Content

Abstract

This project is mainly to complete the palmprint feature extraction and classification tasks. The data set contains 99 people's palm print pictures, in which 3 palm print pictures of each person are distributed in the training set, and the other 3 palm print pictures are distributed in the test set. In this project, I tried the traditional method use SIFT to extract features and KNN for classification which get accuracy of 97.31%, and also tried the convolutional neural network method such as ResNet which get accuracy of 83.16%. In addition, I also tried to use the Gaussian filter, Gabor filter, LBP, etc. to process the palmprint image and extract the texture from the palmprint image, but these methods have not improved the accuracy of palmprint recognition.

Keywords: palmprint recognition, SIFT, KNN, ResNet

1 Project Content

1.1 Introduction of Palmprint Recognition

Automatic palmprint recognition is a biometric recognition technology that has emerged in recent years. On the surface of people's wide palm skin, there are rich texture structures, including main lines, wrinkle lines, ridge lines, singular points, texture and other image features. These features are different from each other and are not related to genes, so palmprints contain a lot of information Can be used for identification. The palmprint image collection is also very convenient, the requirements for the collection equipment are not high, and the general camera or scanner can collect very clear images. Framework of Biometric System as shown in Figure 1, this project requires the realization of the extraction and matching of 99 people's palmprint features, and gives the matching accuracy rate.

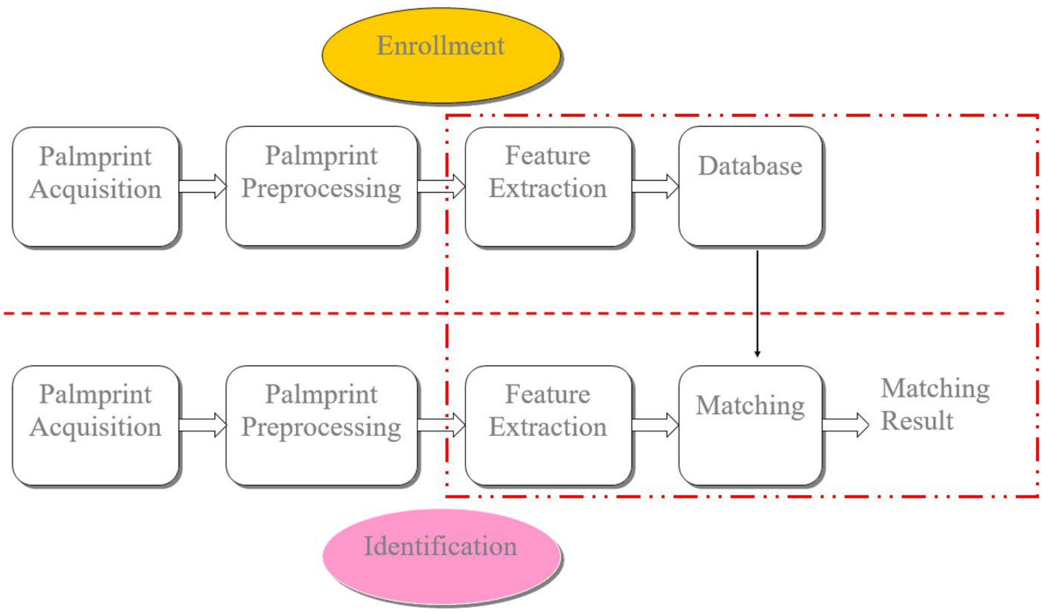


Figure 1. Framework of Biometric System.

1.2 Feature Extraction

1) Main Lines and Wrinkles Extraction

As shown in Figure 2, the main lines and wrinkles in the palmprint image are the most important feature and the main basis for palmprint recognition. The main line is a collective term for "head line", "heart line", and "life line", which everyone has. Wrinkles are not necessarily available to everyone, but if there are, it will be more helpful for recognition.

