Application Letter

Previous Academic Research Papers

For reference only

For Hong Kong University of Science and Technology

RGC Reference Number: PF20-47580 (2021/2022 Intake)

Applicant Name: Siming Zheng

Master University: University of Putra Malaysia

The qualification is obtained by: Research

Computer Graphics, Vision and Visualization (CGV2) Laboratory

Computer Assisted Surgery & Diagnostic (CASD Medical) Laboratory

Augmented Reality Exergame (ARE) Research Group

Natural Language Processing Research Group

Department of Multimedia, Multimedia Computing

Faulty of Computer Science, FCSIT

University of Putra Malaysia

Table of content

PeerJ Computer Science

3D texture-based face recognition system using fine-tuned deep residual networks

Siming Zheng[1], Rahmita Wirza OK Rahmat[2], Fatimah Khalid[3] and Nurul Amelina Nasharuddin[4]

[1] CASD, Department of Multimedia, Putra Malaysia University, Sedang, Malaysia
[2] C1-103 CASD, Department of Multimedia, Putra Malaysia University, Sedang, Malaysia
[3] C2-32, Department of Multimedia, Putra Malaysia University, Sedang, Malaysia
[4] C2-45, Department of Multimedia, Putra Malaysia University, Sedang, Malaysia

ABSTRACT

As the technology for 3D photography has developed rapidly in recent years, an enormous amount of 3D images has been produced, one of the directions of research for which is face recognition. Improving the accuracy of a number of data is crucial in 3D face recognition problems. Traditional machine learning methods can be used to recognize 3D faces, but the face recognition rate has declined rapidly with the increasing number of 3D images. As a result, classifying large amounts of 3D image data is time-consuming, expensive, and inefficient. The deep learning methods have become the focus of attention in the 3D face recognition research. In our experiment, the end-to-end face recognition system based on 3D face texture is proposed, combining the geometric invariants, histogram of oriented gradients and the fine-tuned residual neural networks. The research shows that when the performance is evaluated by the FRGC-v2 dataset, as the fine-tuned ResNet deep neural network layers are increased, the best Top-1 accuracy is up to 98.26% and the Top-2 accuracy is 99.40%. The framework proposed costs less iterations than traditional methods. The analysis suggests that a large number of 3D face data by the proposed recognition framework could significantly improve recognition decisions in realistic 3D face scenarios.

Submitted 3 July 2019
Accepted 18 October 2019
Published 2 December 2019

Corresponding author
Siming Zheng,
gs53626@student.upm.edu.my

Academic editor
Klara Kedem

Additional Information and Declarations can be found on page 17

DOI **10.7717/peerj-cs.236**

Subjects Computer Vision, Graphics, Multimedia

Keywords 3D textures, Face recognition system, Histogram of oriented gradients features, Deep learning, Residual neural networks, Fine-tuning, Tensorboard

INTRODUCTION

With the rapid development of the Internet, smart computing equipment and social networking applications are increasingly used. There are hundreds of millions of 3D images uploaded every day to platforms such as Snapchat and Alipay, on which a large number of 3D face images are generated. Three main problems in creating 3D face recognition systems that many researchers report are the 3D face pose, illumination changes, and variations in facial expression. Extracting better features are a key process for 3D face recognition (*Bagchi, Bhattacharjee & Nasipuri, 2015*; *Zhang et al., 2016*; *Nagi et al., 2013*; *Wang et al., 2015*; *Zhu et al., 2017*). Furthermore, shallow learning (such as machine learning) including only one or no layer of hidden units leads to lack of ability to deal with large-scale data. These challenges have caused persistent problems for the robustness and reliability of such

systems, which has driven many researchers to use deep learning for 3D face recognition tasks.

When deep learning methods are applied in realistic 3D face scenarios, two challenges confronted are as follows: firstly, the accuracy becomes unstable as 3D face images are added; this is because different deep learning networks have different generalization ability extracting images features. When processing a large number of image data, the deeper the layers of deep learning model are, the more problems such as gradient vanishing and gradient exploration will be caused. Secondly, as more and more complex deep learning models will be applied to the actual scenario, the recognition rate may be affected by the depth of a complex model. In this article we explore both issues. How to recognize a large number of 3D face graphics with high precision is the main task of this article.

In this work, the primary objective of the approaches we propose is to create a 3D textures-based end-to-end face recognition system with a high recognition accuracy, a satisfied performance and robustness while remaining practical. In this system, we have developed a residual neural network model base on ResNet for the 3D face recognition task. This model is fine-tuned with different depths using HOG featured 3D face textures. The primary aim is to solve problems of gradient vanishing and gradient exploration. We trained fine-tuned ResNet models with different depths using HOG based 3D texture images to maintain faster calculations and a high accuracy of image growth.

The remainder of this work is prepared as follows. 'Related Works' reminds the related work. 'Materials & Methods' presents methodology of extraction of HOG features and the fine-tuning ResNet model. 'Experiment' shows the experimentation, results and discussion is described in 'Results and Discussion'. The conclusions are finally stated in 'Conclusions'.

RELATED WORKS

Deep learning algorithms have received increasing attention in the face recognition field, and many researchers discovered the importance of studying 3D face recognition (*Maiti, Sangwan & Raheja, 2014*; *Min et al., 2012*; *Pabiasz, Starczewski & Marvuglia, 2015*; *Porro-Munoz et al., 2014*; *Hu et al., 2017*; *Sun et al., 2015*; *Wu, Hou & Zhang, 2017*; *Tang et al., 2013*; *Zhang, Zhang & Liu, 2019*). On one hand, extracting 3D face information is the key step in 3D face recognition: effective face detection and alignment can increase the overall performance of 3D face recognition, which is critical in both security and commercial 3D face recognition systems. On the other hand, researchers have proposed some methods for exploiting and exerting the deep learning for 3D face recognition, and they have demonstrated that the performance of deep learning systems is significantly better than that of machine learning method in the case of a large amount of 3D images.

In recent years, the convolutional neural network (CNN) models have been used for 3D face recognition. *Hu et al. (2017)* has proposed a method of customizing convolutional neural networks. Her CNN's layer configuration uses the same principle to design based on the LeCun model (*LeCun et al., 1989*). The structure of her model, called CNN-2, comprises one convolutional layer, one pooling layer, and a 5×5 filter. However, this structure cannot effectively extract and analyze 3D face data. When the learning rate rose

from 0.034 to 0.038, the classification accuracy increased from 84.04% to 85.15%; while, the accuracy dropped to 81.31% when the learning rate rose to 0.042. Furthermore, using a 7×7 filter increased the classification accuracy significantly to 84.75% with a learning rate of 0.034.

In a follow-up study, *Sharma & Shaik (2016)* have proposed a new methodology for face recognition with a higher accuracy of approximately 98%. They suggested a customized CNN model, including an input layer, a convolutional layer, a pooling layer, and a fully connected layer. They use the above method to recognize the 3D image with resolution of 96×96. According to results, their face recognition system takes twenty epochs for converging the learning rate, which includes the training rate and the testing rate. Especially, the training losses can be decreased to about 0 before the 6th epoch.

Different methods have been proposed to recognize 3D face images. *Kim et al. (2017)* has developed the VGGNet neural network for dealing with 3D face data. The most representative features of face are extracted from the fine-tuned VGGNet model. The model includes two convolution layers and two fully connected layers with random initial weights, using the last fully connected layer with a softmax layer to accommodate the different sizes of the input images. The fine-tuned VGGNet model achieved an accuracy of 95.15% in the experiment.

Nagi et al. (2013) has developed the face alignment algorithm based on the methods of geometric invariants, local binary pattern (LBP), and k-nearest neighbor (kNN). The face landmarks model (22 key points) is used to detect the human face, and the LBP method is used to crop the 3D face areas. The method of kNN calculates the distance between each input data and the training sample, obtaining the k images closest to the training sample. Finally, proposed statistical methods are used to classify and recognize the images. The results show that the model can reach 91.2% in the recognition rate; however, it declined to 84% as the number of datasets increases.

Soltanpour & Jonathan Wu (2017) uses normal vector to study 3D face recognition. She proposed that more detailed distinct information can be extracted from the 3D facial image by using high-order LNDP method. By estimating the three components of normal vectors in x, y and z channels, three normal component images are extracted. The score-level fusion of three high-order LNDP3x, LNDP3y and LNDP3z are used to improve the recognition performance. Experiments use SIFT-based strategy for matching the face features. The results of this study indicate that fusion LNDP3xyz outperforms descriptors, effectively improving the 3D recognition rate to 98.1%.

The study by *Kamencay et al. (2017)* offers probably the most comprehensive empirical analysis of 3D face recognition. In an attempt to build practical and robust face recognition systems, he proposed three main types of layers for CNN architectures: the convolution layer, the pooling layer, and the fully connected layer. He also proposed three machine learning methods for face recognition, such as Principal Component Analysis (75.2%), Local Binary Patterns Histograms (78.1%), and kNN (71.5%). The proposed customized CNN for 3D face recognition outperforms the above machine learning methods, which reaches the average accuracy of approximately 96.35%. The highest accuracy is 98.3% when 80% of the data was used for training model.

Recent advances in HOG feature extraction methods have facilitated investigation of face recognition. In *Singh & Chhabra*'s (*2018*) article, she suggests that HOG features can be used in the recognition system for improving the efficiency and the processing speed. The function of HOG features can capture the edge features that are invariant to the rotation and light. Owing to the fact that both texture and edge information is important for face representation. HOG features and SVM classifier-based face recognition algorithm is presented in *Santosh Dadi & Mohan Pillutla*'s (*2016*) research. His proposed model extracts the HOG features of the face from the image, these features can be given to any classifier. In the testing stage, the test image is obtained and fed to the SVM classifier, which is a non-probabilistic binary classifier and looks for optimal hyperplane as a decision function, for classification. The results show that this method has better classification accuracy for the test data set, about 92%. In addition, compared to the method using standard eigen feature and PCA algorithm as a baseline, SVM also possesses an improved face recognition rate of 3.74%.

To investigate the effect of utilizing HOG features in the CNN model, (*Ahamed, Alam & Manirul Islam, 2018*) developed CNN learning models that using the HOG features as input data to the training model. His model contains of several layers and each layer is repeatedly used, finally a deep neural network is constructed. In order to evaluate the proposed model, a set of images with 160 images are generated for testing the model performance. However, it leads to a low generalization ability since the data set trained by the model is small. The result shows that the accuracy is approximately 89% by using the constructed model.

In the experiment, we used the latest residual deep neural network (ResNet) and the fine-tuning method (*He et al., 2015*). Our preprocessing method uses HOG features of 3D face texture, different layers of ResNet are created during the experiment and whether decision making in face recognition process can be improved or not is investigated. We evaluated these approaches in the context of the same 3D face-recognition experiment as in (*Kamencay et al., 2017*), a more challenging task than the face identification task used in (*Ahamed, Alam & Manirul Islam, 2018*).

MATERIALS & METHODS

The diversity of face poses raises difficulties for 3D face recognition. By detecting key points on the face, such as the tip of the nose, the corners of the mouth, and the corners of the eyes, the face image in an arbitrary pose can be converted into a frontal face image by affine transformation, after which the face features can be extracted, and an identification is performed. This approach shows that after alignment the features can be extracted with greater success, and the recognition accuracy is thus greatly improved. A schematic diagram of face detection and face alignment is shown in Fig. 1. There are three steps for preprocessing of 3D face recognition: 1. 3D face detection, 2. 3D face alignment. 3. 3D human face feature extraction. The first two phases are implemented by using the open-source tool provided by the Dlib, which can monitor the key points of the face real time to obtain the position and posture of the face. Then we developed a module for extracting the HOG features based on 3D face texture images. Key points of the face

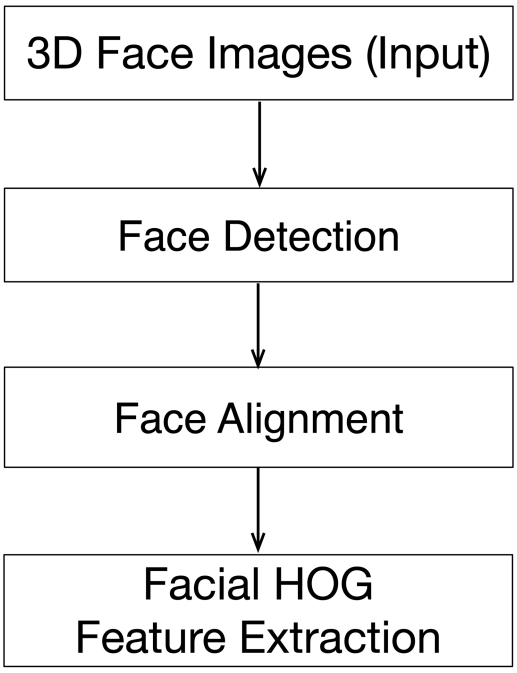

Figure 1 **The pre-processing of 3D face textures.**
Full-size DOI: 10.7717/peerjcs.236/fig-1

are detected using the conditional local neural fields algorithm (*Baltrusaitis, Robinson & Morency, 2013*; *Simonyan & Zisserman, 2015*).

Facial detection and landmarks selection

All 3D images need to be processed before the processing of recognition in order to reduce image noise and redundancy, as shown in Fig. 2. The first step of 3D face recognition is face detection and alignment. We use pre-trained facial landmark detector from the Dlib library, which is used to estimate and predict the location of sixty-eight key points on the human face.

Based on the geometric invariant method, these facial points are marked on the 3D facial images (*Baltrusaitis, Robinson & Morency, 2016*; *Jourabloo & Liu 2015*; *Song et al., 2017*), the subgraphs of A and C is the original 3D images, and the 68 key point distributions are indicated as B and D in Fig. 2 on the right side. These points, including the dominant facial features, such as the tip of the nose, the corners of the mouth, and the corners of the eyes, which are used for further feature extraction and geometric calculations in the recognition stage.

The features of histogram of oriented gradient

In the feature extraction process, we usually try to find the invariant properties and characteristics so that the extraction results do not change significantly due to the specified conditions, this means that the goal of recognition is to find useful discriminative information not the position and size. Regardless of the different changes in the shape and

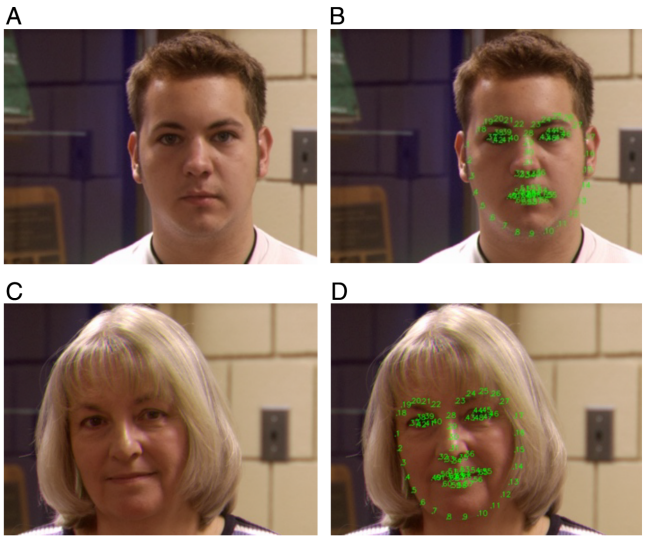

Figure 2 Facial landmarks (68 key points) of Face Recognition Grand Challenge Version 2(FRGC v2.0). The subgraphs of A and C show that the camera takes images of the subjects from different angles. Note that face coordinates are located by using 68 green key points in the subgraphs of B and D.
Full-size DOI: 10.7717/peerjcs.236/fig-2

appearance of the image, we should find reliable and robust discriminative information for improve the recognition rate.

In the field of image processing and computer vision, texture analysis and extraction have a rich history. A method called the Histogram of Oriented Gradient (HOG) has received extensive attention. The core idea of the HOG method is to describe the texture of the detected object by the gradient or distribution of edge directions. Its function is to capture the edge or gradient structure from the image, which is characteristic for the representation of local texture. The benefit of this feature is relatively less affected by the appearance and shape. Essentially, it forms a template and uses learning models to effectively promote recognition.

The HOG descriptor can extract important features from 3D images (*Santosh Dadi & Mohan Pillutla, 2016*; *Kumar, Happy & Routray, 2016*). It captures the local texture information well and has good invariance to geometric and optical changes. Firstly, the target image is divided into small connected regions, which call the cell units. Then, the gradient or edge direction of each pixel in the cell unit are acquired. Finally, the histograms can be combined to form a feature descriptor. In this section, the HOG feature is used as a means of feature extraction in the process of recognition, the purpose is to combine the discriminative 3D face feature in the recognition phase, the specific implementation steps are as follows.

(1) Color and gamma normalization

To reduce the influence of lighting factors, the entire image needs to be normalized in the first step. A compressing process that can effectively reduce shadows, colors and illumination variations of the image, because this information did greatly increase code complexity and demanded the higher performance of processor. At the same time, the

gray image is normalized by Gamma formula. By smoothing part of noises, the influence of local strong light on gradient calculation is reduced. The γ is the symbol of Gamma and its value is 1. The formula of gamma compression Eq. (1) is shown below.

$$I(x,y) = I(x,y)^{\gamma} \tag{1}$$

(2) Gradient computation

The gradient value of each pixel position is calculated in this step. The derivation operation can capture contours, human shadows, and some texture information, which further weakens the influence of illumination. In the operation of computing image gradient, the gradient direction is key to HOG algorithm. The function of H is used for calculating of Histogram of Oriented Gradient. Each pixel point of the transverse gradient $G_x(x,y)$ and the longitudinal gradient of the $G_y(x,y)$ is calculated. It defined as Eqs. (2) and (3).

