A Face Recognition Algorithm Based on Dual-Channel Images and VGG-cut Model

A Face Recognition Algorithm Based on Dual-Channel Images and VGG-cut Model

Dongchu Su[1], Yong Li[1*], Yunping Zhao[1], Rui Xu[1], Bo Yuan[1] and Wenjuan Wu[1]

[1]College of Computer Science, National University of Defense Technology, Changsha 410073, China

[*]Corresponding author's e-mail: yongli@nudt.edu.cn

Abstract. In the embedded system environment, both large amount of face image data and the slow recognition process speed are the main problem facing face recognition of end devices. This paper proposes a face recognition algorithm based on dual-channel images and adopts a cropped VGG-like model referred as VGG-cut model for predicting. The training set uses the same single-layer images of the same person combined into dual channels as a positive example, and single-layer images of different people are combined into dual channels as a negative example. After model training, the trained face verification model is used as the basis of face recognition, and the final recognition result can be obtained through loop matching and similarity ranking. The experimental results on a RISC-V embedded FPGA platform show that compared with the ResNet model called in Dlib, the VGG-like model and MobileFaceNet trained by Keras, our algorithm is increased by 240×, 88×, and 19× in recognition speed, respectively without significant accuracy reduction.

1. Introduction

In recent years, with the development of embedded software and hardware platforms, various end devices can run more applications at a smaller cost. Also, digital image recognition [1], face recognition [2], face verification [3] and other applications that used to be expensive have developed rapidly, which greatly improves the market prospects of low-power embedded applications.

In face recognition, verifying whether two pictures are the same person is the key of face recognition technology, which is also called face verification. The main methods for face verification include classifiers [4], feature point extraction [5], and Siamese network [6] or Triplet network [7].

Previous work [8-10] uses the output of the bottleneck layer as the representation of face feature, but the disadvantage of this method is that the output dimension is too large, often exceeding 1000 dimensions. Some recent work [8] uses the PCA method to reduce the output dimension of the bottleneck layer. But the feature representation of the face does not need to be so complicated. So the FaceNet [11] uses a 128-point output feature vector for representation of face, but it still needs a three-channel image as input, and it also needs to compare the Euclidean distance between two 128-point feature vectors after passing image into the model twice.

The above algorithms have their own advantages, but they all have the disadvantages of high computational and power consumption.

Based on the above shortcomings, this paper proposes a face recognition algorithm based on dual-channel images, which can effectively reduce the amount of data of training sets and model size, improve the recognition speed of the algorithm and reduce computational and power consumption. The experimental results show that compared with the existing embedded face recognition algorithm, the

algorithm adopted in this paper has certain advantages in performance and power consumption, and has certain research value and practical significance.

2. Recognition Algorithm

Facing the problem of large amount of calculation and high cost of existing embedded face recognition algorithms, our algorithm firstly converts a three-channel image to a single-channel grayscale image, thereby reducing the amount of input data so as to reduce calculation in the training and recognition process. In the training process, the two-channel image combined by different images of the same person is set as a positive example, and the two-channel image combined by different people is set as a negative example. According to the proportion of the target face in the image library to be recognized, we generate a training set with a specific proportional load, which can effectively improve the accuracy of recognition. In addition, in order to reduce the size of the recognition model, we have tailored the VGG-like model to effectively reduce the amount of training parameters and recognition calculations while keeping the accuracy basically unchanged.

2.1. Algorithm steps

Take two different people A and B in the data set as an example to generate positive and negative examples of the training set.

•Step 1: Perform a face detection on the face image of A, and obtain a rectangular frame containing the face part as the output picture F_1;

•Step 2: Resize and grayscale F_1 to obtain a single-channel picture F_2;

•Step 3: Repeat steps 1 and 2 for different faces of A to get F_3;

•Step 4: Combine F_2 and F_3 into dual-channel images, and set label to 1, to get a positive example;

•Step 5: Repeat steps 1 and 2 for B to get F_4;

•Step 6: Combine F_2 and F_4 into dual-channel images, and set label to 0 to get a negative example;

•Step 7: Adjust the ratio of positive and negative examples in the training set according to face database to be verified, assuming 1:9, that is, the ratio of the number of positive and negative examples is 1/9;

•Step 8, perform shuffle operation on the training set, perform training to obtain a VGG-cut model called M;

•Step 9. Perform steps 1 and 2 respectively on the target picture T_1 and a face picture F_5 in the face database to obtain a dual-channel picture F_6;

•Step 10, input F_6 into the model, and feed forward to obtain the similarity result R_1;

•Step 11. Repeat steps 9 and 10 with a certain strategy to obtain n similarity results R_1, R_2 ... R_n;

•Step 12, sort the similarity results, the pair of pictures with the highest similarity are to be identified as the same person.