2) Texture Feature Extraction

There are a lot of texture features in palmprints, please design an algorithm to extract them

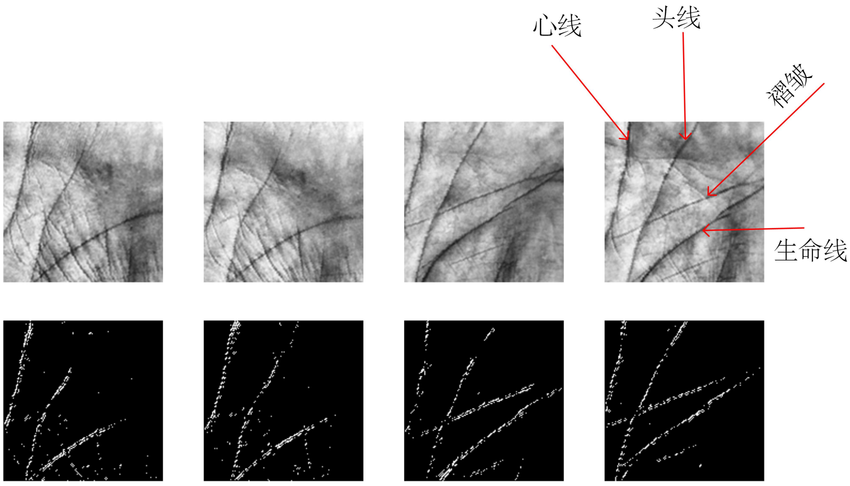


Figure 2. Main Lines and Wrinkles in Palmprint

1.3 Feature Matching

Dataset: there are palmprint images of 99 persons, and each person has 6 samples, 3 of them are in folder “training”, and the rest samples are in folder “testing”. Please using the images in folder “training” for registration, and the images in folder “testing” for testing. Matching the features extracted and give the matching accuracy.

2 Method Description

According to the survey, there are currently two main methods for palmprint recognition. One is to use traditional feature extraction methods such as Wavelet transform, Gabor filters, LBP and SIFT to extract features, and then use KNN or SVM for classification. Another is to train a convolutional neural network to extract and classify image features.

In this project, I tried the traditional method use SIFT to extract features and KNN for classification, and also tried the convolutional neural network method such as ResNet.

2.1 Scale Invariant Feature Transform

Scale invariant feature transform (SIFT) is the size invariant feature transform. It can transform image data into local features independent of size. This feature is very stable, and the image size, rotation angle, spun twists, viewing angle changes, noise, and lighting changes have little effect on this feature.

The main step of SIFT is to extract the key points in the image, and use key point descriptors to describe these key points.

The whole process is mainly divided into four steps:

(1) Scale space peak selection. The purpose of this step is to select potential key points (or points of interest) that satisfy scale invariance and rotation invariance in scale space.

(2) Key point localization. The purpose of this step is to accurately locate

the location of the characteristic key points, which involves the removal of pseudo key points.

(3) Orientation assignment. The purpose of this step is to assign an orientation to each corresponding key point based on the local gradient direction of the key point.

(4) Key point descriptor. The purpose of this step is to describe each key with a high dimensional vector for each key point.

2.2 K-Nearest Neighbors Algorithm

The simplest initial classifier is to record all the categories corresponding to the training data, and when the attributes of the test object and the attributes of a training object completely match, they can be classified. But how is it possible for all test objects to find a training object that exactly matches it, followed by the problem that one test object matches multiple training objects at the same time, resulting in one training object being divided into multiple classes, based on these problems, just K-Nearest Neighbors Algorithm (KNN) was proposed.

KNN classifies by measuring the distance between different feature values. The idea is: if most of the k most similar samples in the feature space (that is, the nearest neighbors in the feature space) belong to a certain category, the sample also belongs to this category, where K is usually not greater than 20. In the KNN algorithm, the selected neighbors are all objects that have been correctly classified. This method determines the category to which the samples to be classified belong based on the category of the nearest sample or samples in the decision of classification.

In KNN, the distance between objects is calculated as a non-similarity index between each object to avoid the matching problem between objects. The distance generally uses Euclidean distance. Besides, KNN makes decisions based on the dominant category of k objects rather than a single object category. These two points are the advantages of the KNN algorithm.

The description of KNN algorithm is:

1) Calculate the distance between the test data and each training data.

2) Sort according to the increasing relationship of distance.

3) Select K points with the smallest distance.

