Received: 20 July 2021 | Revised: 4 February 2022 | Accepted: 28 February 2022

DOI: 10.1049/ipr2.12479

IET Image Processing

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

A single-stage face detection and face recognition deep neural network based on feature pyramid and triplet loss

Tsung-Han Tsai | Po-Ting Chi

Department of Electrical Engineering, National Central University, Taoyuan City, Taiwan (R.O.C)

Correspondence
Tsung-Han Tsai, Department of Electrical Engineering, National Central University, Taoyuan City, Taiwan (R.O.C).
Email: han@ee.ncu.edu.tw

Abstract
A practical deep learning face recognition system can be divided into several tasks. These tasks can be time-consuming if each task is executed with the original image as the input data. And the feature extractors used by different tasks may duplicate its function. In this paper, a multi-task training method based on feature pyramid and triplet loss to train a single-stage face detection and face recognition deep neural network is proposed. As a single-stage work, every task's data is passed through the same backbone network to avoid duplicate computation by sharing the weights and computation. The whole network is established using feature pyramid and anchor boxes to localise the face position, using triplet loss to establish the feature extractor, and finally matching the feature through a simple math function. The benefits of the approach are faster computation speed and less memory usage. On an Nvidia 2080Ti GPU accelerator, this system can achieve 212 FPS for a 640 × 640 resolution input and maintains 92.4% accuracy on the LFW data set.

1 | INTRODUCTION

With the evolution of various technologies, face detection and identification systems has become more and more popular, and the application of the product has become more and more intuitive. A system with face recognition ability has the advantage of being able to identify people by their natural differences in appearance. It is not necessary to carry additional equipment. The face detection and recognition system can be divided into four main tasks: face detection, face correction, feature acquisition, and feature comparison. It is performed from the input of the image lens to the output of identity. However, it should also be able to tolerate noise such as different camera distance, rotation, light, and shadow. In the 1970s, heuristic algorithms have been used for face detection tasks [1], but these methods require some constraints on the application scenario, such as simple background, front face requiring, or other requirements, making it necessary to redesign the entire system once the application scene is changed.

A detection task is defined as obtaining the class and location of a single or multiple objects from a scene, often by using a feature acquisition such as HOG, SIFT, or Haar wavelet transform coupled with a classifier such as SVM, linear regression, or decision tree [2, 3]. These methods usually extract region proposals through the sliding window method and then extract the features and classify, or use the pre-built templates to extract an area that is similar to the target.

The main methods of face recognition are also quite similar. In addition to the differences between images, there are also various human-defined features to obtain the differences between faces. For example, contours, textures, or features mapped to high-frequency space are usually needed. The most common methods are template matching, Harr wavelet transformation etc. But these methods are similar to face detection in that they still require a specific application scenario defined by the network.

Tremendous progress has been made since deep learning, particularly the CNN-based method, has been used in this problem. Face analysis such as unconstrained face detection and face recognition has greatly improved. In neural networks research, object detection begins with the region proposal at various scales and locations is proceeding with a two-stage object classification. For example, the RCNN (regions with CNN features) [4] and its modified version Fast RCNN [5],

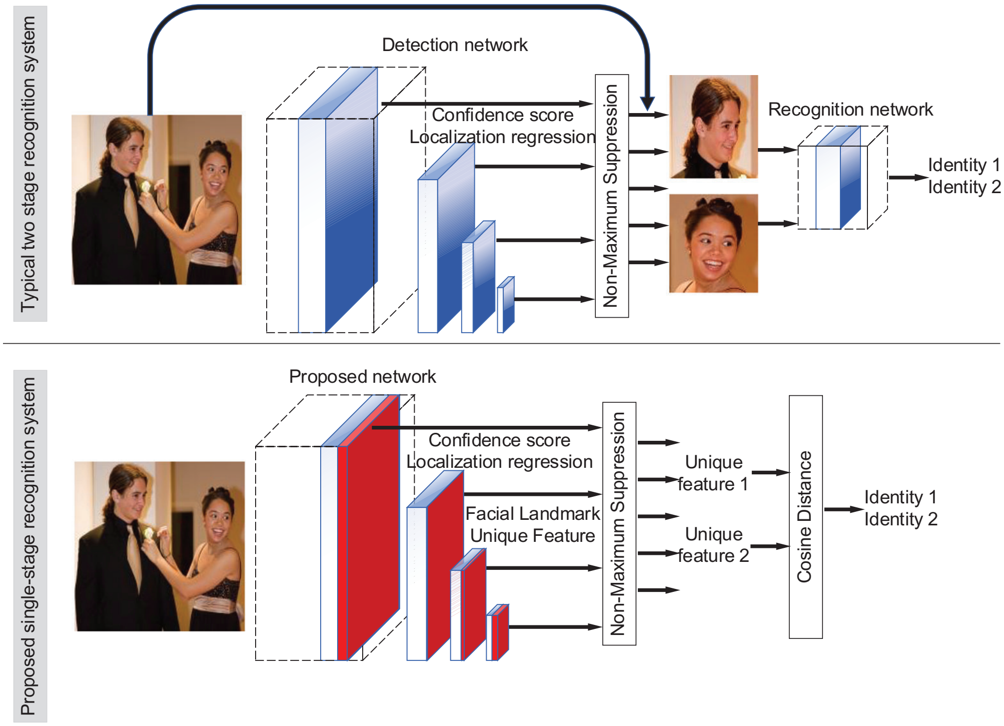

FIGURE 1 A comparison between typical two-stage model and proposed single-stage model