The training diagram for VGG-cut model in our algorithm is shown in Fig.1.

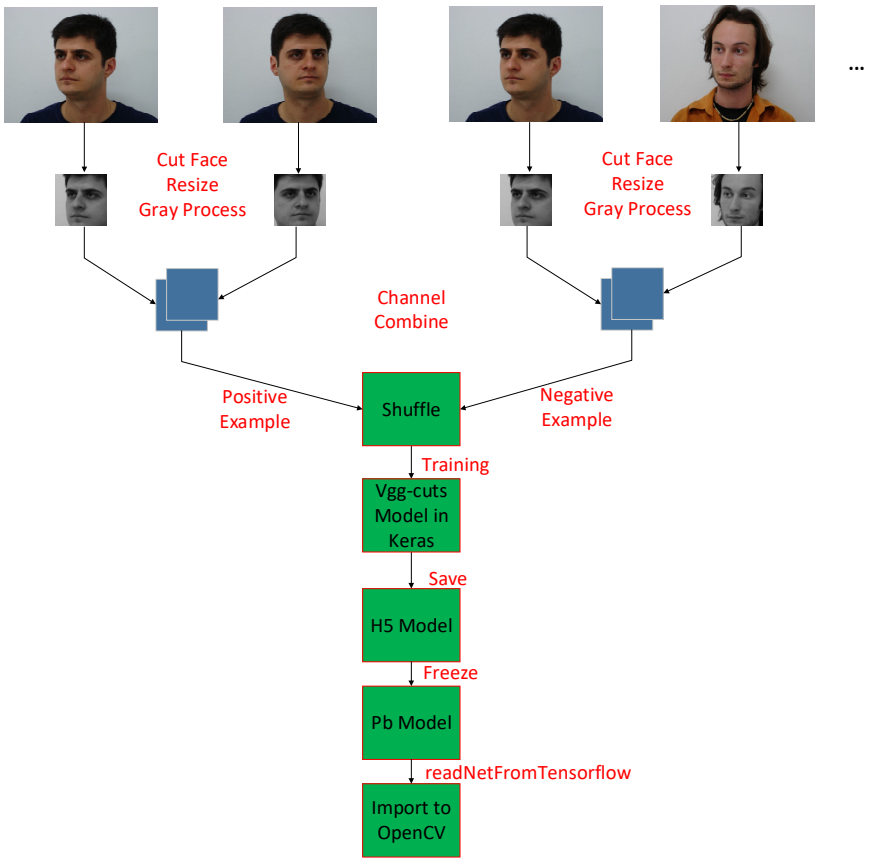

Fig.1 VGG-cut model training process.

It can be seen from Fig.1 that we first performs face detection, resize, and grayscale operations on the positive example to obtain a two-channel image positive example, and then use the same operation to obtain multiple positive and negative examples, after that we scramble the training set and start training the model. Since we want to verify the face recognition program on the embedded platform, we use OpenCV to import trained model.

The feedforward diagram for VGG-cut model in our algorithm is shown in Fig.2.

