Application of Active Learning in Face Liveness detection

XIN ZHAO

2018150801025std.uestc.edu.cn

ABSTRACT

Face anti-fraud attack detection is mainly to help the face recognition system determine whether the collected face is the user's own face, or a printed photo, recorded video, 3D mask and other forgeries, so it is also called liveness detection. This technology is essential for the security of mobile phone unlocking, access control, and face payment. So how to automatically and efficiently distinguish the authenticity of images and resist spoofing attacks to ensure system security has become an urgent problem in face recognition technology.

This article uses CNN neural network to add a dropout layer on a self-made data set that simulates real authentication scenarios, and re-scaling the training image, 45-degree rotation, width offset, height offset, horizontal flip and zoom enhancement. This results in a higher accuracy rate and lower loss.

Keywords: Active Learning, Face Liveness Recognition, Convolutional Neural Network, Image classification

1. Introduction

Compared with biometrics such as fingerprints and iris, facial features are the easiest to obtain. Facial recognition systems have gradually begun to be commercialized, and are developing towards automation and unsupervised trends. However, the current facial recognition technology can identify the identity of a face image but cannot accurately distinguish the authenticity of the input face. So how to automatically and efficiently distinguish the authenticity of images and resist spoofing attacks to ensure system security has become an urgent problem in face recognition technology.

At present, face recognition technology faces three kinds of fraudulent methods: face pictures of legitimate users, facial videos of legitimate users, and 3D models and masks of legitimate users. At present, there are three commonly used ones: blink detection, optical flow detection and multispectral detection。

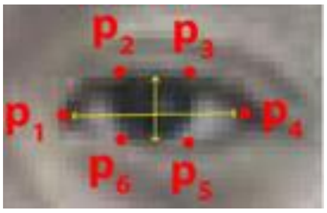


Figure 1-1 blink detection

Blink detection: Find the coordinates of eye feature points by using a face detector (Figure 1), and then calculate the aspect ratio between the two eyes by formula 1-1.

$$EAR = \frac{\left\| p_2 - p_6 \right\| + \left\| p_3 - p_5 \right\|}{2\left\| p_1 - p_4 \right\|} \tag{1-1}$$

Set the standard threshold (Generally, it is 0.25.) If the aspect ratio of the eye is greater than the threshold, it means that eye is open, and if the aspect ratio is less than threshold, it is closed.

Farneback algorithm: For each pixel, we need to calculate its velocity vector v by formula 1-2:

$$v = \left(\frac{dx}{dt}, \frac{dy}{dt} \right) = \left(v_x, v_y \right) \tag{1-2}$$

The polar coordinates and angular velocity of the collected optical flow points are expressed, and expressed by a histogram. The chi-square test is used to determine whether the two optical flow field patterns are consistent. If the threshold is exceeded, it is a living body, and if it does not exceed the threshold, it is a non-living body.

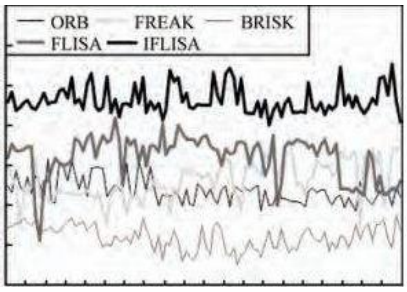


Figure 1-2 farneback algorithm

Multispectral Detection: Figure 1-3 shows the difference in reflectance between human skin and common face forged materials such as silicone masks, photos, PVC, etc. at 380nm~1100nm. It should be pointed out that the above data are measured from the skin part.

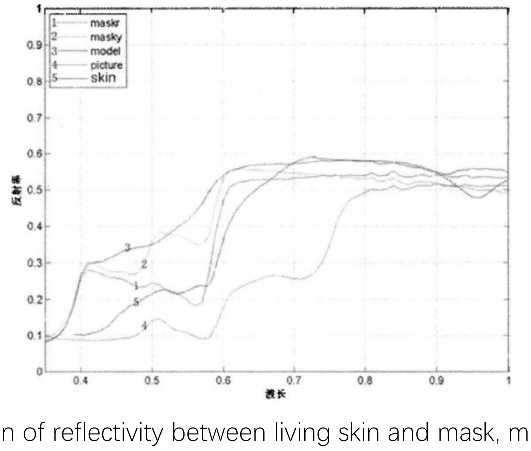


Figure 1-3 Comparison of reflectivity between living skin and mask, model, and photo

Whether it’s a silicone mask, photo or video, they can be easily separated from the skin by reflectivity.