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \tag{2}$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \tag{3}$$

(3) Creating the orientation histograms

The algorithm needs to finish some operations that calculating the direction gradient of the smallest interval. At the beginning, the 3D image is divided into several intervals with different sizes. Starting from the smallest interval, the gradient direction of all the pixels are contained in each interval, which are weighted by the magnitude. The gradient direction with the largest value represents the gradient direction in the current interval. Finally, the gradient magnitude $G(x,y)$ of each pixel point is calculated according to Eq. (4).

$$G(x,y) = [G_x(x,y)^2 + G_y(x,y)^2]^{1/2} \tag{4}$$

Furthermore, the specific operation in the equation above is that the $G(x,y)$ranges from -90° to 90°. Vectors can be evenly divided into nine intervals because each interval is 20°. This means that nine intervals consist of a total of nine feature vectors in a cell. Four cells can form a block, so each block includes 36 feature vectors. In this way, the feature vectors of all cells in a block are concatenated to form the HOG features.

(4) Computing the directional gradient histogram

In the final step of the merging process, the algorithm uses weighted voting to combine the order of all blocks from smallest to largest. In some cases, the algorithm eliminates some detailed features, which are represented by the small amplitude gradient in the intervals. The rest of the blocks are merged into a maximum gradient pattern, in which contains the important representative features in the 3D images. Gradient direction a(x, y) of the pixel point are calculated according to Eq. (5).

$$\alpha(x,y) = tan^{(-1)}[G_y(x,y)/G_x(x,y)] \tag{5}$$

(5) Creating the orientation histograms

After completing above works, the algorithm generated an orientation histogram for the input 3D face image. In this experiment, we make the pixel of 16×16 constitute a

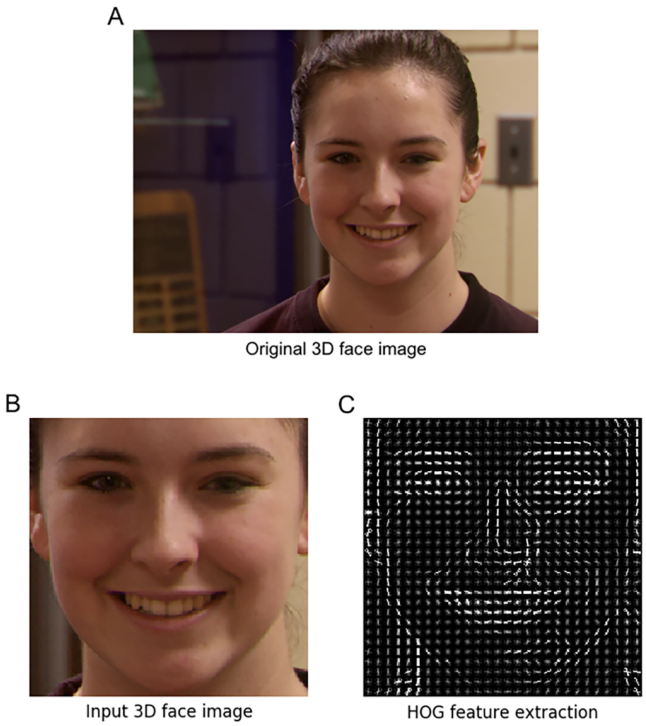

Figure 3 The processing of HOG feature extraction. (A) Original 3D face image (B) input 3D face image; (C) HOG feature extraction.

Full-size DOI: 10.7717/peerjcs.236/fig-3

block in an image with 224×224 image, the stride of scanning window is 8×8, then there are 27 scanning windows in the horizontal and vertical directions in a 3D texture image. Therefore, each 3D texture image $(36 \times 27 \times 27)$ has 26,244 dimensional vectors that can form a complete edge orientation histogram.

After the preprocessing, the face image with extracted HOG features of 3D textures is input into our fine-tuned ResNet classification model. The information contained in the original image is compressed and adjusted, which greatly improves the performance of the subsequent feature extraction network in ResNet neural network. Finally, the whole processing of HOG feature extraction for 3D face image is shown in Figs. 3A, 3B and 3C.

We also demonstrate generation of HOG feature vectors for specific person with different expression, scenarios and various illumination changes. The images based on HOG features extraction are shown from the Figs. 4F to 4J, which are separately correspond to the reprocessing aligned images from Figs. 4A–4E.

The architecture of ResNet neural networks

Convolutional neural networks with multiple layers have several advantages in the research of image classification. The deep network uses a form of end-to-end neural network that automatically integrates the low, medium, and high-level features, and then transmits all of these features to the classifier, extracting different depth features by stacking layers with different depths. Another benefit of the CNN network is that the convolutional layer can

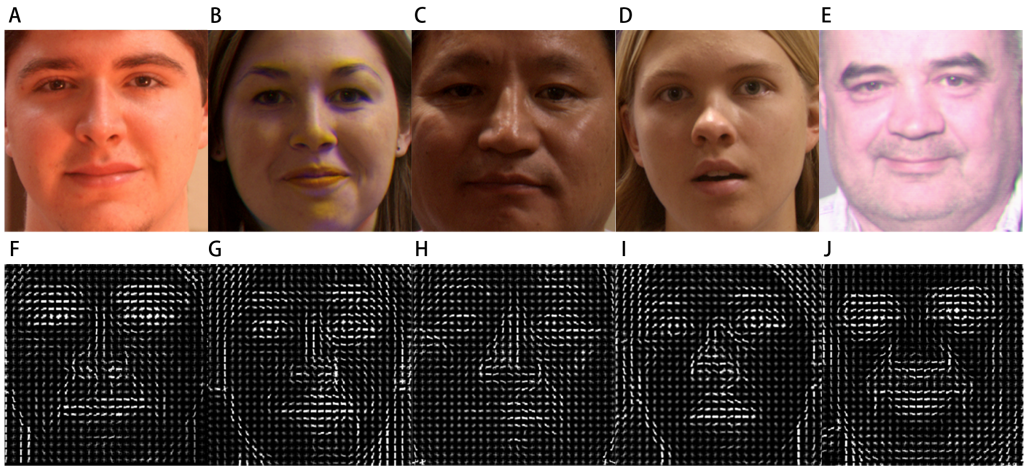

Figure 4 The HOG features of various 3D face images. The subject images (A, B, C, D, and E) were taken under the different conditions of illuminations, expressions, sexes, and ages. Depending on the parameters in the HOG calculation process, the experimental results ensure that the acquired images (F, G, H, I, and J) reveal the discriminative information related to the texture of the original face images.
Full-size DOI: 10.7717/peerjcs.236/fig-4

retain local spatial patterns which may be appropriate for image related tasks (*Zeiler & Fergus, 2014*). Recent research has shown that the depth of the CNN network is critical to model performance (*Szegedy et al., 2014*; *Cheng et al., 2017*). The results of their studies show that the deep convolutional neural network achieves superior performance and significant improvements.

In general, the deeper the neural network is, the worse the recognition performance will be. One major issue in early 3D face recognition research is caused by the use of an error back-propagation algorithm (*Lawrence & Lee Giles, 2000*), which includes the weight coefficient, the derivative of the activation function, and the activation value in the partial derivative. When the number of layers is large, these values are multiplied, easily leading to the vanishing gradient and exploding gradient problems (*Pascanu, Mikolov & Bengio, 2012*; *Hanin, 2018*). Therefore, it is difficult to ensure high-accuracy in the case of growth of 3D face data. In the paper by *He et al. (2015)*, he proposed a theory of deep residual learning, which adopted an approach of shortcut connection to avoid the issues mentioned above. The ResNet architecture for a 152-layer network (a) and a residual block (b) are shown in Fig. 5 above. A residual block with a connections layer can skip a specific layer in the network. The advantages of short connections are that it can reduce the problem of gradient disappearance, thus making the network converge faster and reducing parameters. ResNet-152 also uses the batch normalization operation between each convolution and activation. It allows the researcher to build increasingly deep networks, which have high recognition abilities.

The fine-tuned ResNet neural network model

In deep neural networks, the function of the first layer of training on images is similar to the Gabor filters and color spots operations. First-layer features are not used for specific

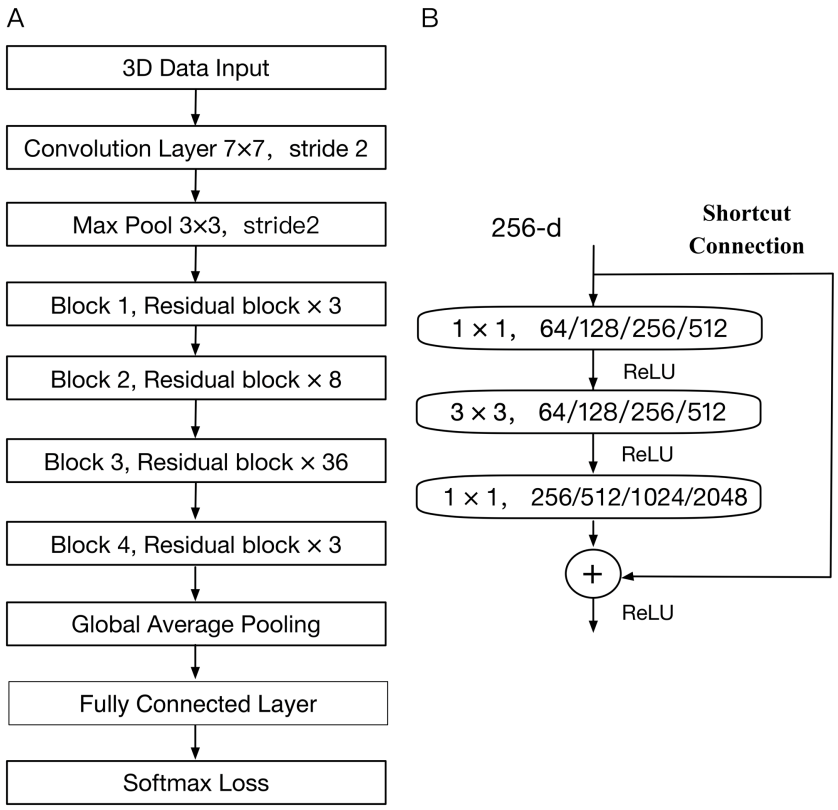

Figure 5 **The architecture of ResNet model.**
Full-size DOI: 10.7717/peerjcs.236/fig-5

data sets or specific tasks but for general ones, because they are applicable to common data sets and tasks. Image features are eventually transmitted from general to specific by the last layer of the network (*Yosinski et al., 2014*). In the big data scenario, we introduce the fine-tuning method in the ResNet neural network, which can greatly shorten the training time, efficiently improve the training loss, and have a stronger generalization ability for getting a good result.

(1) Fine-tuning method

The fine-tuning method can be used to flexibly adjust the architecture of the ResNet model in this 3D face recognition task (*Jung et al., 2015*). In our experiment, four pooling layers with the adaptive average pooling method have been reconstructed. By using the new architecture, it makes the input training data adaptive to the fine-tuned ResNet Model, and the computational complexity of the model is reduced. A softmax layer is created after the fully connected layer to implement the target data classification task of this experiment.

(2) Rectifier Linear Unit

The Rectifier Linear Unit (ReLU) is an important activation function in the ResNet structure, which can overcome the problem of gradient disappearance and speed up the time of training (*Fred & Agarap, 2018*). A ReLU function maps the input value x to 0 if it is negative, and keeps its value unchanged if it is positive, the main ReLU calculation

Structure	Output Size	Layers
Conv layer 1	(Input size: 224 x 224) 112x112	(Input channels: 1) 7x7, 64, stride 2 + BatchNorm
Conv layer 2	56x56	3x3 max pool, stride 2 1x1, 64 3x3, 64 x3 + BatchNorm 1x1, 256
Conv layer 3	28x28	1x1, 128 3x3, 128 x4 + BatchNorm 1x1, 512
Conv layer 4	7x7	1x1, 512 3x3, 512 x3 + BatchNorm 1x1, 2048
	1x1	Adaptive Average Pooling
	1x1	Fully Connected Layers
	1x1	Softmax(466)
Model Output		

Figure 6 **The fine-tuned ResNet neural network.**
Full-size DOI: 10.7717/peerjcs.236/fig-6

expression (Eq. (6)) is shown below.

$$f_{relu}(x) = max(0, x) \tag{6}$$

The convolutional neural network architecture is mainly followed with a combination of all the methods described above. The architecture starts from the input layer of training images and is followed by the convolution layer with the optimum weight and bias for the feature layer. In order to reduce the internal covariate shift (*Ioffe & Szegedy, 2015*) in the deep neural network, the batch normalization algorithm is also added to each convolutional layer to perform the operations of normal normalization and affine transformation on the input of each layer. Finally, our fine-tuned ResNet model were constructed with the proposed method, and the parameters of each convolutional layer are represented in Fig. 6.

In this fine-tuned ResNet model, the layer of adaptive average pooling emphasizes the down-sampling of the overall feature information, its purpose is to reduce the dimension of the feature and retain the effective information, it can integrate features in the feature maps from multiple convolutional layers and pooling layers so that the integrity of the information in this dimension can be more reflected. Through this process, both high-dimensional features and confidence scores can be obtained from each classification. The final full connection layer is used to synthesize the features extracted from the adaptive

average pooling, it can output the probability distribution by using the softmax regression method, which can be divided into more than 1,000 classifications for any tasks, and the value of this parameter was set to 466 in our experiment. In the above structure, Adam function is used as an alternative to the traditional Stochastic Gradient Descent (SGD) optimization algorithm which can iteratively update the weights based on the training data mainly to optimize the neural network and make the training faster.

Datasets

This research received the approval from the University of Notre Dame (henceforth, UND), and the dataset of Face Recognition Grand Challenge version 2 (FRGC-v2) in January 2019. This experiment was performed on FRGC-v2 (*Flynn, 2006*), which is a large number standard face image dataset containing over 50,000 high-resolution 2D and 3D face images, which divided into training and validation partitions in a laboratory setting. Training partition is designed for training algorithms, and validation partitions is used to evaluate the performance of a method in a laboratory environment. All the images were captured by a Minolta Vivid 900/910 series scanner. The datasets used belong to Experiment 3 of FRGC-v2. The experimental 3D face dataset includes 4,007 images of 466 people with different lighting and facial expressions The aim of this dataset is to test the ability to run experiments on large datasets (*Phillips et al., 2005*).

Graphics processing unit

The efficient parallel computing of the graphics processing unit (GPU) makes up for the slow training of deep neural networks (*Chen et al., 2014*). Combined with the CUDA parallel computing platform, it allows the use of larger training datasets and deeper complex neural networks to extract deeper image features (*Huang et al., 2015*; *Singh, Paul & Dr. Arun, 2017*). The model of GPU used in this experiment is NVIDIA GeForce GTX 1080Ti. Its multi-core architecture includes thousands of stream processors, which can perform vector operations in parallel and achieve several times greater throughput in the application. This significantly shortens the calculation time. GPU has therefore been widely used by scientists in deep network learning.

The Fig. 7 shows the processing of our proposed framework. The detailed steps of the proposed framework are as follows: first, the image is detected by the 68 key points of facial landmarks method, and the main multi-channel 3D face regions are extracted, this preprocessing module achieved precision rate of 99.85% and effectively reduces image noise and redundancy from original images, and all images are rescaled to the $224 \times 224 \times 3$. Then, the edges and textures of 3D face images are enhanced by using the HOG method with custom parameters, the HOG face feature images based on 3D textures are obtained, which learned higher discriminative features from 3D face images. Next, fine-tuned deep residual model is proposed by using the HOG textures as the input images. Finally, we generated custom ResNet neural network model. Its image input size is adjusted to the pixel of $224 \times 224 \times 3$, and the structure and quantity of the middle layer in the model are reconstructed, all the operations are performed by using fine-tuning method.

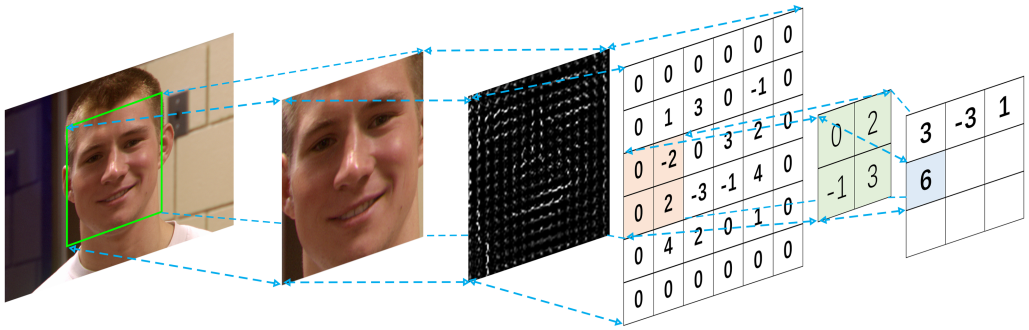

Figure 7 **The fine-tuned ResNet feature extraction operation.**
Full-size DOI: 10.7717/peerjcs.236/fig-7

EXPERIMENT

Key point detection and alignment were carried out on the 3D raw face texture in succession. In the following steps, the dataset applied to the fine-tuned ResNet model comprises 3D face texture with HOG features. Finally, the proposed model conducted for 3D face recognition. In this experiment, an implementation of GPU accelerated training is adopted based on Python and the CUDA architecture, all the HOG-featured images were resized of $224 \times 224 \times 3$ pixel in the dataset, the test model of fine-tuned ResNet with different depth layers (e.g., 50 layers, 101 layers, 152 layers) were then evaluated.