use a highly accurate neural network to identify the category of each proposed region and remove highly overlapping non-background boxes using NMS algorithms. This kind of algorithm still has duplicate computation for the overlapping region. There are three main types of networks, Faster RCNN, SSD (single-shot multiBox detector) [6], and YOLO (you only look once) [7]. Each of them has proposed a different approach to make a single network learn location and categories simultaneously. The widest usage of these algorithms is an architecture called feature pyramid. This architecture can improve the representation of object features of different sizes by mapping downscaled features in the convolutional network to sub-regions of the original picture at different scales. Thus, only one forward operation is required to obtain accurate results.

Many studies have begun to focus on these optimised networks, combining the powerful ability of neural networks to fit input–output pairs with experience in image processing and face recognition to propose a faster and more accurate neural network architecture. Throughout the entire face detection and recognition process, the common approach is to connect the detection model and the recognition model in series into a multi-stage system as the top in Figure 1.Although face identification systems typically use two-stage model: face detection model and recognition model individually, the two-model design may induce some problems:

1. Since two models are concatenated into one system, it will cause a lot of computational waste.
2. Since the recognition model needs to wait for the detection model to complete the task and recognise the face one by one, it will consume a lot of time.
3. As the number of faces in an image increases, the number of iterations of the recognition model execution will also increase.

In this paper, we propose a single-stage face detection and recognition neural network. It can detect and recognise multiple faces in a single picture by using a multi-task learning technique. We also combine the feature pyramids and triplet loss in this multi-task learning approach. A single backbone network with multiple output stages is required to train the output of each task simultaneously. The feature pyramids and anchor boxes are used for localisation, and the triplet loss is used for feature extracting. Eventually, feature matching can be realised by using a simple mathematical function to compare the similarity. The separation between feature extraction and feature matching allows the database to be updated without retraining the network. The comparison between our single-stage system and a typical two-stage recognition system is shown in Figure 1. This approach can have the following contribution:

1. We improve the two-stage model to a one-stage model to obtain faster computation speed on feature extracting.
2. We have less memory usage than the two-stage cascade network.
3. More simple usage in inferencing since it feeds the camera's raw input and outputs multiple identities.

2 | RELATED WORKS

Recent neural network studies have rarely designed the entire system for a single scene-specific task. Most of the research is optimised based on general tasks, such as the face detection method which is always based on general object detection research. The neural network architecture built from the object detection task is improved with the experience gained from the traditional face detection algorithm. The widely used object detector, SSD, uses the neural network to build feature pyramids and anchor boxes, corresponding to a subarea of the original map. By taking advantage of the computational sharing properties of the convolution layer, the amount of repeated computation for multiple regional proposals is greatly reduced.

The main implementation method for SSD is to connect an additional set of convolutions to each different size of output feature map after the fourth downsampling layer and define the output as class scores and box regressions. The number of network outputs is shown in Equation (1).

$$\text{output number } = \sum_{i} x_i * (N+4) * K_i \tag{1}$$

where i is the output layer, x_i is the size of feature map of the output layer i, K_i is the number of anchor size for different output layers, and N is the number of categories on the target network. For each categorical output, it needs to adopt the softmax categorical cross-entropy loss function, and the regression loss for box localisation.

In a recent study, Deng et al.'s RetinaFace represent a robust single-stage face detector [8]. It achieved outstanding results on the WiderFace dataset [9], by using Facebook's RetinaNet [10] as its primary network architecture. This network architecture is still widely used in the current design, such as Zhu et al.'s TinaFace [11]. As a generic object detection model, YOLO is also widely used where Qi et al.'s YOLO5Face [12] is based on and the newest version YOLOv5 as the object detector. They also use feature pyramids and anchor boxes to implement detection frames with multi-scale messages. In the output layer, as in MTCNN, five key points of the face are marked as auxiliary tasks, and a self-supervised face reconstruction method is proposed.

On the face recognition side, the Facebook team proposed DeepFace [13] in 2014, which can be considered as the first CNN system for face recognition tasks. This system aligns the face first in terms of angle and then feeds it into a nine-layer convolutional neural network to achieve good accuracy on the LFW (labelled face in the wild) dataset [14].

After this, Schroff and others in the Google team proposed the FaceNet [15] network, which improves the pre-processing where faces need to be aligned first. It uses the triplet loss to train the network. The main concept is to define that for each face image. There is a high-dimensional feature that could be computed. For the images of the same identity, the distance between features will be smaller than different identities. Each face will output 128-dimensional features after the network calculation, and the similarity is determined by Euclidean distance. Finally, it could reach a high accuracy in the LFW data set.