The above three methods have good performance in live detection, but all have their own problems: blink detection requires the cooperation of users, farneback algorithm requires more complicated calculations and has certain requirements on the instrument, and multispectral detection is very important for the requirements of the instrument and the environment are more demanding.

Although in the CVPR2019 face anti-counterfeiting detection challenge, the top three teams have made a very good accuracy rate (close to 99%), but their results are based on a relatively complete data set and complex neural network. Regardless of the difficulty of obtaining the data set or the complexity of the calculation, considering the hardware foundation at this stage, the top three works are more difficult to promote as a commercial product.

In actual situations, it is difficult to label each image in the data set, and because of the limitation of computing power, it is difficult to perform very complex neural network calculations. Therefore, in this article, the author passes through the processing of several videos, obtained a simple self-made data set that simulates the real live detection scene. Considering that in the real situation, the data label may be less, we use the active learning method to simulate the real scene, and finally get a higher accuracy and the lower loss, compared with the random model, it can be found that the accuracy rate has been greatly improved after active learning.

2. Dataset production

The dataset of this part reference to the work of Adrian Rosebrock about face recognition: https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/#download-the-code , The construction of the data set in this article refers to the ideas and codes of video processing in the above article, but on the basis of the code, the preservation of the generated image files is added, and the normalization of the data set (unified as 150*150) Format) and the division of the data set, as well as some modifications to the parameters of the model.

2.1 the choice of video for dataset

This experiment intercepted clips from the three movies "Black Panther", "Home Alone", and " A Chinese Odyssey ", in which way can we collected face images of men, women, and children of different races at different ages. Among them (Figure 2-1), the real images (above) It is to directly use the original video to capture the face. The false image (below) is to first use the mobile phone to capture the video, and then use the mobile phone to capture the face.

Figure 2-1 the faces in the video

2.2 Processing of the video

Detect whether there are a human face every 16 frames from the MP4 video. If there is a human face, determine the position of the human face through the 68 feature points of the human face, and take a picture of the human face and save it to a specific file path, and then standardize the obtained face images to obtain face image data of the same length and width (150*150). Take a piece of image for example (Figure 2-2).

From the three videos, we obtained 13,540 pictures of three races and different age groups, using 70% of the data set as the training set and 30% of them as the test set for random grouping. Among them, in the training set, there are 3,613 real faces and 5,866 fake faces. In the test set, there are 1,548 real faces and 2,513 fake faces.

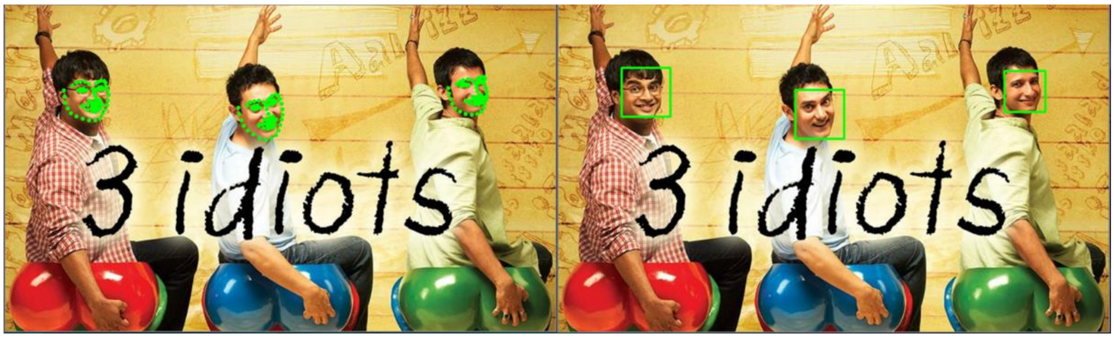
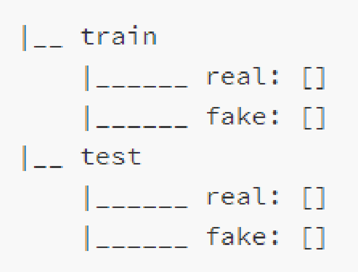


Figure 2-2 68 face feature points selection (left) and face interception (right)

2.3 Dataset structure

After constructing the data set according to the methods in the above two subsections, the final data set structure is as following:



```
|__ train
    |______ real: []
    |______ fake: []
|__ test
    |______ real: []
    |______ fake: []
```

Figure 2-3 dataset structure

3. Model building

This part mainly reference to the work of the article called “Active Learning for Multi-class Image Classification”, and its code in:

https://github.com/mzhao98/ActiveLearning_ImageClassification

based on its code, this article changed the data import method, perform operations such as flipping the photo to prevent the occurrence of over-fitting., and combined some data visualization code, which can make the code clearer. It also change the network and some parameter in the code, to make it fit with our own data set.

