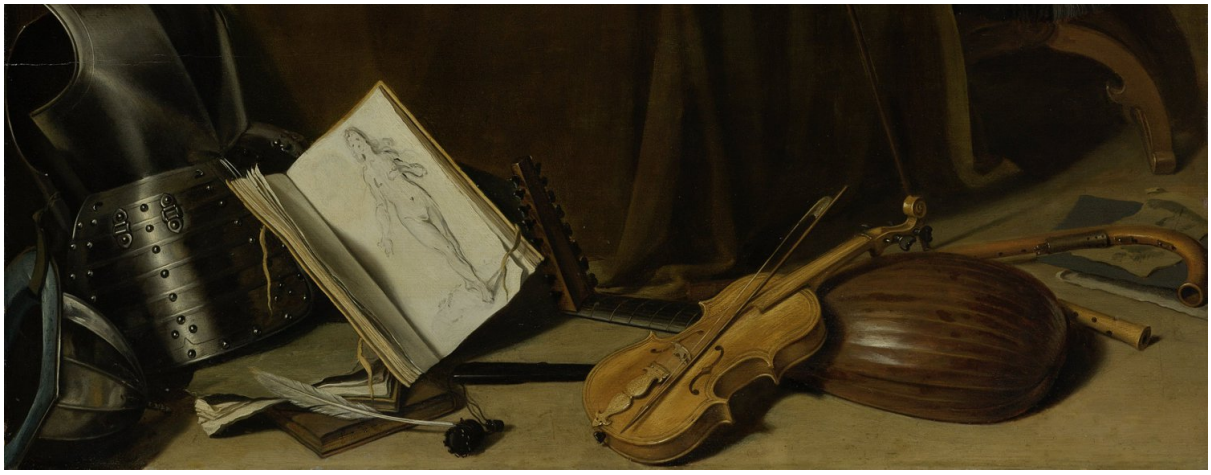


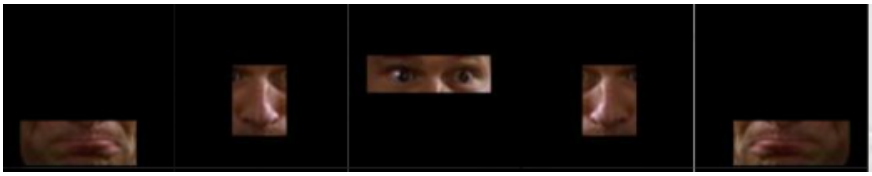
Final Project Report

Data Augmentation

1. Traditional Computer Vision Augmentation:
    - Gaussian Blur : Applying a Gaussian filter to the image can help the model become more robust to small changes in image blur.
    - ColorJitter : Randomly adjusting the brightness and saturation of an image can help the model become more robust to changes in lighting conditions.
    - RandomCrop : Randomly cropping an image can help the model become more robust to changes in the location of the face within the image.
    - RandomAffine : implement random translation and rotation on the image.
    - Horizontal Flip : Horizontal Flipping an image horizontally can help the model become more robust to changes in the direction of the face.
    - Vertical Flip : Vertical Flipping an image vertically can help the model become more robust to changes in the position of the face.
    - GrayScale: Converting an image to grayscale can help the model become more robust to changes in color.

2. Style Transfer : 利用Style Transfer將圖片的texture轉換，希望增強training data的多樣性。

3. LandMark the facial feature：以下為我們認為不受age影響的feature，我們有將這些feature之圖片特別抽出來進行training
    - eyes
    - nose
    - mouth
    - ears



4. DC-GAN & StarGAN ：有嘗試使用DC-GAN與StarGAN合成不同年齡的照片，但由於效果不佳，因此未採用至Training Dataset中。

Model

MobileFaceNet

The MobileFaceNet architecture utilizes residual bottlenecks from MobileNetV2 as its primary building blocks. The detailed structure of the main MobileFaceNet architecture is outlined in Table 1.

