contributions

Name	**Task**	**File names**	**No. Lines of Code**
Fei Gao	Add Virtual Mask Prepare dataset Implement ArcFace Loss Implement ResNet Implement ResSaNet Project Status Report Final report and presentation	/data/transfer_data_to_H5.ipynb /loss/ArcFaceloss.ipynb /model/ResSaNet/IBasic_Transformer.ipynb /train_and_test/train_test_Resnet50.ipynb	442
Shiwen Tang	Prepare dataset Data preprocess and data loader Implement train and test Implement ResSaNet Project Status Report Final report and presentation	/data/data_augmentation.ipynb /data/hdf5_to_dataset.ipynb /data/photos_clean_rename.ipynb /data/train_test_split.ipynb /model/ResSaNet/IBT.ipynb /model/ResSaNet/SE_iBasic_F.ipynb /model/ResSaNet/ResSaNet.ipynb /train_and_test/train_test_ResSaNet.ipynb	1121
Shiyu Hu	Implement IResNet Implement ResSaNet Project Status Report Final report and presentation	/model/ResSaNet/IBasic.ipynb /model/iResNet.ipynb /train_and_test/train_test_IResNet.ipynb	573