A neural network could fit an ideal mathematical function by performing a large number of convolutional operations. In the situation that the input and output data's dimension is almost the same, the difference between the different tasks is only in the definition of the loss function. As a result, it is easy to merge. At the same time, some studies suggest that the problem with neural networks is the unstable gradient descends process, and additional progress needs to be adopted to assist the gradient to descend. For example, in RetinaFace, the backbone network is first trained using the ImageNet dataset and then trained on the face detection task. Another method is to add additional data to the input or output. For example, adding the results of a pixel-level segmentation algorithm to the input will expect the neural network to focus more accurately on the target area. On the segmentation and recognition task, Dadashzadeh et al. proposed HGR-Net [16], which uses the results of a segmentation network to enhance the categorical classification network. But the disadvantage of this approach is also obvious. As the additional part to the network need to be run first as additional inputs, the execution time increases significantly.

In this case, multi-task learning is where multiple training tasks are parallelised on the output. It allows the neural network to train multiple different networks together and is usually in a situation where multiple tasks share the same input. If the network is to be used for a task with different inputs, it is necessary to adjust the input data to meet the requirements for each task or separate the training steps for a different task. Multitask learning has shown its value in many studies [17] and can effectively make the neural network more valuable. In the face detection task, MTCNN and RetinaFace also use auxiliary tasks to perform face alignment and get good results.

A complete face recognition system needs to integrate four tasks: face detection, face alignment, feature extraction, and feature matching. The most common way is to align the face from the face detection task and then input the sub-area of the original image to the neural network for feature extraction and matching. The cascading CNN connection is not efficient in reusing computation results. In this kind of architecture, the execution speed would be affected by the number of faces in an image, and the face recognition network must be repeated multiple times for each detected face area. For smaller computing platforms such as the Nvidia Jetson Nano or DNN Accelerator Stick, real-time computing may not be possible due to memory and speed limitations.

For the integration of face-related tasks, Ranjan et al. [18] proposed a model that can simultaneously perform the alignment task, the recognition task, the expression recognition, the location of key points in the face, and a masked situation, called the all-in-one face. They extracted hundreds of region proposals using the same selective search method in RCNN, and then feed the sub-image into the neural network to obtain the face classify score and features. However, this method still requires different small images and identifies the area separately.

Liao et al. [19] proposed a model with both a location alignment and a recognition task and used feature pyramids and triplet loss to perform a face location regression and identity recognition task. Since the recognition output only has one bounding box, the feature extraction does not work with multiple faces in the same picture.

To satisfy our network, the training data has to contain a massive number of human identity in a complex scene, and each image must contain one or more identities that also appear in other scenes in the dataset. Real data obtained through the internet does not satisfy this requirement, thus, we need to generate virtual data as a temporary solution. For better data generation, we adopt some stable and accurate work and combine them into a simple segmentation model. In the face segmentation part, the FCN architecture [20] proposed by Long et al. is the cornerstone of this task, which is different from the full-connection layer be placed at the end of the network. The output is the heat map of the categories' probability. It also uses multi-scale output, combining layers of different depths to enhance the robustness of the network.

Another famous research on semantic segmentation is the DeeplabV3 architecture proposed by Chen et al. [21, 22]. It features different atrous convolution instead of different sized convolutional kernels. As a result, even the kernel size remains 3 × 3, it can still have different Receptive Fields. To match this convolutional architecture, the authors also propose a network architecture called ASPP (atrous spatial pyramid pooling) to optimally adapt the characteristics of this architecture by running different atrous convolution and then concatenating the output of each layer. In this way, each different convolution branch can be treated as a branch of a different sized convolution kernel of Inception-v3 [23], but with a reduced computation amount.

3 | MODEL DESIGN

Since manual labelling is a very time-consuming task, it is easier to build a database by using existing open-source data. This paper will use different public datasets for training as well as validation. The detection network requires the dataset with the location of the face to train with the designed detection network, while the recognition network requires the dataset with the identity of the picture of the face. Typical face detection and recognition system uses a detection network to detect faces in an image, and then slices each region containing face and put them into the recognition network.

The main problem in this paper is how to make the face detection task and the recognition task share the same training data. The data required for the face detection network must label the location of a human face in a large number of complex scenes, and the face must appear at different scales in different locations. For face recognition tasks, the data must label the identity of the person, and the same person must appear multiple times. Since no publicly available dataset satisfies both requirements, this paper will use synthesised data to train the model.

For the face recognition task, we use LFW as the verification set and VGGFACE2 [24] as the training set. The LFW data set contains 13,000 face images, but only 1,680 people have more than two images. It is insufficient for neural networks that require large amounts of data. Thus this dataset is often only used for verification. VGGFace2 contains 3.31 million face pictures and 9131 identities, where each identity has more than 300 pictures on average. The face data covers different factors in time, makeup, accessories, expressions, and other changes. Since the content is very complex, the network trained by this dataset can be well covered and extrapolated to the real world.

In this paper, face segmentation is used as an aid to generate datasets that require training data to contain a face segmentation label. We use the part label annotation in the LFW dataset, which divides the image into geometric blocks and labels the three categories of face, hair, and background.

