Face Anti-Spoofing Using Multi-Branch CNN

Nguyen Cong Tin *, Bach-Tung Pham *, Thi Phuong Le *, Tzu-Chiang Tai†, and Jia-Ching Wang *
*Department of Computer Science and Information Engineering, National Central University, Taiwan
† Department of Computer Science and Information Engineering, Providence University, Taiwan

***Abstract*— We propose a face classification system based on deep learning algorithm. This system is capable of distinguishing real and fake faces from RGB images taken by a normal camera. To do that, we have built a system composed of 4 parts: RGB image processing, HSV image processing, YC_rC_b image processing, and classification. In order to achieve optimal processing performance, we include encoder and decoder structure models, which eliminate unnecessary components and help the model focus only on the components it gives. Most importantly, this structure helps reduce the complexity of the model. In addition, we have applied a number of special tweaks to the training data. Experimental results indicate that our system gives very good results on the public database.**

I. INTRODUCTION

In many face recognition systems, a user's face image is required for verification as described in Fig. 1. This image, normally, is an RGB image. The system compares this image to the database to make sure that the object in this image is presented in the database and they belong to the same object.

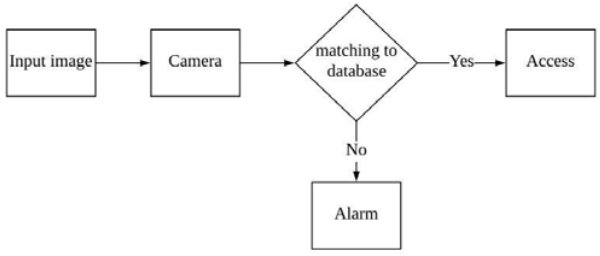

Fig 1: Flowchart of a typical face recognition system.

However, the input image may not be the real object. To solve the weakness of the color-based face recognition system, we proposed a method to help computer have the ability to learn differences between the real and fake face based on Convolution neural network. Besides, it is easy to collect real face images, but hard to collect enough fake face images. So, we applied a method to synthesize data, and then we can generate enough data for training.

The major problems to be solved in this paper are listed below:

1. Enhance classification performance by using multiple color spaces.
2. Solve data missing problem by synthesizing data.
3. Analyze performance of autoencoder structure in the verification system.

II. RELATED WORK

A. Single Shot Detector (SSD) based on ResNet

The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections. The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which we will call the base network. The key difference between training SSD and training a typical detector that uses region proposals, is that ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Some version of this is also required for training in YOLO[1] and for the regional proposal stage of Faster R-CNN[2] and MultiBox[3]. Once this assignment is determined, the loss function and back propagation are applied end-to-end.

B. ResNet

ResNet [4] was unleashed in 2015 by Kaiming He. et.al. through their paper Deep Residual Learning for Image Recognition and bagged all the ImageNet challenges including classification, detection, and localization.

C. Autoencoder

Autoencoders (AE) [5] are neural networks that aims to copy their inputs to their out-puts. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. Figure 2 shows the general structure of autoencoders.

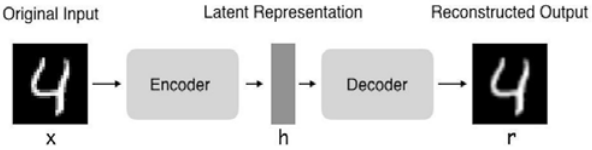

Fig 2: General structure of autoencoders.

III. PROPOSED METHOD

The block diagram of our proposed model is shown in Fig. 3. In our model, the input images are converted to 3 color spaces, RGB, HSV, and YC_rC_b. Then these images are passed into 3 Encoder structures. The outputs of 3 Encoders are concatenated. The output of concatenation has the same width and height as the input image but different depth. This output is passed into the Decoder structure to be decompressed. The output of Decoder moves to the next 2 fully connected layers and is classified to obtain the final output.

Figure 4 shows the Encoder structure in our model. The activation function in each Convolution layer is RELU [6]. It

is proven that RELU gives results better and helps converging models faster than the other activation function like tanh or sigmoid etc.

