applied sciences

MDPI

Article

Deep Learning Mask Face Recognition with Annealing Mechanism

Wen-Chang Cheng [1], Hung-Chou Hsiao [2,*] and Li-Hua Li [2]

[1] Department of Computer Science & Information Engineering, Chaoyang University of Technology, Taichung City 413310, Taiwan
[2] Department of Information Management, Chaoyang University of Technology, Taichung City 413310, Taiwan
* Correspondence: s10814902@gm.cyut.edu.tw

Abstract: Face recognition (FR) has matured with deep learning, but due to the COVID-19 epidemic, people need to wear masks outside to reduce the risk of infection, making FR a challenge. This study uses the FaceNet approach combined with transfer learning using three different sizes of validated CNN architectures: InceptionResNetV2, InceptionV3, and MobileNetV2. With the addition of the cosine annealing (CA) mechanism, the optimizer can automatically adjust the learning rate (LR) during the model training process to improve the efficiency of the model in finding the best solution in the global domain. The mask face recognition (MFR) method is accomplished without increasing the computational complexity using existing methods. Experimentally, the three models of different sizes using the CA mechanism have a better performance than the fixed LR, step and exponential methods. The accuracy of the three models of different sizes using the CA mechanism can reach a practical level at about 93%.

Keywords: mask face recognition (MFR); face recognition (FR); deep learning; artificial intelligence (AI); convolutional neural network (CNN); FaceNet; cosine annealing

check for updates

Citation: Cheng, W.-C.; Hsiao, H.-C.; Li, L.-H. Deep Learning Mask Face Recognition with Annealing Mechanism. *Appl. Sci.* **2023**, *13*, 732. https://doi.org/10.3390/app13020732

Academic Editor: Hendry Hendry

Received: 28 November 2022
Revised: 26 December 2022
Accepted: 29 December 2022
Published: 4 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

To prevent the spread of the virus effectively, the World Health Organization (WHO) recommends that people wear masks and maintain social distance when going out since COVID-19 began to ravage the world in 2020. However, wearing a mask will cover most of the face. This increases the security risks in places that originally required people to reveal their full face (such as banks, shops, etc.). The face information that can be used for face recognition includes eyes, nose, mouth, and profile, etc. This technology has matured thanks to deep learning. However, the effect of wearing a mask means less of the face can be identified, making the existing face recognition system another challenge. According to a 2020 National Institute of Standards and Technology (NIST) [1] test of numerous face recognition algorithms, wearing a mask can reduce the performance of an otherwise stable face recognition system because less information about the face can be recognized. For example, leading biometric research institutes such as NEC [2] and Thales [3] were also forced to re-examine and adjust their algorithms to improve the accuracy of face recognition systems wearing masks after the COVID-19 pandemic. This highlights the shortcomings of the current mask face recognition (MFR) algorithm.

After the outbreak of COVID-19, many related retrospective studies of MFR by Ahmad Alzu'bi et al. [4] until 2021 show that MFR has become a hot field. Hongxia Deng et al. [5] proposed a MFR algorithm based on a large margin cosine loss called MFCosface. The sample features are mapped into a feature space, allowing a larger distance between classes. An Att-inception module that combines the Inception-Resnet module and the convolutional block attention module is added to enhance the feature weights of unobscured regions. Chenyu Li et al. [6], inspired by amodal perception [7], used a module composed

of two different neural networks for MFR. The first module is de-occlusion using GAN for face restoration. The second module is distillation, which uses a pre-trained face recognition model for transfer learning to learn face features from the pre-trained model. The two modules are combined for MFR. Lingxue Song et al. [8] proposed a mask learning strategy to find and discard damaged (obscured) features for Face Recognition. Firstly, they use Pairwise Differential Siamese Network (PDSN) to train the difference between masked and unmasked face features and build a mask dictionary. The correspondence between the masked face area and the corrupted feature elements in each item of the dictionary is called the Feature Discarding Mask (FDM). The corrupted features are removed for face recognition by combining the relevant dictionary to generate the FDM and the original feature calculation. Fadi Boutros et al. [9] proposed a face recognition model called Embedding Unmasking Model (EUM) based on the convolutional neural network (CNN) model and a novel Self-restrained Triplet Loss to evaluate the results of recognition by EUM. Walid Hariri [10] combined deep learning with traditional image processing to solve the MFR. First, the features of the unobscured areas of the eyes and forehead were obtained using a trained CNN model, and then a bag-of-features paradigm was used in the last layer of the network to quantify the features for classification. HUA-QUAN Chen et al. [11] first obtained the unmasked eyes and forehead areas in the face image and used ESRGAN [12] to perform an image super-resolution. Next, the steps are divided into two parts based on the neural network. One is to use the YCbCr color space in the image for the frequency domain broadening followed by Fast Independent Component Analysis feature reduction to obtain the features. The second is to use image RGB and enhance MBConvBlock in EfficientNet [13], to obtain the spatial fine-grained feature using the semantic grouping layer and group bilinear layer. Finally, the two features are combined and connected to the multi-layer perceptron (MLP) output. Shiming Ge et al. [14] proposed a model called LLE-CNN for masked face detection. The model is divided into three parts. The first part combines two pre-trained models for the high-dimensional feature description of the masked face in the image. The second part uses the locally linear embedding (LLE) algorithm to perform a similarity transformation with the synthesized normal face, in order to repair the obscured or damaged areas of the face. The third part is the validation module, which improves the face area of the retouched candidates and performs classification and regression validation. Maxim Markitantov et al. [15] proposed a multimodal corpus called Biometric Russian Audio-Visual Extended MASKS (BRAVE-MASKS). It is used to analyze the facial features and voice of the face of the person wearing the mask. Weitao Wan et al. [16] proposed a CNN model called MaskNet that automatically assigns higher weights to unmasked face features. Occluded parts are given lower weights, and this is validated against a dataset of real-life faces and a synthetic occlusion face. Haibo Qiu et al. [17] proposed a deep learning model called Face Recognition with Occlusion Masks (FROM) that uses a large number of occluded face images to learn the features of the occluded area and dynamically clean these features. Govind Jeevan et al. [18] used a multitude of existing face recognition models for MFR and performed analytical tests. It is proposed that the model performance can be improved by adding a masked face image during the training process, and that the parameters of face recognition are different from those of mask face recognition and need to be reset.

The above research on MFR is full of novel algorithms incorporating deep learning, and all of them have good performance. However, as the study [19,20] reminds us, the computational cost of deep learning is extremely high, and high economic and environmental costs are required to reduce the error rate. From the research of Mario Lucic et al. [21], it is concluded that the factors that affect the performance the most are hyperparameters and random restarts weight, and the network architecture does not have much influence. Therefore, if the random restarts weight and hyperparameters are set properly, different models of the architecture yield a similar performance. To summarize the above, in order to simplify the complex computation, this study uses the CNN framework validated by the imagenet [22] competition and FaceNet [23] as the training method to complete the MFR.

The reason for using FaceNet is that its main network architecture is not fixed, and can be combined with transfer learning to use different CNN architectures, which is more flexible. The FaceNet training method mainly uses the triplet loss function to allow the network weights to be converged and not to retrain the model in the future if new samples are added for classification. The most important concept of transfer learning is to apply a problem solvable model to different but related problem domains [24], which also saves training time and solves the problem of random restarts weighting. As for the hyperparameter adjustment, this study tries to start from the learning rate (LR) which is not easy to adjust. The neural network model selects different optimizers for gradient descent during the learning procedure, intending to find the best solution in the global domain. Choosing the right optimizer allows the model to converge and find the best weights easily, and whichever optimizer is chosen requires LR to allow the model to update the parameter weights. In general, LR is a fixed value, but the disadvantage is that if it is set too large, the model does not converge easily. If the setting is too small, the model will converge too slowly. Improperly set values tend to make the model find only the local optimal solution, rather than the ideal global optimal solution. In this study, we will use the cosine annealing (CA) mechanism proposed by Ilya Loshchilov et al. [25] that can dynamically adjust the LR of the optimizer. The concept is derived from the simulated annealing (SA) algorithm, an optimization algorithm invented by Kirkpatrick et al. [26], which is derived from the annealing process of metal processing and is often applied in the software domain to find the best approximate solution within a certain time and space range. The CA mechanism has also made achievements in other fields. For example, Pingchuan Ma et al. [27] and Denis Ivanko et al. [28] have applied it to the study of lip reading. Therefore, this study uses FaceNet as the training method, combined with transfer learning using three different sizes of validated CNN architectures: InceptionResNetV2 [29], InceptionV3 [30], and MobileNetV2 [31]. With the addition of CA dynamic adjustment of the LR mechanism for MFR, it is expected that a generalized MFR method can be found without increasing the computational complexity using the existing method.