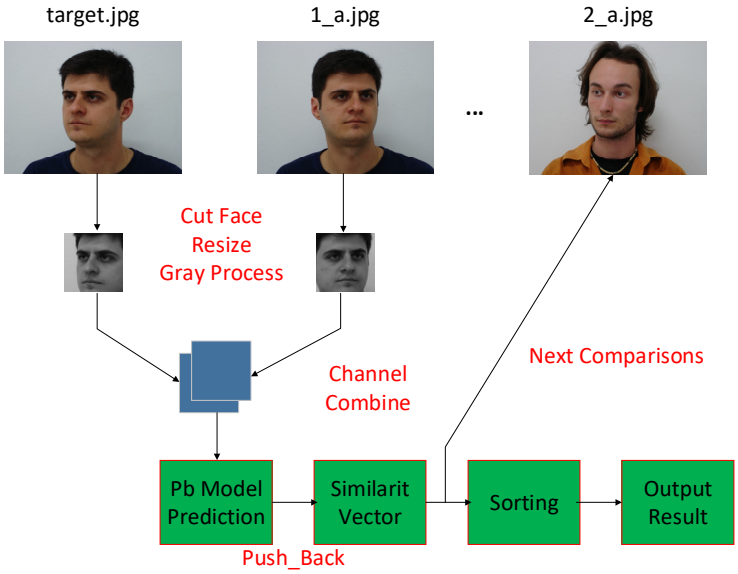

Fig.2 VGG-cut model prediction process.

Fig.2 presents the prediction process of the OpenCV program. It is also necessary to perform face detection, resize, and grayscale operations on the two pictures to be compared, and then pass the data into the model to obtain a similar value, and perform the same operation on the pictures in the face database. Finally, recognition results are output after sorting the similarity vectors.

In these two processes, the images we input are all 3-channel images with a width of 640 pixels and a height of 480 pixels. After grayscale and resize, the width and height of the output image is 100 pixels, and it is a single-channel image.

2.2. VGG-cut models

The VGG-cut model is obtained by cutting the heavy VGG model whose model summary is shown in Table 1.

Table 1. VGG-cut Model Summary

Layer	Input Shape	Output Shape	Kernel	Param	Input Shape
conv2d_1	(100, 100, 1)	(98, 98, 16)	(3, 3, 16)	304	(100, 100, 1)
maxpooling2d_1	(98, 98, 16)	(49, 49, 16)	(2, 2, 16)	0	(98, 98, 16)
dropout_1	(49, 49, 16)	(49, 49, 16)	-	0	(49, 49, 16)
conv2d_2	(49, 49, 16)	(24, 24, 32)	(3, 3, 32)	4640	(49, 49, 16)
maxpooling2d_2	(24, 24, 32)	(12, 12, 32)	(2, 2, 32)	0	(24, 24, 32)
dropout_2	(12, 12, 32)	(12, 12, 32)	-	0	(12, 12, 32)
flatten_1	(12, 12, 32)	(1, 1, 4608)	-	0	(12, 12, 32)
dense_1	(1, 1, 4608)	(1, 1, 32)	-	147488	(1, 1, 4608)
dropout_3	(1, 1, 32)	(1, 1, 32)	-	0	(1, 1, 32)
dense_2	(1, 1, 32)	(1, 1, 1)	-	33	(1, 1, 32)
Total	-	-	-	152,465	-

The main principle of cutting heavy VGG in this paper is that minimize the number of convolution kernels and the depth of the convolution layer while maintaining the characteristics of the VGG convolutional layer. The experimental results show that even if it is cropped to only 2 layers of convolution and 2 layers of fully connected, the accuracy of the model can still be maintained above 96%.

3. Experimental Results

3.1. Experimental environment

In order to verify the algorithm proposed in this paper, we compared the VGG-cut model with the ResNet call by Dlib [12], original VGG-like model and MobileFaceNet. The results show that the algorithm in this paper effectively improves the recognition accuracy and recognition speed. The experimental environment uses the Nexy-Vedio FPGA platform to program an RISC-V soft core called Rocket [13]. The soft core has two hardware threads, the core clock frequency is 50MHz, the on-board DDR3 size is 512MB, and the Debian operating system is stored in a 128G SD for booting by RISC-V SoC, the VGG-cut model is trained and frozen under the Keras framework, the recognition algorithm is implemented on OpenCV and the model obtained by Keras is referenced in OpenCV programs, due to the small memory resources of the FPGA, we cross-compiled both the OpenCV lib and program and then copy them into the SD card. All programs run and called under the environment of the operating system. The FPGA platform and system stack are shown in Fig.3 and Fig.4 respectively.