(1) Firstly, a convolution layer in fine-tuned ResNet architecture multiplies the 2×2 filter with a highlighted area (also 2×2) of the input feature map, and all the values are summed up to generate one value in the output feature map, as shown in Fig. 7.

(2) After the 3D data are processed through the first convolution layer, the next layer is max-pooling. The filter window of the max-pooling is moved across the input feature with a step size defined by the stride (the value of stride is 2 in the case of ResNet-152).

The advantage is that it can reduce errors and preserve more texture information. In the max-pooling, the maximal value is selected from four values in the filter window. The size of the detection region is $f \times f$, with a stride of s, so the output features h' and w' are given through the Eq. (7) below.

$$h' = [\frac{h - f + s}{s}], w' = [\frac{w - f + s}{s}]. \tag{7}$$

(3) The residual block consists of two convolution layers each a 1×1 filter and one convolution with a 3×3 filter. The 1×1 layer mainly reduces and restores dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions. Two 1×1 convolutions effectively reduce the number of convolution parameters and the amount of calculation. The residual block is used for ResNet-50/101/152.

(4) The fine-tuned ResNet model uses a global average pool and then categorizes 3D face images at the end of the network through fully connected layers. The global average pooling layer provides faster calculations with more accurate classification and fewer parameters. It serves to sum up all the values from the filter window and then average them, which can reduce errors and retain 3D background information of the image.

(5) Finally, the fully connected layer reassembles the previous local 3D features into a complete graph through the weight matrix. The classification y is defined as follows:

$$y = f(W^T x + b) \tag{8}$$

(6) However, having a good neural network model in specified dataset does not necessarily imply that the model is perfect or that it will be reproduced when tested on external data. In order to make sure it is robust, reproducible and unbiased for testing future new datasets under non-ideal conditions, accuracy metrics is adopted to evaluate the fine-tuned ResNet model's performance.

The indicator accuracy is a measurement of the correct proportion of image classifications. This study is accurate in its ability to differentiate the 3D face recognition cases correctly. To estimate the accuracy of tests, the proportion of true positives (TP) and true negatives (TN) in all evaluated cases are calculated in this experiment. Mathematically, this can be stated as following.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

The sub formula of TP+TN+FP+FN is the total number of observations. Moreover, the respective tests of Top 1 and Top 2 accuracy are used to evaluate the performance of our proposed model. The following hypothesis will be tested: the increasing network depth improves the accuracy of 3D face recognition. This study contributes to this developing area of research by exploring how different depth of our fine-tuned ResNet networks affect the outcome of 3D recognition.

RESULTS AND DISCUSSION

Tensor Board is a data visualization tool that can be used to visualize computational graph structure, provide statistical analysis, and plot the values captured as summaries during the execution of computational graphs. In this research, different types of pre-trained ResNet neural network with the same structure but different depths are proposed. As shown in Fig. 8, the three-subgraph shown below are the Tensor Board graph of linear regression corresponding to the 50 (A), 101 (B), and 152 (C) layers of structure of the ResNet neural networks. The 3D face recognition rates of the three different structural models were recorded objectively in real time use of Tensor Board.

There were 21 times of training and testing in all test cases. According to the results of the ResNet50 testing model, the maximal accuracy was 97.02% for the validation set in the 21th epoch, which corresponds to the similar accuracy of 97.22% in the 6th epoch of the ResNet101 model. This accuracy rate is close to 97.10% in the same epoch in the ResNet152 model. The highest accuracies were 98.05% for ResNet 101 and 98.26% for ResNet 152.

The most accurate indicator of Top 2 can be used to further evaluate the performance of the trained ResNet model. In the ResNet 152 model, the recognition rate fluctuates at first and then becomes regular with the increase of test sets. The accuracy rate was maintained at an average of 99.30% for ResNet 152 after the 12th epoch. The results show that the

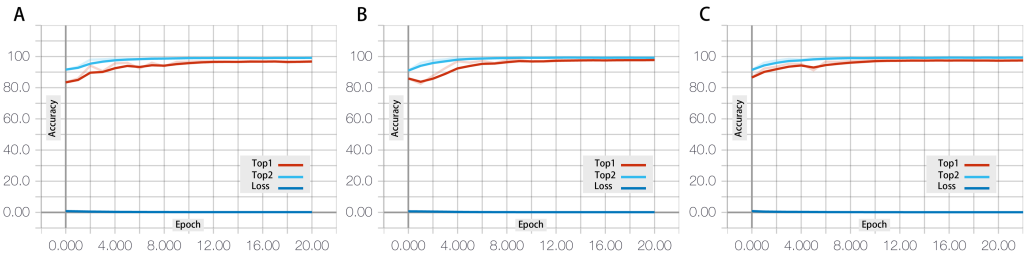

Figure 8 The accuracy rate of different layer numbers in the fine-tuning ResNet architecture. The performances are shown for the fine-tuned ResNet-50 layer (A), the fine-tuned ResNet-101 model (B), and the fine-tuned ResNet-152 model (C).

Full-size DOI: 10.7717/peerjcs.236/fig-8

ResNet model has strong generalization ability. The recognition rate reaches its peak of 99.40% in ResNet-152 in the 12th epoch of the 3D face recognition experiment.

This experiment explores the benefits and effect of different numbers of neural network layer through the fine-tuning method on 3D face texture recognition research with high accuracy. The Fig. 8 has shown that the increasing of layers of fine-tuned ResNet neural network model, the proposed framework can improve the accurate through the HOG method based on 3D face textures.

To eliminate the effects of interference factors, the 3D face dataset was processed beforehand (3D face detection, alignment, and HOG feature extraction). Studies have shown the importance of the fine-tuned convolutional neural network model (ResNet) with depth layers that have more highly discriminative features. The model is advantageous in that although the depth is significantly increased, the ResNet model is less complex with a higher accuracy rate. The Fig. 8 presents the inter-correlations among the three recognition rates of the ResNet model with different layers. The 3D face recognition rate is positively correlated with the number of layers in the ResNet model, which is also a principal factor determining the computing time.

With qualitative modes of enquiry employed, the experiments show that the proposed method achieves promising results, demonstrating that the ResNet-152 neural network model described in this paper can have a recognition accuracy of 98.26% (Top 1); compared with the most accurate, the accuracy of the second accurate test was improved by 1.14% (at 99.40%) with the FRGC-v2 datasets. Practical results proved the validity of the proposed method in 3D face recognition.

The classification performance of methods applied to the FRGC-v2 dataset seem superior to the seemingly impressive results of published studies utilizing different methods in Table 1 (*Hu et al., 2017*; *Sharma & Shaik, 2016*; *Soltanpour & Jonathan Wu, 2017*).

In previous researches, custom CNN is a commonly used deep learning algorithm used for 3D image recognition tasks. Firstly, in custom CNN network training, it is necessary to constantly adjust network parameters. The customized parameters such as weights and biases in CNN network results in a very slow convergence of training, and thus greatly increasing the training time and the number of epochs (*Hu et al., 2017*; *Sharma & Shaik, 2016*). In addition, when the dimension increases with the increase of data volume, it

Table 1 Performance comparisons between the proposed method and state-of-the-art methods based on the FRGC-v2 dataset.

Method	Features	Classifier	Accuracy
Huiying Hu et al.	Raw image	Custom CNN-2	85.15%
S Sharma et al.	Constrained Local model	Custom CNN	98%
Sima Soltanpour et al.	$LNDP^{3}{}_{xyz}$ Based Normal Component Images	SIFT-based matching Method	98.10%
Proposed Methodology	HOG features	ResNet 152 layers ResNet 101 layers ResNet 52 layers and Fine-tuning	**Top1: 98.26%** **Top2: 99.40%** Top1: 97.77% Top2: 99.40% Top1: 97.02% Top2: 99.12%

will lead to a curse of dimensionality problems and cause a drop in the performance of the classifier (*Soltanpour & Jonathan Wu, 2017*). The Fine-tuning method speeds up the convergence and shortens the training period, thus adapting to 3D processing. The purpose of multi-layer convolution is that the features acquired through one-layer convolution are often local, and the more layers there are, the more global the features will be acquired. Then, how to maintain good performance and improve accuracy is a key in larger numbers of 3D face recognition scene. Therefore, fine-tuning depth residual network is proposed based on HOG features to effectively solve the problem of large numbers of 3D face recognition.

To the best of our knowledge, our work is to examine a fine-tuned Deep Residual Networks model on the recognition task of FRGC-v2 dataset. To increase accuracy during ResNet training, several methods were considered in this paper: (1) a fine-tuning deep residual network was adopted, taking advantage of its intrinsic features, such as shortcut connection, weights sharing and pooling architectures, and these can be improved through the deepening of the network structure; (2) the number of layers is carefully designed with smaller filter size to avoid overfitting while there is sufficient capacity for the network to solve the complex large number classification problems (*Hawkins, 2004*; *Lawrence & Lee Giles, 2000*); (3) data extraction was performed via the HOG method at the image preprocessing that contains higher discriminative features in the 3D images. As a result, the proposed methods were well trained and yielded state-of-the-art classification accuracy.

CONCLUSIONS

In this study, in-depth investigations were conducted on end-to-end 3D face textures recognition. We first review the previous studies on 3D face recognition and then summarize the critical research questions to be solved. The 3D face detection and alignment modules are implemented and flexibly applied in 3D face raw data, which achieves a precision rate of 99.85%. In addition, the detailed steps of the HOG extraction pattern were presented. 3D face images with HOG features can significantly minimize the descriptor size for reducing computation load and economizing the memory in the

recognition process. We trained the fine-tuned ResNet models combined with HOG features; the discriminative power of the deeply learned features can highly enhance recognition ability. This study implemented every important subcomponent, which can effectively reduce 3D image noise and greatly increase the robustness of our proposed recognition system.

The experiment showed that the degradation problem was efficiently solved by increasing the number of layers in our fine-tuned ResNet neural networks, which improves the recognition rate within a short time, and the accuracy is maintained at a certain level. However, although the performance of the algorithm is unexceptional in practical application, we think that several aspects of the model should still be studied and improved. Firstly, although the HOG algorithm is advantageous for less calculation time and faster detection speed, when the pose of the 3D face is changed drastically, there is a target loss in the face image, which leads to low processing efficiency. The 3D face alignment can be pre-processed with the CNN detection method, and a multi-processing or multithreading method can be used to speed up face alignment, which ensures that the pre-processing module can process data quickly. Secondly, the recognition rate may be adversely affected by certain conditions. For instance, the ResNet-152 model exhibited the phenomenon of overfitting, in which the accuracy rate dropped and remained at around 97% after the 9th epoch. This phenomenon is caused by two conditions: too few datasets and the excessive complexity of the neural network model. This can be solved by increasing the amount of 3D face data in the future works via a data augmentation method (*Perez & Wang, 2017*; *Wong, Gatt & Stamatescu, 2016*). This also shows that the ResNet network has a more powerful data processing capability for a large number of data. Overall, the development of large number 3D face recognition classification system is a challenging work, and there is still a long way to go to apply these theories and methods in large-scale scenes. The results suggest that fine-tuned deep residual networks classification approach based on HOG features will be a promising direction to improve 3D face recognition rate.

ACKNOWLEDGEMENTS

I am indebted to Prof. Rahmita Wirza O.K. Rahmat for assistance with data collection, providing helpful discussions of the 3D face analyses. I thank Fatimah Khalid for her help in the depth analysis and comments on the face recognition theory, as well as Prof. Dr. Nurul Amelina Nasharuddin for fruitful advices.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Siming Zheng conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, performed the computation work, authored or reviewed drafts of the paper, approved the final draft.
- Rahmita Wirza O.K. Rahmat conceived and designed the experiments, performed the experiments, contributed reagents/materials/analysis tools, authored or reviewed drafts of the paper, approved the final draft, checked the logic of this article.
- Fatimah Khalid conceived and designed the experiments, analyzed the data, contributed reagents/materials/analysis tools, authored or reviewed drafts of the paper, approved the final draft, checked the fluency of this article.
- Nurul Amelina Nasharuddin analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, performed the computation work, authored or reviewed drafts of the paper, approved the final draft, examined the experimental codes.

Data Availability

The following information was supplied regarding data availability:

The raw data is available at zheng, siming (2019): 1-step-FRGC2-Original-Dataset-.zip. figshare. Figure. DOI: 10.6084/m9.figshare.9791951.v1.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.236#supplemental-information.

REFERENCES

Ahamed H, Alam I, Manirul Islam MD. 2018. HOG-CNN based real time face recognition. In: *Proc. International conference on advancement in electrical and electronic engineering (ICAEEE), vol. P-045*. 1–4 DOI 10.1109/ICAEEE.2018.8642989.

Bagchi P, Bhattacharjee D, Nasipuri M. 2015. 3D face recognition using surface normal. In: *TENCON 2015*. Piscataway: IEEE DOI 10.1109/TENCON.2015.7372819.

Baltrusaitis T, Robinson P, Morency L-P. 2013. Constrained local neural fields for robust facial landmark detection in the wild. In: *ICCV*. DOI 10.1109/ICCVW.2013.54.

Baltrusaitis T, Robinson P, Morency L-P. 2016. OpenFace: an open source facial behavior analysis toolkit. In: *IEEE winter conference on applications of computer vision*. DOI 10.1109/WACV.2016.7477553.

Chen Z, Wang J, He H, Huang X. 2014. A fast deep learning system using GPU. In: *IEEE international symposium on circuits and systems (ISCAS)*. Piscataway: IEEE DOI 10.1109/ISCAS.2014.6865444.

Cheng Z, Shi T, Cui W, Dong Y, Fang X. 2017. 3D face recognition based on kinect depth data. In: *4th international conference on systems and informatics (ICSAI), vol. 1*. 2–3 DOI 10.1109/ICSAI.2017.8248353.

Flynn P. 2006. *Face recognition grand challenge biometrics database (v2.0) license agreement*. Notre Dame: University of Notre Dame DOI 10.6084/m9.figshare.9791951.

Fred A, Agarap M. 2018. Deep learning using rectified linear units (ReLU). ArXiv preprint. arXiv:1803.08375v2:1-6.

Hanin B. 2018. Which neural net architectures give rise to exploding and vanishing gradients? In: *32nd conference on neural information processing systems (NeurIPS), vol. 1*. 580–589.

Hawkins DM. 2004. The problem of overfitting. *Journal of Chemical Information and Computer Sciences* **44**:1–12 DOI 10.1021/ci0342472.

He K, Zhang X, Ren S, Sun J. 2015. Deep residual learning for image recognition. In: *IEEE conference on computer vision and pattern recognition (CVPR)*. Piscataway: IEEE DOI 10.1109/CVPR.2016.90.

Hu H, Ali Shah SA, Bennamoun M, Molton M. 2017. 2D and 3D face recognition using convolutional neural network. In: *TENCON 2017*. Piscataway: IEEE DOI 10.1109/TENCON.2017.8227850.

Huang Y-B, Li K, Wang G, Cao M, Li P, Zhang Y-J. 2015. Recognition of convolutional neural network based on CUDA Technology. ArXiv preprint. arXiv:00074 (1506) 2-4.

Ioffe S, Szegedy C. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning* **37**:2–5.

Jourabloo A, Liu X. 2015. Pose-invariant 3D face alignment. In: *IEEE international conference on computer vision (ICCV)*. Piscataway: IEEE DOI 10.1109/ICCV.2015.421.

Jung H, Lee S, Yim J, Park S, Kim J. 2015. Joint fine-tuning in deep neural networks for facial expression recognition. In: *IEEE international conference on computer vision (ICCV)*. Piscataway: IEEE DOI 10.1109/ICCV.2015.341.

Kamencay P, Benco M, Mizdos T, Radil R. 2017. A new method for face recognition using convolutional neural network. *Digital Image Processing and Computer Graphic* **15**:664–670.

Kim D, Hernandez M, Choi J, Medioni G. 2017. Deep 3D face identification. In: *2017 IEEE international joint conference on biometrics (IJCB)*. Piscataway: IEEE DOI 10.1109/BTAS.2017.8272691.

Kumar P, Happy SL, Routray A. 2016. A real-time robust facial expression recognition system using HOG features. In: *International conference on computing, analytics and security trends*. DOI 10.1109/CAST.2016.7914982.

Lawrence S, Lee Giles C. 2000. Overfitting and neural networks: conjugate gradient and backpropagation. In: *IJCNN*. DOI 10.1109/IJCNN.2000.857823.

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* **1**:541–550 DOI 10.1162/neco.1989.1.4.541.

Maiti S, Sangwan D, Raheja JL. 2014. Expression-invariant 3D face recognition using K-SVD method. *Applied Algorithms* **8321**:268–275.