2. Dataset

The dataset on mask faces. MaskedFace-Net was proposed by Adnane Cabani et al. [32] The images in the dataset were created by Flickr-Faces-HQ (FFHQ) [33] as a composite of faces wearing masks, and faces not wearing masks properly. Gibran Benitez-Garcia et al. [34] proposed the TFM dataset. The dataset collection showed the masked faces in the wild wearing a variety of masks and at different angles. Other masked face in the wild datasets include the MAsked FAces dataset (MAFA) proposed by Shiming Ge et al. [14] and the Real-World Masked Face Dataset (RMFD) proposed by Zhongyuan Wang et al. [35]. However, the above study was mainly used to detect whether people were wearing masks or wearing them correctly. The quality of each dataset varies and requires a lot of time for data cleaning. This study focuses on MFR, so the above dataset was not selected. This study uses VGGFace2_HQ_CROP [36], which is a Dataset modified from VGGFace2 [37]. VGGFace2 is a large collection of 3.31 million images, divided into 9131 subjects, each representing a different face. The average pixel size of each image in the VGGFace2 dataset is 137 $\times$ 180 pixel [38]. Since the size and quality of each image in the VGGFace2 dataset varies, some scholars have high-quality the VGGFace2 dataset and named it VGGFace2_HQ [39]. In VGGFace2_HQ, each image is face alignment, and the size is standardized to 512 $\times$ 512. VGGFace2_HQ_CROP is the VGGFace2_HQ face retrieval version. There are three main reasons for using VGGFace2_HQ_CROP dataset: 1. the number of images is large enough, 2. the image quality is stable, and 3. the images have already completed face acquisition, so there is no need to perform related operations during the experiment. Table 1 shows the relevant data for the three datasets. Although the total number and subjects of VGGFace2_HQ_CROP images were significantly reduced, they were still sufficient to allow this study to proceed.

Table 1. VGGFace2 Dataset corresponding table.

Datasets	Subjects	Numbers	Image Size
VGGFace2	9131	3.311 M	137 × 180 (average)
VGGFace2_HQ	8631	1.160 M	512 × 512
VGGFace2_HQ_CROP	4605	0.624 M	160 × 160

Figure 1 is an example of the corresponding image of the dataset. Figure 1a is the original VGGFace2 image, Figure 1b is VGGFace2_HQ, and Figure 1c is VGGFace2_HQ_CROP. Both VGGFace2_HQ and VGGFace2_HQ_CROP have already performed pre-processing such as face alignment and cropping on the original VGGFace2 to make the image quality of the whole dataset more stable.

(**a**) (**b**) (**c**)

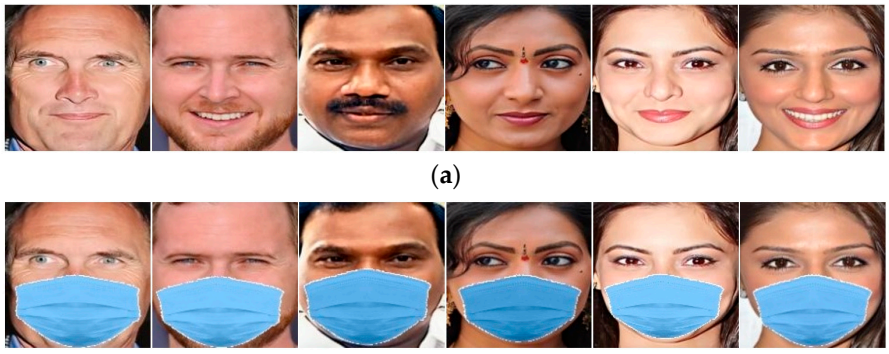
Figure 1. Example of dataset correspondence. (**a**) VGGFace2 [37]; (**b**) VGGFace_2_HQ [39]; (**c**) VGGFace2_HQ_CROP [36].

In this study, the top 600 classifications from VGGFace2_HQ_CROP were selected. One subject represented one person, 500 subjects were the training set, and 100 subjects were the test set. MaskTheFace [40] was used to synthesize all the images of 600 different subjects of masks and was named VGGFace2_HQ_CROP_MASK_600 (MASK600), and the final data obtained is shown in Table 2. In Table 2, the Train set is 500 subjects, and the Test set is 100 subjects, totaling 600 subjects. The number of images is the total number of original and synthetic images. Take the Train set as an example. The number of original images is 46,758, and each original image corresponds to one synthetic image, so the total number of images is 93,516.

Table 2. MASK600 Information.

Sets	Subjects	Numbers
Train set	500	93,516
Test set	100	19,056

Figure 2 is an example of MASK600 correspondence image, Figure 2a above is the original image, and Figure 2b below is the synthesized image.



(**a**)

(**b**)

Figure 2. MASK600 example images (**a**) original images [36]; (**b**) synthesized images (synthesized with [40]).

3. Research Methods

This study combines CNN architecture, FaceNet, and CA as research methods. It is expected to find a generalized MFR method. The research methods are explained below.

3.1. CNN Architecture

FaceNet does not bind the CNN architecture, so this study will use the neural network architecture with good results in the imagenet [22] competition to experiment. Three different model architectures were used for training and testing: InceptionResNetV2 [29], InceptionV3 [30], and MobileNetV2 [31]. Table 3 shows the size of the model and the number of parameter weights after training in this study (calculated using keras in TensorFlow2). These three models can represent three different sizes of models: large, medium, and small, and are used to see how this issue performs for models of different sizes. All three are well-known models. In the statistics of Google Scholar, InceptionResNetV2 paper has 12,698 citations [41], InceptionV3 paper has 23,879 citations [42], and MobileNetV2 paper has 13,034 citations [43], and are still increasing. These three models are recognized by many studies, so we choose them for our study. Since this study only uses the model directly, the model architecture is not modified except for the final output layer. We do not repeat the model details here, but refer to the original paper for more information on the three models [29–31].

Table 3. CNN model information after training.

Models	Parameters	Sizes
InceptionResNetV2	56,106,336	214.0 MB
InceptionV3	24,162,208	92.5 MB
MobileNetV2	6,354,112	24.4 MB

3.2. FaceNet

FaceNet is a neural network-based face recognition training method proposed by the Google team in 2015 (the architecture is shown in Figure 3). The output of traditional neural network models can be divided into two main types: regression values and classification results. FaceNet defines the output layer as a layer of N dimensions (commonly 128 dimensions) of regression numerical vector output. The training method is shown in Figure 3, where the sample A is computed by the CNN model to obtain a set of N-dimensional vector values of X_i^A. The vectors $X_i^{A'}$ and X_i^B are the N-dimensional vector values of the samples A′ and B. where A and A′ are different face images of the same person, and B is the face image of another person. These vectors are exported and calculated by a loss function called Triplet Loss.

