*electronics*

MDPI

*Article*

# Attention-Mechanism-Based Face Feature Extraction Model for WeChat Applet on Mobile Devices

**Jianyu Xiao [1], Hongyang Zhou [1], Qigong Lei [1], Huanhua Liu [2,*], Zunlong Xiao [1] and Shenxi Huang [2]**

[1] School of Computer Science and Engineering, Central South University, Changsha 410075, China; xiaojianyu@csu.edu.cn (J.X.); 8208211430@csu.edu.cn (Q.L.)
[2] School of Information Technology and Management, Hunan University of Finance and Economics, Changsha 410205, China
* Correspondence: liuhuanhua@hufe.edu.cn

**Abstract:** Face recognition technology has been widely used with the WeChat applet on mobile devices; however, facial images are captured on mobile devices and then transmitted to a server for feature extraction and recognition in most existing systems. There are significant security risks related to personal information leakage with these transmissions. Therefore, we propose a face recognition framework for the WeChat applet in which face features are extracted in WeChat by the proposed Face Feature Extraction Model based on Attention Mechanism (FFEM-AM), and only the extracted features are transmitted to the server for recognition. In order to balance the prediction accuracy and model complexity, the structure of the proposed FFEM-AM is lightweight, and Efficient Channel Attention (ECA) was introduced to improve the prediction accuracy. The proposed FFEM-AM was evaluated using a self-built database and the WeChat applet on mobile devices. The experiments show that the prediction accuracy of the proposed FFEM-AM was 98.1%, the running time was less than 100 ms, and the memory cost was only 6.5 MB. Therefore, this demonstrates that the proposed FFEM-AM has high prediction accuracy and can also be deployed with the WeChat applet.

**Keywords:** WeChat applet; mobile device; deep learning; MobileNet; efficient channel attention



**Citation:** Xiao, J.; Zhou, H.; Lei, Q.; Liu, H.; Xiao, Z.; Huang, S. Attention-Mechanism-Based Face Feature Extraction Model for WeChat Applet on Mobile Devices. *Electronics* **2024**, *13*, 201. https://doi.org/10.3390/electronics13010201

Academic Editor: Byung-Gyu Kim

Received: 31 October 2023
Revised: 17 December 2023
Accepted: 28 December 2023
Published: 2 January 2024



## 1. Introduction

With the advancements in convolution neural networks [1–4], face recognition technology has been widely used in various fields [5], such as identity verification [6], security and surveillance [7], access control [8], and mobile device unlocking [9]. Recently, mobile devices have begun to play a central role in modern communication, productivity, and entertainment, since they allow users to access information, communicate, and perform various tasks while being mobile. WeChat has become one of the most popular social media platforms, since it can provide instant messaging, voice and video calls, social networking, and the ability to share photos and videos. WeChat also provides developers with a fast and effective development platform, which has become one of the most important for mobile applications [10–12]. It has a wide range of applications for the facial recognition system based on WeChat, which is deployed on mobile devices.

However, there are some limitations with existing facial recognition systems. Figure 1 shows the framework for most existing facial recognition systems, in which the front-end aims to capture and preprocess facial images, and the server is designed to extract features, measure the distance between these features, and perform recognition tasks. As shown in the blue box, the original facial image is transmitted over the Internet/Intranet, which leads to a high risk of facial image leakage. If the original face image is leaked, a large amount of personal information can be easily obtained either directly or by inferring, such as age, gender, location, health condition, and even personality. This may lead to a series of privacy and data security issues, e.g., identity theft, privacy infringement, social engineering attacks,

location tracking, and discrimination. Moreover, facial images' collection, processing, and transmission must adhere to special rules for information protection.