This architecture uses smaller expansion factors than those found in MobileNetV2. The non-linearity function employed is PReLU, which has been found to perform better in face verification tasks when compared to ReLU. The network uses a fast downsampling strategy at the start, an early dimension-reduction technique at the final convolutional layers and a linear 1x1 convolution layer as its feature output layer. Batch normalization is applied during training, and batch normalization folding is utilized before deployment.

MobileFaceNet with FC512 is a lightweight deep neural network architecture for facial recognition that uses a fully connected layer with 512 neurons (FC512) at the end of the network.

Feature Embedding

The final output of the network is a 512-dimensional vector, also known as a feature embedding, which represents the input image in a compact and discriminative feature space (將input圖片通過mobile-facenet model，投影到512 dimension之超球) .

The 512-dimensional vectors generated by MobileFaceNet with FC512 can be compared to other vectors generated from other images to calculate the cosine similarity. This method is also known as "ArcFace" and it is a loss function for training deep neural networks for face recognition tasks.

Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine similarity is a value between -1 and 1, where -1 represents completely dissimilar vectors, 0 represents orthogonal vectors and 1 represents identical vectors. The cosine similarity metric is often used in face recognition tasks as it is relatively fast to compute and can provide a good measure of similarity between two feature embeddings.

ArcFace Loss

在進行training時，ArcFace Loss會將MobileFaceNet Output的512 dimension embedding轉換成one - hot vector，並與gound truth之one hot vector計算cross-entropy loss。計算完loss以後再將結果backpropogate回MobileFaceNet update training的參數

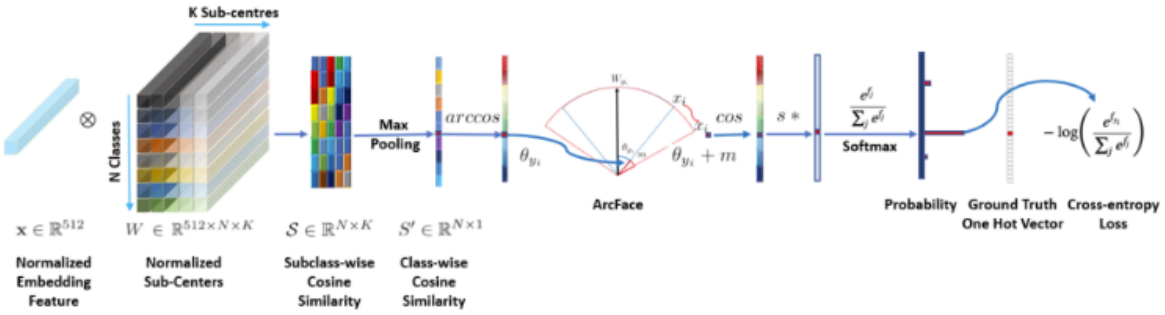


Table 1

Each line describes a sequence of operators, repeated n times. All layers in the same sequence have the same number c of output channels. The initial layer of each sequence utilizes a stride of s, while all the other layers use a stride of 1. All the spatial convolutions in the bottlenecks have a kernel size of 3x3.The expansion factor t is always applied to the input size. GDConv7x7 denotes a Global Depthwise Convolution with 7x7 kernels.

Input	Operator	t	c	n	s
$112^2 \times 3$	conv3×3	–	64	1	2
$56^2 \times 64$	depthwise conv 3×3	–	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1×1	–	512	1	1
$7^2 \times 512$	linear GDConv 7×7	–	512	1	1
$1^2 \times 512$	linear conv1×1	–	128	1	1

Experiment

1. **SEResNet50 + ArcFace**

   Combining the SEResNet50 architecture with the ArcFace loss function can be a powerful combination for face recognition tasks. Firstly, without implementing any data augmentation to train the model. We surprisingly got an unacceptable results that the training accuracy could reach about 0.8, but the validation accuracy can't even reach 0.1. It is obviously an overfitting problem due to the following cause:

   - The dataset is not that large. (Each persons has only few images , but with the large age difference among their face image.)