Min R, Choi J, Medioni G, Dugelay J-L. 2012. Real-time 3D face identification from a depth camera. In: *ICPR*. DOI 10.1109/CRV.2018.00020.

Nagi GM, Rahmat R, Taufik M, Khalid F. 2013. Multimodal 2D-3D face recognition. *International Journal of Future Computer and Communication* **2(6)**:687–691 DOI 10.7763/IJFCC.2013.V2.253.

Pabiasz S, Starczewski JT, Marvuglia A. 2015. SOM vs FCM vs PCA in 3D face recognition. *Artificial Intelligence and Soft Computing* **9120**:120–126 DOI 10.1007/978-3-319-19369-4_12.

Pascanu R, Mikolov T, Bengio Y. 2012. Understanding the exploding gradient problem. ArXiv preprint. arXiv:1803.08375v2:1-6.

Perez L, Wang J. 2017. The effectiveness of data augmentation in image classification using deep learning. ArXiv preprint. arXiv:04621 (1712) 2-7.

Phillips PJ, Flynn PJ, Scruggs T, Bowyer KW, Chang J, Hoffman K, Marques J, Min J, Worek W. 2005. Overview of the face recognition grand challenge. In: *CVPR*. DOI 10.1109/cvpr.2005.268.

Porro-Munoz D, Silva-Mata FJ, Revilla-Eng A, Talavera-Bustamante I, Berretti S. 2014. 3D face recognition by functional data analysis. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* **8827**:819–826.

Santosh Dadi H, Mohan Pillutla GK. 2016. Improved face recognition rate using HOG features and SVM classifier. *IOSR Journal of Electronics and Communication Engineering* **11**:36–37 DOI 10.9790/2834-1104023642.

Sharma S, Shaik S. 2016. Real time face authentication using convolutional neural network. In: *International conference on signal processing (ICSP)* DOI 10.1049/cp.2016.1455.

Simonyan K, Zisserman A. 2015. Very deep convolutional networks for large-scale image recognition. In: *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. DOI 10.1109/ACPR.2015.7486599.

Singh G, Chhabra I. 2018. Effective and fast face recognition system using complementary OCLBP and HOG feature descriptors with SVM classifier. *Journal of Information Technology Research* **11**:34–33.

Singh S, Paul A, Dr. Arun M. 2017. Parallelization of digit recognition system using deep convolutional neural network on CUDA. In: *Third international conference on sensing, signal processing and security (ICSSS)*. DOI 10.1109/SSPS.2017.8071623.

Soltanpour S, Jonathan Wu QM. 2017. High-order local normal derivative pattern (LNDP) for 3d face recognition. In: *IEEE international conference on image processing (ICIP)*. Piscataway: IEEE DOI 10.1109/ICIP.2017.8296795.

Song A, Li L, Atalla C, Cottrell GW. 2017. Learning to see faces like humans: modeling the social dimensions of faces. *Journal of Vision* **17(10)**:837 DOI 10.1167/17.10.837.

Sun Y, Liang D, Wang X, Tang X. 2015. DeepID3: face recognition with very deep neural networks. ArXiv preprint. arXiv:1502.00873v1:2-4.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. 2014. Going deeper with convolutions. In: *2015 IEEE conference on computer vision and pattern recognition*. Piscataway: IEEE DOI 10.1109/CVPR.2015.7298594.

Tang H, Yin B, Sun Y, Hu Y. 2013. 3D face recognition using local binary patterns. *Signal Processing* **93(8)**:2190–2198 DOI 10.1016/j.sigpro.2012.04.002.

Wang X, Ruan Q, Jin Y, An G. 2015. 3D face recognition using closest point coordinates and spherical vector norms. In: *ICWMMN*. DOI 10.1049/cp.2015.0943.

Wong SC, Gatt A, Stamatescu V. 2016. Understanding data augmentation for classification: when to warp? *IEEE* **1**:1–4 DOI 10.1109/dicta.2016.7797091.

Wu Z, Hou Z, Zhang J. 2017. Research on the 3D face recognition based on multi-class classifier with depth and point cloud data. In: *IEEE advanced information management, communicates, electronic and automation control conference (IMCEC), vol. 1*. Piscataway: IEEE, 398–402 DOI 10.1109/IMCEC.2016.7867242.

Yosinski J, Clune J, Bengio Y, Lipson H. 2014. How transferable are features in deep neural networks? In: *The 27th international conference on neural information processing systems, vol. 2*. 3320–3328.

Zeiler MD, Fergus R. 2014. Visualizing and understanding convolutional neural networks. In: *ECCV LNCS, 8689 1*. 818–833.

Zhang D, Zhang M, Liu Y. 2019. Three dimension face recognition based on gabor transformation and support vector machine. *Journal of Applied Science and Engineering* **22**(**1**):163170.

Zhang J, Hou Z, Wu Z, Chen Y, Li W. 2016. Research of 3D face recognition algorithm based on deep learning stacked denoising autoencoder theory. In: *ICCSN*. DOI 10.1109/ICCSN.2016.7586606.

Zhu X, Liu X, Lei Z, Li SZ. 2017. Face alignment in full pose range: a 3D total solution. ArXiv preprint. arXiv:1804.01005v1:9-10.

Learning Scale-variant Features for Non-holistic Iris Authentication with Robust Deep Ensemble Learning Framework

Siming Zheng[1]*, Rahmita Wirza O.K. Rahmat[2‡], Fatimah Khalid[3‡], Nurul Amelina Nasharuddin[4‡]

[1] *Computer Assisted Surgery and Diagnostic (CASD), Department of Multimedia, Putra Malaysia University, 43400 UPM Serdang, Malaysia*

[2] *C1,Department of Multimedia, Putra Malaysia University, 43400 UPM Serdang, Malaysia*

[3] *C2,Department of Multimedia, Putra Malaysia University, 43400 UPM Serdang, Malaysia*

[4] *C2,Department of Multimedia, Putra Malaysia University, 43400 UPM Serdang, Malaysia*

Abstract

In recent years, mobile Internet has accelerated the proliferation of smart mobile development. The mobile payment, mobile security and privacy protection have become the focus of widespread attention. Iris recognition evolves into a high-security authentication technology in these fields, and widely used in distinct science fields in biometric authentication researches. The Convolutional Neural Network (CNN) is one of the conventional deep learning approach for image recognition, whereas its anti-noise ability is weak and needs a certain amount of memory to train in image classification tasks. Under these conditions we improved the architecture of Mask R-CNN and put forward the fine-tuning neural network architectures based on mobile Inception V4, which integrate every component in an overall system that combines the iris detection, extraction, and recognition function as an iris recognition system. The proposed framework has the characteristics of scalability and high availability; it not only can learn the scale-variant features by the zero-padding normalization but also enhancing the robustness of the whole learning framework. Importantly, our custom architectures can be trained by using different spectrum of samples, such as Visible Wavelength (VW) and Near Infrared (NIR) iris biometric image data. The recognition average accuracy of 99.10% is achieved while executing in the mobile edge calculation device of the Nvidia Jetson Nano.

1. Introduction

The essential characters of iris informatization are digitalization and recessiveness. The iris is one of the most complex organs of the human body; the hidden password in the eyeball is richer, and contains much more random texture patterns than using the Personal Identification Number (PIN), fingerprint, or the human face [1] [2]. Under these favorable conditions, iris characteristics can provide value for identifying encryption technology. More and more iris authentication technologies have been applied to mobile devices, such as smart mobile phones, tablets, and human-machine interactive devices, due to the graphics processing units (GPU) with high-performance graphics processing capabilities [3] [4] [5].

Over the last two decades, smart mobile devices have been embedded with built-in high-resolution imaging sensors. The sensor can be used to perform iris recognition tasks and allow researchers to explore appropriate solutions for all the stages of iris recognition in a mobile environment. Some iris authentication functions are supported in earlier mobile devices. For example, the first smartphone designed with an iris authentication, Arrowsnxf-04G [6],is equipped with an infrared camera and a light-emitting diode (LED). The camera can scan and decode the images of the iris of the user. To support iris authentication development, Samsung S8 series mobile phones added Infrared Radiation (IR) and an iris camera in its front lens; using the front camera to assist the iris camera with infrared LED to determine the approximate general outline of the user. The iris camera scans the iris information through the light source and then converts this information into a specific code. Finally, the system compares the code with a known password to determine whether to unlock. Huawei introduced GPU Turbo technology in 2019, which greatly improved the performance of the graphics processing on several smartphones (P30 Pro series [7], up to 60%. In addition, Nvidia Shield tablet K1 [8] equipped Kepler architecture with 192 cores streaming graphics multiprocessors, which supports thousands of threads to implement high-performance calculations in parallel.

Although the processing power of these mobile devices is growing, system robustness is still the main concern. Currently, there are two research problems that need to be addressed in a complex mobile environment [9] [10]. In this research, the first problem addressed is how to process and verify the high quality images of the iris with scale-variant features, because most mobile phones are equipped with a high-definition camera but in an uncooperative

environment. The second problem relates to how to improve the recognition accuracy on target iris images under a different spectrum for application in different practical scenarios. Therefore, our research objective is to investigate the structure of the multi-learning model to solve the research problems above in a mobile environment.

The remainder of this paper is organized as follows: We briefly introduce related works about iris authentication in Section II. Section III puts forward the proposed iris authentication framework with multiple critical components for iris region extraction and matching. Section IV shows experimental findings in favor of using the proposed method, and the analysis is in Section V. Finally, we discuss and conclude the research in Section VI and VII.

2. Related Works

Iris authentication is a process of identifying individuals based on their iris shape and texture distribution[31] [32]. We also have reviewed the strengths and weaknesses of current iris authentication studies. As a result of the survey, from which we were able to gain valuable conclusions, we found that several studies demonstrated that some scholars used similarity computation methods to measure the similarity between the two iris templates, as posited in [11] [12] [13] [14] [15].

The most critical work of the iris verification system is to detect the iris and outer boundary correctly. Deshpande et al. [16] used Daugman's integral and differential algorithm [17] in their experiment. Once the iris boundary is detected, the program converts the iris to a standard size and encodes it in the iris template to enable matching between the iris templates. The Rubber Sheet model is used for the normalization of the iris image. In the final matching phase, an 1D logarithmic Gabor filter is applied for the iris feature extraction, while the Hamming distance is used as a matching algorithm to compare two biometric templates for iris verification. One of the challenges in seeking the boundaries are iris images with low contrast or low lighting during detection. Deshpande's algorithms overcome some difficulties in non-uniform illumination and reflections. While his works enhance the performance of the segmentation and normalization process in iris authentication systems and, his proposed process flow contains many different individual sub-components, which increase the complexity in the system with weaker coupling. When Daugman's Rubber Sheet Model and 1D log Gabor filter completed the

normalization and extraction of the iris, the important detailed texture data was lost. His proposed system achieves an overall accuracy of 95% with robust characteristics.

In Mohammed Hamzah Abed's work [18], he adopted a Circular Hough transform to detect an iris in the recognition system with a precision rate of 98.73% on average. The Haar wavelet then transforms the extracted raw features from iris images for fast computing and low storage. However, he applied the method of Principal Component Analysis (PCA) to alleviate a variety of noises for image reduction. The principle of PCA is mainly to eliminate the correlation between variables. It cannot improve discriminative information and achieve good results for nonlinear dependence. In the verification phase, the method of Cosine distance measured the similarity between two non-zero vectors of inner product space; the result of the experiment shows that the method is effective with a classification accuracy of 91.14%.

Subsequent works have continued to use the same experimental data as a way to explore the performance of the proposed framework. Kaudki [19] introduced some of the new concepts in iris preprocessing. Rubber-Sheet Unwrapping Normalization has the ability to deal with different sizes of iris images extraction due to the changes in pupil size caused by external lighting. This kind of normalization mainly improves the clarity of acquired features. Furthermore, the Haar wavelet transform is applied for feature extraction because of its computational simplicity. The Hamming distance, as a measure of the characteristic distance, is used to match and validate the target data. The recognition accuracy does not continue to increase substantially and remains at around 97%. This is because that recognizing high-quality iris image is the weakness of the traditional template matching method. Iris images processed in a series of transformations suffer from significant degradation [20], making iris recognition between the training set and testing set less relevant. This might be due to image degradation by some image fusion techniques [20] [21] [22] [23], and feature extraction operations like Hough transform [24] [25] [26]. Besides, it is important to point out that author adopted a non-robust method for iris localization - Circular Hough Transform(CHT), which cause the poor positioning accuracy of 96% due to the limited circle candidates produced by "voting" in the Hough parameter space. These findings reinforce the importance of researching the influence of image quality.

Currently, some of the attention models have been proposed and applied in fine-grained representation learning tasks to efficiently extract non-holistic

subtle features and the key parts from the human body, and different data types were used to extract multi-pattern features to enhance the performance of visual features of local and global regions. For instance, Kai Han et al.[27] proposed Attribute-Aware Attention Model (A3M) combining fine-grained classification and retrieval to represent semantics for improving the recognition abiity. The global features and attribute-specific features were utilized to derive the local information of the image for more discriminating representation in object re-identification. The RA-CNN [28] proposed a recurrent attention convolutional neural network to discover critical parts for the fine-grained prediction tasks automatically. The process recursively learned the multiple scales of attentional regions and region-based feature representation. To improve robustness of model learning, Guo et al. [29] proposed a novel two-branch network, the human body and potential part branches were composed of a modern human body analysis model and a self-focus mechanism, which aims to solve the problem of misalignment of human and nonhuman body parts. Similarly, the most relevant iris recognition methods published over the years are "phase-based", such segmentation and normalization of the iris are to obtain dimensionless representations. Hugo P.[30] think this can be avoided and propose a non-holistic iris recognition method that does not require iris segmentation. Their experiments reveal that the proposed method is particularly valuable in the case of low-quality data.

As mentioned in the previous reviews, there are still many challenges and requirements in the iris authentication system, such as the high detection performance, low system coupling, robust extraction, high precision recognition rate. All of these become the main factors restricting the development of iris authentication system. Taking into the account the research questions and the background information mentioned above, it is possible to suppose the most appropriate method of the investigation here as qualitative and quantitative. Regarding current iris authentication system, there is a need to provide a practical evaluation of a self-determinative motivational decision making of the learning model, which also stimulate and motivate us to implement a deep learning-based ensemble learning framework for the modern biometric information security field.

3. Material and methods

A robust iris authentication system includes detection, extraction, normalization, and recognition submodules [33]. Every submodule is sequentially performed in the Graphics Processing Unit (GPU) in our experiments. Fig1 below shows the flow of pre-processing required for the training model. We added two types of layers, extraction and normalization, as the robust components, which are implemented respectively.

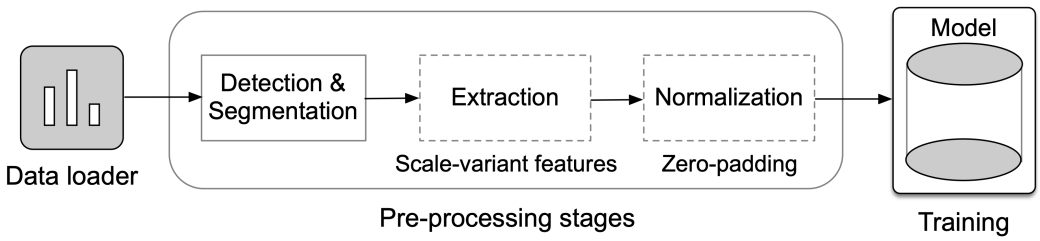

Fig 1. The flow of pre-processing. The flowchart shows the steps needed to be performed before the iris recognition model (From *left* to *right*). We propose the custom sub-components of extraction and normalization, which are both within the scope of the research. See the boxes with a dotted border area.

3.1. The Improved Mask R-CNN Learning Model

The preprocessing of iris automatic detection is a crucial stage in the iris authentication system. We found that some potential factors may be susceptible to interference, which could affect recognition performance [34]. For instance, the natural iris texture can be easily obscured by cosmetic contacts, having a significant impact on the extraction of the iris area. The experimental dataset consists of thousands of iris images taken from different angles and under various conditions which are affected by a variety of internal and external factors.