The experiment is by use of SegNet [25] architecture since this is now a very common encoding–decoding neural network. We trained this network using the LFW part labels dataset. The deconvolution layer used in the FCN paper performs well for small object classes according to the experiments in the SegNet paper. But the inference scene where this task focuses does not contain any mini-objects. There is no need to augment face images into multiple scaling. Only the images of a single face that have been aligned to 160 × 160 are used. Therefore, all deconvolution layers are replaced with linear upscaling functions to speed up the training session of the network.

We collected 2,000 images from public datasets on the Internet that did not contain humans and generated the mask image of the faces in VGGFace2 by the segmentation network. The data synthesis process is as follows.

1. Randomised the size of face samples. Our network input size is 640 × 640. Since face recognition results will have a significant decrease in accuracy when the size of the face picture is smaller than 80 × 80, we limit the minimum size of the face to no less than 0.2 times the size of the background image and the maximum size of the sample does not exceed 0.8 times.
2. Randomly position different samples on the background image with the overlap between samples not exceeding 0.2.
3. Computing the new bounding box as the detection task's ground truth.

As mentioned in the FaceNet paper, this approach also uses the mini-batch method but it takes more time to synthesise the images. Thus we mixed it with pre-generated images. If only training the image with 30 K image, the model will have a serious overfitting problem.

A face recognition system needs to perform both face detection and face recognition tasks simultaneously. If the computational resources are limited, the design needs to be smaller for a two-stage network. This may affect the final accuracy. Based on the experiment on RetinaFace, we perform an additional triplet loss to train a feature extractor with the location. The network architecture is shown in Figure 2. The backbone network adopts the Mobilenet V2 architecture with a reduced number of layers.

Following the official version of RetinaFace's work, a good result is obtained on three feature maps. The only difference

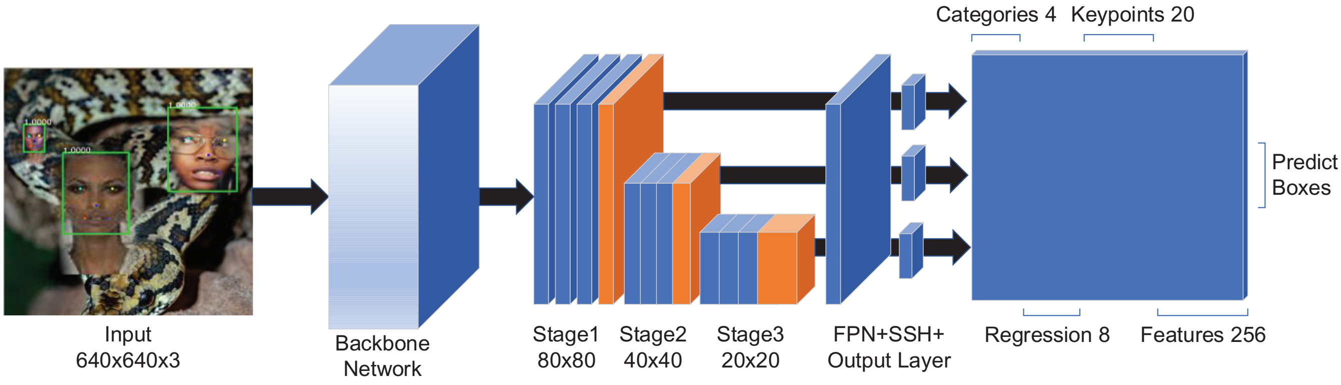

FIGURE 2 The proposed model architecture

is that the proposed model appends the feature extractor layer to three output layers than the original one. The number of parameters is only 3.1% higher and the number of computations is only 3.5% higher than the original model. It has almost no speed effect during inference sessions. The network is designed to take advantage of weights and computation sharing by convolution layer, and avoid the problem of duplicate computation. This is also the concept used in feature pyramid networks such as SSD.

4 | TRAINING STRATEGY

The objective of this paper is to generate a single-stage neural network with both detection and recognition functionalities, instead of generating the confidence score of the prediction bounding box first and then recognising the sub-regions from the original image. The experimental detail of the detection tasks is the same as that described in RetinaFace, except the way to generate the triplet set for the recognition task.

In the training session, to increase the complexity of the training set and to avoid the overfitting problem of the neural network model, we use different image processing methods to augment the images. In this paper, image cropping, deformation, flipping, filling, and normalisation methods are used. The reason for not using deformation is that this computation will remove the important information of human identity, making the face recognition task unstable.

The training parameters are shown in Table 1. The reason for using MobileNetV2 for training is that the lite network structure is much faster than the Resnet. It allows us to quickly verify different parameters and check the robustness of each training strategy.

To increase the robustness of the network, we adopt a pre-training model, which achieves a top-1 accuracy of 46.58% on ImageNet. The training process used two data selection strategies and corresponded to two different triplet matching methods. The first one is to place a single face with a multiple of 3 on a single picture, which is simpler and can be pre-generated easily. The other one is to place the face randomly on the background image. The data set is shown in Figure 3.