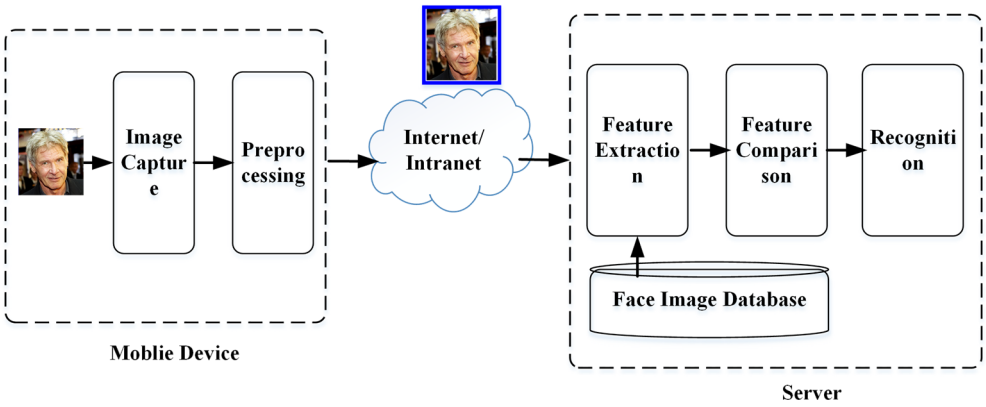



**Figure 1.** Framework of the existing facial recognition systems, where the face image is from [13].

To prevent the leakage of original facial images, we propose a framework in which the WeChat front-end is designed for image capturing, preprocessing, and feature extraction, and only the extracted features are transmitted to the server's end for feature comparison and recognition. Certainly, this information on the extracted features that is transmitted over the Internet/Intranet is far less in size than that of an original facial image, and it is too difficult to obtain personal information even if the feature information is leaked. In WeChat environments, the program's caches are limited to 10 MB, and the computational power and development environments are also constraints. It is challenging to construct a feature extractor that not only has high prediction accuracy but can also be employed with WeChat. Many light-weight structures have been proposed for computer vision tasks, such as SqueezeNet [14,15], Xception [16], ShuffleNet [17,18], and MobileNet [19]. However, it is also challenging to deploy them with WeChat. Therefore, we propose the Face Feature Extraction Model based on Attention Mechanism (FFEM-AM) to extract features of facial images. Inspired by [20], we constructed a smaller model FFEM-AM using fewer layers and incorporating depth-wise separable convolution. To improve the prediction accuracy, the Efficient Channel Attention (ECA) module [21], inverted residuals structure, and linear action function were incorporated.

The main contributions of this work are threefold:

1. To prevent the leakage of original facial images, we propose a framework for a facial recognition system in which facial features are extracted with the WeChat front-end, and only the features, rather than an original facial image, are transmitted to the server for recognition.
2. We propose the light-weight feature extractor FFEM-AM, and depth-wise separable convolution and the ECA module are introduced so that the FFEM-AM can be deployed with WeChat but also with high accuracy.
3. We constructed a large-scale facial image database in a real-world environment and evaluated the proposed FFEM-AM using this self-built database and the WeChat applet for mobile devices; the experiments show that the prediction accuracy was 98.1%, which is better than those of popular light-weight models; the running time was less than 100 ms, and the memory cost was only 6.5 MB.

The paper is organized as follows: the method is proposed in Section 2; the experiments and analysis are provided in Section 3; and Section 4 concludes the paper.

## 2. The Proposed Method

### *2.1. Framework*

As shown in Figure 2, we propose a new framework to prevent the leakage of original facial images during transmission. With WeChat for mobile devices, the facial image is

captured and preprocessed (e.g., face detection and image enhancement), and discriminative features are extracted by the FFEM-AM, which learned on facial image big data. The extracted features, rather than original facial images, are transmitted to a server for feature comparison and recognition. On the server's end, the FFEM-AM extracted features of facial images stored in the database, and then the similarities among the features extrated by WeChat and that extracted by server's end are measured for recognition. In the proposed framework, only the features required for recognition are transmitted over the Internet/Intranet, and this amount of information is far less than that of an original facial image, therefore, reducing the risk of personal information leakage effectively. The biggest challenge with the framework is the feature extractor, which not only has high prediction accuracy but can also be run with WeChat. The memory, computation, and development environment of WeChat are constraints. Therefore, we propose a light-weight feature extractor FFEM-AM, which is discussed in the following sections.

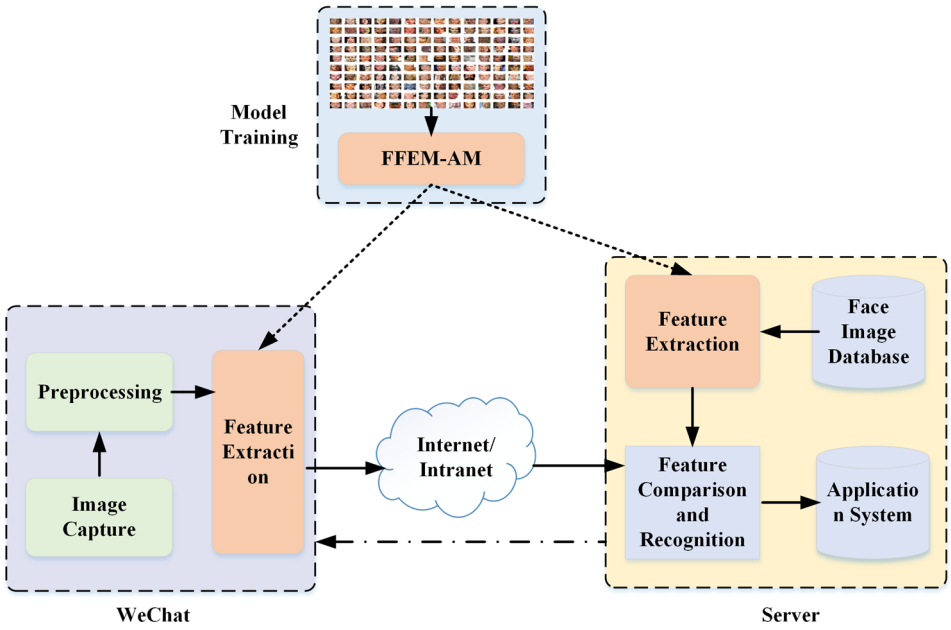



**Figure 2.** Framework of the proposed method, the feature extraction on the WeChat and Server share the identical FFEM-AM model, where the face images are from [13].