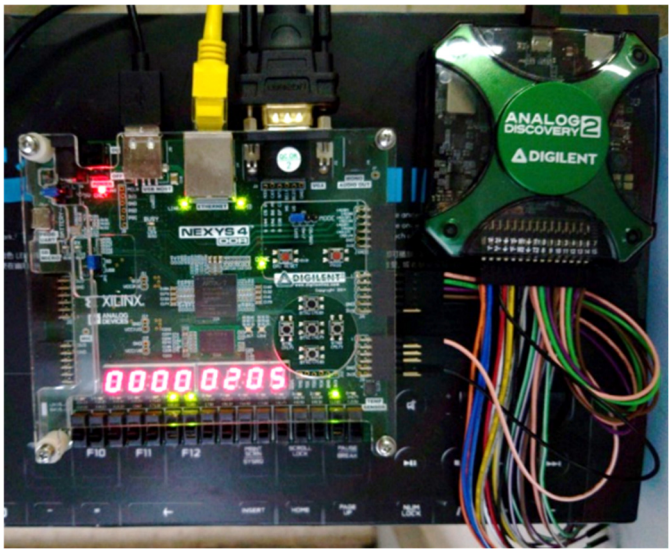

Fig.3 RISC-V SoC on FPGA Platform.

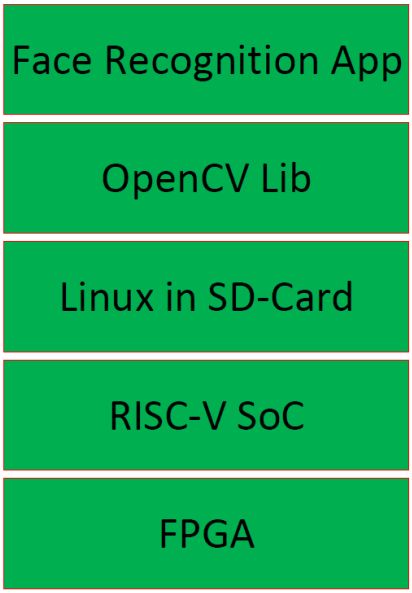

Fig.4 System Stack.

3.2. Recognition performance and analysis

Take a target to be recognized in 20 faces database as an example. There are 10 people in the 20 faces database, and each person has two pictures. Only one person is the target and both two pictures are different angle picture of the target. The result of the recognition as shown in Fig.5.

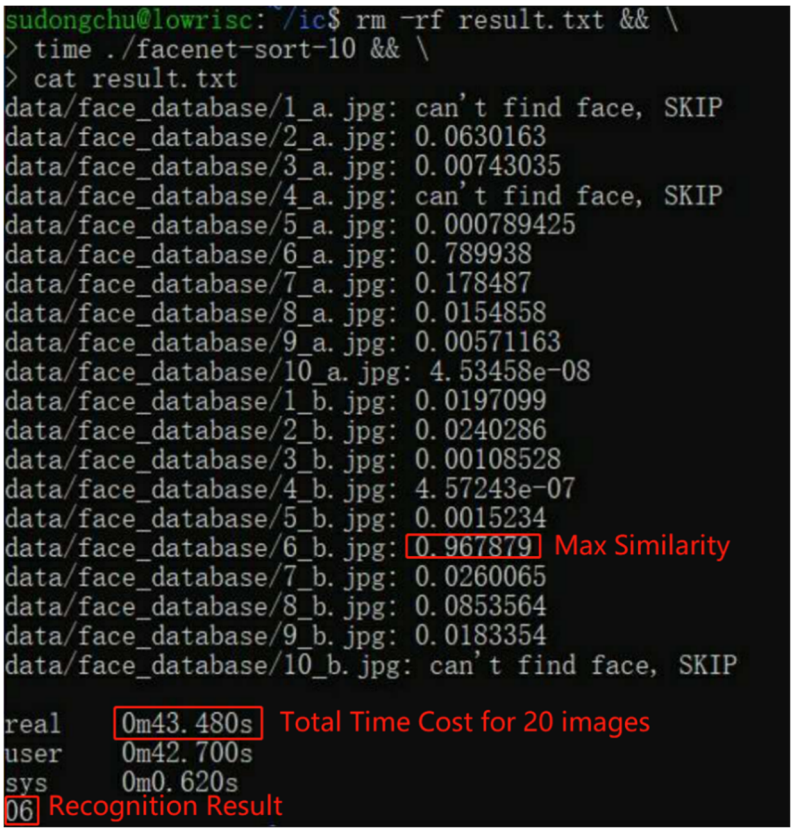

Fig.5 Face Recognition Results of 10 People Database in 50Mhz.

In order to visualize the results, we will output the similarity of the recognized pictures and the final recognized ID. In order to further reduce the recognition time, we make a judgment that the two pictures are the same person when the similarity is greater than 95%, and there is no need for further recognition.