3.1 Image data processing

Format the image as a floating-point tensor that has been properly preprocessed and then fed into the network:

1. Read the image from the disk
2. Decode the internal synchronization of these images and convert them to the correct grid format according to their RGB content
3. Convert them to floating point tensors
4. Rescale the value of the tensor from 0 to 255 to a value between 0 and 1, because neural networks prefer to handle smaller input values

Since the training volume of the self-made data set is relatively small, overfitting usually occurs. Therefore, we expand the data set by re-scaling the training image, 45-degree rotation, width offset, height offset, horizontal flip and zoom enhancement, so that it has a sufficient number of training examples. This helps expose the model to more aspects of the data and can be better generalized.

3.2 Neural network construction

This article uses a relatively simple CNN network structure, which consists of the following basic layers:

1. **Convolutional layer**: Each convolutional layer in a convolutional neural network is composed of several convolutional units, and the parameters of each convolutional unit are optimized through the backpropagation algorithm.
2. **ReLU layer**: The ReLU layer uses Rectified Linear Units, ReLU. Used as the activation

function of this layer of nerves. It can enhance the non-linear characteristics of the decision function and the entire neural network without changing the convolutional layer itself.

3. **Pooling layer**: In the pooling layer, there are many different forms of non-linear pooling functions, among which "Max pooling" is the most common. It divides the input image into several rectangular areas, and outputs the maximum value for each sub-area.
4. **The dropout layer**, which will randomly filter (set to zero) the number of output units from the applied layer during the training process. Dropout takes the form of fractions as its input value, in the form of 0.1, 0.2, 0.4, etc. This means randomly exiting 10%, 20% or 40% of the output unit from the applied layer. When a dropout of 0.1 is applied to a certain layer, it will randomly kill 10% of the output units in each training period.

My convolutional neural network is constructed in the way shown in Figure 3-1:

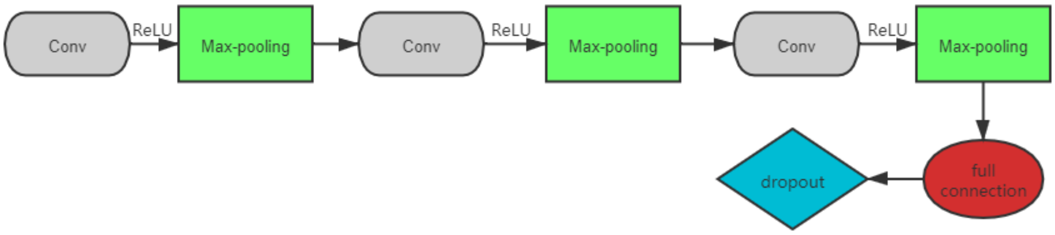


Figure 3-1 CNN structure

3.3 Active learning

Active learning refers to the use of automatic machine learning algorithms to automatically select some data request tags from the data set, which is also known as query learning or optimal experimental design in statistics. Active learning continuously selects data from unlabeled data and adds it to the training set after designing a reasonable query function. Effective active learning data selection strategy can effectively reduce the cost of training and improve the recognition ability of the model at the same time.

This article uses the largest marginal uncertainty method to select the data that needs to be labeled. The largest marginal uncertainty method is to compare the best and worst uncertainties. The maximum marginal uncertainty (LMU) is the classification probability of the most probable category minus the classification probability of the least likely category. If the probability of the most probable class is significantly greater than the probability of the least probable class, then the classifier is more certain about the class membership of the example. Similarly, if the probability of the most probable category is not much greater than the probability of the least likely category, the classifier is not sure about the class membership of the example. The active learning algorithm will select the example with the smallest LMU value.

This paper compares the use of maximum marginal uncertainty with randomly selected samples to be labeled, showing the superiority of the maximum marginal uncertainty algorithm.

4. conclusion

After an epoch of about 400 steps, an accuracy rate of nearly 90% was finally obtained on the training set (Figure 4-1), which is a big improvement compared with the accuracy rate of 50% obtained by the random model, indicating that the maximum margin of use Uncertainty can greatly improve the prediction accuracy of the model. And this model maintains a trend of gradual improvement in accuracy with the increase of epoch.