TABLE 1 Training hyperparameter

Backbone	MobileNet V2
Embedding dimension	128
Threshold	0.5
Batch size	16, 64
Optimiser	SGD
Initial learning rate	0.01
Momentum	0.9
Weight decay (L2 penalty)	5e-4
Input size	640 × 640
NMS threshold	0.35
Training epoch	250
Training data	The face synthesis data generated by the VGGFace2 dataset
Number of data	3000× batchsize for each epoch
Learning rate decay	Starting from the 40th epoch, every 20 epoch drops to 0.2 times
Pretrained model	ImageNet (46.58% top 1)

The anchor box used is all square in shape, as defined in Table 2. Each output stage corresponds to two sizes of anchor frames. The size of the corresponding original image from smallest to largest is 16, 32, 64, 128, 256, and 512. The size of the three output stages is 80 × 80, 40 × 40, and 20 × 20, corresponding to an output step of 8, 16, and 32.

The definition of triplet loss function for the two training strategies is shown in Equations (2) and (3). The losses for the other outputs of the network are the same as in the RetinaFace paper.

Hard Triplet in one sample:

$$\mathrm{Loss}_{\mathrm{same}} = \frac{1}{N}\sum_{i}^{N} \begin{array}{l} \max\Big(0, \left\| f\left(x_i, \mathrm{anchor}_i\right) - f\left(x_i, \mathrm{positive}_i\right)\right\|_2^2 \\ -f\left\|\left(x_i, \mathrm{anchor}_i\right) - f\left(x_i, \mathrm{negative}_i\right)\right\|_2^2 + a\Big) \end{array} \quad (2)$$

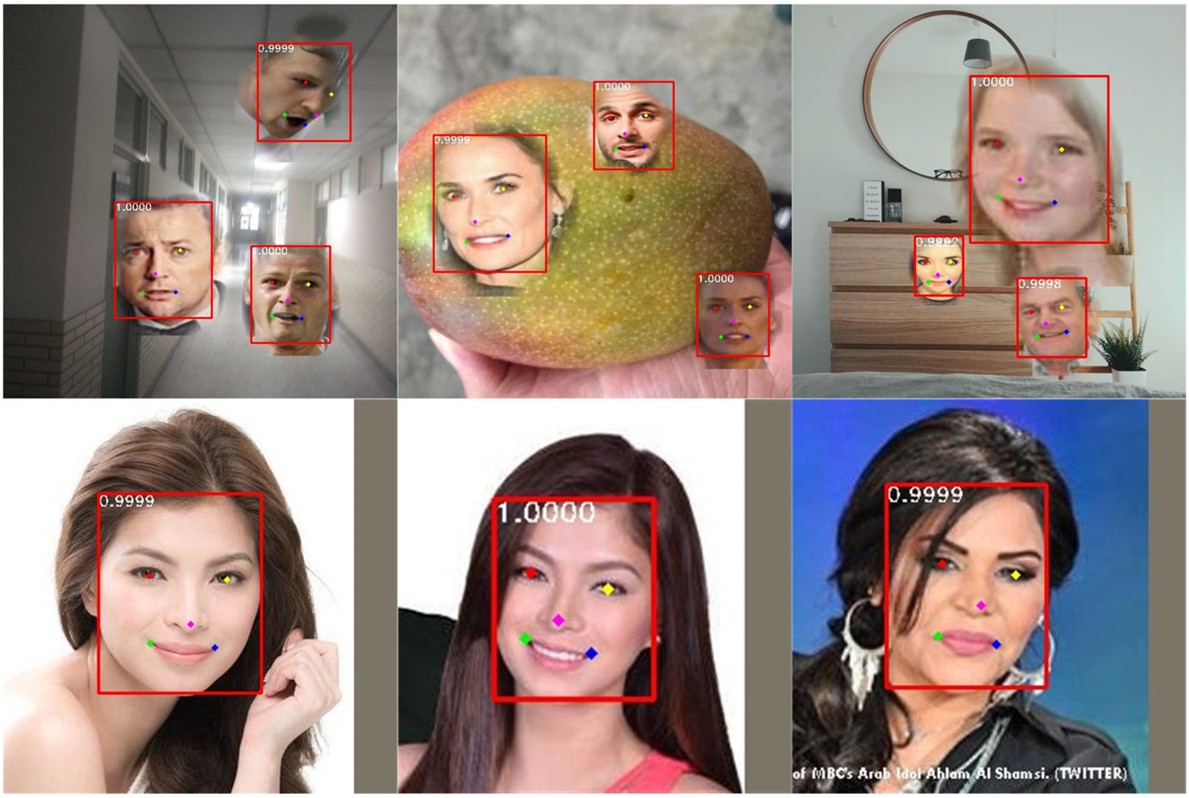

FIGURE 3 The dataset used to train the proposed network

TABLE 2 Rough parameter of proposed network

layer	Input size	Input channel	Kernel size	Stride	Parameter	FLOPs
Retina Net	640 × 640	3			~3.4 M	~5.0 G
Output_stage1	80 × 80	64	1 × 1 × 64, 2 × 2 1 × 1 × 64, 4 × 2 1 × 1 × 64, 10 × 2 1 × 1 × 64, 128 × 2	1	256 512 1280 16.3K	1.6 M 3.2 M 8.1 M 104 M
Output_stage2	40 × 40	128	1 × 1 × 128, 2 × 2 1 × 1 × 128, 4 × 2 1 × 1 × 128, 10 × 2 1 × 1 × 128, 128 × 2	1	512 1 K 2.5K 32.7 K	819 K 1.6 M 4.0 M 52 M
Output_stage3	20 × 20	256	1 × 1 × 256, 2 × 2 1 × 1 × 256, 4 × 2 1 × 1 × 256, 10 × 2 1 × 1 × 256, 128 × 2	1	1 K 2 K 5.1 K 65.5 K	409 K 8192K.0 M 26 M
Total					3.6 M	5.2G