### *2.2. Network*

Motivated by MobileNetV2 [22], the proposed model utilizes depth-wise separable convolution [23] to construct the backbone and reduce the computational cost. The model's structure is shown in Table 1. In face recognition with the WeChat applet, most facial images are captured using the front-facing camera, and they are well-aligned and of high quality; therefore, the input image resolution was set to $96 \times 96$, which can encompass most facial features. Then, standard convolutional kernels with a stride of 2 are first used to perform convolutions to obtain $48 \times 48 \times 32$ output features. There are 14 bottleneck layers after the convolutional kernels, which are followed by an average pooling layer to convert the $3 \times 3 \times 1280$ feature map to $1 \times 1 \times 1280$. Finally, the model utilizes a $1 \times 1$ convolutional kernel to output the extracted features.

To construct a light-weight model, the proposed FFEM-AM model only employs 17 layers for feature extraction, which is fewer than that of MobileNetV2. In the bottleneck, a $1 \times 1$ convolutional layer is first used to increase the number of channels of the feature map. Then, a $3 \times 3$ depth-wise separable convolution follows. Finally, a $1 \times 1$ convolutional layer is used to reduce the number of channels, which aims to enhance the feature representation. The dimensions of the feature map first increase and then decrease in the proposed structure; the reason for this is that feature maps with high dimensionality contain most of the information, even if the dimensions are reduced. The ReLU activation [24] can lead to

greater information loss for features with low dimensionality. To preserve more information, we used a linear activation function instead of the ReLU, which ensures that all of the activated values are positive. The powerful feature representation of Convolutional Neural Networks (CNNs) depends heavily on nonlinear activation functions; therefore, it is useful to increase the dimensions before ReLU activation, which can prevent information loss while maintaining the characteristics of nonlinear activation. Therefore, we used an inverted residual structure to construct the bottleneck, which employs a high-dimensional expansion layer for nonlinear transformation, deploying residual connection layers at the first and last bottleneck layers, which further reduces the computational complexity. Moreover, the bottleneck layer can work with different resolutions of input images. If the input resolution is low, we can reduce the number of inverted residual structures to reduce the computational cost. When the input resolution is high, we should increase the number of inverted residual structures to enhance the feature representation ability.

**Table 1.** Network of the proposed Face Feature Extraction Model based on Attention Mechanism (FFEM-AM), where *t* denotes the expansion factor of the inverted residual structure, *c* represents the number of convolutional kernels, *n* represents the repetitions, and *s* represents the stride.

| Input | Operator | t | c | n | s |
|---|---|---|---|---|---|
| 962 × 3 | Conv2d | - | 32 | 1 | 2 |
| 482 × 32 | bottleneck | 1 | 16 | 1 | 1 |
| 482 × 16 | bottleneck | 6 | 24 | 2 | 2 |
| 242 × 24 | bottleneck | 6 | 32 | 2 | 2 |
| 122 × 32 | bottleneck | 6 | 64 | 3 | 2 |
| 62 × 64 | bottleneck | 6 | 96 | 2 | 1 |
| 62 × 96 | bottleneck | 6 | 160 | 3 | 2 |
| 32 × 160 | bottleneck | 6 | 320 | 1 | 1 |
| 32 × 320 | conv2d 1 × 1 | - | 1280 | 1 | 1 |
| 32 × 1280 | pool 3 × 3 | - | - | 1 | - |
| 1 × 1 × 1280 | conv2d 1 × 1 | - | k | - | - |

## *2.3. Bottleneck*

### 2.3.1. Efficient Channel Attention Module

We used a depth-wise separable convolution layer and reduced the number of network layers to compress the model and introduce ECA, as shown in Figure 3, into the bottleneck layer of the proposed FFEM-AM to improve the recognition accuracy. The ECA module utilizes an adaptive one-dimensional convolution layer for interchannel interactions, which can achieve significant improvements with only a minor increase in the parameters.