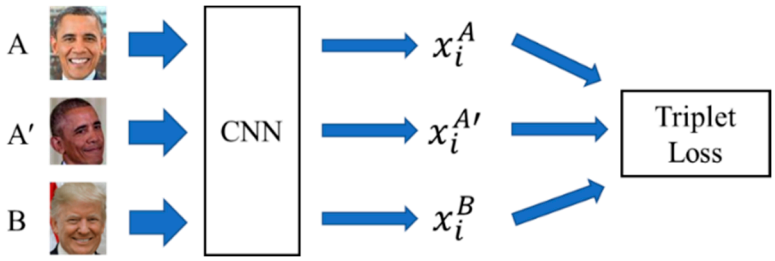


Figure 3. FaceNet architecture [23].

The principle of Triplet Loss is shown in Figure 4. In Figure 4, Anchor (Figure 3 sample A) is the basic sample, Positive is the positive sample which is similar to Anchor (Figure 3 sample A′), and Negative is the negative sample which is different from Anchor (Figure 3 sample B). The Triplet Loss calculation allows Anchor's distance to be as close as possible to Positive and as distant as possible to Negative, in order to learn the best result and find the best vector value in N dimensions. Equation (1) is the calculation of Triplet Loss. X_i^A is the basic sample, $X_i^{A'}$ is the positive sample which is similar to the basic sample, and X_i^B is the negative sample which is different from Anchor. $||, ||_2^2$ means that the vector will be normalized by L2 after the calculation. α is a margin that is enforced between positive and negative pairs, and according to the original paper, α is set to 0.2 [23].

$$L = \sum_i^N max([||X_i^A - X_i^{A'}||_2^2 - ||X_i^A - X_i^B||_2^2 + \alpha],\ 0) \tag{1}$$

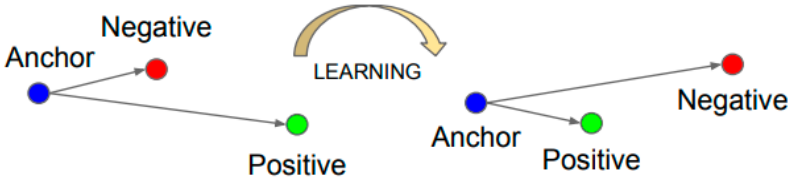


Figure 4. The principle of Triplet Loss [23].

3.3. Cosine Annealing

The LR of the optimizer plays an important role in the model training process. If the setting is too large, the model will not converge easily. If the value is set too small, the model will converge too slowly. Therefore, this study applied the CA mechanism proposed by Ilya Loshchilov et al. [25] that can dynamically adjust the LR, as shown in Equation (2).

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right) \tag{2}$$

where η_t represents the LR value of the t-th epoch, η_{max} and η_{min} are the set interval range values of LR, η_{max} refers to the maximum value and η_{min} is the minimum value. T is to set the total number of epochs to be executed. Figure 5 is an example where η_{max} is assumed to be 1×10^{-4} and η_{min} is 1×10^{-6}, and the LR converges from a maximum value of 1×10^{-4} to a minimum value of 1×10^{-6} during the training process of 100 epochs. According to this mechanism, LR automatically adjusts from large to small with the number of training of the model. The model needs to have a larger LR during initial training to speed up the gradient descent process of the optimizer to find the best solution in the global domain. As the number of model training increases and the LR decreases, the scope of the optimizer gradient descent search for the global best solution becomes smaller, so that the optimizer is less likely to miss the global best solution in the model, and the model is more likely to converge.

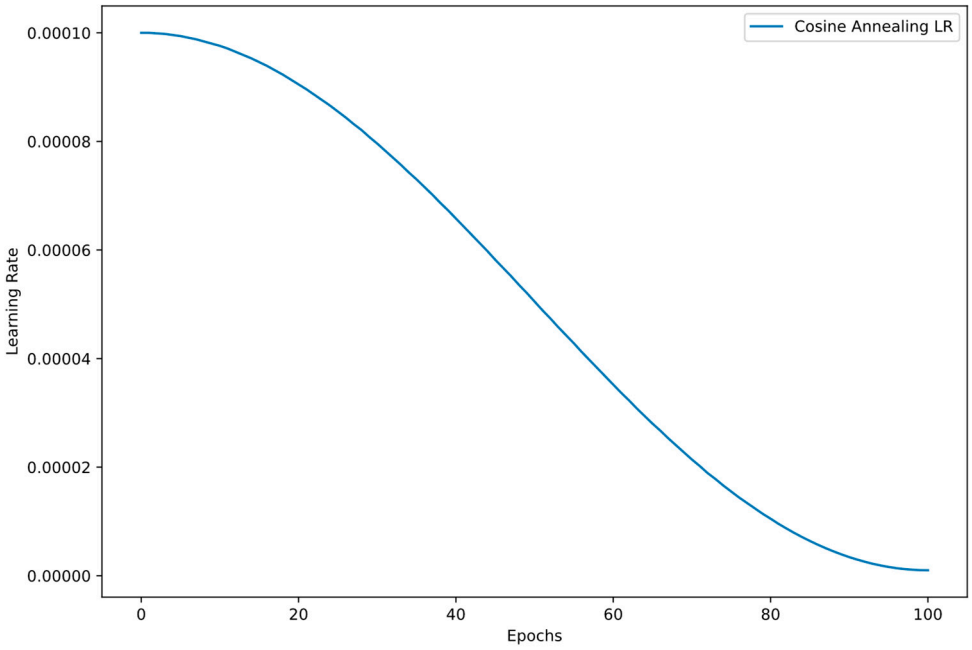


Figure 5. Cosine Annealing Learning Rate Convergence Chart.

4. Experiment Results

The experimental environment of this study is Nvidia 2080Ti with Ubuntu 18.04 and Tensorflow 2.3.4 using the FaceNet training method combined with the CA mechanism. The performance evaluation criteria, training, and testing are explained in detail below.

4.1. Performance Evaluation Criteria

In this study, *Accuracy* and *F1-Score* were used as the performance criteria of the model during the experiment. *Accuracy* is shown in Equation (3), and *F1-Score* is shown in Equation (4).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3}$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$recall = \frac{TP}{TP + FN} \tag{6}$$

TP is the result of the correct classification of the positive sample, and *FN* is the result of incorrect classification of the positive sample. *TN* is the correct result after negative sample comparison, and the last *FP* is the incorrect result after negative sample comparison. *Accuracy* is suitable when the positive and negative samples are relatively average. However, if the positive and negative samples are not balanced, *F1-Score* is a more suitable benchmark for performance evaluation than *Accuracy*. In the *F1-Score* equation, the *precision* represents the predicted performance of the positive sample. The *recall* is the number of correct predictions out of the total number of samples with true results. Therefore, the *F1-Score* is a composite rating of *precision* and *recall*.

4.2. Training

Figure 6 shows the training flow of this study, combining FaceNet and CA training models. First, all the training samples in the MASK600 train set (including the original image and the synthetic image) are randomly disordered, and each image is horizontally

flipped and input into the CNN model together with the original image for training. The last layer obtains 128-dimensional vector values for L2 normalization and calculates Triplet Loss. Then, the CA mechanism is used to dynamically adjust the LR of the optimizer to update the model parameter weights and complete the training of the MFR model.

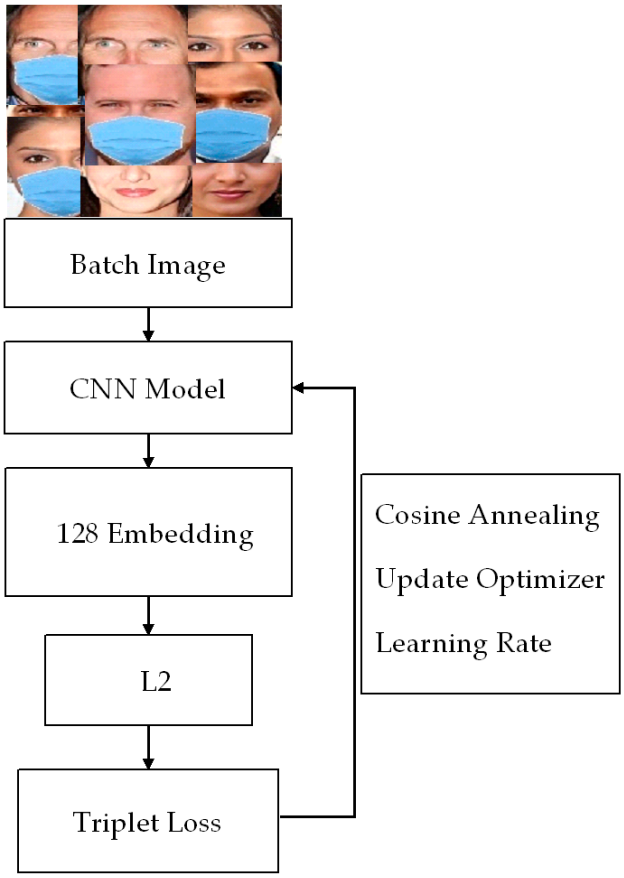


Figure 6. Training Flow Chart.