Soft triplet split randomly in different samples:

$$\mathrm{Loss}_{\mathrm{diff}} = \frac{1}{N*N}\sum_{i}^{N}\sum_{j}^{N} \begin{array}{l} \max(0, \left\| f\left(x_i^a, \mathrm{anchor}_i\right) - f\left(x_i^p, \mathrm{positive}_i\right)\right\|_2^2 \\ - \left\| f\left(x_i^a, \mathrm{anchor}_i\right) - f\left(x_j^n, \mathrm{negative}_j\right)\right\|_2^2 + a) \end{array} \tag{3}$$

where N is the total number of pairs in the batch, x is the input image, i, j are the indexes of the images, f is the output of the anchor box with the highest overlap with this sample, a is the force margin between positive and negative sample, and anchor, positive, and negative are the indexes of the corresponding anchor box for each output. The difference is whether the comparison is between different outputs from the same sample or different samples. However, the comparison between different samples and the comparison between the same sample consume different computational resources for the backward operation. Thus, the batch size used for the comparison between the same sample is 64 and the batch size used for the comparison between different samples is 16.

The loss function of the entire network is called multi-object triple loss (MOTL) and is defined in Equation (4).

$$\text{Loss} = L_{\text{cls}}\left(p_i, p_i^*\right) + p_i^* L_{\text{box}}\left(t_i, t_i^*\right) + p_i^* L_{\text{pts}}\left(l_i, l_i^*\right) + L_{\text{tri}} \tag{4}$$

where L_{cls} is the loss of face category, $p_i$is the face type score predicted by anchor box i, p_i^* is the box based on hard negative mining. If it is a positive sample, set it to 1. And if it is a negative sample, set it to 0. In the case of only two categories, L_{cls} is Softmax binary cross-entropy (face/not face). L_{box} is a localisation loss for detecting box coordinates and size, and uses a smooth function to adjust the regression after standardising it like in SSD. L_{pts} is the differences proposed by the five key points of the face, and L_{tri} is the triplet loss.

In this experiment, since the final application is still based on recognising multiple faces at once, the focus is to make the output of smaller predict boxes converge successfully. We mainly train the synthesised data, with original VGGFace2 data as an extra training step to retain real-world stability.

In making the selection, we select the furthest positive sample and some negative samples and synthesise them into virtual data. The selection process is similar to the online training method in FaceNet. Unlike single output tasks in FaceNet, in this experiment, each predict box will correspond to a different output. To avoid a serious overfitting problem, a face must contain samples in different scales and locations. Therefore a part of the training sample is pre-generated in advance to speed up the training session, and the rest of them is generated during the training session.

In FaceNet, all training pairs are dynamically selected with static pictures. But in this paper, training images need to be composited in a batch. If the same selecting strategy is used, the data processing time will consume a very large part of the training time. Therefore, we scale the faces in VGGFace2 to 1, 1/2, and 1/4, and regard them as a different sample to speed up the synthesis process.

The loss curve for this experimental training is shown in Figure 4. The blue line is the triplet loss, the orange line is the localisation regression loss, the green line is the categorical loss, and the red line is the landmark regression loss. We can see that there is a significant decrease at the time in which the learning rate decreases. It proves that the learning rate decrements are effective in this training session.

Since feature pyramids generate quite a few outputs, most networks require further filtering of the predict boxes. The network outputs each anchor box's category, box regression coefficient, facial features, and confidence score. It sorts the output by confidence score, removes boxes with confidence below the threshold, and performs an NMS operation to remove boxes with an overlap of 0.35 or more.

The output features will be normalised by L2 normalisation to make the length of the output vector equal to a unit length. After normalisation, there is not much difference between Euler distance and cosine similarity. To make the recognition system easy to use, the output features are compared by cosine similarity, and the output similarity score is scaled into a range of [−1, 1], preventing features with wider ranges.