As can be seen from example A in Fig 2, a slender specular highlight is shown in the iris area near the nose. Thus, the performance is reduced owing to the light reflection or uneven illumination in the area of the iris. Example B shows a permeable contact lens covered in the iris region; this can generate some large artifacts in the detection region. Example C shows a

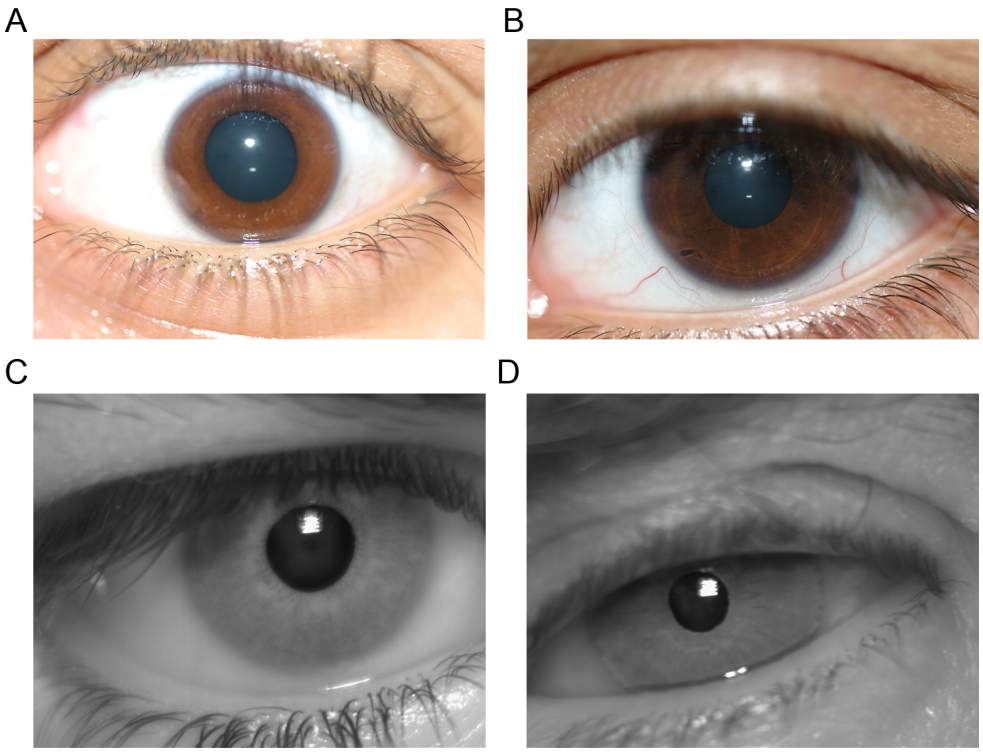

Fig 2. The challenges of UTiris datasets in a different spectrum. The color iris images (A and B) are acquired in the VW and grey iris images (C and D) are acquired under the NIR. The images were taken from [40].

small portion of eyelashes that have a considerable influence on structural iris texture. Example D is exhibiting a non-ideal iris region due to poor coordination of human-machine interaction. This causes many iris features to be lost. The challenges we have described in the detection and extraction phases can be summarized as follows: contact lens; eyelash occlusion, specular highlights of illumination, uncooperative action. Besides, the shape of the eyelids varies from one individual to another. All of these factors make the localization of the eyelids more difficult [35] [36] [37].

To detect the iris area at a fine-grained level, we custom the Mask R-CNN [38] neural network architecture. The model of Mask R-CNN is an extended structure of Faster R-CNN [39], including a function of pixel-to-pixel

alignment. The proposed Mask R-CNN is constructed by stacking the layer to predict the iris location in each Region of Interest (ROI); a mask branch. This layer is similar to the existing boundary layer and classification layer. Our Mask R-CNN architecture mainly performs two different operations: object detection and semantic segmentation, which are used for the iris region extraction base on iris datasets, and are aimed at extracting and normalizing the iris region from the object images.

3.1.1. The RPN Component

In the process of iris detection, there are irises of different sizes in the raw image, and the detection of iris of different sizes requires different features. The multi-scale pyramid is a good solution, such as FPN component. First, the component of ResNet performs the deep convolution operations on the input iris image with 3x3 convolution kernel, and then generates five different feature maps with different sizes in each layer. Second, the component of FPN performs the upsampling operations on the last 4-layer convolutional feature map, and then the results are merged with the data generated by the previous convolution layer. Thus, combining these two types of feature maps, which can further enrich the ability to express features at different scales, thereby achieving robust iris detection capabilities.

The Region Proposal Networks (RPN) is an essential component of Mask R-CNN. They are lightweight neural networks that are used to replace the selective search in the model of Faster-RCNN. Similar to Faster R-CNN, the purpose of RPN is to seek and generate the region proposals effectively.

In the RPN architecture, a sliding window is used to scan the image and to find the area where the target of the iris exists; a rectangle distributed over the image area, as demonstrated. The center of the sliding window is the anchor, as shown in Fig 3, and every anchor is implicated in the aspect ratio and the scale [38] [39]. The sliding windows are implemented by RPN convolution, and they scan the surrounding area based on the anchor points at high speed. Sliding windows scan all areas in parallel mode by using the GPU acceleration.

By default, we used three aspect ratios and three scales, resulting in 9 anchors at each sliding position. For example, the 3x3 convolution feature map networks slide on WxH using the padding operation and the stride is 1; thus, the sliding window has WxH slides, each slide with 9 anchors in each

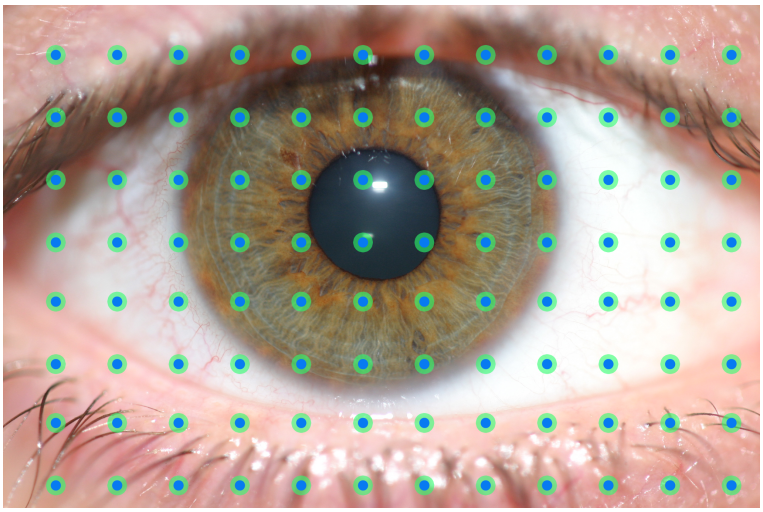

Fig 3. The generated anchor points distributed on the raw iris image. The figure above gives a visual representation of the effect of anchor distribution. The number of anchors is obtained through the RPN network, which is used to select positive and negative samples and finally calculate the difference between the positive sample and the ground truths [38].

scanning operation. Finally, the number of WxHx9 anchors are generated, and the role of RPN is to use these anchors to determine the location of the feature map and the size of its bounding box.

At each sliding window position, multiple scanned regions are considered as candidates, predicted simultaneously. In our setting, a limited number of the highest potential candidate regions for each iris image is counted as k boxes. In the region layer of Mask R-CNN, the outputs of 4k parameters are needed to encode the coordinates of 4 different points for the k boxes, and the classification layer requires 2k values to evaluate the probability of whether each region is the target.

Fig 4. illustrates the processing to seek and pin-point the ROI of the iris. Using the sliding window and anchors, we obtained W×H×9 proposals from one original iris image. Each proposal generated six parameters: two parameters (*0* and *1*) which are used to label the foreground and the background probability, calculating the target by comparing each proposal and ground

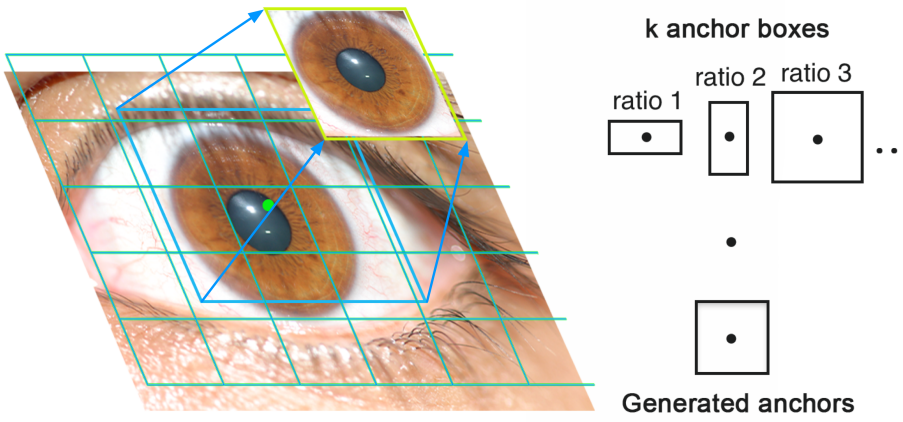

Fig 4.The processing of iris region extraction. The figure demonstrates iris localization and iris region extraction, which are performed to provide discrimination information for a detailed neural network. The scale and aspect ratio of the anchor is controlled by the *"SCALES"* and *"RATIOS"* parameters in configuration.

---

truth. Meanwhile, each proposal is transformed into ground truth size by translation and contraction operation. There are four parameters (*upper-left, upper-right, lower left and lower right*) for the locating four coordinates due to the differences in the position and size of each proposal and ground truth. Once six parameters are appropriately set, all the scale-variant features of the detected region can be output, including the various coordinates of iris detection. This enables us to implement further normalization function by using the extracted iris image with the highest probability.

3.1.2. The Iris Detection on Experimental Data

The experimental iris images not only contain the regions of interest (ROI) of the iris but also existing redundant identifying information, such as eyelashes and sclera, etc. Thus, the current raw iris images cannot be used directly in the training model. We focused on iris preprocessing and recognition implementation in the robustness and practicability of the system. The crucial ability of our framework is to extract unique properties from

the iris images, making it easier to create a specific code for each individual. In our experiment, the reconstructed Mask R-CNN is used for locating and extracting the iris with scale-variant features. Before inputting unique scale-variant iris features and texture into the recognition model, two steps need to be performed, including iris extraction and normalization.

In the iris region detection, we propose an efficient detection method specifically for the iris region, and limbic and boundary of the eyelid on our dataset. To collect some training-testing samples, we create some ground truth in the dataset of UTiris for building iris ground truth. The experimental data is a set of color and grey iris images with high-resolution 2048×1360 or 1000×776 pixels. Firstly, we randomly select 158 iris images from the whole UTiris dataset [40], and each has two images for training the Mask R-CNN model. Iris data annotation is the important part of iris detection, we mark the outline of the iris to provide the ground truth for training the proposed Mask R-CNN. The generated ground truth image is labeled as 1, if there is evidence of the iris in the sliding window of 299×299 pixel or 0 background, as presented in Fig 5 below.

Then, to enhance the robustness of the model input, we generated a multi-channel space that duplicates the same array from grey space to a representative grey image. The new arrays are formed by stacking the given arrays in three dimensions. We adopt the pre-trained model to the COCO dataset [41]. By matching the testing set and ground truths, the proposed Mask R-CNN calculates their matched probability to output the best one, which is used as a referencing position to locate the iris region of the raw iris images, as indicated in Fig 6 (B). Ground truths matched across different iris examples can be seen in (DOI 10.6084/m9.figshare.10280492). If the raw iris image is successfully matched by the top probability of iris ground truth, it can generate four coordinates. They correspond to four corners of a square in Fig 6 (C). The iris position can be routed through the squares with different sizes; the results of iris localization are displayed in Fig 6 (D). The results show good localization performance (DOI 10.6084/m9.figshare.10280501). After the iris is located, if the iris outer boundary contains some periocular textures, such as sclera, eyelid and pupil, further analysis correlation may be adopted [42]. In addition, all of these textures play a significant role expected to complement the iris as auxiliary features to improve the recognition effect in non-cooperative environments [43] [44]. The same calculation mode is applied for the grey iris images arraying from Fig 6 (E) to 6 (H).

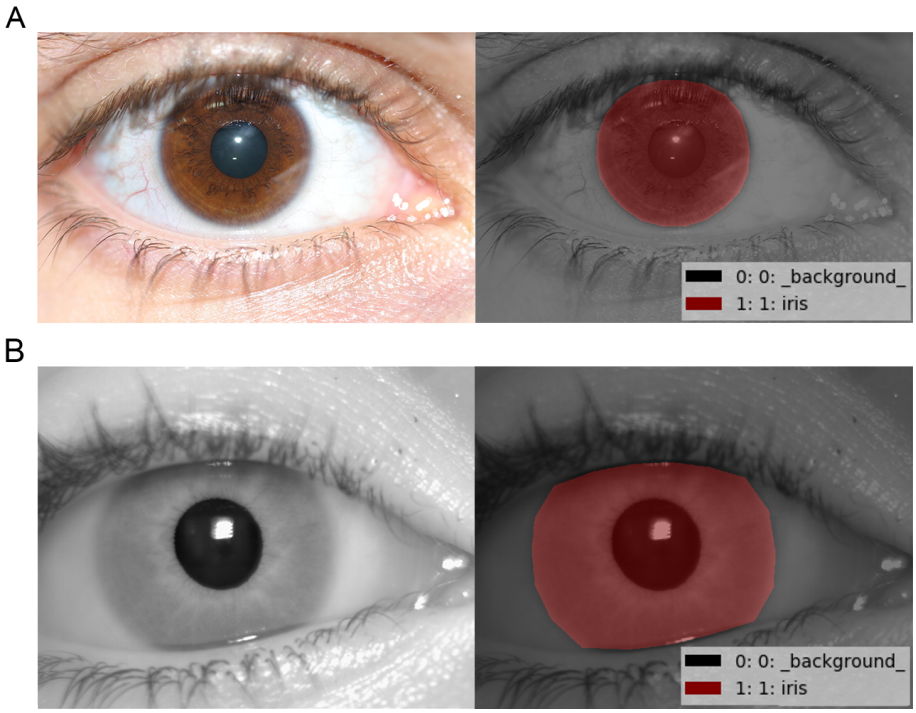

Fig 5. The annotation on the fine-grained iris images. Corresponding pairs of iris ground truth acquired in the *VW* and *NIR* session. The figure given at the *left side* is original iris dataset and the *right side* is iris ground truth layer with red region.

3.2. The Iris Extraction and Normalization on the Scale-variant Iris Images

During the testing recognition model, the fully connected layer needs to reshape the pooled results after the convolutional layers, so the input images must be a set of fixed size images. If the dimension of the input vector is not fixed, then the number of the weight parameters of the full connection is variable, which results in slow testing and dynamic changes of the network, and has almost no effect on parameter training. In the previous step, all of the extracted iris region images are the same width of 299 pixels, are diverse in height size (less than 299 pixels). To fill these vacant pixels is an effective way to enhance and train the recognition model.

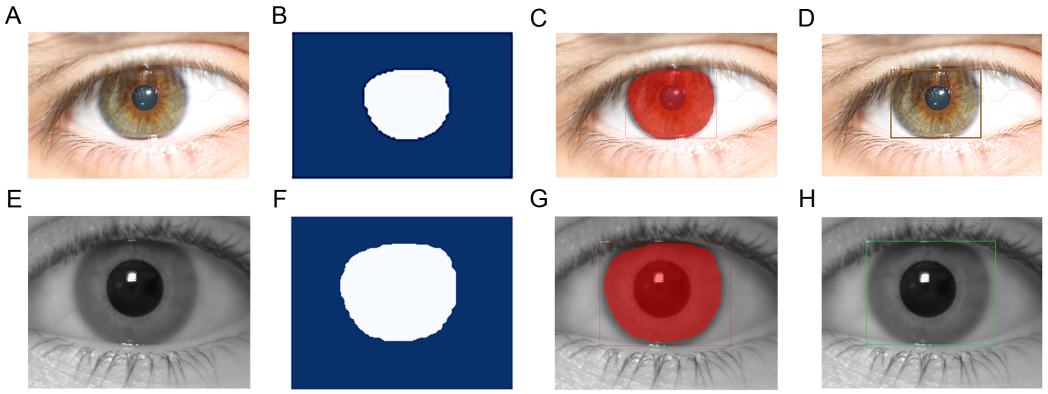

Fig 6. The iris localization and Extraction with iris ground truths. To achieve iris region localization, three normalized steps corresponding to the top iris ground truth (*2nd column*), iris region (*3rd column*), iris periocular (*4th column*), are obtained under the different spectrum images applied by the proposed Mask R-CNN. The *upper* row in each sub-diagram shows successful iris region extraction in the *VW* session, and the *bottom* row shows the segmentation in the *NIR* session. All of these extracted images are non-holistic iris.

After the iris extraction, these non-holistic iris images have scale-variant features, this is because the variables of width and height of each image are various reply on different size of the extracted iris region. The effective processing range of each iris image is different and non-holistic. The purpose of the proposed zero padding layer is to determine whether to fill the additional edge pixels of the input image matrix when performing convolution or pooling operations. Given this, we have customized a zero-padding layer in the architecture of Mask R-CNN for normalizing the iris images with different width and high. Consequently, the shapes match the output image as needed, and directly input into the further recognition model, as shown in Fig 8 below.