To verify the effect of dynamically adjusting the LR of the optimizer using the CA mechanism, the experimental was trained using a fixed LR value, step, exponential and the CA mechanism. Table 4 shows the hyperparameters of the training. There are two groups of CAs, CA1 for 1×10^{-3}~1×10^{-5} and CA2 for 1×10^{-4}~1×10^{-6}. The reason for choosing these two groups is that we set the Max Learning Rate of CA1 to 1×10^{-3} as suggested in the paper by Diederik P. Kingma et al. [44]. The Max Learning Rate of CA2 is then set to 1×10^{-4}, using 0.1 as a unit of magnitude. When selecting the Min Learning Rate, in order to allow the number range between the two groups to cross, the Min Learning Rate of CA1 is set to 1×10^{-5}, and the Min Learning Rate of CA2 is set to 1×10^{-6}. Step and exponential are also divided into 1×10^{-3} and 1×10^{-4} for the Initial LR, following the CA approach. The Decay Rate of step is set to 0.8, and every 5-epoch decay 1 time so the Epoch Decay is set to 5. The k value of exponential is the decline smoothing degree and is set to 0.05. The four magnitudes of the CA method, FLR1 = 1×10^{-3}, FLR2 = 1×10^{-4}, FLR3 = 1×10^{-5} and FLR4 = 1×10^{-6}, were tested in a training with fixed LR values. In addition, to make the model learn better, the initialization weights of the training are all initialized using the pre-trained weights of the imagenet, instead of random initialization of the weights.

Table 4. Hyperparameters for model training.

Hyperparameters	Fixed Learning Rate				Step		Exponential		Cosine Annealing	
	FLR1	**FLR2**	**FLR3**	**FLR4**	**ST1**	**ST2**	**EXP1**	**EXP2**	**CA1**	**CA2**
Image Size	$160 \times 160 \times 3$									
Initial weights	imagenet									
Epochs	100									
Batch Size	192									
Optimizer	Adam									
Decay Rate	0.8									
Epoch Decay	5									
k	0.05									
Fixed Learning Rate	1×10^{-3}	1×10^{-4}	1×10^{-5}	1×10^{-6}						
Initial Learning Rate					1×10^{-3}	1×10^{-4}	1×10^{-3}	1×10^{-4}		
Max Learning Rate									1×10^{-3}	1×10^{-4}
Min Learning Rate									1×10^{-5}	1×10^{-6}

Figure 7 shows the comparison of the four types of convergence. The initial LR is 1×10^{-4} as an example. During the 100 training epochs, it can be seen that the decreasing process of LR value of CA mechanism is more stable. The LR convergence process is also more stable with the step's Decay Rate set to 0.8, Epoch Decay set to 5 and exponential's k value set to 0.05. The LR annealing rates of both methods are not too fast or too slow, and the final LR values are similar to the LR values of the CA mechanism.

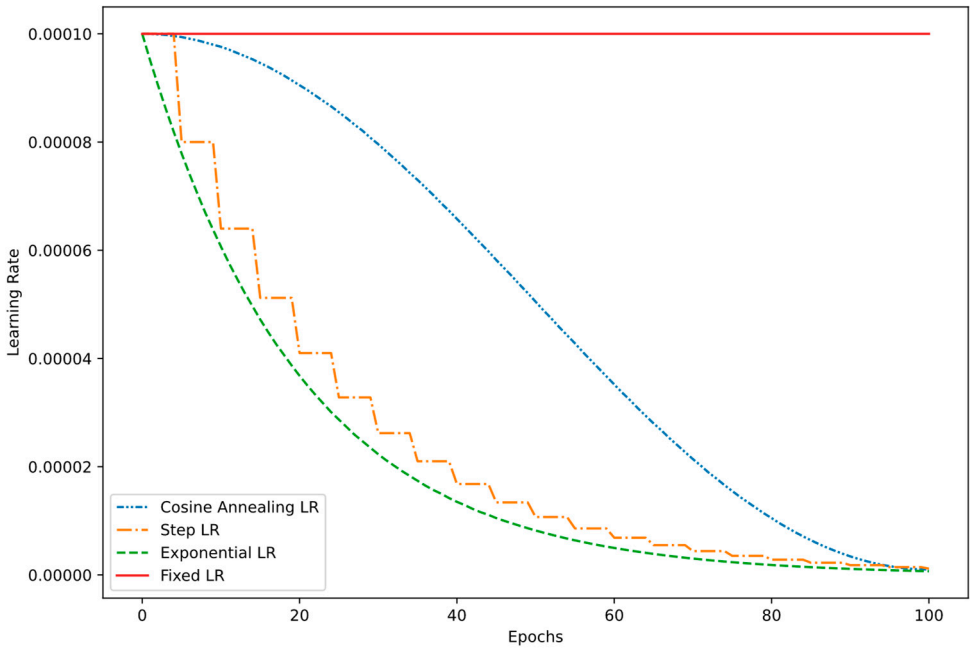


Figure 7. Learning Rate Convergence Comparison Chart.

Figures 8–10 represent the variation of Loss values during the training of Inception-ResNetV2, InceptionV3, and MobileNetV2 models.

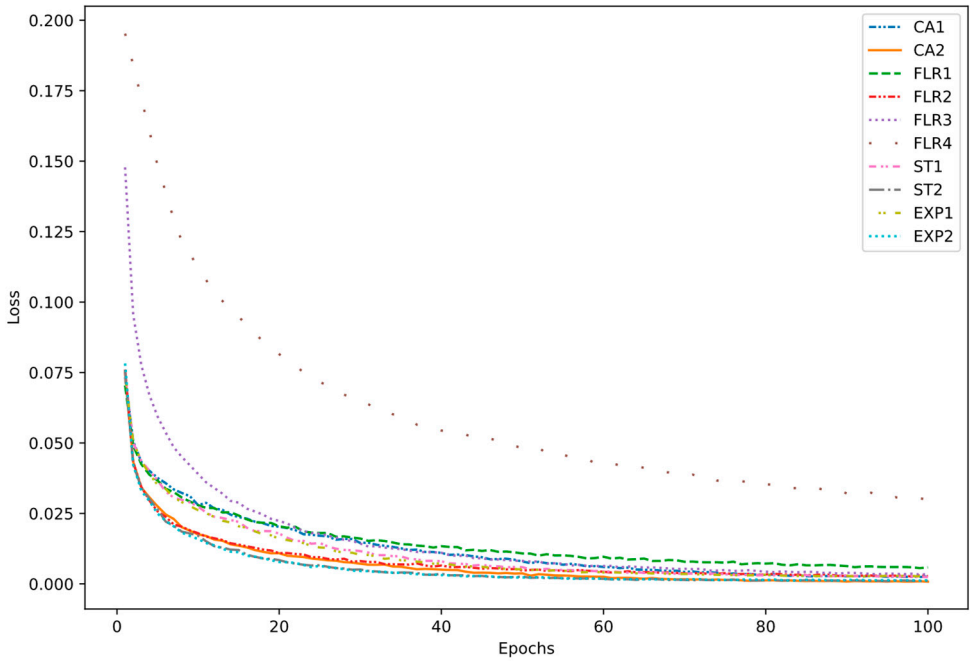


Figure 8. InceptionResNetV2 Training Loss Comparison Chart.