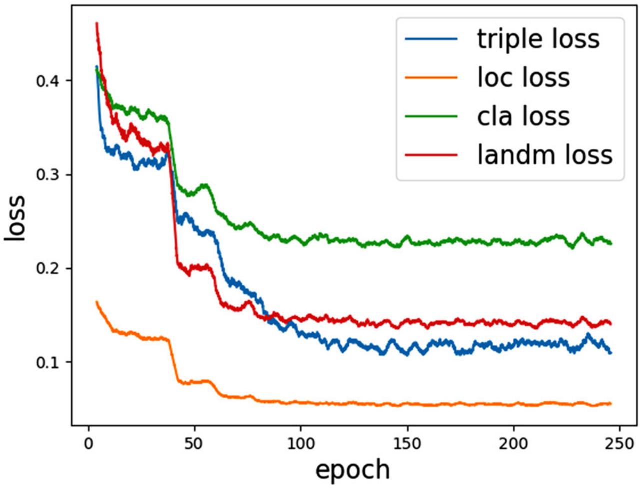

FIGURE 4 Training loss curves

5 | EXPERIMENTAL RESULT

The segmentation network parameters used in this paper are the same as in SegNet, except that the input resolution is adjusted to 128 × 128. The segmentation result achieves 97.2% mean pixel accuracy on the LFW part labels dataset. Furthermore, we do not want any facial features to be ignored during the synthesis process. To improve the recall rate, we performed a Gaussian blur filter with 25 × 25 kernel size on all predicted masks and normalised the pictures again to a maximum value of 1. The network achieves a 99.2% recall rate, which means it works well without losing too many facial regions.

The results of the Mobilenet V2 detection network achieve an accuracy of 98.5% on the FDDB dataset [27] and an F1-score of 80% on the hard set of WiderFace validation. The comparison with other networks is shown in Table 3 and Figure 5. Table 3 and Figure 5 also show the speed comparison of the proposed network. Table 4 shows the parameter and FLOPs with two-stage system architecture and the proposed method. Compared to the two-stage system, the model we proposed is fully eliminating the second stage, and instead of only adding the several output stage, the comparison of the append layers and FaceNet is added in Table 4. The main advantage of the work is that we can inference the identity among multiple faces that exist in a single image by only one forward computing, while the two-stage system needs to crop each face as a different sample.

The different triplet selection setup results are shown in Table 5. It is from pre-generating 30,000 random triples and treating them directly as training sets. It also adjusts the pictures contained in the mini-batches as the online-training method in FaceNet's work. The loss was stable and can converge for both

TABLE 3 The evaluation result of the proposed face recognition system

		Detection task (WiderFace easy)		Recognition (synthesised testing data)
	Model	**Accuracy**	**FPS**	**FPS (10 face)**
All-in-one face [18]	Single-stage	–	0.3	0.3
UniFace [19]	Single-stage	0.91	110	11
SSD+ FaceNet	Two-stage	0.85	41	24
SFD [26] + FaceNet	Two -stage	0.927	35	22
RetinaFace(mobilenetv2-0.25)+ FaceNet	Two -stage	0.901	215	48
RetinaFaceNet (mobilenetv2-0.25)	Single-stage	0.891	212	207

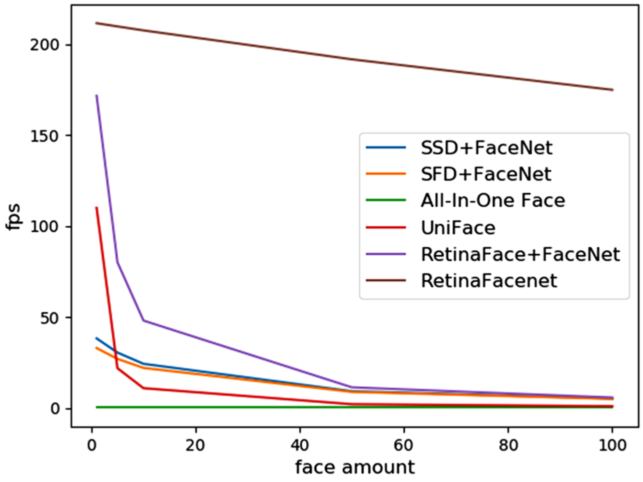

FIGURE 5 Inference speed comparison

TABLE 4 Comparison with two stage network

Layer	Parameter	FLOPs
Retina Net	3.4 M	5.0 G
Proposed network	3.6 M	5.2 G
Difference	14.5 K	1 182 M
FaceNet (MobileNetV2)	3.3 M	133 M

TABLE 5 The result of different triple setup

Setup	Train Set accuracy	LFW accuracy
MobileNetV1, no dynamic triplet	93.2%	70.5 %
MobileNetV2, no dynamic triplet	96.1%	75.5 %
MobileNetV2, dynamic triplet	94.1%	92.4 %

methods. But it was found that if only the hard triplet was used, there would be a serious overfitting problem, generating a larger number of pictures, and selecting a suitable triplet-set is required.

The neural network's architecture and its training methods proposed in this paper is focused on reducing the time consumed by scenes containing a quite number of faces. To better meet the requirements of embedded systems, we reduce the amount of memory used too. Since the feature pyramid output stage barely affects the speed of the entire network, almost all of the recognition task time is saved. As compared to a two-stage system, we only require the final comparison with the pre-generated features of the face sample in the database. Eventually, the system can achieve 212 FPS for 640 × 640 input with the Nvidia RTX 2080Ti acceleration. The speed comparison between this network and the two-stage system with the face-detection-only network and the FaceNet task recognition network which also uses MobileNetV2 as the backbone. Input pictures from one face to one hundred faces, the execution speed for searching through a database containing 100 faces and their features is shown in Figure 5. As shown in Figure 5, all the other methods suffer from the increase in the number of faces. However, our method can still maintain a high inference speed for a large number of faces. This means that if there are more faces in a picture, the single-stage network will be faster compared to a multi-stage system, reducing repetitive computation is beneficial for recognition tasks performed on platforms with fewer computing resources.