The proposed zero-padding layer can fill the vacant area with pixels in the vertical direction, and the width kept the same, the whole process completed by TensorFlow session, as shown in Fig 7 and 8. The details include four major steps: 1). first get the width and height information on each extracted

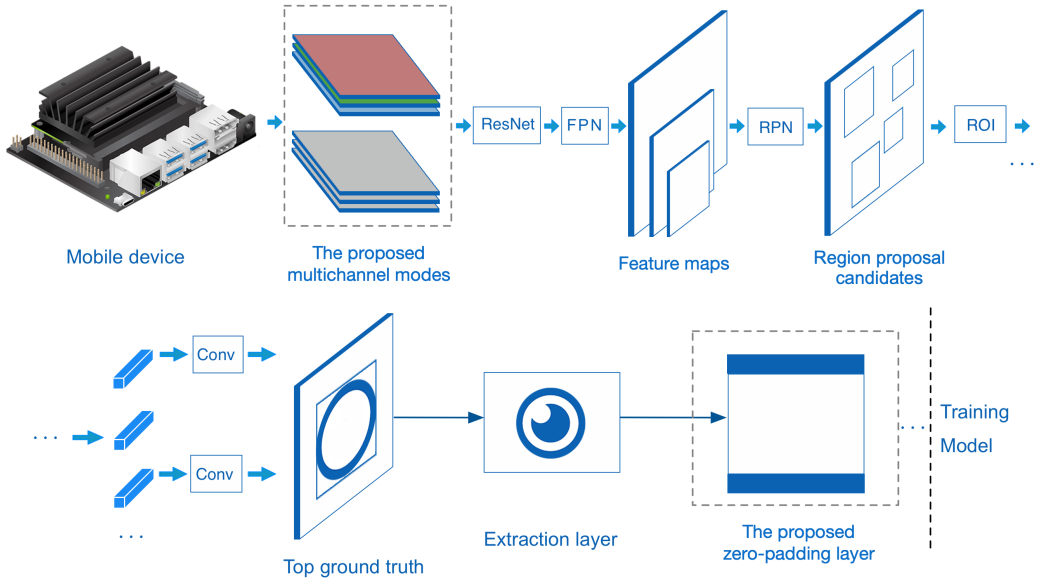

Fig 7. The optimized Mask R-CNN architectures in the mobile environment. The *first* dotted box illustrates that we implement two input modes are applied during the preprocessing of the system, *VW(Color)* and *NIR(Grey)* modes. The *second* dotted box illustrates that the non-holistic iris features map with discriminative information was obtained by Mask R-CNN and those scale-variant images are normalized by the custom *zero-padding layers*. Each individual component proposed in our Mask R-CNN architecture does influence iris recognition performance.

---

iris image, 2). if the width is greater than the height, then fill the pixels vertically with the center of the image, 3). the height of the area covered by the filled pixels is to be consistent with the width, 4). repeat the above process until all the extracted iris images are filled. Through this method, we can align the images and effectively maintain the scale invariance of the extracted iris images, thereby increasing the robustness of the current system. Besides, all the values of filled pixels are zeros with black color [45] [46], the advantage is that it can reduce GPU computation during the training model, and the stability of the recognition model for image processing is enhanced[] [].

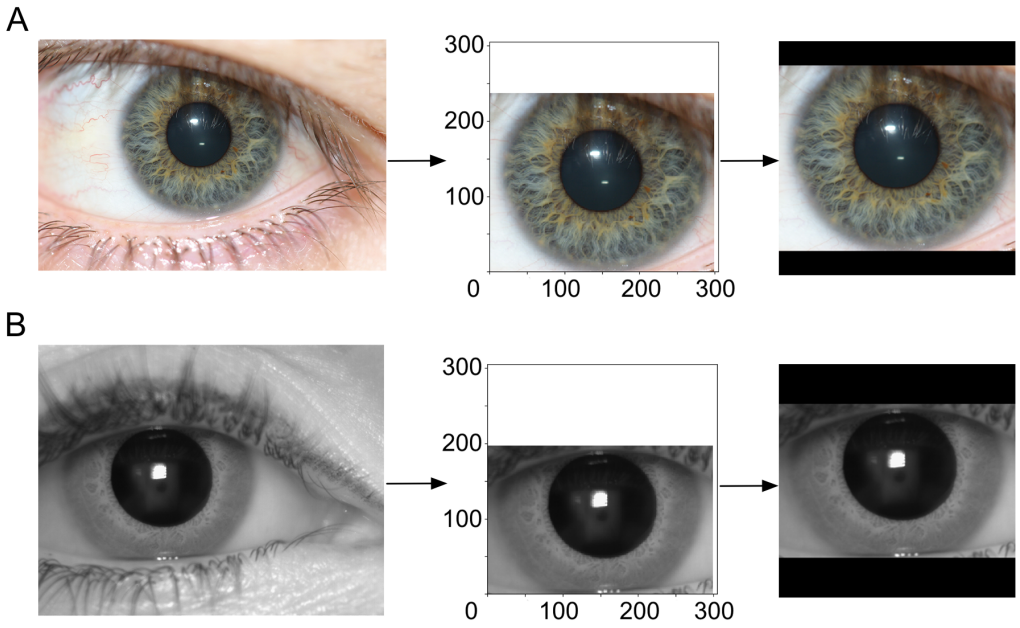

Fig 8. The proposed zero-padding normalization on extracted iris images with different sizes. The experimental results show that the iris region is extracted from the raw data (*left*, original iris image), then the extracted non-holistic iris images with scale-variant features in different width and high variables (*middle*, iris region) are transferred into a standard training set(*right*, processed iris image with padding layer).

3.3. The Fine-Tuned Mobile Inception V4 Architecture

The fine-tuned mobile Inception V4 is executed to identify the human iris. For our experimental dataset, the iris image represents a specific domain, such as iris periocular and iris characters, which belong to the human central visual system. For these particular texture images, a strategic priority for us is that we would fine-tune the Inception V4 neural networks and continue training them on the iris dataset we have.

Many novel models and efficient learning techniques have been introduced to make CNN's model deeper and more powerful [47] [48], achieving revolutionary performance in a wide range of inputted data. Szegedy et al. [49]

proposed an improved mobile version of Inception v4 based on Inception v3 [50]. Inception v4 primarily consists of an input stem, three different inceptions and two reduction modules. Based on the structure of Inception v4, we have developed a fine-tuned model. The proposed model includes a final classifier layer, and its dimensions differ from the original model. Fig 9 shows the complete architecture of the proposed model; the fine-tuning elements are shown in the dashed boxes. The implementation of iris recognition on a mobile device is significantly different due to a different environment on a dedicated device; the former relies on computational power and has limited space for storage.

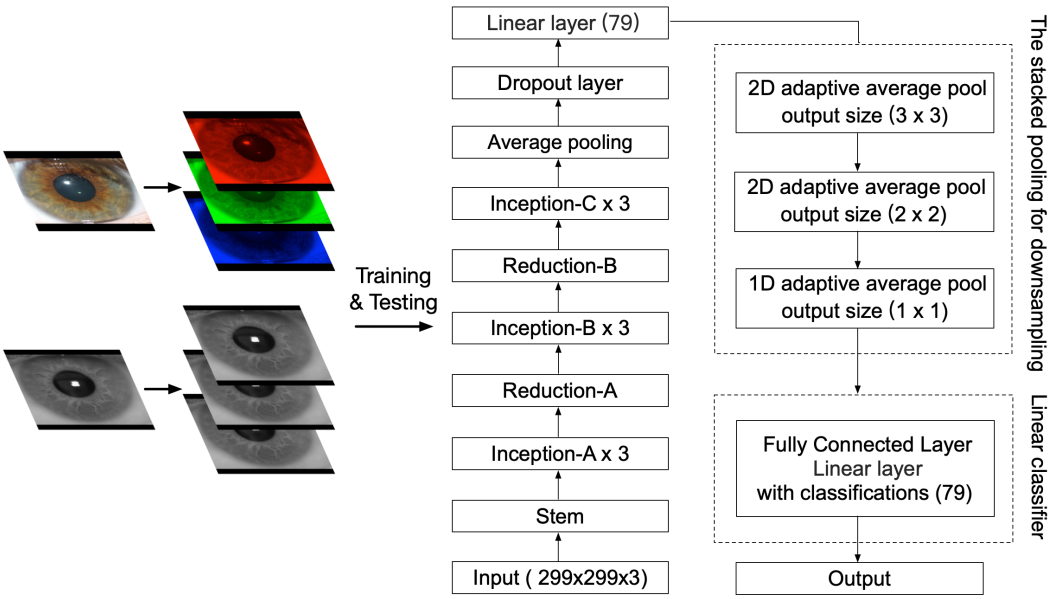

Fig 9. The fine-tuned Inception V4 architecture. The final pooling procedure is described as a series of adaptive pooling combinations, where the filter is with different output sizes. The final full connection layer is adjusted according to the number of classifications in this experiment. These two subcomponents (*with two dotted boxes above*) are the optimized structures generated by our fine-tuning method.

Below are the implementation details of our fine-tuned Inception V4 neural

network model:

Here, we fine-tuned the first layer of our model based on the original Inception V4 model. The input size is adjusted to 299X299 pixels. Different networks require different input sizes. For example, the Inception V4 requires that the image size of the input network is 299x299 pixels. Accordingly, the image size was adjusted to 299x299 pixels, so the feature mapping dimension output by the intermediate convolution module can remain the same.

So that the model recognizes the iris more efficiently in a mobile environment, we loaded pre-training weights and custom classification layers to develop our model. Our task is for the network to focus on learning scale-variant features in the final linear layer. Siyu Huang et al. propose simple but effective variants of the pooling module - stacked pooling [51]. Stacked pooling is an equivalent form of multi-kernel pooling [52] and works by stacking smaller kernel pooling. All the pooling operations are calculated on down-sampled feature maps except for its first kernel pooling, which can reduce the computing cost for training the learning model. Their empirical studies reveal that stacked pooling shows a better computing efficiency than multi-kernel pooling. In our fine-tuned model, we stacked three adaptive average pooling layers; the first two layers are two-dimensional, and the output sizes are 3×3 and 2×2, respectively. The last layer's output size is 1 and applies an 1D adaptive average pooling over an input signal composed of several input planes.

Our iris recognition is a classification task. The fully connected layer of the proposed network is set to 79 categories for the current UTiris datasets instead of the 1000 categories of default classification capability. Given this, we also make some improvements before the model output. First, the adjustment is to fine-tune the last linear layer of model with 79 classes features output. Next, a combination of three pooling functions is proposed and followed by the previous linear layer, which is implemented by the adaptive average pooling functions. The output sizes of each are 3, 2, 1, respectively. Finally, we fine tune the last fully connected layer with the classification number of 79 for the current classification task, this further improves the model's nonlinear expression ability. This kind of architecture is to reduce the calculation parameters during the training, at the same time, all the high-level discriminative features learned by the previous layers can be retained, which can efficiently improve the convergence of the model during the training.

Selecting the correct level of activation function is an important part of the design in a neural network. In the process of backpropagation, the gradient

has direction and size. The gradient descent algorithm multiplies a variable called the learning rate to determine the location of the next point. A paper [53] by Leslie N. Smith describes some very instructive learning rate settings to find the initial learning rate. If the learning rate was low, the gradient would decline slowly, and the training took a long time. By contrast, if the learning rate setting was high, then it was difficult to converge to the extreme value. Considering the experiment in practice, we apply the value of 1e-4 as the learning rate in our proposed model.

Finally, we need to use an optimization algorithm to iterate over the model parameters to minimize the loss of function value. In many fields, such as computer vision, the most commonly used is the gradient descent method to find the optimal loss during the training phase [54]. The Adam algorithm is currently the mainstream optimization algorithm, some researchers also pointed out the defect of Adam's convergence [55] [56]. Combine with the parameter of AMSGrad gradient descent function, which is more robust to the parameter changes and make more stable in the training process [57]. In this research, the AMSGrad is substituted for the original optimizer without the momentum parameter in our proposed model.

Our proposed model adjusts and adds learning layers to optimize the Inception V4 and investigate whether they can improve decision making in the recognition process in the mobile environment. The methodologies above detail the implementation of multi-tasks deep learning architectures for single iris object detection on our edge computing device.

3.4. The Iris Datasets

To evaluate the performance of the proposed framework, we selected one well-known iris dataset of UTiris, which includes the iris image acquisition scheme, using different devices [40]. The database consists of a total of 1540 iris images, from Visible-Wavelength (VW) and Near-Infrared (NIR) sessions. The aim of proposed learning models is to train and validate the different images from the same individual's iris under the different wavelength environments. All the iris images are captured under non-constraint conditions and simulate mobile device's environment, such as non-ideal imaging, different imaging distances and illumination conditions. There are 804 images of 2048×1360 pixels in the VW session, and another 736 images of 1000×776 pixels in the NIR session, which are taken from 79 individuals demonstrated in 158

classifications. The dataset was collected by the University of Tehran from 24-27th of June 2007.

3.5. The K-fold Cross Validation

The cross-validation [58] emphasizes the objective evaluation of the matching degree of the model to the data. This data analysis method [59] divides the original data into k groups, creating a verification for each subset and takes the rest of the k-1 subset data as the training set; thus, the cross-validation can train K models. The K models are evaluated in the validation set, and the final loss is obtained by the weighted mean operation via the loss of each model. The loss function of our recognition model is implemented by cross-entropy loss function, combining the log softmax and negative log likelihood functions to calculate the model loss. This strategy is capable of measuring subtle differences and is suitable for applying in fine-grained image classification tasks.

3.6. The Edge Calculation Devices

The integrated framework above is executed on a GPU-based edge calculation device - Jetson Nano. The Jetson Nano is proposed by Nvidia [60], which is intended for low-power applications requiring high computational performance in mobile environments. It is equipped with a Maxwell GPU with Quad ARM Cortex-A57 processor, and 4GB of LPDDR4 memory. Learning models can run with a Linux kernel 3.10.96 on the Ubuntu 18.04 system. Design of low-power consumption (5 watts) and integrated GPU makes the Jetson Nano an ideal candidate for conducting the proposed methods. Due to its strong computing capabilities [61], researchers can perform multiple neural networks by parallel mode in certain fields, such as object detection, extraction and image classification in the complex mobile environment.

4. Results

4.1. Evaluation of Proposed Mask R-CNN Architecture

In our experiment, we randomly use a portion of the iris image as the training, then use the Callback function to save the best weights and load them, before predicting the new iris images. There is approximately 20%

of data for training and 80% for testing the proposed Mask R-CNN. After finished the preprocessing phase, all iris images were fixed to 299×299 pixel by the normalization operation in order to meet the model input size; this can efficiently reduce the capacity of the training set and increase the training speed for the further training model, as shown in Fig 10.

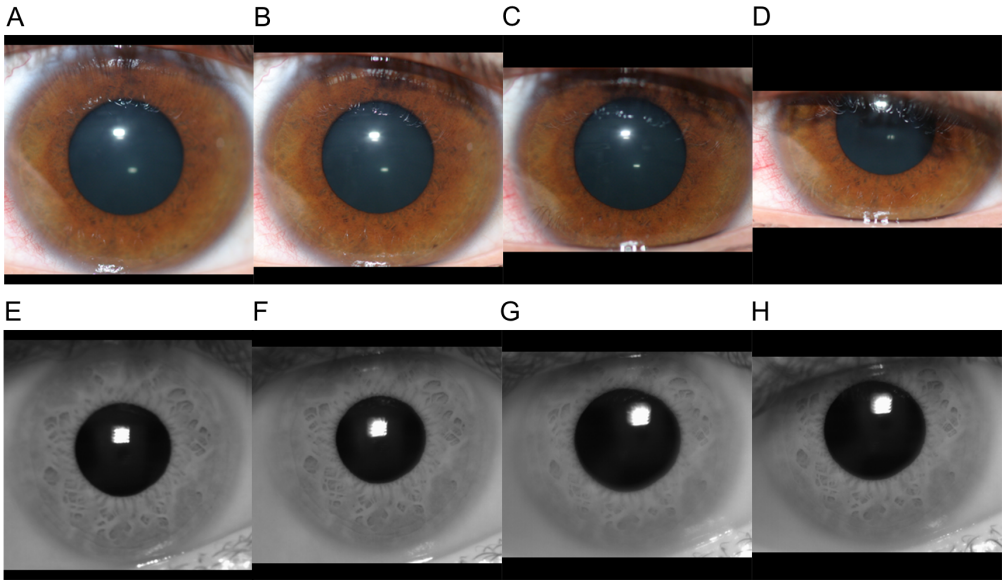

Fig 10.Different degrees of padding processing effects on the extracted non-holistic iris images. Comparing the different degrees of image processing effects with the color (Fig 10A-10D) and gray iris images (Fig 10E-10H). The *left*-to-*right* images represent the padding effect sorted the non-holistic iris by size, from smallest to largest in padding area.

The results in Fig 10 show that the effect of experimental images, demonstrating the effectiveness and practicality of the proposed Mask R-CNN architecture. Different noise factors can also be distinguished such as multiple scales, color distortion, and insufficient light. Among the most important is that one eye is open while the other eye is half-open or closed, which affects the performance and stability in the biometric identification process. Adding the zero-padding layer in the architecture allows efficient learning of the scale-variant features from each iris training set and enhances the robustness of the iris authentication system. The proposed Mask R-CNN also

uses a multi-tasks loss function to calculate the loss, combining the loss of classification, localization and extraction mask. The equation is defined as below.

$$Loss = L_{cls} + L_{box} + L_{mask}$$

The formula consists of Loss functions in each ROI region: classification loss and position regression loss of the bounding box. They were inherited from Faster R-CNN and the Loss of the mask is proposed by [38]. During our experiments, the number of classification classes is set to 1 since this is the single-class classification task. We change the non-maximum suppression of RPN to 120 (from a default value of 200). The maximum number of ground truth instances and final detections are adjusted to 70 and 50, respectively. We also train the model with the anchor scale (16, 32, 64, 128, 256) and it performs worse than the original setting of (32, 64, 128, 256, 512). Finally, the learning rate was set at 0.001 and the momentum rate at 0.9. Given this, the total Loss drops to 0.0156%. With the improvement of Loss weights(1,2,1,2,3), Loss is decreased by 0.0143%.While training the model, the Loss evidently dropped to nearly 0.01% and to the lowest of 0.0066% at the 203rd and 233rd of total 250 epochs, respectively. Our proposed Mask R-CNN architecture correctly identified 1539 of 1540 iris images in Utiris datasets; the one failure is due to the improper capture of partial iris images. Finally, the precision results obtained with iris detection is more than 99.99%.