4) Compute the frequency of occurrence of the category of the first K points.

5) Return the category with the highest frequency among the first K points as the predicted classification of the test data.

2.3 Deep residual network

Residual networks (ResNet) was proposed by the MSRA He Kaiming team in 2015 and won the first place in many competitions. Its related paper ***Deep Residual Learning for Image Recognition*** also won the best paper of CVPR2016. In the paper, the author proposed the residual structure as shown in Figure 3. Add an identity mapping (identity mapping), the original function H(X) need to learn convert to F(X) + X. The article believes that compared to H(X), the optimization of F(x) is simpler, and the optimization difficulty of the two is not the same. This idea comes from the residual vector coding in image processing. Through a reformulation, a problem is decomposed into residual problems between multiple scales. Can get better optimization results.

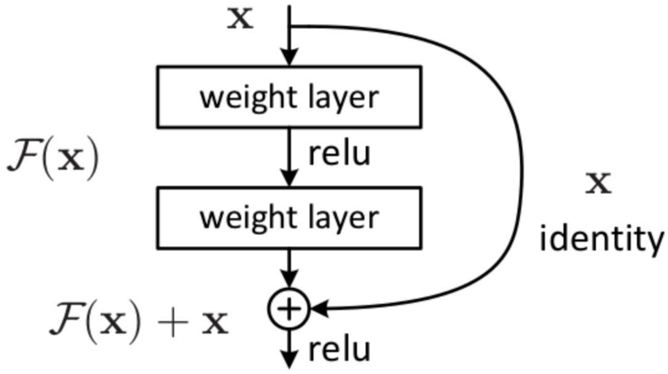


Figure 3. Residual Block

3 Experiment Results and Analysis

In this project, I tried the traditional method use SIFT to extract features and KNN for classification which get accuracy of 97.31%, and also tried the convolutional neural network method such as ResNet which get accuracy of 83.16%.

3.1 Scale Invariant Feature Transform and KNN

I tried the traditional method use SIFT to extract features and KNN for classification which get accuracy of 97.31%. Algorithm flow:

1) Preprocess the picture to equalize the picture grayscale.

2) Obtain the key point kp and descriptor des (feature description vector) generated by the SIFT algorithm.

3) Use KNN to calculate the number of matching points between the query image and the training image, and use k (k=2) nearest neighbor matching.

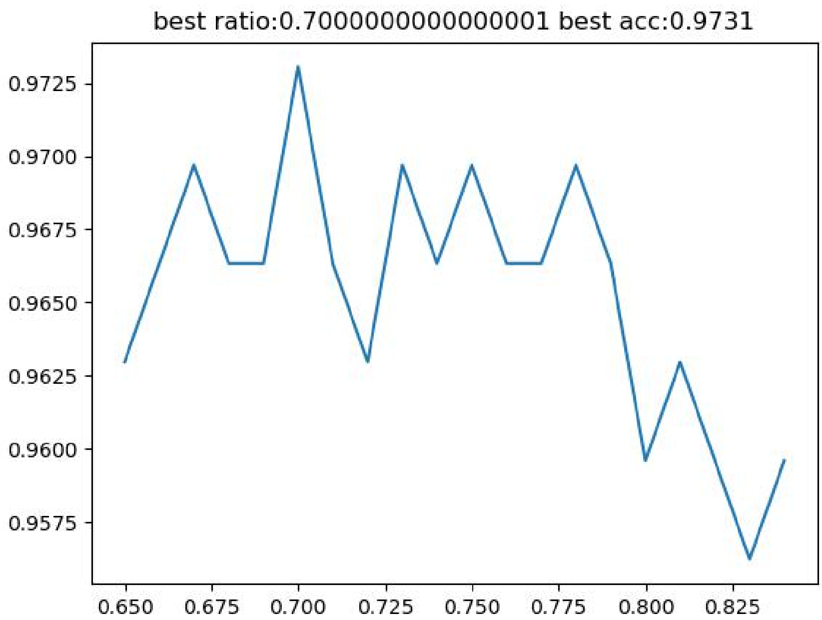
4) According to the maximum number of matching points, determine the

category of the query picture.