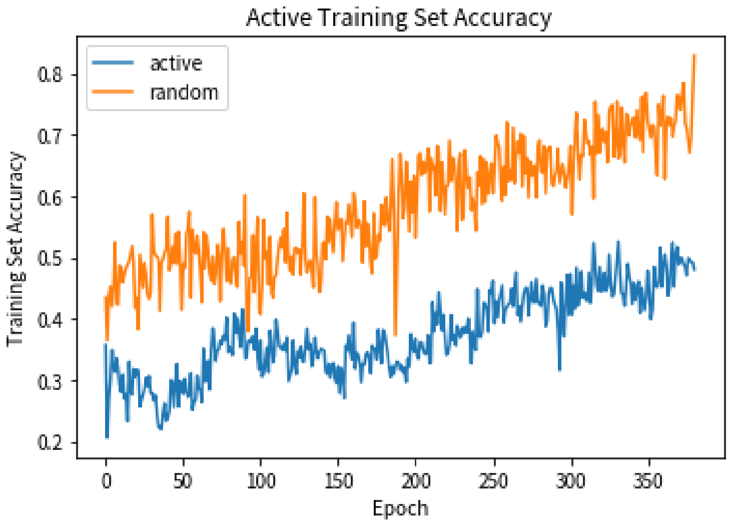


Figure 4-1 accuracy of the model

Compared with traditional methods, this model has lower requirements for user interaction, equipment and environment. Compared with neural networks, the method used in this article can greatly reduce the number of samples that need to be labeled. The accuracy rate will continue to rise.

However, the model in this experiment is still inadequate, and the accuracy is relatively low. It may be necessary to increase the complexity of the neural network or adjust the parameters in the future. In addition to selecting the data to be labeled through the Largest Margin Uncertainty, there are three methods for selecting the Smallest Margin Uncertainty, Least Confidence Uncertainty, and Entropy Reduction. In the future, you can consider using other three methods to select the data to be labeled.

The code of this article can be download in:

https://github.com/XqsZX/MCU_active_learning_for_face_liveness_detection

5. Acknowledgement

The author would like to express his sincere thanks to the Dr. Robert F. Murphy for introducing the active learning and inspire the author to finish this experience. Thanks Adrian Rosebrock, for his article about opencv face recognition, his work let author get to know this area, and made his own dataset. Thanks RuYi Wang, my TA for answering questions about the code.

6. References

[1] O. Kahm and N. Damer, “2d face liveness detection: An overview,” inb Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the. IEEE, 2012, pp. 1–12.

[2] A. Anjos, M. M. Chakka, and S. Marcel, “Motion-based countermeasures to photo attacks in face recognition,” IET Biometrics, vol. 3,no. 3, pp. 147–158, 2013.

[3] Wild P, Radu P, Chen L, et al. Robust multimodal face and fingerprint fusion in the presence of spoofing attacks[J]. Pattern Recognition, 2016, 50: 17-25.

[4] Boulkenafet Z, Komulainen J, Hadid A. face anti-spoofing based on color texture analysis[C]//Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015: 2636-2640.

[5] Galbally J, Marcel S, Fierrez J. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition[J]. Image Processing, IEEE Transactions on, 2014, 23(2): 710-724.

[6] Chingovska I, Rabello dos Anjos A, Marcel S. Biometrics evaluation under spoofing attacks[J]. Information Forensics and Security, IEEE Transactions on, 2014, 9(12): 2264-2276.

[7] Pinto A, Robson Schwartz W, Pedrini H, et al. Using visual rhythms for detecting video-based facial spoof attacks[J]. Information Forensics and Security, IEEE Transactions on, 2015, 10(5): 1025-1038.

[8] Gragnaniello D, Poggi G, Sansone C, et al. An investigation of local descriptors for biometric spoofing detection[J]. Information Forensics and Security, IEEE Transactions on, 2015, 10(4): 849-863.

[9] Kim W, Suh S, Han J J. Face Liveness Detection From a Single Image via Diffusion Speed Model[J]. Image Processing, IEEE Transactions on, 2015, 24(8): 2456-2465.

[10] Pinto A, Pedrini H, Robson Schwartz W, et al. Face Spoofing Detection Through Visual Codebooks of Spectral Temporal Cubes[J]. Image Processing, IEEE Transactions on, 2015, 24(12): 4726-4740.

[11] Wen D, Han H, Jain A K. Face spoof detection with image distortion analysis[J]. Information Forensics and Security, IEEE Transactions on, 2015, 10(4): 746-761.

[12] Joshi A J , Porikli F , Papanikolopoulos N . Multi-class active learning for image classification[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2009.

[13] https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/