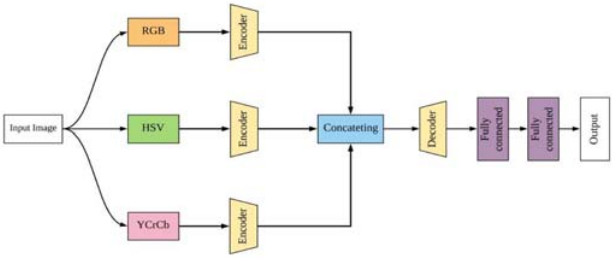

Fig 3: Our proposed model.

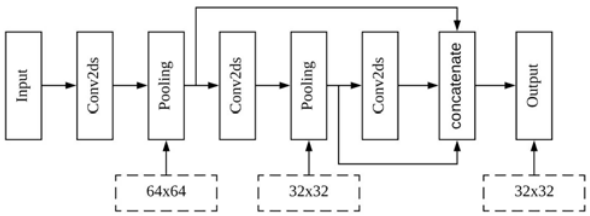

Fig 4: Encoder structure.

To train this model we use the training strategy with 2 steps shown as below:

Step 1: 3 branches are split into 3 models. Each model is an Autoencoder model. Each of these is trained as a normal Autoencoder, and the output label is input. The purpose is to force the model to choose the most important features to rebuild the input image.

Step 2: 3 branches are concatenated together to form the full model as shown in Fig. 3. Besides, 2 fully connected layers and output layer are added next to the Decoder. In this step, all 3 branches and Decoder are frozen, and only fully connected layers and output layer are trained.

In the end, our purpose is to know if it is a real or fake face, so a Binary Cross-Entropy loss function is used to determine probability of the input image. This function has a Sigmoid activation plus a Cross-Entropy loss [7]. Thus it is also called Sigmoid Cross-Entropy loss function. Unlike Softmax loss function, it is independent for each vector component (class), meaning that the loss computed for every CNN output vector component is not affected by other component values. Binary Cross-Entropy loss function is used for multi-label classification, where the insight of an element belonging to a certain class should not influence the decision for another class.

IV. EXPERIMENT

A. Data overview

To evaluate the proposed method, we used the Replay-Attack database. The Replay-Attack database for face spoofing consists of 1300 video clips of photo and video attack attempts to 50 clients, under different lighting conditions. This database was produced at the Idiap Research Institute, in Switzerland. Table 1 shows distribution of videos per attack-protocol in the Replay-Attack database. Figure 5 shows examples of real and fake samples from the Replay-Attack database.

Table 1: Distribution of videos per attack-protocol in the Replay-Attack database.

	Hand-Attack			Fixed-Support			All Supports		
Protocol	train	dev	test	train	dev	test	train	dev	test
Print	30	30	40	30	30	40	60	60	80
Mobile	60	60	80	60	60	80	120	120	160
Highdef	60	60	80	60	60	80	120	120	160
Digitalphoto	60	60	80	60	60	80	120	120	160
Photo	90	90	120	90	90	120	180	180	240
Video	60	60	80	60	60	80	120	120	160
Grandtest	150	150	200	150	150	200	300	300	400

Fig 5: Examples of real and fake samples from the Replay-Attack database. The images come from videos acquired in two illumination and background scenarios (controlled and adverse). The first row belongs to the controlled scenario while the second row represents the adverse condition. (a) Shows real samples, (b) shows samples of a printed photo attack, (c) corresponds to a LCD photo attack, and (d) corresponds to a high-definition photo attack.

B. Preprocessing

To improve the accuracy of our model, we applied the set of transformations to the training data: Rotation, Increase and reduce brightness, Shift image by width or height, Horizontal flip, Vertical flip. We observed that this model was not so accurate in real work. So we applied the perspective transform to get a more accurate simulation under real work conditions. The work in [8] shows a great way to synthesize data. The process was as follows: Add reflection effect, Implement

perspective transform, Fill in the background. Figure 6 shows an example of perspective transform.