1. Code: SWIT_DIP.py

2. Result:

```
Processing image 293 ...
Predict result: 98 Groud truth: 98
Processing image 294 ...
Predict result: 98 Groud truth: 98
Processing image 295 ...
Predict result: 99 Groud truth: 99
Processing image 296 ...
Predict result: 99 Groud truth: 99
Processing image 297 ...
Predict result: 99 Groud truth: 99
Predict the correct number of pictures: 289 Accuracy: 0.9730639730639731 ratio: 0.7
best ratio:0.7 best acc:0.9731
```



3.2 Residual Networks

I tried the convolutional neural network method such as ResNet which get accuracy of 83.16%. This method needs to be run on the GPU server.

It mainly includes 5 steps:

1) Establish a data set.

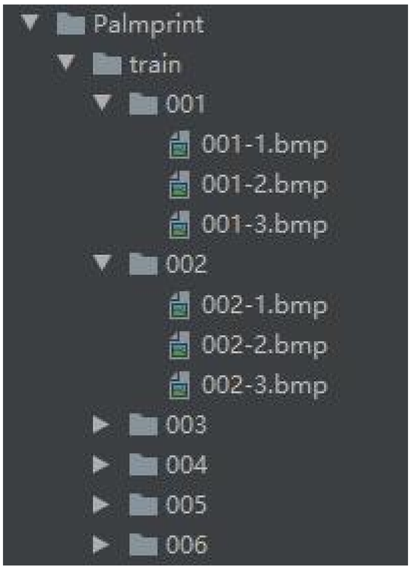
2) Data enhancement.

3) Load data.

4) Transfer learning.

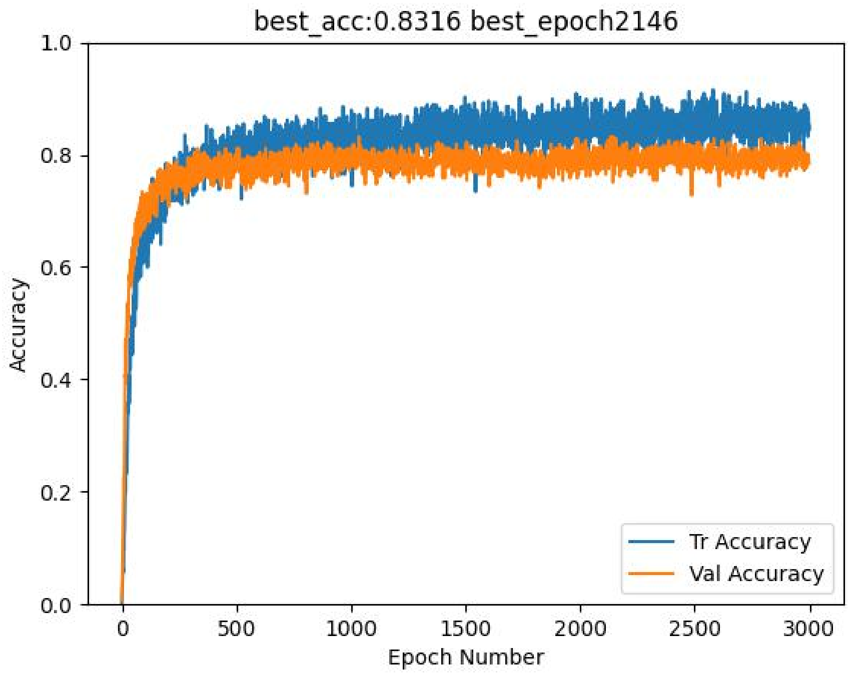
5) Training.