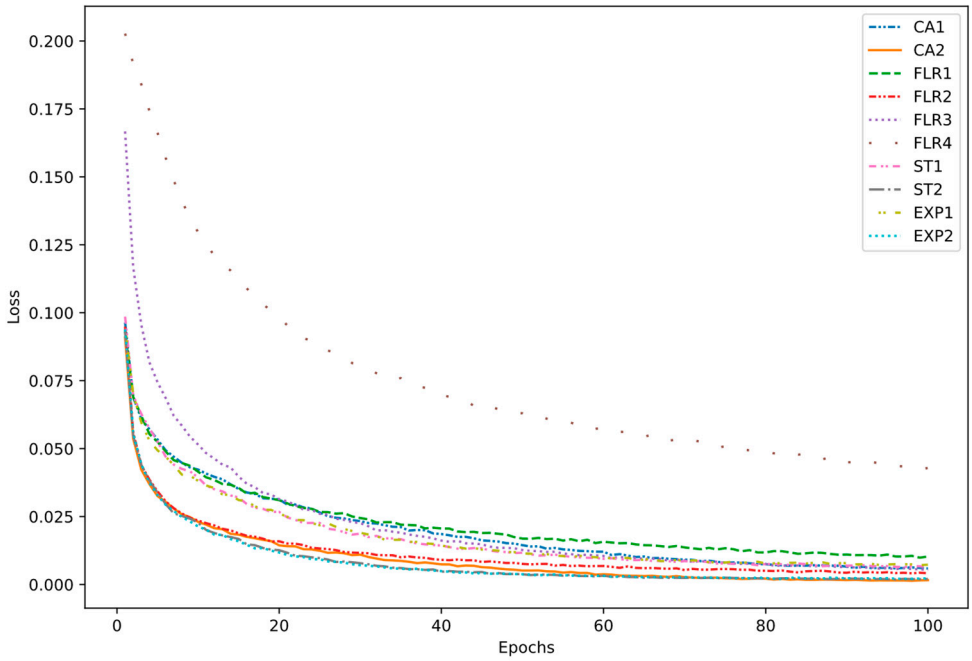


Figure 9. InceptionV3 Training Loss Comparison Chart.

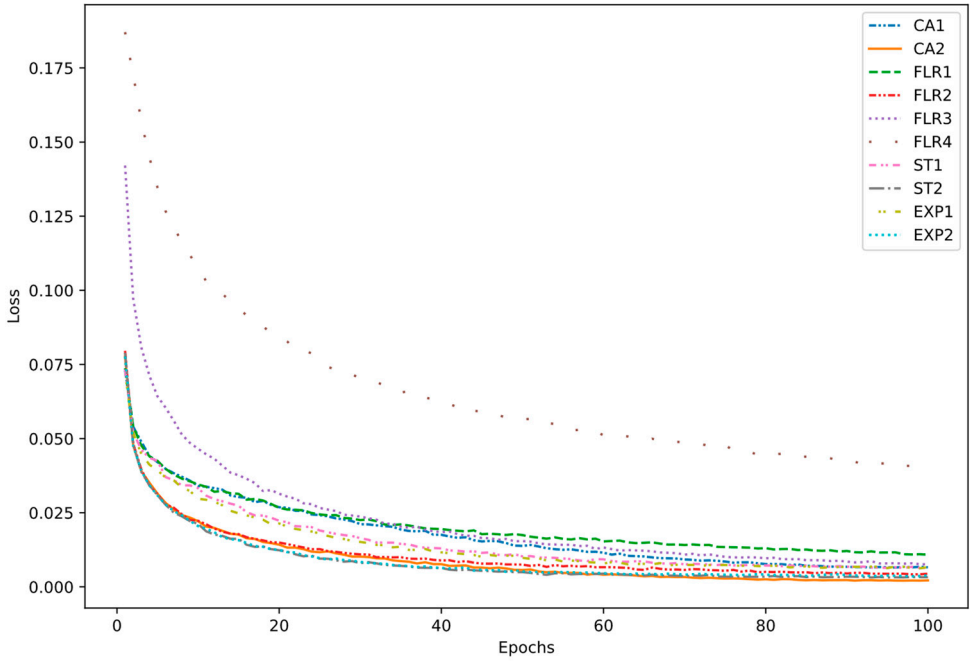


Figure 10. MobileNetV2 Training Loss Comparison Chart.

Among the three models, the best loss values were obtained using CA2 range LR values from 1×10^{-4} to 1×10^{-6}. Step and exponential are the next best in terms of ST2 and EXP2. The third is a fixed LR value of FLR2 = 1×10^{-4}. The worst case is the fixed LR value of FLR4 = 1×10^{-6}. Therefore, we conclude that using the CA mechanism in training is better than step, exponential and fixed LR. CA1, ST1 and EXP1 are not as effective as CA2, ST2 and EXP2. This proves that the convergence from 1×10^{-3} is too slow, and even if we use annealing to allow LR to converge automatically, we need to find an optimal boundary value. The best result was obtained by using 1×10^{-4} in all four methods. In the following test, four sets of hyperparameters with CA2 range of 1×10^{-4}~1×10^{-6}, ST2 and EXP2 with Initial LR of 1×10^{-4} and fixed LR of FLR2 = 1×10^{-4} were used.

In the comparison procedure, the Euclidean distance of 128-dimensional vector values between two images is used as a criterion to determine whether they are the same face, and a threshold(th) is used to define in advance as a benchmark to determine whether they are the same face. In the face recognition method of Dlib [45], the same face is identified when the Euclidean distance is less than equal to th = 0.6, and greater than th = 0.6 is not [46]. However, this study is an MFR, which is different from the general face recognition, so we need to find a suitable th value again. This study refers to the practice of [47]. The appropriate way to find th is to use the model after training, with four sets of parameters: FLR2, CA2, ST2 and EXP2. The 500 subjects in the MASK600 training set are sequentially divided into 50 groups, with one group for every 10 subjects. 10 images from each subject were randomly selected and compared with the group of image subjects for 128-dimensional Euclidean distance comparison. Therefore, there are 10 subjects and 100 randomly selected images (including synthetic images and original images) in each group. After 50 groups of calculations, the average was taken. The comparison th values ranged from 0.3 to 0.9 (interval 0.01). Since each subject is randomly selected, the search procedure for each model is performed five times and the final total average is calculated. In addition, due to the unbalanced sample size, the negative samples will be more than the positive samples in a single comparison. Therefore, Accuracy cannot be used as a basis for finding th. Here, the average F1-Score of each model group is used as the evaluation criterion instead.

For more rigorous experiments, each model is trained twice. Figure 11 shows the average F1-Score and Accuracy of each model after the first training. It can be observed that due to the imbalance of the positive and negative samples, there is no significant difference

in Accuracy for each model. The F1-Score differs significantly, so the best F1-Score is used as the basis for finding the appropriate th value. Table 5 shows the best F1-Score of each model after the first training is the corresponding th value. The th values for each model are different, and in most cases the th values are between 0.5 and 0.7.

Table 5. Best average F1-Score and Threshold for the first training model.