Also, such a task can be simply implemented by treating each person's identity as a category. However, the disadvantage is also obvious: it can be time-consuming when updating the database, as the model needs to be fine-tuned to append new classification outputs. Another advantage of a feature extractor network is that a sample can be executed first. Then it saves the predicted features into a database which can be manually assigned an ID to the features of the identity. When the inference model has performed afterward, the face feature and its identity can be read directly. There is no need to repeat the unique feature computation for the samples in the database.

6 | CONCLUSION

In this paper, a single-stage face detection and recognition neural network using feature pyramids and multi-object triplet loss is proposed. We combined two tasks into a single-stage face detection and recognition neural network to facilitate the application system. The training data is synthesised using face slicing technology to adapt the network to realistic scenarios using different training strategies. The processing speed of the system, from camera input to the end of identification, has a dramatic speed improvement compared to the approaches that use two neural networks. When a scene contains a large number of faces, it can have a great FPS (frame-per-second) improvement. With the acceleration of NVidia GeForce RTX 2080 Ti, the 640 × 640 network input achieves 212 FPS and maintains 92.4% accuracy on the LFW data set.

CONFLICT OF INTEREST STATEMENT

All authors have seen and agree with the contents of the manuscript and there is no financial interest to report. The authors certify that the submission is original work and is not under review at any other publication.

DATA AVAILABILITY STATEMENT

Data openly available in a public repository that issues datasets with DOIs.

ORCID

Tsung-Han Tsai https://orcid.org/0000-0001-7524-0621

REFERENCES

1. Sakai, T., Nagao, M., Kanade, T.: Computer analysis and classification of photographs of human faces. In: Proceedings of First USA-JAPAN Computer Conference, pp. 55–62. Springer, Berlin, Heidelberg (1972)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893. IEEE, Piscataway, NJ (2005)
3. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, pp. 1150–1157. IEEE, Piscataway, NJ (1999)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. IEEE, Piscataway, NJ (2014)
5. Girshick, R.B.: Fast R-CNN. In: International Conference on Computer Vision (ICCV), pp. 1440–1448. IEEE, Piscataway, NJ (2015)
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision (ECCV), pp. 21–37. Springer, Cham (2016)
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: IEEE Conference Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE, Piscataway, NJ (2016)
8. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: single-stage dense face localization in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5202–5211. IEEE, Piscataway, NJ (2020)
9. Yang, S., Luo, P., Loy, C.C., Tang, X.: WIDER FACE: a face detection benchmark. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5525–5533. IEEE, Piscataway, NJ (2016)
10. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007. IEEE, Piscataway, NJ (2017)
11. Zhu, Y., Cai, H., Zhang, S., et al.: Tinaface: strong but simple baseline for face detection. arXiv:201113183v3 (2021)
12. Qi, D., Tan, W., Yao, Q., Liu, J.: YOLO5Face: why reinventing a face detector. arXiv:2105.12931 (2021)
13. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708. IEEE, Piscataway, NJ (2014)
14. Huang, G.B., Ramesh, M., Berg, T., Miller, E.L.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Boston, MA (2007)
15. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823. IEEE, Piscataway, NJ (2015)
16. Dadashzadeh, A., Targhi, A.T., Tahmasbi, M., Mirmehdi, M.: HGR-Net: a fusion network for hand gesture segmentation and recognition. arXiv:1806.05653 (2018)
17. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv:1706.05098 (2017)
18. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: An all-in-one convolutional neural network for face analysis. In: 2017 12th IEEE International Conference on IEEE Automatic Face & Gesture Recognition (FG 2017), pp. 17–24. IEEE, Piscataway, NJ (2017)
19. Liao, Z., Zhou, P., Wu, Q., Ni, B.: Uniface: a unified network for face detection and recognition. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3531–3536. IEEE, Piscataway, NJ (2018)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. IEEE, Piscataway, NJ (2015)
21. Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
22. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv:1802.02611 (2018)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. IEEE, Piscataway, NJ (2016)
24. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67–74. IEEE, Piscataway, NJ (2018)
25. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(12), 2481–2495 (2017). doi:https://doi.org/10.1109/TPAMI.2016.2644615
26. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: S^3FD: single shot scale-invariant face detector. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 192–201. IEEE, Piscataway, NJ (2017)
27. Jain, V., Learned-Miller, E.: FDDB: A Benchmark for Face Detection in Unconstrained Settings. University of Massachusetts, Boston, MA (2010)

How to cite this article: Tsai, T.-H., Chi, Po-T.: A single-stage face detection and face recognition deep neural network based on feature pyramid and triplet loss. IET Image Process. 16, 2148–2156 (2022). https://doi.org/10.1049/ipr2.12479