This section summarizes our Mask R-CNN architecture, compare to the [16] [18] and [19] works, we proposed the stronger and more robustness of detecting the iris location and the extracting the precise region of the iris region, are presented. A baseline dataset has been completed that serves as a basis to measure evaluation targets. Next, we comprehensively validate the performance of the proposed framework by using a pre-processed iris dataset.

4.2. The Evaluation of Fine-tuned Mobile Inception V4 Architecture

In the authentication phase, our framework was conducted on UTiris datasets and the performance of the fine-tuned mobile Inception V4 is analyzed. The total size of the training set is 1540 images, which were randomly sorted in the memory. The learning rate of the whole network was initialized to 1e-4. The testing iteration was set to 17 epochs on the mini batches of eight images

during the training, using the AMSGrad optimization algorithm. All of these super-parameters were set properly so that the network would be able to generalize well. Simultaneously, these factors could be used to improve the performance and accuracy of our iris authentication systems.

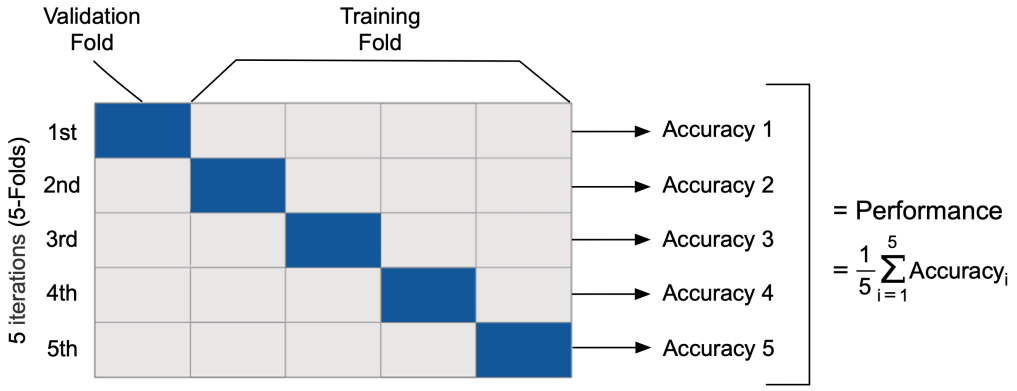

Fig 11. The 5-fold cross validation

The cross-validation strategy was employed to verify the performance. As illustrated in Fig 11, we applied *Five fold cross validation* [62] to evaluate the performance of our proposed mobile Inception V4 on the Nvidia Jetson Nano. Every sub dataset of UTiris were evaluated, and the accuracy rate verified the measurement of the recognition learning model.

The final curves of validation accuracy and loss were used to evaluate the proposed framework, as displayed in Fig 12. Through the training of the proposed learning model with 17 epochs, all experimental results are visualized by the Tensorboard tool. Fig 12 (A) and Fig 12 (C) illustrate that the accuracy is proportional to the increasing epochs and maintains a steady accuracy after the 12th epoch. In each cross-validation, we chose the highest accuracy as the benchmark to represent the best performance. The results are presented in Fig 12(A) which manifests that while mobile inception V4 is applied on the VW session, the best result was achieved by the fine-tuned method with an accuracy of 99.37%, peaking at a 100% recognition rate in the 5th cross-validation of the 11th epoch on the NIR session. Here, the difference between the average accuracy of the proposed framework with VW and NIR was 0.3%. Gradually, with the help of optimization functions, the Loss function learned to reduce the error in prediction. The results showed

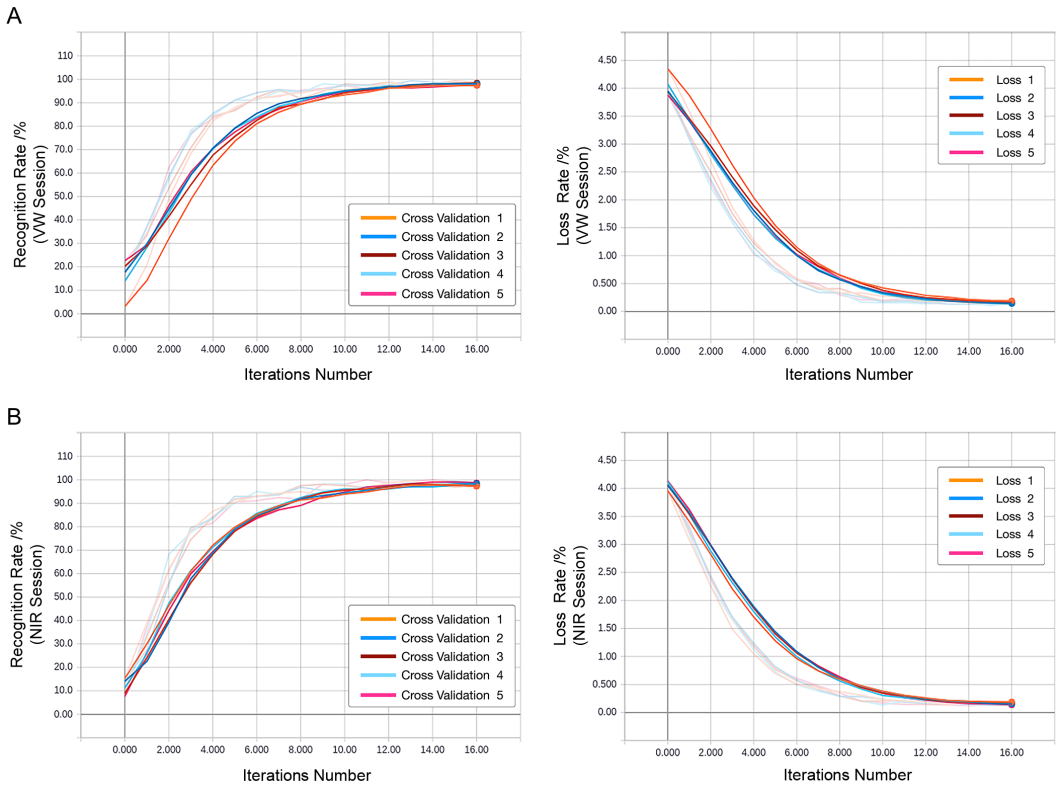

Fig 12. The curve plot for the experimental result of 5-fold cross validation. The X-axis displays the number of epochs ranging from *0* to *16*, and the Y-axis displays the validation accuracy from *0* to *110*. Curve plot for the categorical accuracy for *17* epochs. The accuracy was defined as the total frequency of the correct acceptance and the imitators with correct rejection.

---

that the average loss rate is decreased by roughly 4%, from 4.05% to 0.14%. We found that increasing the number of training iterations could significantly improve the proposed learning framework performance.

5. Discussion

In this study, an iris authentication system based on Nvidia Jetson Nano was introduced. The proposed system consists of two distinct functions: iris detection, non-holistic iris extraction and iris recognition. We consider the iris database, which consists of 2 classes. The dataset is split into training data and testing data sets and organized into 2 folders: train and test. The 5- fold cross-validations are calculated with a batch size of 8 for 17 epochs, demonstrating that the fine-tuned mobile Inception V4 model was steady in every validation phase, as explained in Tables 1 and 2.

Table 1: The table presents the intercorrelations among the 5-fold cross- validations on VW session.

Fine-tuned Inception V4	Epoch 0	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10	Epoch 11	Epoch 12	Epoch 13	Epoch 14	Epoch 15	Epoch 16
Cross Validation 1	3.163	20.89	49.37	68.35	82.28	87.97	91.77	93.04	94.30	95.57	95.57	96.20	98.71	96.84	98.10	97.47	97.47
Validation Loss	4.347	3.584	2.691	1.865	1.247	0.8529	0.5879	0.4328	0.3355	0.3285	0.2723	0.2568	0.1840	0.2100	0.1538	0.1724	0.1787
Cross Validation 2	17.72	37.34	58.86	76.58	85.44	91.14	94.30	95.57	94.94	95.57	97.47	97.47	97.47	98.37	98.73	98.10	98.10
Validation Loss	3.947	3.095	2.354	1.652	1.146	0.7699	0.4740	0.3398	0.3281	0.2530	0.1713	0.1869	0.1559	0.1331	0.1291	0.1373	0.1269
Cross Validation 3	20.25	34.18	53.80	70.89	84.18	86.71	92.41	95.57	91.77	94.94	98.10	97.47	98.73	97.47	98.10	99.37	98.73
Validation Loss	3.940	3.164	2.489	1.753	1.217	0.8743	0.5731	0.4074	0.4083	0.2827	0.1922	0.1877	0.1621	0.1836	0.1559	0.1482	0.1365
Cross Validation 4	14.01	36.93	57.32	78.34	84.71	91.08	91.08	94.90	94.27	98.09	97.45	97.45	98.73	96.82	97.45	98.73	98.73
Validation Loss	4.073	3.067	2.219	1.596	1.054	0.7190	0.5629	0.3877	0.3307	0.1651	0.1555	0.1583	0.1314	0.1817	0.1509	0.1040	0.1358
Cross Validation 5	22.67	33.33	62.67	77.33	83.33	87.33	92.00	92.67	95.33	96.00	97.33	96.00	98.67	96.00	97.33	98.00	98.00
Validation Loss	3.879	3.133	2.200	1.593	1.031	0.7751	0.5743	0.4867	0.2955	0.2077	0.1971	0.2256	0.1518	0.1867	0.1798	0.1435	0.1333

Table notes that all the accuracy rates/% and loss rates/% of 17 epochs measures on the proposed framework in columns, and the rows show five iterations for each k cross-validation(k=5).

Considering the results in Tables 1 and 2, we can observe that increasing the number of training epochs can slightly improve the recognition rate. In the initial training phase, the recognition accuracy of the model in both sessions is relatively low. This is because the model needs to learn from the training data, and the sample of training data has not fitted the proposed model. On the whole, the average accuracy of VW sessions (15.56%) is higher than the NIR session (11.43%) in the first epoch since the color images contain a large

Table 2: The table presents the intercorrelations among the 5-fold cross- validations on NIR session.

Fine-tun Mobile Incep	Epoch 0	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10	Epoch 11	Epoch 12	Epoch 13	Epoch 14	Epoch 15	Epoch 16
Cross Validation 1	15.29	38.85	61.78	78.34	86.62	90.45	92.99	94.27	94.90	93.63	96.18	96.18	98.73	99.36	98.09	98.09	96.18
Validation Loss	3.958	3.083	2.254	1.488	1.046	0.6934	0.5036	0.4207	0.3747	0.3080	0.2304	0.2041	0.1837	0.1500	0.1783	0.1879	0.1806
Cross Validation 2	14.10	27.56	55.13	79.49	83.33	92.95	92.95	93.95	96.79	95.51	96.15	96.79	97.44	98.72	98.08	99.36	98.72
Validation Loss	4.061	3.282	2.411	1.710	1.236	0.8172	0.5813	0.4011	0.2987	0.2820	0.2378	0.1842	0.1670	0.1650	0.1443	0.1249	0.1407
Cross Validation 3	8.91	33.12	56.05	74.52	84.08	91.72	92.99	93.63	97.45	98.09	97.45	96.82	98.73	99.36	98.09	96.82	97.45
Validation Loss	4.061	3.237	2.437	1.695	1.179	0.7826	0.5643	0.4510	0.2988	0.2027	0.2073	0.2127	0.1461	0.1474	0.1594	0.1854	0.1444
Cross Validation 4	11.25	34.38	68.13	77.50	84.38	90.00	95.00	93.75	97.50	98.13	98.13	95.63	98.75	96.88	96.88	98.75	98.13
Validation Loss	4.124	3.183	2.256	1.657	1.125	0.7234	0.4915	0.3818	0.2901	0.2054	0.1326	0.2061	0.1663	0.2056	0.1669	0.1421	0.1258
Cross Validation 5	7.59	37.34	61.39	78.48	81.65	95.51	91.14	92.41	91.77	98.10	98.10	100	98.73	99.37	100	99.37	98.10
Validation Loss	4.134	3.319	2.374	1.681	1.192	0.7588	0.6120	0.4739	0.3495	0.2026	0.1707	0.1466	0.1456	0.1354	0.1218	0.1355	0.1128

Table notes that all the accuracy rates/% and loss rates/% of 17 epochs measures on the proposed framework in columns, and the rows show five iterations for each k cross-validation(k=5).

amount of different discrete information in three channels. In the same eighth epoch, the curves (in Fig.12) show that the performance of the model on the NIR session (93.01%) is significantly beyond that of the VW session (92.31%), this reveals that our proposed learning model is easier and better to fit the NIR session data set rather than VW session. Based on the comparative analysis of the two models, this research reveals that the convergence speed of our fine-tuned Inception V4 processing on the NIR session is better than the VW session in overall epochs.

Concurrently, we observe that the validation accuracy fluctuates after the 10th epoch. The reason for this is that the amount of data in the experimental data set is still relatively small, and training on the model with more layers may bring some problems such as overfitting due to the disparate sizes of pre-processed iris images and of the effective recognition area within the range. Under the mobile environment, this seems inevitable. To solve this problem, we used the early stopping method; calculating the accuracy of validation data at the end of each epoch, and stopping training when the accuracy was no longer improved.

In the fine-tuned mobile Inception V4 architecture, we propose a module of three adaptive average pooling layers with various properties and sizes. This module emphasizes the down sampling of the overall feature information, mainly including two positive roles: 1) reduction of parameters is reflected more in the extraction of higher-order information. 2) more useful discriminative information is passed to the next layer for feature extraction while reducing the dimension. These changes effectively speed up the fitting of depth network and data and decrease considerable computing costs so that our fine-tuned model can achieve successive and ideal recognition rates in the first few epochs. Fig 9 illustrates an example of the stacked pooling, with a kernel set of K ={3, 2, 1} and a stride of 1. In empirical studies, this configuration shows the best performance in most cases. During our observation, we found that the combination of average pooling layers can significantly maintain the recognition rate, and efficiently improve the convergence speed of the proposed model by stacking pooling layers.

We also investigate how to train a fine-tuned CNN to classify non-holistic iris images with high accuracy. The hyperparameters of the learning rate are closely associated with model performance. Weight decay can be effective at preventing the problem of over-fitting. It controls the size of network parameters updated after each iteration. Our experiment shows that when the learning rate is modulated by 1e-4, the proposed framework has a notable recognition rate of 100% in the 5th verification set of the VW session. Comparatively, the accuracy rate of assessment is over 99% in the NIR session. Likewise, we also try to use the learning rate of 1e-3 and 1e-5; however, the higher learning rate can accelerate the model learning speed, but it was easily led to the gradient explosion of loss value and the fluctuation of the accuracy and loss rate. Conversely, the lower learning rate can cause the slow learn speed of the model, which easily leads to overfitting, and it was difficult for the learning model to converge.

Lastly, compared to preprocessing and recognition approaches used by authors [16] [18] [19] in Table 3, we give the framework more agility and flexible space to adapt to the change of system, and research demonstrates the effectiveness of the proposed framework. Starting with good quality images, we do not adopt any degradation operation in our preprocessing phase. Our Mask R-CNN architecture is reconstructed to implement robust detection and scale-variant feature extraction, respectively. The model has the advantage that image space is unaffected by the image degradation process. Furthermore, our proposed recognition model always produces

Table 3: Comparing the original results obtained from experiments with the state-of-the-art methodologies. The best results in bold.

References	Preprocessing	Localization Precision	Methodology	Recognition Accuracy /%	
Bhagyashree Deshpande and Deepak Jayaswal [16]	Daugman's integro-differential + Daugman's Rubber Sheet + 1D log Gabor filter	×	Hamming Distance	**95.00%** (VW)	
Mohammed Hamzah Abed [18]	Circular Hough transform + Haar wavelet transform + PCA	98.73%	Cosine Distance	**91.14%** (NIR)	
Onkar Kaudki and Kishor Bhurchandi [19]	Circular Hough Transform + Rubber-Sheet Unwrapping + Haar wavelet transform	96.21%	Hamming Distance	**97.00%** (NIR)	
Proposed Frameworks	Optimized Mask R-CNN	99.99%	Fine-tuned Inception V4	**Accuracy/%**	
			Session	VW	NIR
			CV 1	**98.71%**	**98.73%**
			CV 2	**98.73%**	**99.36%**
			CV 3	**99.37%**	**99.36%**
			CV 4	**98.73%**	**98.75%**
			CV 5	**98.67%**	**100%**
			Average accuracy	**98.84%**	**99.24%**
			Overall accuracy	**99.04%**	

top performance for all levels of methodologies, demonstrating the robust adaptation and excellent performance, and the highest performance in iris detection. Our results demonstrate that our framework provides the best

average recognition accuracy of 98.84% and 99.24% for the VW and NIR session, respectively. The overall accuracy of the model in the UTiris dataset is 99.04%, are summarized as follows.