Models	F1-Score	Threshold
InceptionResNetV2+CA2	91.866% ± 0.303	0.60
InceptionResNetV2+FLR2	85.264% ± 0.359	0.57
InceptionResNetV2+ST2	90.191% ± 0.283	0.65
InceptionResNetV2+EXP2	89.417% ± 0.130	0.64
InceptionV3+CA2	89.349% ± 0.196	0.55
InceptionV3+FLR2	81.388% ± 0.371	0.48
InceptionV3+ST2	87.457% ± 0.152	0.58
InceptionV3+EXP2	86.565% ± 0.113	0.57
MobileNetV2+CA2	86.492% ± 0.239	0.66
MobileNetV2+FLR2	78.422% ± 0.394	0.57
MobileNetV2+ST2	83.270% ± 0.328	0.71
MobileNetV2+EXP2	82.058% ± 0.232	0.72

Figure 12 shows the average F1-Score and Accuracy trend of each model after the second training. Table 6. shows the best average F1-Score and its corresponding th value for each model after the second training. It can be observed that the changes are similar to the situation in the first training of the value. Most of the th values are the same or only differ by 0.01. The larger differences were 0.03, 0.02 and 0.05 for InceptionResNetV2+EXP2, InceptionV3+EXP2 and MobileNetV2+FLR2, respectively. Subsequent model testing experiments will be conducted using the most suitable th value found for each model after two training experiments.

Table 6. Best average F1-Score and Threshold for the second training model.

Models	F1-Score	Threshold
InceptionResNetV2+CA2	91.374% ± 0.260	0.6
InceptionResNetV2+FLR2	85.658% ± 0.232	0.56
InceptionResNetV2+ST2	90.537% ± 0.146	0.65
InceptionResNetV2+EXP2	89.817% ± 0.211	0.67
InceptionV3+CA2	88.971% ± 0.288	0.54
InceptionV3+FLR2	82.320% ± 0.406	0.49
InceptionV3+ST2	88.183% ± 0.130	0.58
InceptionV3+EXP2	86.445% ± 0.399	0.59
MobileNetV2+CA2	86.258% ± 0.241	0.65
MobileNetV2+FLR2	79.502% ± 0.555	0.62
MobileNetV2+ST2	83.307% ± 0.364	0.72
MobileNetV2+EXP2	81.997% ± 0.279	0.72

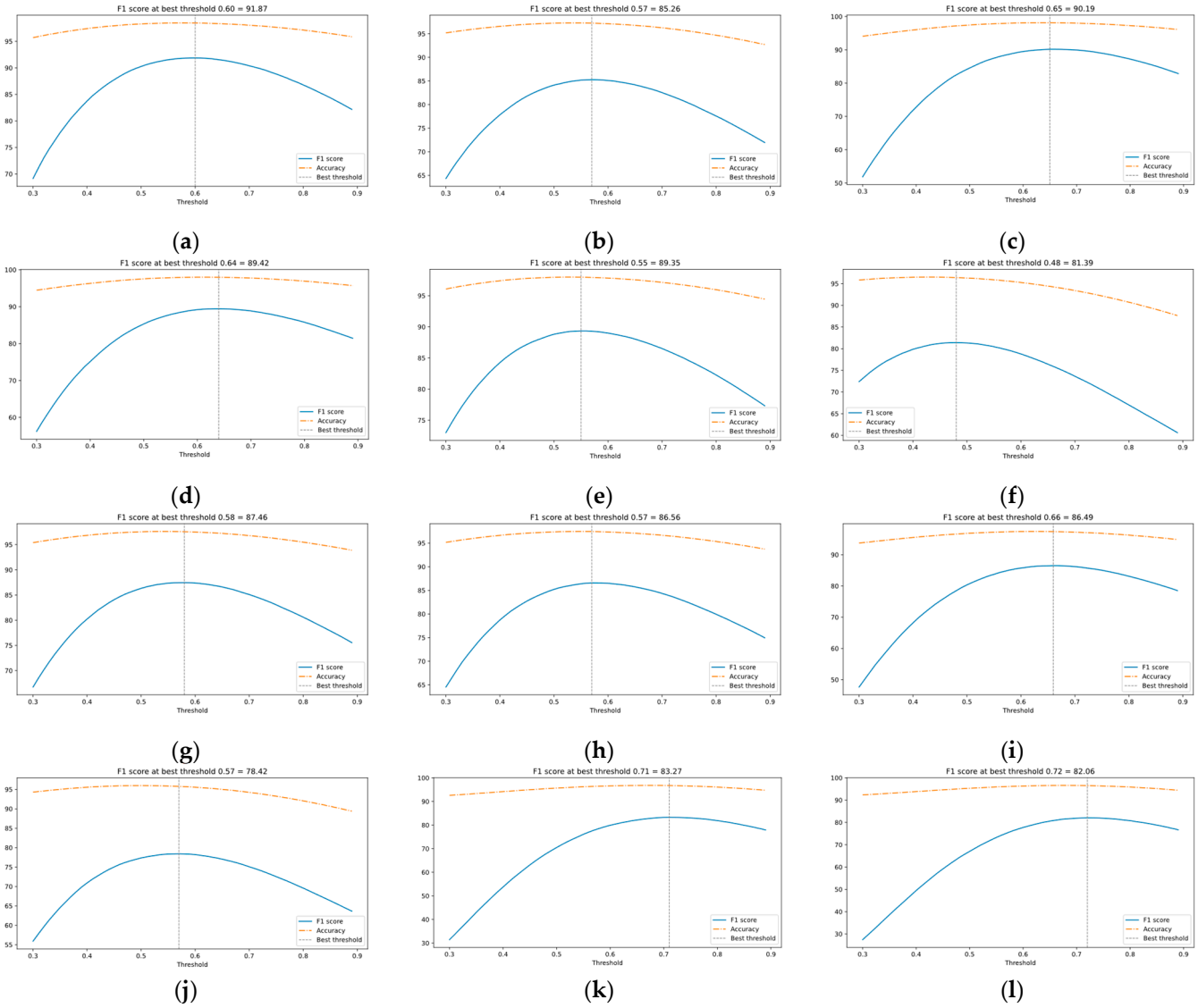


Figure 11. The average Accuracy and F1-Score of the first training model. (**a**) InceptionResNetV2+CA2; (**b**) InceptionResNetV2+FLR2; (**c**) InceptionResNetV2+ST2; (**d**) InceptionResNetV2+EXP2; (**e**) InceptionV3+CA2; (**f**) InceptionV3+FLR2; (**g**) InceptionV3+ST2; (**h**) InceptionV3+ST2; (**i**) MobileNetV2+CA2; (**j**) MobileNetV2+FLR2; (**k**) MobileNetV2+ST2; (**l**) MobileNetV2+EXP2.