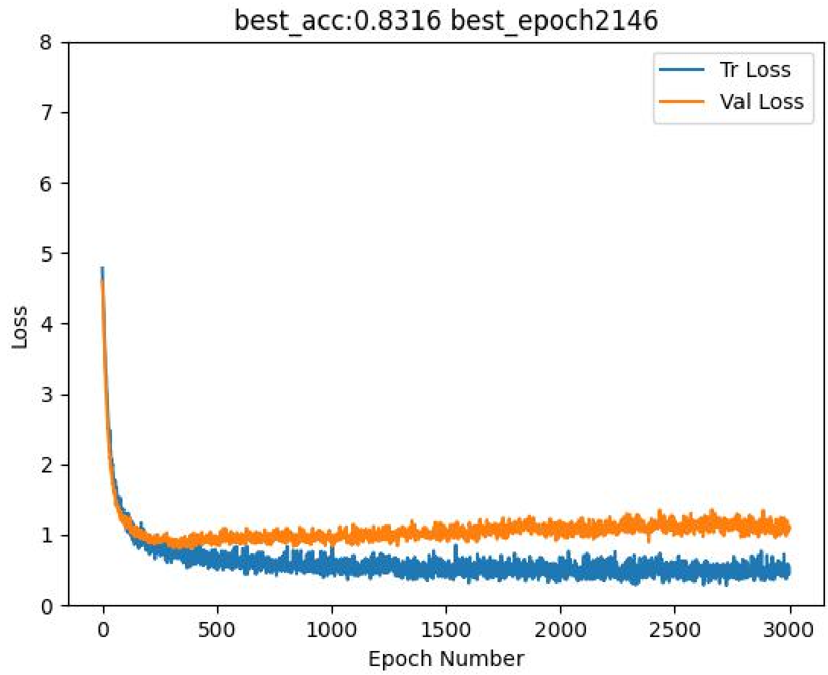
1. Data directory structure:



2. Code: resnet18_DIP.py

3. Result: accuracy: 83.16%. The amount of data is small, so the training is fast. The experiment ran 3,000 epochs, but 500 epochs can reach the peak of the quasi-rate.



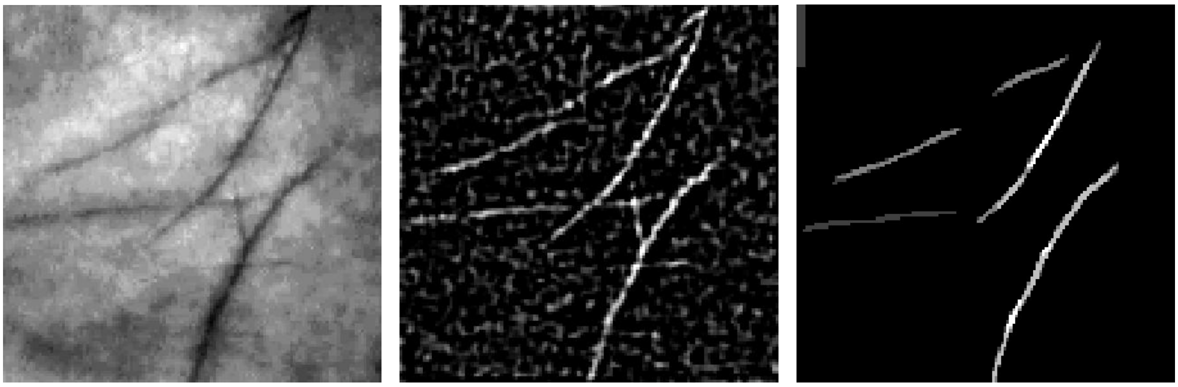


3.3 Texture Extraction

In addition, I also tried to use the Gaussian filter, Gabor filter, LBP, etc. to process the palmprint image and extract the texture from the palmprint image, but these methods have not improved the accuracy of palmprint recognition.

1. Code: texture_extraction_DIP.py

2. Image processing results:



(a) Grayscale (b) Low-pass Gaussian filter (c) Gabor filter

4 Summary

This project is mainly to complete the palmprint feature extraction and classification tasks. The data set contains 99 people's palm print pictures, in which 3 palm print pictures of each person are distributed in the training set, and the other 3 palm print pictures are distributed in the test set. In this project, I tried the traditional method use SIFT to extract features and KNN for classification which get accuracy of 97.31%, and also tried the convolutional neural network method such as ResNet which get accuracy of 83.16%. In addition, I also tried to use the Gaussian filter, Gabor filter, LBP, etc. to process the palmprint image and extract the texture from the palmprint image, but these methods have not improved the accuracy of palmprint recognition.