EE301FZ Signals and Systems

Signature Verification

Li ZongTan 19105690

Wei ZhiChen 19103887

Luo Tian 19104642

Yang YanBo 1910337912345

Wang Ning 19103964

Wang YanXiang 19104278

Project-2021 Supervisor: Chin Hong Wong

Degree Major -BSc in Robotics and Intelligent Devices

Maynooth University

National University of Ireland Maynooth

Maynooth International Engineering College

福州大学梅努斯国际工程学院

A project submitted in partial fulfilment of the requirements for the BSc in Robotics and Intelligent Devices

Declaration

We hereby certify that this material, which we now submit for assessment on the program of study as part of junior qualification, is entirely our own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of our work.

We hereby acknowledge and accept that this thesis may be distributed to future first year students, as an example of the standard expected first year projects.

Signed: Date:

Abstract

Nowadays, bio-metric technology is used to prevent unauthorized access to handwritten documents from either signature verification. In some situations, real-time identifying can not be realized and only the off-line signature can be used. People are facing the problem that they need to verify different signatures by identifying the eigenvalue of the signature's features. An improved off-line signature verification system is proposed in the paper. The off-line signature verification method has two stages: the data prepossessing , the feature classification based. Three feature classification approaches were proposed: first are a BP neural network with a Gabor filter, second are Yolo v2 is implemented using the embedded system, and third are the HOG identification method The image prepossessing has finished perfectly with two captivities and the three feature classification methods are capable to achieve about 90 percent false acceptance rate(**FAR**), 10 percent false rejection rate(**FRR**) percent average equal rate(**AER**) ,for the test set. From our test, we can conclude that the verification method is reliable and the BP neural network, k210 and HOG methods are capable to achieve great performances on our testing set with the two standards(FAR,FRR,AER).

1 Introduction

1.1 Topic

Signature is a behavioral trait of an individual and forms a special class of handwriting in which legible letters or words may not be exhibited. Signature verification is a method to prevent unauthorized access, which is widely adopted in the bank and legal system. Signature verification systems are divided into two categories: ONLINE (dynamic) and OFFLINE (static). In the online case, an acquisition device, such as a digitizing table, is used to acquire the user's signature. The data is collected as a sequence over time, containing the position of the pen, and in some cases including additional information such as the pen inclination, pressure, etc. In offline signature verification, the signature is acquired after the writing process is completed. In this case, the signatures were processed as image files. Facing the situation, we looked through many articles on the internet and decide to utilize two algorithms. In order to exclude other potentially influential factors, the raw image files were prepossessed to get a useful part of the signature. The prepossessing of data includes assigning x,y axis coordinates to the image files and adjustments of the x,y axis coordinates according to the rotation angle. Furthermore, establishments of the BP neural network and the Gabor filter follow the standards shown in GitHub and possess our own parameter choices, the k210 follow the existed algorithms to handle the real-time verification emergency, and the hog methods is based on opencv to improve verification performances. The evaluation of our works is based on the accuracy of verification and the similarities between the right answer and another disturbance term.

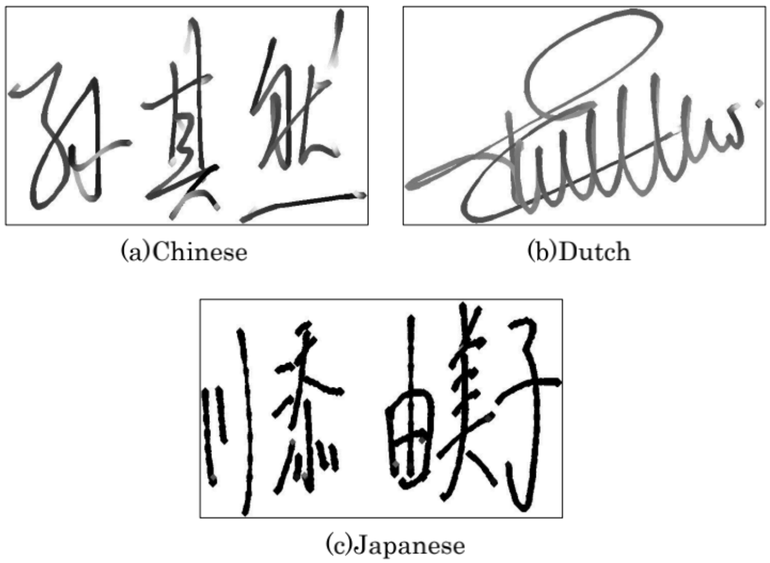


Figure 1: Examples of generated images for Chinese, Dutch, and Japanese signatures. [1]

1.2 Metrics

There are many target we need to evaluate our algorithm/model which has been discussed completely [2].

