Project Report:

Face Recognition Attendance System Framework

Khang Vo

Swinburne University of Technology

COS30082: Applied Machine Learning

Grace Tao

3 June 2024

Introduction

Face recognition is a crucial technology for various applications, including security and attendance system. This project focuses on building a face recognition attendance system with an integrated liveness detection technique to prevent spoofing. The system utilizes convolutional neural networks (CNNs) for face verification and anti-spoofing methods to ensure the authenticity of the faces being recognized. This report details the methodology and results of the implemented models, including the training schemes, evaluation metrics, and performance comparisons.

Methodology

To deal with the dataset of images of faces with a 64x64 size, we decided to use ResNet-18 as a based embedding model for all methods. The ResNet-18 model was almost the same as the original model, except that we substituted the classification head with a fully connected layer of 128 nodes to output face embeddings. We trained the base embedding CNN model to extract low-dimensional face embeddings from images. In this project, we used two approaches: metric learning and classification-based learning.

Method 1: Metric Learning

Metrics

We used a Siamese network to model the similarity between pairs of face images. With this method, we used two different metrics to build two distinct models:

Euclidean distance: This metric calculates the sum of the squared differences between the corresponding coordinates of the two vectors.

Cosine distance: This metric calculates the 1 - (cosine of the angle between the two vectors). This metric ranges from 0 (closest) to 2 (furthest)

Training Scheme

The given dataset of 64x64 images of human faces were used to train the network. The dataset was divided into batch size of 16 and was resized and normalized for ResNet model before training. The networks were trained with random triplet sets of anchors, positive and negative images to minimize the distance between the embeddings of anchor and positive while maximizing the distance between the anchor and negative. A margin was added to the anchor-negative distance to make sure that the model could separate the images. For the Euclidean model, the used margin value was 0.5, while the margin was 1 for the Cosine model.

The used loss function was the triplet loss function, which returns the maximum between (*anchor-positive distance - anchor-negative distance + margin*) and 0. Both models were compiled with Adam optimizer with a learning rate of 0.0001 and were trained for 5 epochs each.

Method 2: Classification-based learning

Method

While using the same base embedding model of ResNet-18, in this method, we added two extra fully connected layers with 1000 and 4000 nodes respectively to perform classification tasks (the dataset has 4000 people). The first fully connected layer was followed by a batch normalization layer and a ReLU activation, while the later layer only had softmax activation.

Training Scheme

The same dataset was used to train the classification network, except that instead of forming triplets, the dataset just had 4000 classes. We also used batch size of 16 and performed image preprocessing and normalization for ResNet model before training.

The model was compiled with Adam optimizer with a learning rate of 0.0001 and sparse categorical cross-entropy loss function. It was trained for 10 epochs.

Extra: Anti-spoofing techniques

Researched ideas

As I researched, there are two main approaches to liveness detection:

- Texture analysis (passive liveness detection): this method uses such methods as Local Binary Pattern (LBP) or frequency-based analysis to classify real face or printed face on 2D photograph. (Das & Chakraborty, 2014)
- Motion analysis (active liveness detection): this method actively checks for and requires user to perform some actions, such as blink or turn left/right.

Due to the limitation of time, hardware and dataset, we decided to use Motion analysis to avoid spoofing attack in the system, as it requires few additional data.

Approach

To prevent spoofing attacks, we decided to check for blink action when the user checks in. The blink action was divided into two phases: eyes closed, and eyes opened. To check if user has closed their eyes or not, we had to use facial landmarks to calculate the aspect ratio of each eye. In the 68-point facial landmark system, each eye consists of 6 points, forming 2 connected vertical lines and a horizontal line. The eye aspect ratio (EAR) can be calculated as the sum of the length of the two vertical lines, divided by double the length of the horizontal line (Soukupová & Čech, 2016). Then we could take the average EAR of both eyes to get the final ratio. If the resulting ratio is bigger than a threshold, then the user is identified as opening eyes and vice versa.

Due to technical limitations, we could not perform the detection (which requires several frames to detect) in real time along with the face recognition module (which requires exactly one frame). Therefore, we resorted to requiring user to check in twice, one with eyes closed and one with eyes opened. If a user can perform both actions consecutively (no other faces recognized in between the two times), then that user will be detected as real face and check in successfully.