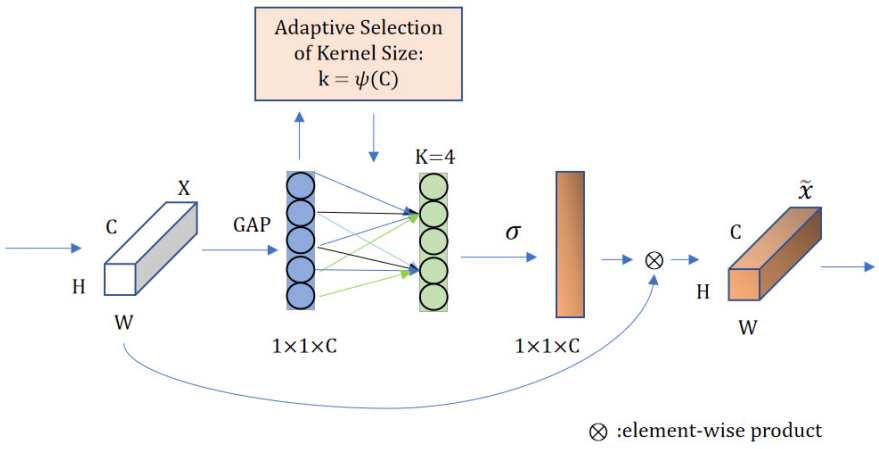



**Figure 3.** Structure of the Efficient Channel Attention (ECA) [21].

Attention mechanisms [25] can improve performance with only a minor increase in the parameters. Firstly, attention mechanisms can improve the accuracy by helping the model pay more attention to crucial information rather than irrelevant information. Secondly, it makes the model more robust to variations in an input image, since it allows the model to concentrate on representative regions, and the impact of variations in other regions is reduced. Thirdly, it enhances the model's interpretability; for example, it can explicitly indicate which regions played a significant role in a decision.

One of the most popular attention mechanisms is the Squeeze-and-Excitation (SE) module [26], which learns the weight coefficients for each channel and weights the features of different channels to achieve better feature extraction. The SE module applies global average pooling for each channel and uses two Fully Connected (FC) layers to estimate the channel weights, where the FC layers operate based on nonlinear interchannel interactions. However, the dimension reduction of features in the process may have negative effects; it is inefficient to capture the dependencies among channels, and the FC layers require many parameters. Therefore, we introduce the ECA module, optimized using SE, into the proposed FFEM-AM. The ECA replaces the FC layers in the SE with $1 \times 1$ convolutions, which avoids the dimension reduction in the SE. Meanwhile, it can also maintain the dimensionality in inter-channel interactions and can adaptively select the size of the 1D convolution kernel. Compared to the SE module, the ECA module not only enhances the model's performance but also significantly reduces the model's complexity.

The ECA module requires determining the coverage range for interactions to capture local cross-channel interactions. However, manually adjusting the optimization coverage range for interactions can result in significant computational overhead. The high-dimensional channels of the grouped convolution's architecture share long-distance convolutions among a fixed number of groups. Consequently, the coverage range of interactions (i.e., size of the convolution kernel), denoted as $k$, is proportional to the channel dimension $C$. There exists the following mapping between $k$ and $C$:

$$C = \phi(k) \tag{1}$$

The simplest mapping is a linear function, but the relationships represented by linear functions are often too limited. On the other hand, the channel dimension "$C$" is often a power of 2. Therefore, the linear function can be extended to a nonlinear function:

$$\phi(k) = 2^{(\gamma \times k - b)} \tag{2}$$

When the channel dimension $C$ is given, the convolution kernel size $k$ can be determined adaptively using the following formula:

$$k = \psi(C) = \left| \frac{\log_2(c)}{\gamma} + \frac{b}{\gamma} \right|_{od} \tag{3}$$

Here, $|t|$ represents the nearest odd integer to $t$, and $\gamma$ and b are set to 2 and 1, respectively. By using a nonlinear mapping, high-dimensional channels exhibit longer distance interactions, while low-dimensional channels exhibit short-distance interactions.

#### 2.3.2. Bottleneck with ECA Modules

We introduced ECA modules at the first and last bottlenecks for the proposed FFEM-AM model. Adding the ECA module into the first bottleneck layer, serving as the network's entrance, aims to enhance the representation capability. By incorporating the ECA module into the last bottleneck layer (i.e., the output layer), the model can weight the output feature vectors to enhance the network's responsiveness to different features. Finally, the channel attention module is embedded into the shallow and deep features to obtain better channel features. Figure 4 shows the bottleneck layers combined with attention modules. Only bottleneck layers with a stride of 1 are modified, since the first and last bottleneck layers

have a stride of 1. After the modification, the dimensions of an input image are expanded using a 1 × 1 convolution layer, and the depth-wise separable convolution layer is used to extract features. Then, an ECA module is applied to enhance the channel features, which are then combined with the original input feature map. Finally, a 1 × 1 convolution layer is used to reduce the feature dimensions and obtain the output. In the bottlenecks, we introduce a feature fusion scheme, since the position information, shallow features, and deep features are crucial, by combining them, which can preserve the comprehensive feature maps during convolutional computations.