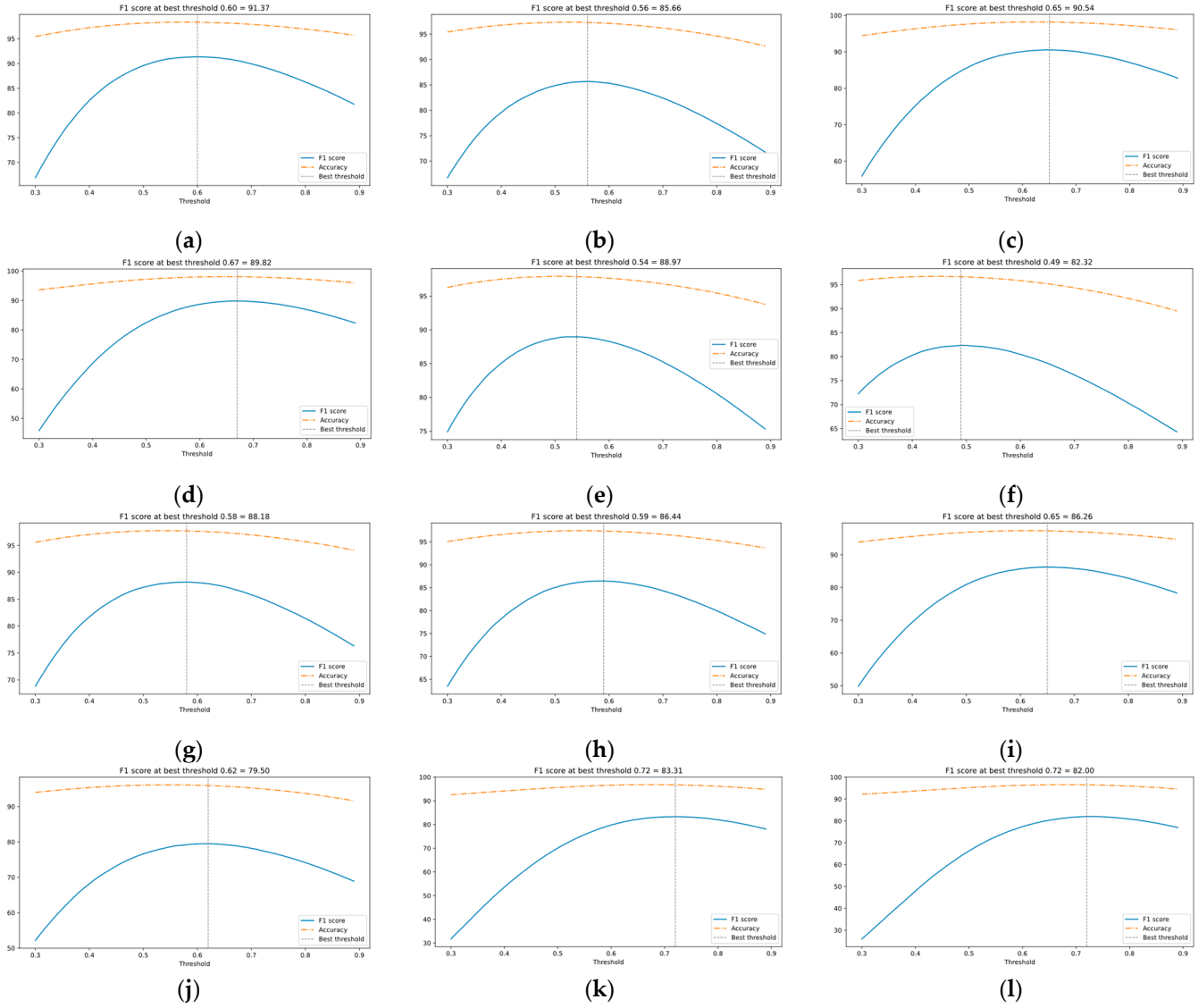


Figure 12. The average Accuracy and F1-Score of the second training model. (**a**) InceptionResNetV2+CA2; (**b**) InceptionResNetV2+FLR2; (**c**) InceptionResNetV2+ST2; (**d**) InceptionResNetV2+EXP2; (**e**) InceptionV3+CA2; (**f**) InceptionV3+FLR2; (**g**) InceptionV3+ST2; (**h**) InceptionV3+ST2; (**i**) MobileNetV2+CA2; (**j**) MobileNetV2+FLR2; (**k**) MobileNetV2+ST2; (**l**) MobileNetV2+EXP2.

4.3. Testing

The models are evaluated using the most appropriate th value found for each model during the training procedure, and Accuracy is used as the performance criterion. The test procedure is to divide the 100 subjects from the MASK600 test set into 10 groups, and each 10 subjects are divided into 1 group. Ten images were selected at random for each subject. Therefore, a total of 100 randomly selected images (including synthetic images and original images) from 10 subjects were compared in 128-dimensions by Euclidean distance comparison. The 10 groups of Accuracy calculations were averaged after completion. Since each subject was randomly selected, the final average Accuracy was calculated for each model after 5 times of testing. Tables 7 and 8. show the average Accuracy of each model after the two tests. We can see that the Accuracy between the same model and parameters is similar after two tests. From Table 7, it can be observed that the best Accuracy of the models of different sizes are using CA2, 93.586% $\pm$ 0.101 for InceptionResNetV2+CA2, 93.025% $\pm$ 0.189 for MobileNetV2+CA2 and InceptionV3+CA2. 92.930% $\pm$ 0.084. In Table 8, the best Accuracy of the models of different sizes is also using the CA2 approach, 93.596% $\pm$ 0.088 for

InceptionResNetV2+CA2, 93.165% ± 0.197 for MobileNetV2+CA2 and 93.037% ± 0.073 for InceptionV3+CA2 respectively. Therefore, we concluded that the overall effect of using the LR value of CA2 in the range of 1×10^{-4}~1×10^{-6} in the appropriate th value is more significant than that of using the fixed LR value, step and exponential methods.

Table 7. The average Accuracy of each model after the first test.

Models	Threshold	Accuracy
InceptionResNetV2+CA2	0.6	93.586% ± 0.101
InceptionResNetV2+FLR2	0.57	92.709% ± 0.166
InceptionResNetV2+ST2	0.65	93.453% ± 0.173
InceptionResNetV2+EXP2	0.64	93.240% ± 0.204
InceptionV3+CA2	0.55	92.930% ± 0.084
InceptionV3+FLR2	0.48	91.552% ± 0.117
InceptionV3+ST2	0.58	92.348% ± 0.046
InceptionV3+EXP2	0.57	92.829% ± 0.179
MobileNetV2+CA2	0.66	93.025% ± 0.189
MobileNetV2+FLR2	0.57	91.603% ± 0.129
MobileNetV2+ST2	0.71	92.491% ± 0.084
MobileNetV2+EXP2	0.72	92.437% ± 0.064

Table 8. The average Accuracy of each model after the second test.

Models	Threshold	Accuracy
InceptionResNetV2+CA2	0.6	93.596% ± 0.088
InceptionResNetV2+FLR2	0.56	93.015% ± 0.268
InceptionResNetV2+ST2	0.65	93.376% ± 0.115
InceptionResNetV2+EXP2	0.67	93.469% ± 0.149
InceptionV3+CA2	0.54	93.037% ± 0.073
InceptionV3+FLR2	0.49	92.104% ± 0.106
InceptionV3+ST2	0.58	92.407% ± 0.206
InceptionV3+EXP2	0.59	92.379% ± 0.162
MobileNetV2+CA2	0.65	93.165% ± 0.197
MobileNetV2+FLR2	0.62	92.113% ± 0.201
MobileNetV2+ST2	0.72	92.147% ± 0.077
MobileNetV2+EXP2	0.72	92.525% ± 0.155

5. Discussion

From the experiments, even if the CA mechanism is used to allow the model to dynamically adjust the optimizer's LR, it needs to be within a suitable range. For example, in this study, CA2 in the range 1×10^{-4}~1×10^{-6} is more suitable than CA1 in the range 1×10^{-3}~1×10^{-5}. The performance of the models trained with CA mechanism is also better than those trained with fixed LR, step and exponential. The Accuracy obtained from the CA mechanism is also more stable than other methods when tested. In addition, the optimal th value found by using the CA mechanism in the two training models is also more stable. The process of finding the best th revealed that the most suitable th values for different models are different values. It means that the use of th should not be limited to a fixed value, and the appropriate th value should be used depending on the situation. The next goal is to consider and find how to extend the current method to improve the overall score and obtain a more practical and generalized MFR model.

6. Conclusions

In this study, we used the FaceNet training method with CA to dynamically adjust the optimizer's LR for better convergence of the model. In the experiments of three different sizes of CNN models, large, medium, and small, it was found that using CA2, a set of LR hyperparameters between 1×10^{-4} and 1×10^{-6}, gave better results than using fixed LR, step and exponential. in all models. In the test experiment, the accuracy of the CA2

method for three different sizes of models, large, medium, and small, was around 93%, which is a practical level. In particular, the small MobileNetV2 model with fewer parameter weights than InceptionResNetV2 and InceptionV3 yield Accuracy that is only slightly behind InceptionResNetV2 and better than InceptionV3. It is proved that if a suitable hyperparameter value (in this paper, LR) can be found, the models can all have a similar performance. The ability to accomplish the same task with a low-complexity model reduces economic and environmental costs.