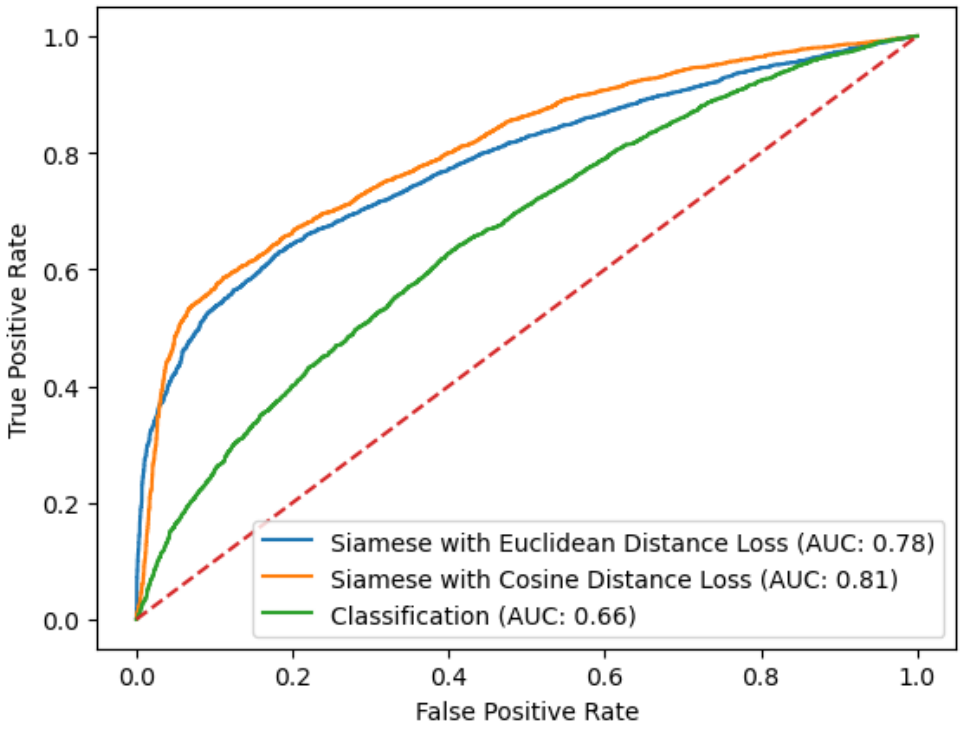

Figure 1. ROC curve and AUC of each model

Conclusion and Discussion

Face Verification

Evaluation metric

To evaluate the effectiveness of the three models, we used the Receiver Operating Characteristics (ROC) with the Area Under the Curve (AUC) metric. The ROC curve is plotted as the true positive rate against the false positive rate at different threshold. The area under the ROC curve represents the probability that a classifier will rank a positive pair of faces higher than a negative pair of faces in term of similarity score. A good model will have the AUC close to 1. (Google for Developers, n.d.)

Evaluation results

As the classification model overfitted at around epoch 5, the result will be calculated with the best classification model at epoch 5.

After training all three models, we took out each embedding model and attached a simple Siamese head to calculate the distance between a pair of faces. We used the verification set that came with the dataset to evaluate all models. The result ROC curve and AUC of each model are presented in figure 1.

As can be seen from the graph, the two Siamese models return promising results, with the one using Cosine Distance metric slightly better than the other. Meanwhile, the classification-based model's performance is not good. This is due to the nature of classification models, as they only arrange embeddings so that the embeddings can be dividable, but the distance between the same class is not necessarily shorter than the distance to other classes. On the other hand, metric learning models learn to group embeddings to form clusters and push other classes away, making it separable.

As a result, the best model for face recognition is the metric learning model using Cosine distance, with the AUC of 0.81.

Anti-spoofing techniques

Through many experiments, we identified an effective threshold for the Eye Aspect Ratio (EAR) at approximately 0.25, which accurately recognizes genuine eye actions most of the time. However, the performance is inherently linked to the accuracy of the facial landmark predictor from dlib. Thus, while the system reliably detects eye movements as intended, its effectiveness is partially dependent on the precision of the underlying facial landmark detection. Moreover, since the check-in system is not continuous (due to system design and huge delay when inferencing), it remains vulnerable to attacks using two different photos or video playback.

References

Das, D., & Chakraborty, S. (2014). Face liveness detection based on frequency and micro-texture analysis. *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, 1-4. doi:10.1109/ICAETR.2014.7012923

Google for Developers. (n.d.). *Classification: ROC Curve and AUC*. Retrieved 6 3, 2024, from Google for Developers: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Soukupová, T., & Čech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. *L. Čehovin, R. Mandeljc, & V. Štruc (Eds.), 21st Computer Vision Winter Workshop*.