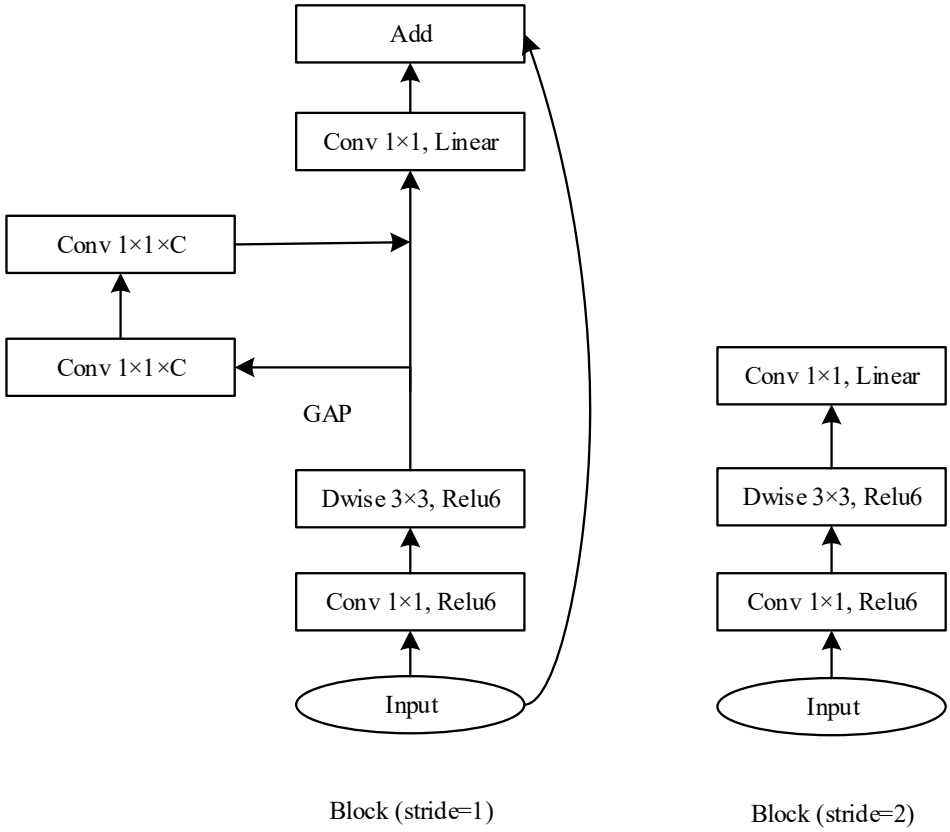



**Figure 4.** Bottleneck structure with ECA modules.

### *2.4. Loss Function and Recognition*

We modeled the facial recognition as a multiclass classification problem to train the proposed FFEM-AM, in which all facial images of the same person belong to one class, and the number of classes equals the number of persons. Let $x = g(I)$ denote the feature extraction of FFEM-AM, and I and x denote the facial image and extracted features, respectively. We used an FC to map the extracted features from the FFEM-AM to the scores for each class, denoted as $z = f(x)$, where $z$ denotes the scores for each class. After the FC layer, we used SoftMax as the classification layer, as follows:

$$P_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e_j}, \tag{4}$$

where $P_i$ represents the probability that the input belongs to class $i$, and $k$ denotes the number of classes. The SoftMax loss was selected as the training loss:

$$Loss = -\sum_{j=1}^{k} y_i \log P_j, \tag{5}$$

where $y_i$ denotes the ground truth.

After training, the FC and SoftMax layers were removed. We use the Euclidean distance to measure the similarity of the features, written as:

$$d(x_i, x_j) = \sqrt{\sum_{t=1}^{M} (x_i(t) - x_j(t))^2}, \tag{6}$$

where $x_i$ and $x_j$ represent the extracted feature from the $i$-th and $j$-th facial image, and $M$ denotes the dimensionality of the feature. The image $I$ stored in the facial image database has the smallest distance between it and test image $I_t$; meanwhile, a distance that is smaller than a given threshold will be used to recognize them as the same person, as per the following:

$$\begin{gathered} I = argmin\, d\big(g(I_t), g(I_j)\big)\ (I_j \in D), \\ s.t\ d\big(g(I_t), g(I_j)\big) \leq \sigma \end{gathered} \tag{7}$$

where $D$ denotes the facial image database, and σ denotes the given threshold that is set as 0.5 (normalized).