We randomly generate the target face and faces database, and compare ResNet called by Dlib, original VGG-like model, MobileFaceNet and our VGG-cut model on parameters of model size, recognition accuracy. All models are tested on the same test set. The results are shown in Table 2.

Table 2. Performance Comparison with Previous Published Face Verification Models

Method	Model Size	Train Acc	Val Acc	Recognition Time
ResNet[14]	99.72MB	99.2%	99.1%	8m25s
VGG-like[15]	23.63MB	98.7%	98.2%	3m6s
MobileFaceNet[16]	4.23MB	99.3%	99.0%	41s
VGG-cut1	4.04MB	97.8%	97.9%	29s
VGG-cut2	3.76MB	96.9%	93.8%	22s
VGG-cut3	1.04MB	96.5%	96.5%	8s
VGG-cut4	**153KB**	**96.3%**	**96.1%**	**2.1s**

The Val Acc in Table 2 represents the accuracy when the actual test set is used for verification, and the recognition time represents the average time cost for judging a pair of pictures.

From the model size column in Table 2, it can be inferred that the accuracy of the validation set of VGG-cut model can be maintained above 96% even when the model is reduced to 153KB. It can also be inferred that the model with face recognition capability is not necessarily huge in size.

It can be also known from Table 2 that the shallower the depth of the convolution layer of the model, the smaller the number of convolution kernels, the fewer the number of parameters of the model, and the faster the recognition speed of the model.

4. Conclusion

This paper proposes a face recognition algorithm based on dual-channel images. The three-channel picture is grayed out to output a single-channel picture, and combined with other single-channel pictures to obtain the training set and the test set. The cropped VGG model is adopted and trained by dual-channel images training set. We compare ResNet called by Dlib, the original VGG-like model, MobileFaceNet, and our VGG-cut model in performance of model size, recognition accuracy and recognition time on a RISC-V SoC in order to simulate embedded environment. The result show that the small model can still maintain high recognition accuracy and has a faster recognition speed. In the future, we will use a more realistic test environment to test and optimize our face recognition algorithms, and do more researches on reducing hardware power consumption.

Acknowledgments

This work was fully supported by NUDT-NF5 RISC-V development team.

References

[1] Zhou Jianjun and Zhou Jianhong. (2009) Research on embedded digital image recognition system based on ARM-DSP. 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, pp. 524-527.
[2] C. Liu. (2014) The development trend of evaluating face-recognition technology. International Conference on Mechatronics and Control (ICMC), Jinzhou, pp. 1540-1544.
[3] R. Ranjan et al. (2019) A Fast and Accurate System for Face Detection, Identification, and Verification. IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 1, no. 2, pp. 82-96.
[4] M. Belahcene, A. Chouchane and H. Ouamane. (2014) 3D face recognition in presence of expressions by fusion regions of interest. 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, pp. 2269-2274.
[5] Ding, C., & Tao, D. (2017). Pose-invariant face recognition with homography-based normalization. Pattern Recognition, 66, 144–152.
[6] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. (2015) Siamese neural networks for one-shot image recognition. ICML deep learning workshop. Vol. 2.
[7] G. Kertész and I. Felde. (2020) One-Shot Re-identification using Image Projections in Deep Triplet Convolutional Network. IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, pp. 597-602.
[8] Y. Taigman, M. Yang, M. Ranzato and L. Wolf. (2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, pp. 1701-1708.
[9] C. Szegedy et al. (2015) Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 1-9.
[10] Y. Sun, X. Wang and X. Tang. (2015) Deeply learned face representations are sparse, selective, and robust. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 2892-2900.
[11] F. Schroff, D. Kalenichenko and J. Philbin. (2015) FaceNet: A unified embedding for face recognition and clustering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 815-823.
[12] King, Davis E. (2009) Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research 10: 1755-1758.
[13] Asanovic, Krste, et al. (2016) The rocket chip generator. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17.
[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778.
[15] François., Chollet., Keras, (2020). Accessed on: September 3, 2020. [Online]. Available:

https://keras.io/zh/getting-started/sequential-model-guide/

[16] Chen, S., Liu, Y., Gao, X., & Han, Z. (2018) Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In Chinese Conference on Biometric Recognition. pp. 428-438.