- False Acceptance Rate (FAR): own signatures but reject
- False rejection Rate (FRR): other signatures but accept
- Average Error Rate (AER):the average error considering only FRR and FAR
- Equal Error Rate(EER): Threshold when FRR and FAR curve intersection point.

1.3 Project Statement

By now, we already have some achievements: verification of the reliability of the signature recognition on the MATLAB platform by using the neural network, the comparisons between BP neural network and gross number classifier, optimization of the parameter collection, etc.

2 Literature Review

Signature verification is an important research area in the field of person authentication. It has been broadly researched in the last decades. We search a lot of paper, which includes 5 in Chinese and 5 in English. Most of the papers are based on the online signature which has more data for analysis [3] or based on the screens with more data.[6] Some online methods can not be adopted with limited data. [4]. So, many papers put their mind on the genuine signature and others' signature, because some papers think the similarity is not easy to distinguish with image data merely. In this way, we determine our research only based on the image file of the signature. Also, most papers based on huge online data sets, utilize our own image format signature to figure out whether the relatively small amount of data can train our model well.

After the objects of study and size of the data set as defined. We have more restrictions on the choices of reference articles. The solution of the prepossessing comes from the article only refers the image prepossessing. [5]. Due to the demand of huge data of other classifiers. Gross number classifier was the chosen classifier which was relatively simple that can be trained with our limited data. Also, the BP neural network and hog methods were been chosen because of its similarities to classifiers and the low time, GPU, knowledge costs for training. the `K210` can effectively to handle the real-time emergency for signature verification,

2.1 Feature extraction

For feature extraction, some paper used transforms to finish this task. There are: **Gabor Filter** is used to doing the data dimensional reduction to get the texture feature [6], the method of **LPP**(Locality Persevering Projection) is used to linear data dimensional to get the effective feature vector and feature dimension [7]. Some papers proposed based on the On-line signature using RGB information and **HOG**(Histogram of Oriented Gradient) to distinguish different signature [1], and eigen-signature been proposed which use the **PCA**(Principle Component Analysis) to extract feature of signature [8]. And **LBP**(local binary pattern) is an improvement of binary image, which is used to extract feature in block [9].

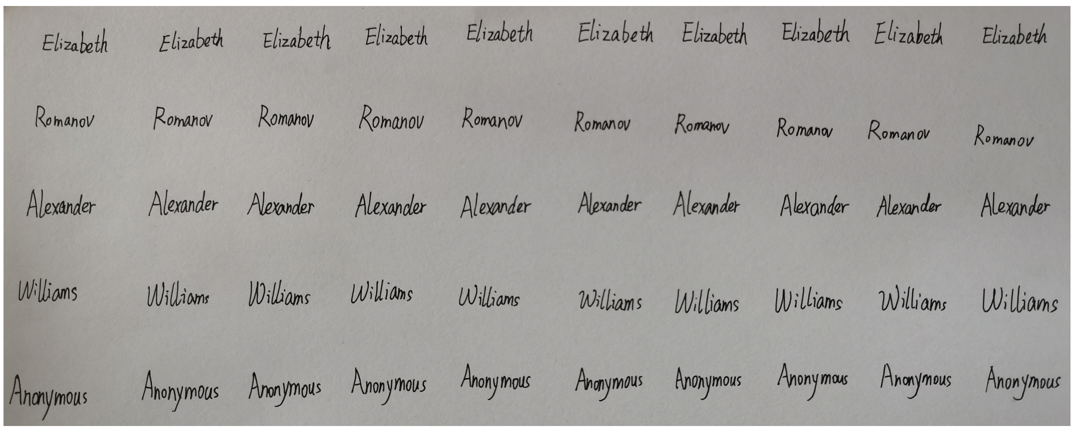
2.2 Training

For training part, **SVM**(support vector machine) is widely used for most classifier [1] [7] , And the other important part is neural network, **BP**(Back Propagation) is the simplest method to classify different feature [6], and **DBN** (Deep Belief Network)also has high accuracy performance [9].