## 3. Experimental Results and Analysis

### *3.1. Set-Up*

#### 3.1.1. Model Training Platform and Metrics

The proposed FFEM-AM was trained on a Windows 10 operating system, and Python 3.7 and TensorFlow 2.1.0 were used to build the deep learning network. The hardware platform was equipped with an Intel Core TM i7-11700F CPU with 32 GB memory and a NVIDIA GeForce GTX 1650 GPU with 4 GB. The number of epochs was set to 30, the batch size was 32, the initial learning rate was 0.1, and the learning factor was 0.96 using an SGD optimizer.

Accuracy is one of the most common evaluation metrics in classification, which can be obtained as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

where $TP$, $TN$, $FN$, and $FP$ denote true positives, true negatives, false negatives, and false positives. The running time and memory cost are also crucial in mobile devices due to its limited computation power and memory. Therefore, Accuracy, running time, and memory cost were also taken as metrics for evaluating performance.

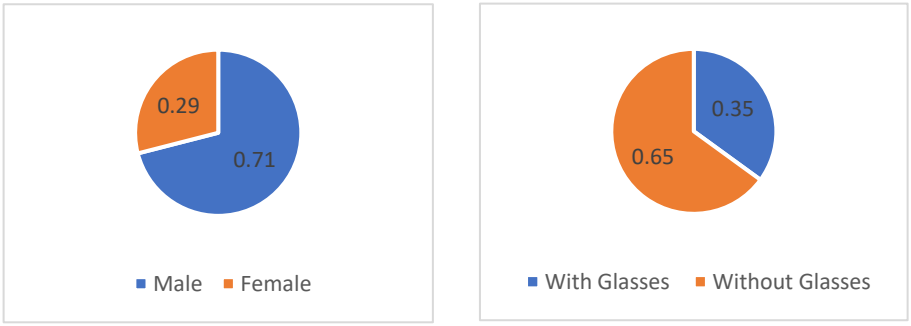
#### 3.1.2. Database

Since the proposed algorithm will be subsequently used in the real-world application systems, we collected facial images taken in real environments and constructed the face database for model training and test. We collect 1440 facial images of 52 persons in various lighting conditions and face postures, aiming to cover as many scenarios of real-world environments as possible. Figure 5a shows the distribution of males and females, and Figure 5b shows the distribution of persons with glasses and without glasses. In the database, the number of male faces and female faces are 1022 and 418, respectively; the male-to-female ratio is close to 7:3. The ratio of persons with glasses to persons without glasses is 13:7. In the experiments, two facial images for each person were left for testing, and the rest of the facial images were used for training.

#### 3.1.3. Model Deployment

JavaScript is the most popular runtime environment for mobile devices, and we used TensorFlow.js to build the deep learning model. TensorFlow.js [27] is a JavaScript version of TensorFlow that supports GPU hardware acceleration and can run in applet environments. It not only aids in deploying well-trained models but can also be used to train models. In this work, we used TensorFlow.js v2.0 containing the tfjs-core, tfjs-converter, tfjs-layers, tfjs-backend-webgl, tfjs-backend-cpu, and tfjs-data subpackages. The localStorage cache was used in the applet to reduce the bandwidth and amount of time related to model.

Figure 6 shows the development flow: Firstly, the model was trained and converted to the *.h5 model format using model.save(), where * denotes the file name. Secondly, the *.h5 model was converted into a format that can be loaded by TensorFlow.js, obtaining model.json. Finally, the related npm packages were introduced in the applet, and the model could be loaded using the tf.loadLayersModel().




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

**Figure 5.** Distribution of facial images in the database: (**a**) males and females; (**b**) persons with glasses and without glasses.

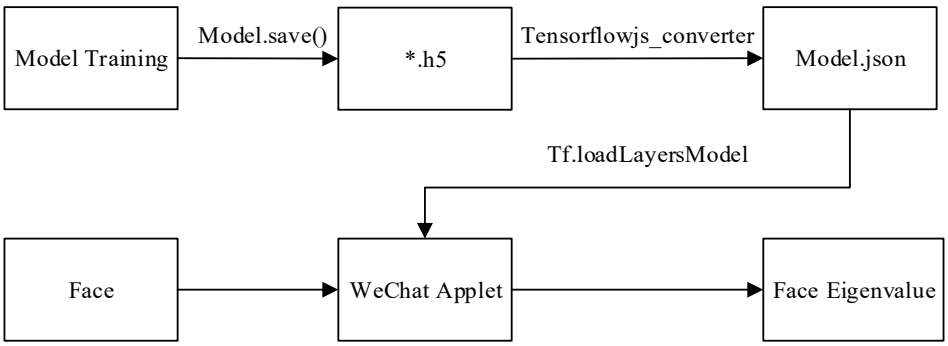
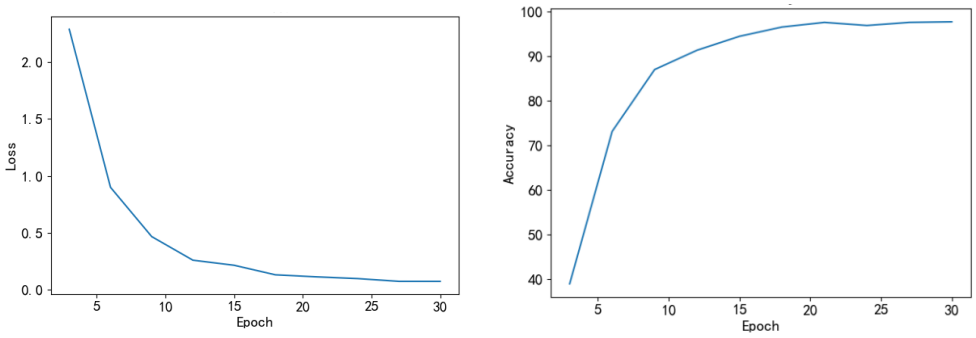