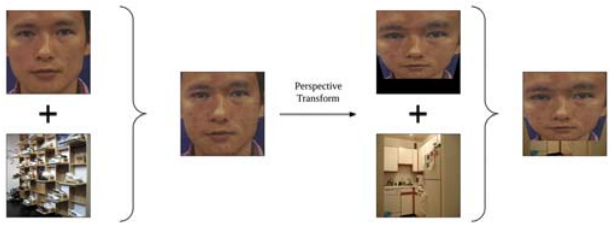

Fig 6: Example of perspective transform.

C. Evaluation metrics

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is often used to describe the performance of a classification model. The following formula is used to calculate the accuracy of a model.

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

True Positive (TP): Observation is positive, and is predicted to be positive.
False Negative (FN): Observation is positive, but is predicted to be negative.
True Negative (TN): Observation is negative, and is predicted to be negative.
False Positive (FP): Observation is negative, but is predicted to be positive.

A system's False Acceptance Rate (FAR) typically is stated as the ratio of the number of false acceptances divided by the number of identification attempts.

$$FAR = \frac{FP}{FP + TN}$$

A system's False Rejection Rate (FRR) typically is stated as the ratio of the number of false rejections divided by the number of identification attempts.

$$FRR = \frac{FN}{TP + FN}$$

Equal error rate (EER) is a biometric security system algorithm used to predetermine the threshold values for its false acceptance rate and its false rejection rate. When the rates are equal, the common value is referred to as the equal error rate.

After obtaining FAR and FRR at various threshold values, the Half Total Error Rate (HTER) is calculated by using the following formula.

$$T = argmin_T(|FAR - FRR|)$$

$$HTER = \frac{FAR_T + FRR_T}{2}$$

We use a set of 100 threshold values from 0.001 to 1 with a difference of 0.001.

D. Results

In this section we will show the results on the Replay-Attack dataset. Figure 7 shows the receiver operating characteristic (ROC) curve on the Replay-Attack dataset. Figure 8 shows EER on the Replay-Attack dataset. It is seen that we get high accuracy on the Replay-Attack dataset.

Figure 9 shows the output of Decoder. In this figure, 3 images are the 3 layers of Decoder's output. We can see a significant difference between real and fake face. Clearly, the real face has been reconstructed with more accuracy than the spoofing face. In the real face, the model focuses on the detail like eyes and lips; in contrast, in the spoofing face, the model focuses on wide areas of the image, especially near edges and face areas. These areas are the places which have a slight reflection of light in the original image.

Table 2 shows the comparison of EER and HTER on Replay-Attack dataset with different methods. Apparently, our proposed method obtains better results than other methods.

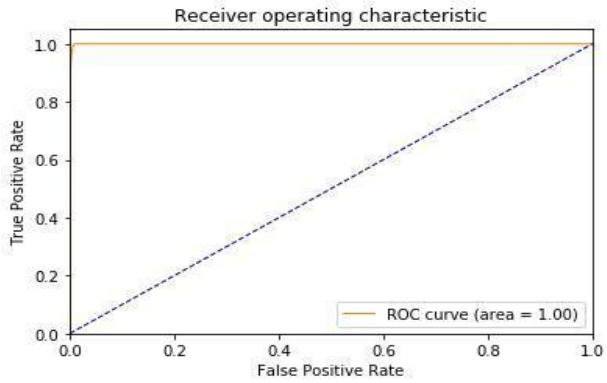

Fig 7: ROC curve on Replay-Attack dataset.

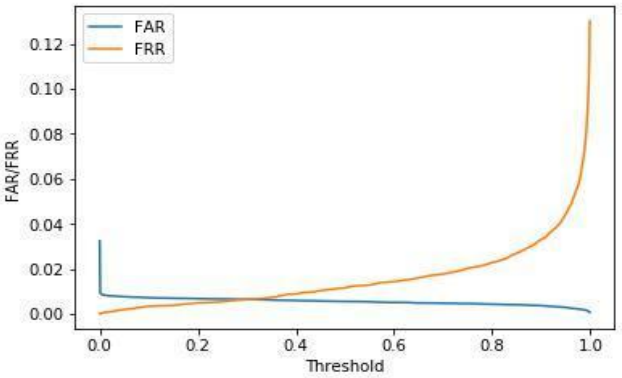

Fig 8: EER on Replay-Attack dataset.