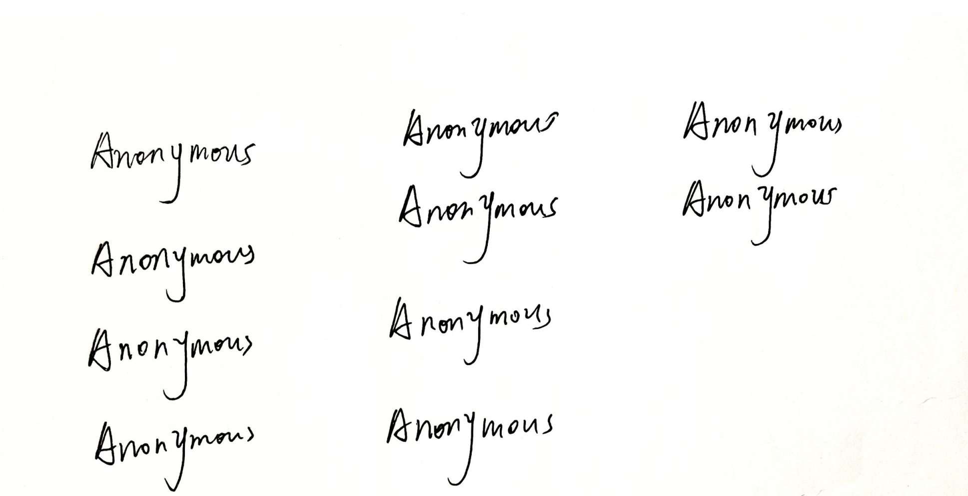
3 Methodology

3.1 Training Set Making

After discussion, we decide to sign the verbs `Anonymous` used as our training set. Added other four verbs as supplementary verbs to verification the Robustness. Project members is 6, we can verification 5 person's signature.



(a)