**Figure 6.** The process of deploying the model on the WeChat applet.

### *3.2. Training Loss and Validation Accuracy*

Figure 7a,b show the loss and validation accuracy during training. We can see that with the increase in training epochs, the training loss decreased and validation accuracy increased rapidly. After 25 epochs, the training loss and validation accuracy stabilized, which indicates that the proposed model converged.




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

**Figure 7.** Loss and validation accuracy during training: (**a**) training loss; (**b**) validation accuracy.

### *3.3. Evaluating the Prediction Accuracy*

To evaluate the performance, the FFEM model, which was obtained by removing the attention mechanism, MobileNetV2 model, and ShuffleNetV2 model were taken as the comparison models. The experimental results are shown in Table 2. It can be seen that the accuracy of the FFEM (90.4%) was less than that of MobileNetV2 (96.1%) and ShuffleNetV2 (97.1%), since the structure of the FFEM is lightweight and the number of CNN layers is less than that of MobileNetV2 and ShuffleNetV2. Although the number of layers for both MobileNetV2 and ShuffleNetV2 is more than that of the FFEM-AM, the accuracies of MobileNetV2 and ShuffleNetV2 were less than that of the proposed FFEM-AM (98.1%). The reason for this is that the proposed FFEM-AM introduced the ECA module, which has the ability to obtain information on the interactions among different feature channels. Moreover, the memory costs of MobileNetV2 and ShuffleNetV2 were close to 10 M, which cannot be deployed on the WeChat Applet.

**Table 2.** Prediction accuracy of the comparison models.

| Models | Accuracy (%) |
|---|---|
| FFEM | 90.4 |
| MobileNetV2 [21] | 96.1 |
| ShuffleNetV2 [17] | 97.1 |
| FFEM-AM | 98.1 |

To test the stability of the proposed model, the recognition accuracies for males, females, individuals wearing glasses, and individuals without glasses were validated. The experimental results are shown in Table 3. It can be seen that the recognition accuracies for males, females, individuals wearing glasses, and individuals without glasses were 98.6%, 96.6%, 97.2% and 97.0%, respectively. It can be observed that the recognition accuracies for the four categories were fairly consistent, which indicates that the proposed model exhibits good stability.

**Table 3.** Prediction accuracy for different persons.

| | Accuracy (%) |
|---|---|
| Males | 98.6 |
| Females | 96.6 |
| With Glasses | 97.2 |
| Without Glasses | 97.0 |

### *3.4. Performance on the WeChat Applet*

In this section, we test the memory and computational cost for the proposed model on a mobile platform. We removed the classification layer, converted the feature extraction model into a JavaScript-compatible format, and deployed it on the WeChat applet. The average memory and running time were taken as metrics. We tested the proposed model on the Honor V30 and iPhone12 platforms, and the experimental results are shown in Table 4. We can see that the memory cost of the proposed model was 6.5 M, which is less than 10 M; this demonstrates that the proposed model can be deployed on the WeChat applet. The running times were less than 90 ms for the Honor V30 and the iPhone12, which shows that the proposed model can run effectively on mobile devices.

**Table 4.** Prediction accuracy of the proposed model using different mobile device platforms.

| | Memory (MB) | Time (ms/Sheet) |
|---|---|---|
| Honor V30 | 6.5 | 86 |
| iPhone12 | 6.5 | 31 |

## 4. Conclusions

In this work, we proposed a face recognition framework to protect personal information, in which features of facial images are extracted at the front-end and only the extracted features are transmitted to the server for recognition. We proposed the Face Feature Extraction Model based on Attention Mechanism (FFEM-AM) for face feature extraction on mobile device front-ends. The proposed FFEM-AM uses a light-weight structure to reduce the memory cost and introduced Efficient Channel Attention (ECA) to improve the prediction accuracy. The proposed FFEM-AM was evaluated using the WeChat applet for mobile devices, and the experiments show that the prediction accuracy on a self-built database was 98.1%, the running time was less than 100 ms, and the memory cost was only 6.5 MB. Therefore, the proposed FFEM-AM can be used for face recognition with the WebChat applet.