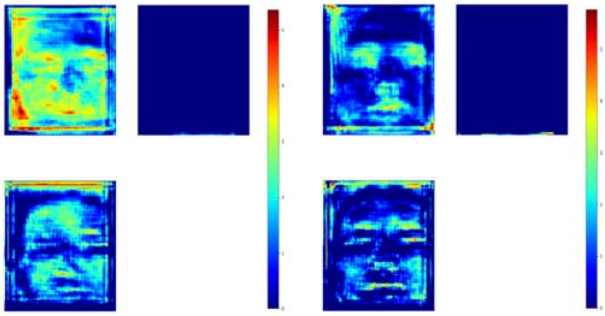

Fig 9: Left: Reconstruction image of fake face. Right: Reconstruction image of real face

Table 2: Comparison of EER and HTER on Replay-Attack dataset with different methods.

Methods	EER (%)	HTER (%)
Fine-tune VGG-Face [9]	8.40	4.30
DPCNN [9]	2.90	6.10
Multi-Scale [10]	2.14	-
YcbCr+HSV-LBP [11]	0.40	2.90
Fisher vector [12]	0.10	2.20
Moire pattern [13]	-	3.30
Patch-based CNN [14]	4.44	3.78
Depth-based CNN [14]	3.78	2.52
Patch&Depth Fusion [14]	0.79	0.72
FASNet [15]	-	1.20
3D synthesis [16]	0.25	0.63
Proposed	0.00	0.00

V. Conclusions

We proposed a method to detect the attack on the vulnerable systems based on convolution neural network. This method extracts features from 3 color spaces, RGB, HSV and YC_rC_b. The combination of ResNet's shortcut into Autoencoder has improved it's efficieny. This combination not only helps the model extract the most important features but also helps the model avoid some problems like vanishing gradients and curse of dimensionality, if we have sufficiently deep networks. Our method uses multi-branch structure to model the multidimensional view point, which helps evaluate input images not only in single color spaces but also many color spaces. This increases accuracy of our model.

Through this research, we have seen the potential of multi-branch structure, so our orientation in the future is to improve this structure. Concretely, we will concentrate on studying the harmony between branches.

References

[1] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[2] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, pp.91-99.

[3] Erhan, D., Szegedy, C., Toshev, A. and Anguelov, D., 2014. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154).

[4] Wu, Z., Shen, C. and Van Den Hengel, A., 2019. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, *90*, pp.119-133.

[5] Rubenstein, P.K., Schoelkopf, B. and Tolstikhin, I., 2018. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*.

[6] Agarap, A.F., 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

[7] Mao, X., Li, Q., Xie, H., Lau, R.Y. and Wang, Z., 2016. Multi-class generative adversarial networks with the L2 loss function. *arXiv preprint arXiv:1611.04076*, *5*, pp.1057-7149.

[8] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A., 2014. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*.

[9] Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M. and Hadid, A., 2016, December. An original face anti-spoofing approach using partial convolutional neural network. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)* (pp. 1-6). IEEE.

[10] Yang, J., Lei, Z. and Li, S.Z., 2014. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*.

[11] Banerji, S., Verma, A. and Liu, C., 2011. Novel color LBP descriptors for scene and image texture classification. In *Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV)*(p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[12] Simonyan, K., Parkhi, O.M., Vedaldi, A. and Zisserman, A., 2013, September. Fisher vector faces in the wild. In *BMVC*(Vol. 2, No. 3, p. 4).

[13] K. Patel, H. Han, A. K. Jain, and G. Ott. Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks. In ICB, 2015. 7

[14] Atoum, Y., Liu, Y., Jourabloo, A. and Liu, X., 2017, October. Face anti-spoofing using patch and depth-based CNNs. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (pp. 319-328). IEEE.

[15] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo. Transfer learning using convolutional neural net works for face anti-spoofing. In ICIAR, 2017. 7

[16] Guo, J., Zhu, X., Xiao, J., Lei, Z., Wan, G. and Li, S.Z., 2019, June. Improving face anti-spoofing by 3d virtual synthesis. In *2019 International Conference on Biometrics (ICB)* (pp. 1-8). IEEE.