   → so we implement data augmentation

2. **SEResNet50 + ArcFace + traditional augmentation**

   Combining the SEResNet50 architecture with the ArcFace loss function can be a powerful combination for face recognition tasks. And using traditional augmentation techniques on input images. The performance of the training accuracy is 0.99,the validation accuracy is 0.25, the testing accuracy is 0.56.

   However we can notice that the performance on validation set and test set aren't that good. That's because it becomes overfitting due to the gradient is broken (i.e. the gradient is stuck into the local minimum). We assume that it may results from the following reasons.

   - The SEResNet50 is too complex for handling this kind of problems.

   → so we change the model backbone

3. **MobileFaceNet + ArcFace + traditional augmentation**

   Combining the MobileFaceNet architecture with the ArcFace loss function can be a powerful combination for face recognition tasks. And using traditional augmentation techniques on input images.The performance of the training accuracy is 0.99,the validation accuracy is 0.26, the testing accuracy is 0.68.

   We can notice that MobileFaceNet architecture got the better performance than SEResNet50 does. We assume that MobileFaceNet architecture is more suitable for solving our case.

   → make more efforts on data augmentation

4. **MobileFaceNet + ArcFace + traditional augmentation + style transfer + face features landmarks**

   Combining the MobileFaceNet architecture with the ArcFace loss function can be a powerful combination for face recognition tasks. And using traditional augmentation techniques in conjunction with style transfer and landmarks (利用較有特徵的部位進行training，如 : 眼睛) to serves as input images. The performance of the training accuracy is 0.99,the validation accuracy is 0.35, the testing accuracy is 0.72.

   Based on the observation, we find out that the implementation on data augmentation can significantly raise the model performance to a better level.

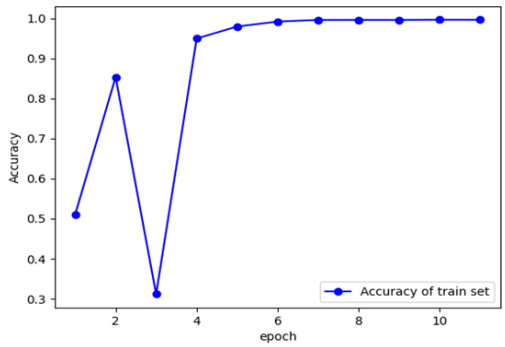
   <p.s.> By the way we have also tried the following method to generate the aging face image of each person.

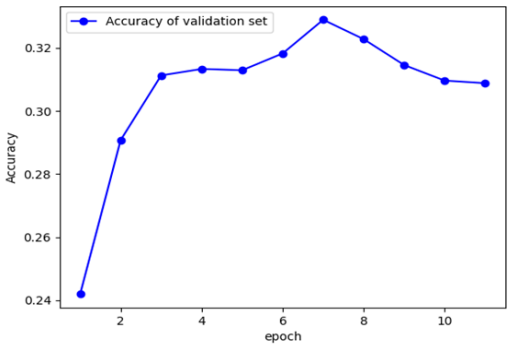
(1) Apply the CNN model to predict the age of each face image in order to labelling the age of each face image in the CALFW dataset.

(2) Apply the GAN method to generate the face image with different age of each person to achive the goal of augmenting.

Unfortunately, our CNN model can't predict the age of face precisely(e.g. It predict a face image of young teenager to be 70 yr), so the GAN model can't generate the acceptable results also.

Result





Training Accuracy : 0.99

Validation Accuracy : 0.33

Testing Accuracy : 0.72

AUC : 0.78

Reference

1. **ArcFace: Additive Angular Margin Loss for Deep Face Recognition**
2. **MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices**

分工

蘇勇誠、吳宇文、謝念恩共同寫code、training、讀paper、寫report