**Author Contributions:** Study conception and design, J.X., H.Z. and H.L.; data collection, Q.L., Z.X. and S.H.; analysis and interpretation of results, J.X., H.Z. and H.L.; draft manuscript preparation, J.X., H.Z. and H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Natural Science Foundation of Hunan Province (Grant No. 2022JJ30002), Scientific Research Project of Hunan Provincial Education Department (Grant No. 22A0663), Scientific Research Project of Hunan Provincial Education Department (Grant No. 21B0833), Scientific Research Key Project of Hunan Education Department (Grant No. 21A0592), Special Funds for High-Tech Industry Technology Innovation Leading Plan of Hunan (Grant No. 2022GK4009), and National Social Science Foundation (Grant No. 23BTQ056).

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author, upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Simonyan, K.; Zisserman. A very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
2. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
5. Kortli, Y.; Jridi, M.; Al Falou, A.; Atri, M. Face recognition systems: A survey. *Sensors* **2020**, *20*, 342. [CrossRef] [PubMed]
6. Sun, Y.; Chen, Y.; Wang, X.; Tang, X. Deep learning face representation by joint identification-verification. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. Available online: https://proceedings.neurips.cc/paper/2014/file/e5e63da79fcd2bebbd7cb8bf1c1d0274-Paper.pdf (accessed on 30 October 2023).
7. Mahdi, F.P.; Habib, M.; Ahad, A.R.; Mckeever, S.; Moslehuddin, A.; Vasant, P. Face recognition-based real-time system for surveillance. *Intell. Decis. Technol.* **2017**, *11*, 79–92. [CrossRef]
8. Radzi, S.A.; Alif, M.M.F.; Athirah, Y.N.; Jaafar, A.S.; Norihan, A.H.; Saleha, M.S. IoT based facial recognition door access control home security system using raspberry pi. *Int. J. Power Electron. Drive Syst.* **2020**, *11*, 417. [CrossRef]
9. Patel, K.; Han, H.; Jain, A.K. Secure face unlock: Spoof detection on smartphones. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2268–2283. [CrossRef]
10. Wei, M.; Liu, K. Development and design of the wechat app. *J. Web Syst. Appl.* **2022**, *4*, 19–24.
11. Wang, Z. Short video applet based on wechat. Application of intelligent systems in multi-modal information analytics. In Proceedings of the International Conference on Multi-modal Information Analytics (MMIA 2021), Huhehaote, China, 23–24 April 2021; Volume 2, pp. 844–849.
12. Ding, Y.; Lu, X.; Xie, Z.; Jiang, T.; Song, C.; Wang, Z. Evaluation of a novel wechat applet for image-based dietary assessment among pregnant women in China. *Nutrients* **2021**, *13*, 3158. [CrossRef] [PubMed]
13. Kumar, N.; Berg, A.C.; Belhumeur, P.N.; Nayar, S.K. Attribute and simile classifiers for face verification. In Proceedings of the International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 365–372.
14. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. Squeezenet: Alexnet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.

15. Gholami, A.; Kwon, K.; Wu, B.; Tai, Z.; Yue, X.; Jin, P.; Zhao, S.; Keutzer, K. Squeezenext: Hardware-aware neural network design. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1638–1647.
16. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
17. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
18. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 116–131.
19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
20. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
21. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [CrossRef]
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
23. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. Depth wise separable convolution neural network for high speed sar ship detection. *Remote Sens. Environ.* **2019**, *11*, 2483. [CrossRef]
24. Han, Z.; Yu, S.; Lin, S.-B.; Zhou, D.-X. Depth selection for deep ReLU nets in feature extraction and generalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1853–1868. [CrossRef] [PubMed]
25. Yan, C.; Tu, Y.; Wang, X.; Zhang, Y.; Hao, X.; Zhang, Y.; Dai, Q. STAT: Spatial-temporal attention mechanism for video captioning. *IEEE Trans. Multimed.* **2019**, *22*, 229–241. [CrossRef]
26. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
27. Smilkov, D.; Thorat, N.; Assogba, Y.; Nicholson, C.; Kreeger, N.; Yu, P.; Cai, S.; Nielsen, E.; Soegel, D.; Bileschi, S.; et al. Tensorflow. js: Machine learning for the web and beyond. In Proceedings of the Machine Learning and Systems, Stanford, CA, USA, 31 March–2 April 2019; pp. 309–321.