As a result, we propose an ensemble learning system of two learning models for the personal iris authorization. Among them, the fined tuned-based Inception V4 architecture to classify and verify the iris, which is validated by 5-fold cross validation method. With the improvements, the proposed Inception V4 can efficiently improve the convergence of the learning model under the prerequisite of high accuracy guaranteed. When processing the NIR images, the proposed classification learning model can process the grey-scale image by stacking the same dimensions. In the aspect of robustness, sections of the research are implemented by multiple different functional components [16] [18] [19], and this would increase the system complexity and coupling in the iris verification process. To address this, the proposed Mask R-CNN can preform multi-tasks learning, which simultaneously detect, locate and extract non-holistic iris feature region. Considering the diversity of iris samples, its architecture can also apply the local iris ground truth to achieve robust iris localization and extraction. In addition, the proposed zero-padding layer can flexibly normalize the different scale-variant iris region feature images. Our experimental results and observations for the proposed framework indicate that the performance outperforms the current state-of-the-art methodologies in detection capability, recognition rate, and robustness of the system.

6. Conclusions

A comprehensive overview of mobile environment devices and the implementation of an non-holistic iris authentication system is presented, and the merits and drawbacks of the methods used in the state of art approaches are analyzed. The current investigations showed that the method of deep learning is moving towards a complex hierarchical structure. We propose a learning framework with multi-tasks for detection, extraction, normalization and recognition for the high-resolution iris images in our iris authentication system. Thus, the whole framework demonstrates dynamic and flexible characteristics. All of these improvements significantly improve the practicability of the proposed system in the actual scene.

In consequence, qualitative and quantitative research designs were adopted to provide experimental results which endeavor to explain the scale-variant

features of models of learning through iris recognition. The proposed solutions are suitable for high performance built-in GPU mobile devices and aid researchers in the estimation of analysis results for further research in the mobile environment.

Acknowledgments

We thank the support of the Multimedia Department of Computer Science and Information Technology at the University of Putra Malaysia, and the laboratory of Computer-Assisted Surgery and Diagnostic (CASD). This research was undertaken at CASD, and the support of CASD is gratefully acknowledged. I would like to thank all the contributors to this study for their prompt responses to different opinions, their continued support and encouragement and the vibrant research atmosphere that they have provided.

References

1. Folorunso C. O., Asaolu O. S. & Popoola O. P. A Review of Voice-Base Person Identification: State-of-the-Art. Covenant Journal of Engineering Technology (CJET) Vol.3 No.1, June 2019 ISSN: p 2682-5317 e 2682-5325. https://doi.org/10.20370/2cdk-7y54

2. Hui, D. O. Y., Yuen, K. K., Zahor, B. A. F. B. S. M., Wei, K. L. C., & Zaaba, Z. F. (2018). An assessment of user authentication methods in mobile phones. https://doi.org/10.1063/1.5055518

3. Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. Proceedings of the IEEE, 1–20. https://doi.org/10.1109/jproc.2019.2921977

4. Noruzi, A., Mahlouji, M. & Shahidinejad, A. Artif Intell Rev (2019). https://doi.org/10.1007/s10462-019-09776-7

5. Deng, Y. (2019). Deep learning on mobile devices: a review. In Mobile Multimedia/Image Processing, Security, and Applications 2019, vol. 10993, 109930A, International Society for Optics and Photonics.

6. ARROWSNX F-04G (2015). http://www.fujitsu.com/global/about/resources/news/press-releases/2015/0525-01.html Accessed 15 October 2016.

7. HUAWEI P30 Pro (2019). Accessed October 2019. https://consumer.huawei.com/en/phones/p30/specs/

8. NVIDIA. Tegra K1 Next-Get Mobile Processor., 2014. http://www.nvidia.com/object/tegra-k1- processor.html Accessed Ovtober 2019.

9. Hess, Christopher David, "Design of an embedded iris recognition system for use with a multi-factor authentication system." (2019). Electrical Engineering Undergraduate Honors Theses. 60. https://scholarworks.uark.edu/eleguht/60

10. Gorodnichy, D. O., & Chumakov, M. P. (2019). Analysis of the effect of ageing, age, and other factors on iris recognition performance using NEXUS scores dataset. IET Biometrics, 8(1), 29–39. https://doi.org/10.1049/iet-bmt.2018.5105

11. M. M. Khaladkar and S. R. Ganorkar, "A Novel Approach for Iris Recognition," vol. 1, no. 4, 2012.

12. J. Daugman and C. Downing, Epigenetic randomness, complexity and singularity of human iris patterns, Proceedings of the Royal Society of London B: Biological Sciences, no. December 2000, pp. 1737–1740, 2001.

13. N. Ahmadi, M. Nilashi, Iris texture recognition based on multilevel 2-D haar wavelet decomposition and hamming distance approach, J. Soft Comput. Dec. Support Sys. 5 (3) (2018) 16–20.

14. N.Y. Tay, K. M. Mok, A review of iris recognition algorithms in information technology international symposium on vol. 2, pp. 1-7 Aug 2008.

15. Samant, P., Agarwal, R., & Bansal, A. (2017). Enhanced discrete cosine transformation feature based iris recognition using various scanning techniques. 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON). https://doi.org/10.1109/upcon.2017.8251128

16. Deshpande, B., & Jayaswal, D. (2018). Fast and Reliable Biometric Verification System Using Iris. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). https://doi.org/10.1109/icicct.2018.8473300

17. J. G. Daugman, "How iris recognition works," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1, pp. 21–30, Jan. 2004.

18. Mohammed Hamzah Abed. Iris recognition model based on Principal Component analysis and 2 level Haar wavelet transform: Case study CUHK and UTIRIS iris databases. Journal of College of Education/Wasit, VL-27, 2017.

19. Kaudki Onkar & Bhurchandi Kishor. (2018). A Robust Iris Recognition Approach Using Fuzzy Edge Processing Technique. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). https://doi.org/10.1109/icccnt.2018.8493855

20. K. W. Bowyer, S. E. Baker, A. Hentz, K. Hollingsworth, T. Peters, and P. J. Flynn, "Factors that degrade the match distribution in iris biometrics," Identity in the Information Society, vol. 2, no. 3, pp. 327–343, 2009.

21. Bowyer, K. W., Baker, S. E., Hentz, A., Hollingsworth, K., Peters, T., & Flynn, P. J. (2009). Factors that degrade the match distribution in iris biometrics. Identity in the Information Society, 2(3), 327–343. https://doi.org/10.1007/s12394-009-0037-z

22. E. Garea, J.M. Colores, M.S. García , L.M. Zamudio , A .A . Ramírez, Cross-sensor Iris verification applying robust fused segmentation algorithms, in: IEEE. Pro- ceedings of International Conference on Biometrics. ICB 2015, 2015, pp. 17–22.

23. Llano, E. G., García Vázquez, M. S., Vargas, J. M. C., Fuentes, L. M. Z., & Ramírez Acosta, A. A. (2018). Optimized robust multi-sensor scheme for simultaneous video and image iris recognition. Pattern Recognition Letters, 101, 44–51. https://doi.org/10.1016/j.patrec.2017.11.012

24. Sunil Chawla and Aashish Oberoi, "A Robust Algorithm for Iris Segmentation and Normalization using Hough Transform," Global Journal of Business Management and Information Technology, Volume 1, Number 2 (2011), pp. 69-76, 2011.

25. Anand Deshpande, Prashant Patavardhan, "Segmentation And Quality Analysis Of Long Range Captured Iris Image," ICTACT Journal on Image and Video Processing, 2016.

26. Reddy, N., Rattani, A., & Derakhshani, R. (2016). A robust scheme for iris segmentation in mobile environment. 2016 IEEE Symposium on Technologies for Homeland Security (HST). https://doi.org/10.1109/ths.2016.7568948

27. Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-aware attention model for fine-grained representation learning. In ACMMM, 2018 https://arxiv.org/abs/1901.00392

28. Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In CVPR, pages 4438–4446, 2017. https://doi.org/10.1109/CVPR.2017.476

29. Jianyuan Guo, Yuhui Yuan, Lang Huang, Chao Zhang, JinGe Yao, and Kai Han. Beyond human parts: Dual partaligned representations for person re-identification. In the IEEE International Conference on Computer Vision (ICCV), October 2019. https://arxiv.org/abs/1910.10111

30. Proenca H, Neves JC (2019) Segmentation-less and non-holistic deep-learning frameworks for iris recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 0–0 https://doi.org/10.1109/CVPRW.2019.00283

31. Nguyen, K., Fookes, C., Ross, A., Sridharan, S. (2018). Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective. IEEE Access, 6, 18848–18855 https://doi.org/10.1109/ACCESS.2017.2784352

32. Arsalan, M.; Hong, H.G.; Naqvi, R.A.; Lee, M.B.; Kim, M.C.; Kim, D.S.; Kim, C.S.; Park, K.R. Deep Learning-Based Iris Segmentation for Iris Recognition in Visible Light Environment. Symmetry 2017, 9, 263.

33. Lamiaa A. Elrefaei, Doaa H. Hamid, Afnan A. Bayazed1 & Sara S. Bushnak1 and Shaikhah Y. Maasher. "Developing Iris Recognition System for Smartphone Security," Springer, July 2017.

34. Hajari, K., & Bhoyar, K. (2015). A review of issues and challenges in designing Iris recognition Systems for noisy imaging environment. 2015 International Conference on Pervasive Computing (ICPC). https://doi.org/10.1109/pervasive.2015.7087003

35. Yooyoung Lee, Ross J. Micheals, James J. Filliben and P. Jonathon Phillips, "Vasir: An Open-Source Research Platform for Advanced Iris Recognition Technologies", Journal of Research of the National Institute of Standards and Technology, Vol. 118, pp. 218-259, 2013.

36. Zhaofeng He, Tieniu Tan, Zhenan Sun and Xianchao Qiu, "Robust Eyelid, Eyelash and Shadow Localization for Iris Recognition", Proceedings of 15th IEEE International Conference on Image Processing, pp. 265-268, 2008.

37. Kevin W. Bowyer, Karen Hollingsworth and Patrick J. Flynn, "Image Understanding for Iris Biometrics: A Survey", Computer Vision and Image Understanding, Vol. 110, No. 2, pp. 281-307, 2008.

38. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN.The IEEE. International Conference on Computer Vision (ICCV), 2017, pp. 2961-2969.

39. Ross Girshick; The IEEE International Conference on Computer Vision (ICCV), 2015, Fast R-CNN. pp. 1440-1448.

40. [J] Mahdi S. Hosseini, Babak N. Araabi and H. Soltanian-Zadeh, Pigment Melanin: Pattern for Iris Recognition, IEEE Transactions on Instrumentation and Measurement, vol.59, no.4, pp.792-804, April 2010. https://doi.org/10.6084/m9.figshare.10279592

41. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In European Conference on Computer Vision, pp. 740–755, 2014.

42. Aginako, N., Castrillón-Santana, M., Lorenzo-Navarro, J., Martínez-Otzeta, J. M., & Sierra, B. (2017). Periocular and iris local descriptors for identity verification in mobile applications. Pattern Recognition Letters, 91, 52–59. https://doi.org/10.1016/j.patrec.2017.01.021

43. Padole, C. N., & Proenca, H. (2012). Periocular recognition: Analysis of performance degradation factors. 2012 5th IAPR International Conference on Biometrics (ICB). https://doi.org/10.1109/icb.2012.6199790

44. Anis Farihan Mat Raffei, Tole Sutikno, Hishammuddin Asmuni, Rohayanti Hassan, Razib M Othman, Shahreen Kasim, Munawar A Riyadi. Fusion Iris and Periocular Recognitions in Non-Cooperative Environment. IJEEI, 2019. https://doi.org/10.11591/ijeei.v7i3.1147

45. Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285, 2016.

46. Montanari L, Basu B, Spagnoli A, Broderick BM. A padding method to reduce edge effects for enhanced damage identification using wavelet analysis. Mech Syst Signal Pr. 2015;52-53:264-277.

47. Tang X., Xie J., Li P. (2017) Deep Convolutional Features for Iris Recognition. In: Zhou J. et al. (eds) Biometric Recognition. CCBR 2017. Lecture Notes in Computer Science, vol 10568. Springer, Cham.

48. Ribeiro, E., Uhl, A., & Alonso-Fernandez, F. (2018). Iris Super-Resolution using CNNs: is Photo-Realism Important to Iris Recognition? IET Biometrics. https://doi.org/10.1049/iet-bmt.2018.5146

49. Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. 2016.

50. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In CVPR, 2016:2818-2826.

51. Siyu Huang, Xi Li, Zhiqi Cheng, Zhongfei Zhang, and Alexander G. Hauptmann. Stacked pooling: Improving crowd counting by boosting scale invariance. CoRR, abs/1808.07456, 2018.

52. Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y., & Belongie, S. (2017). Kernel Pooling for Convolutional Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.325

53. Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). https://doi.org/10.1109/wacv.2017.58

54. Ruder, S. An overview of gradient descent optimization algorithms. CoRR, abs/1609.04747, 2016.

55. Wilson, A. C, Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. arXiv preprint arXiv:1705.08292, 2017.

56. I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101, 2017.

57. Reddi, Sashank J, Kale, Satyen, and Kumar, Sanjiv. On the convergence of adam and beyond. In International Conference on Learning Representations (ICLR), 2018.

58. Kohavi, R., et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI, volume 14.

59. Yadav, S., & Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. 2016 IEEE 6th International Conference on Advanced Computing (IACC). https://doi.org/10.1109/iacc.2016.25

60. N. Corp., "Jetson nano," developer.nvidia.com/embedded/buy/jetsonnano-devkit, 2019, [Online; accessed 09/27/19].

61. Ramyad Hadidi, Jiashen Cao, Yilun Xie, Bahar Asgari, Tushar Krishna, Hyesoon Kim. Characterizing the Deployment of Deep Neural Networks on Commercial Edge Devices. IEEE International Symposium on Workload Characterization (IISWC), Orlando, Florida (2019).

62. Wong, T.T.; Yeh, P.Y. Reliable Accuracy Estimates from k-fold Cross Validation. IEEE Trans. Knowl. Data Eng. 2019.

Turnitin Report

by Siming Zheng

Submission date: 23-Aug-2020 10:25AM (UTC-0400)
Submission ID: 1372888998
File name: Learning Scale-variant Features for Non-holistic Iris Authentication with Robust Deep Ensemble Learning Framework (18.05M)
Word count: 8975
Character count: 46815

Turnitin Report

ORIGINALITY REPORT

5%	1%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, Alexander Hauptmann. "Stacked Pooling for Boosting Scale Invariance of Crowd Counting", ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020 Publication	1%
2	essayshark.com Internet Source	<1%
3	Bhagyashree Deshpande, Deepak Jayaswal. "Fast and Reliable Biometric Verification System Using Iris", 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018 Publication	<1%
4	Hugo Proenca, Joao C. Neves. "Segmentation-Less and Non-Holistic Deep-Learning Frameworks for Iris Recognition", 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),	<1%

2019
Publication

5	"Computer Vision – ECCV 2014", Springer Nature, 2014 Publication	<1%
6	Submitted to Monash University Student Paper	<1%
7	wiyn.org Internet Source	<1%
8	Submitted to Queen Mary and Westfield College Student Paper	<1%
9	Fujia Wei, Gang Yao, Yang Yang, Yujia Sun. "Instance-level recognition and quantification for concrete surface bughole based on deep learning", Automation in Construction, 2019 Publication	<1%
10	"Intelligent Computing Theories and Application", Springer Science and Business Media LLC, 2019 Publication	<1%
11	Hanlin Tan, Huaxin Xiao, Xiaoyu Zhang, Bin Dai, Shiming Lai, Yu Liu, Maojun Zhang. "MSBA: Multiple Scales, Branches and Attention Network With Bag of Tricks for Person Re-Identification", IEEE Access, 2020 Publication	<1%

12	"Advances in Machine Learning and Computational Intelligence", Springer Science and Business Media LLC, 2021 Publication	<1%
13	pytorch.org Internet Source	<1%
14	Submitted to University of Strathclyde Student Paper	<1%
15	Zhiming Hu, Ahmad Bisher Tarakji, Vishal Raheja, Caleb Phillips, Teng Wang, Iqbal Mohomed. "DeepHome", The 3rd International Workshop on Deep Learning for Mobile Systems and Applications - EMDL '19, 2019 Publication	<1%
16	Submitted to University College London Student Paper	<1%
17	www.mdpi.com Internet Source	<1%
18	R. T. Al-Zubi, D. I. Abu-Al-Nadi. "Automated personal identification system based on human iris analysis", Pattern Analysis and Applications, 2006 Publication	<1%
19	Muhammad Zaigham Zaheer, Jin-Ha Lee, Marcella Astrid, Seung-Ik Lee. "Old Is Gold:	<1%

Redefining the Adversarially Learned One-Class Classifier Training Paradigm", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020
Publication

---

20 Xiaohong Gao, Barbara Braden, Stephen Taylor, Wei Pang. "Towards Real-Time Detection of Squamous Pre-Cancers from Oesophageal Endoscopic Videos", 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019 <1%
Publication

---

21 Chunsheng Guo, Ruizhe Li, Meng Yang, Xianghong Tang. "Deep neural network with FGL for small dataset classification", IET Image Processing, 2019 <1%
Publication

---

22 Chi Xu, Wendi Cai, Yongbo Li, Jun Zhou, Longsheng Wei. "Accurate Hand Detection from Single-Color Images by Reconstructing Hand Appearances", Sensors, 2019 <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		