Author Contributions: Conceptualization, W.-C.C. and H.-C.H.; methodology, H.-C.H.; software, H.-C.H.; validation, H.-C.H.; formal analysis, W.-C.C. and H.-C.H.; investigation, H.-C.H.; resources, H.-C.H.; data curation, H.-C.H.; writing—original draft preparation, H.-C.H.; writing—review and editing, W.-C.C., H.-C.H. and L.-H.L.; visualization, H.-C.H.; supervision, W.-C.C. and L.-H.L.; project administration, W.-C.C. and L.-H.L.; funding acquisition, W.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is funded by the National Science and Technology Council (Restructuring of the former Ministry of Science and Technology), Taiwan. The No. is MOST-111-2637-E-324-001, Taiwan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://www.kaggle.com/datasets/zenbot99/vggface2-hq-cropped (accessed on 20 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ngan, M.L.; Grother, P.J.; Hanaoka, K.K. *Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face Recognition Accuracy with Masks Using Pre-COVID-19 Algorithms*; NIST Interagency/Internal Report (NISTIR); National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020.
2. Face Recognition: Biometric Authentication. Available online: https://www.nec.com/en/global/solutions/biometrics/face (accessed on 25 July 2022).
3. Biometric Technology to Control COVID-19. Available online: https://www.thalesgroup.com/en/spain/magazine/biometrictechnology-control-covid-19 (accessed on 25 July 2022).
4. Alzu'bi, A.; Albalas, F.; Tawfik, A.H.; Lojin, B.Y.; Bashayreh, A. Masked Face Recognition Using Deep Learning: A Review. *Electronics* **2021**, *10*, 2666. [CrossRef]
5. Deng, H.; Feng, Z.; Qian, G.; Lv, X.; Li, H.; Li, G. MFCosface: A Masked-Face Recognition Algorithm Based on Large Margin Cosine Loss. *Appl. Sci.* **2021**, *11*, 7310. [CrossRef]
6. Li, C.; Ge, S.; Zhang, D.; Li, J. Look Through Masks: Towards Masked Face Recognition with De-Occlusion Distillation. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 3016–3024.
7. Michotte, A.; Thinès, G.; Crabbé, G. *Les Compléments Amodaux des Structures Perceptives. Michotte's Experimental Phenomenology of Perception*, 2nd ed.; Thinés, G., Costall, A., Butterworth, G., Eds.; Publications U. Louvain: Louvain-la-Neuve, Belgium, 1964; pp. 140–167.
8. Song, L.; Gong, D.; Li, Z.; Liu, W. Occlusion Robust Face Recognition Based on Mask Learning with Pairwise Differential Siamese Network. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 773–782.
9. Boutros, F.; Damer, N.; Kirchbuchnera, F.; Kuijper, A. Self-restrained Triplet Loss for Accurate Masked Face Recognition. *Pattern Recognit.* **2022**, *124*, 108473. [CrossRef]
10. Hariri, W. Efficient masked face recognition method during the COVID-19 pandemic. *Signal Image Video Process.* **2022**, *16*, 605–612. [CrossRef] [PubMed]
11. Chen, H.Q.; Xie, K.; Li, M.R.; Wen, C.; He, J.B. Face Recognition with Masks Based on Spatial Fine-Grained Frequency Domain Broadening. *IEEE Access* **2022**, *10*, 75536–75548. [CrossRef]
12. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
13. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
14. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting Masked Faces in the Wild with LLE-CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

15. Markitantov, M.; Ryumina, E.; Ryumin, D.; Karpov, A. Biometric Russian Audio-Visual Extended MASKS (BRAVE-MASKS) Corpus: Multimodal Mask Type Recognition Task. In Proceedings of the INTERSPEECH 2022, Incheon, Korea, 18–22 September 2022; pp. 1756–1760.
16. Wan, W.; Chen, J. Occlusion robust face recognition based on mask learning. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3795–3799.
17. Qiu, H.; Gong, D.; Li, Z.; Liu, W.; Tao, D. End2End Occluded Face Recognition by Masking Corrupted Features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 6939–6952. [CrossRef] [PubMed]
18. Jeevan, G.; Zacharias, C.G.; Nair, S.M.; Rajan, J. An empirical study of the impact of masks on face recognition. *Pattern Recognit.* **2022**, *122*, 108308. [CrossRef]
19. Thompson, N.C.; Greenewald, K.; Lee, K.; Manso, G.F. The Computational Limits of Deep Learning. *arXiv* **2020**, arXiv:2007.05558v2.
20. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 28 July–2 August 2019; pp. 3645–3650.
21. Lucic, M.; Kurach, K.; Michalski, M.; Gelly, S.; Bousquet, O. Are GANs Created Equal? A Large-Scale Study. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 3–8 December 2018; pp. 700–709.
22. IMAGENET. Available online: https://www.image-net.org/challenges/LSVRC (accessed on 24 August 2022).
23. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
24. West, J.; Ventura, D.; Warnick, S. *Spring Research Presentation: A Theoretical Foundation for Inductive Transfer*; Brigham Young University, College of Physical and Mathematical Sciences: Provo, UT, USA, 2007.
25. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
26. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
27. Ma, P.; Martinez, B.; Petridis, S.; Pantic, M. Towards Practical Lipreading with Distilled and Efficient Models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7608–7612.
28. Ivanko, D.; Ryumin, D.; Kashevnik, A.; Axyonov, A.; Karnov, A. Visual Speech Recognition in a Driver Assistance System. In Proceedings of the 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; pp. 1131–1135.
29. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI), San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
30. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
32. Cabani, A.; Hammoudi, K.; Benhabiles, H.; Melkemi, M. MaskedFace-Net—A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health* **2021**, *19*, 100144. [CrossRef] [PubMed]
33. Flickr-Faces-HQ Dataset (FFHQ). Available online: https://github.com/NVlabs/ffhq-dataset (accessed on 20 November 2022).
34. Benitez-Garcia, G.; Takahashi, H.; Jimenez-Martinez, M.; Olivares-Mercado, J. TFM a Dataset for Detection and Recognition of Masked Faces in the Wild. In Proceedings of the 4th ACM International Conference on Multimedia in Asia (MMAsia'22), Tokyo, Japan, 13–16 December 2022; pp. 1–7.
35. Wang, Z.; Wang, G.; Huang, B.; Xiong, Z.; Hong, Q.; Wu, H.; Yi, P.; Jiang, K.; Wang, N.; Pei, Y.; et al. Masked Face Recognition Dataset and Application. *arXiv* **2020**, arXiv:2003.09093v2.
36. VGGface2_HQ_cropped. Available online: https://www.kaggle.com/datasets/zenbot99/vggface2-hq-cropped (accessed on 20 June 2022).
37. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. VGGFace2: A dataset for recognizing faces across pose and age. In Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG), Xi'an, China, 15–19 May 2018; pp. 67–74.
38. Massoli, F.V.; Amato, G.; Falchi, F. Cross-Resolution Learning for Face Recognition. *Image Vis. Comput.* **2020**, *99*, 103927. [CrossRef]
39. VGGFace2-HQ. Available online: https://github.com/NNNNAI/VGGFace2-HQ (accessed on 20 November 2022).
40. Anwar, A.; Raychowdhury, A. Masked Face Recognition for Secure Authentication. *arXiv* **2020**, arXiv:2008.11104v1.
41. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. Available online: https://scholar.google.com/scholar_lookup?arxiv_id=1602.07261 (accessed on 20 November 2022).

42. Rethinking the Inception Architecture for Computer Vision. Available online: https://scholar.google.com/scholar_lookup?arxiv_id=1512.00567 (accessed on 20 November 2022).
43. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. Available online: https://scholar.google.com/scholar_lookup?arxiv_id=1801.04381 (accessed on 20 November 2022).
44. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
45. Dlib C++ Library. Available online: http://dlib.net (accessed on 23 August 2022).
46. face_recognition.py—Dlib. Available online: http://dlib.net/face_recognition.py.html (accessed on 23 August 2022).
47. Deep Face Recognition with Keras, Dlib and OpenCV. Available online: https://github.com/krasserm/face-recognition/blob/master/face-recognition.ipynb (accessed on 25 October 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.