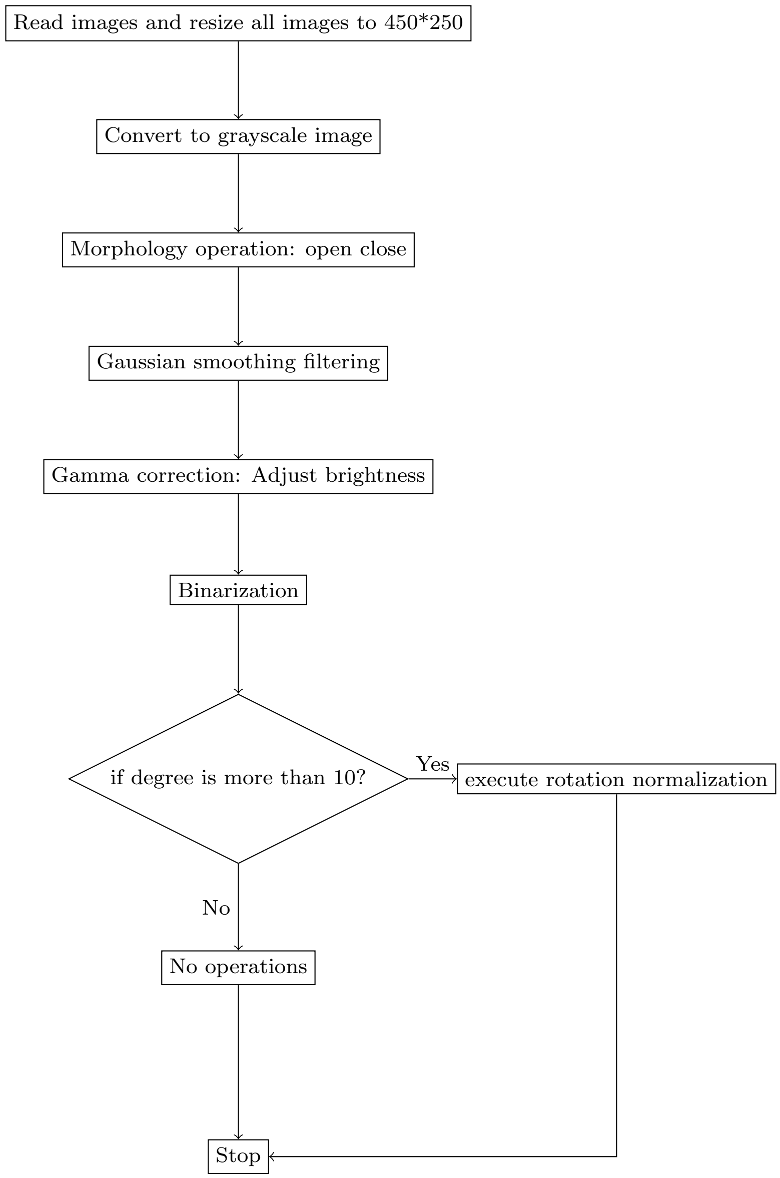


(b)

Figure 2: The hand write testing set

3.2 Prepossessing

Environment of `Python3.9` with `OpenCv` for prepossessing the images after cutting. The processing flow chart is below:



The main idea of rotation normalization is: The center of mass is calculated below [2]:

$$\begin{aligned} \bar{u} &= 1/N \sum_{i=1}^{N} u(i) \\ \bar{v} &= 1/N \sum_{i=1}^{N} v(i) \end{aligned} \tag{1}$$

Where the

$u(i)$ is x-coordinate of the ith pixel in the signature curve.

$v(i)$ is y-coordinate of the ith pixel in the signature curve.

N is the number of pixels in the signature.

Shift the center of the curvature to the center of image. Center normalization is finished. The Second order momentum and cross momentum is calculated below:

$$\begin{aligned} \bar{u^2} &= 1/N \sum_{i=1}^{N} (u(i) - \bar{u})^2 \\ \bar{v^2} &= 1/N \sum_{i=1}^{N} (v(i) - \bar{v})^2 \\ \overline{uv} &= 1/N \sum_{i=1}^{N} (v(i) - \bar{v})(u(i) - \bar{u}) \end{aligned} \tag{2}$$

Then calculated the eigenvalue of the matrix below

$$I = \begin{pmatrix} \overline{u^2} & \overline{uv} \\ \overline{uv} & \overline{v^2} \end{pmatrix} \tag{3}$$

The least inertia axis angle is given by

$$\theta = \arctan(y_{eig}, x_{eig}) \tag{4}$$

Once θ obtained, all points on signature can rotate with rotation matrix below:

$$R = \begin{pmatrix} \cos(\theta) & -sin(\theta) \\ sin(\theta) & \cos(\theta) \end{pmatrix} \tag{5}$$

3.3 Main algorithm

3.3.1 2D Gabor Transform and BP Neural Network

the three methods all achieve ideal results, so we can choose the most suitable algorithm according to different environments to test the signatures.

BP which have 2 processes:The forward propagation of the signal and the backward propagation of the error.

In the forward propagation, the input samples are passed in from the input layer, processed by each hidden layer layer by layer, and then passed to the output layer. If the actual output of the output layer is inconsistent with the expected output (teacher signal), it will turn to the error back propagation stage.

In back propagation, the output is transmitted back to the input layer layer by layer through the hidden layer in some form, and the error is allocated to all units of each layer, so as to obtain the error signal of each layer unit, and this error signal is used as a correction unit The basis of the weight.We have decided to do testing and these follows what we have done, testing result attached: It is base on the method of literature [6], it used methods to extract characteristic of Chinese.

Gabor filter can extract relevant features in different scales and directions in the frequency domain within the operations like model selection, feature dimension reduction, feature normalization and feature selection. In the spatial domain, a two-dimensional Gabor filter is the product of a sinusoidal plane wave and a Gaussian kernel function. The former is the tuning function and the latter is the window function

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}} e^{i\left(2\pi \frac{x'}{\lambda}+\psi\right)} \tag{6}$$

Where

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases} \tag{7}$$

Which is basic idea of Gabor Filter.

PCA has the function for dimensional reduction , by taking the largest individual differences shown by the principal components and finding features that are more easily understood by network and reduce the number of variables.

3.3.2 YOLO v2

Yolo(you only look once) [10]uses convolutional networks to extract features, and then uses the full connection layer to obtain predicted values. The network structure refers to GooLeNet model, which contains 24 convolutional layers and 2 fully connected layers. Finally, the predicted value of each boundary box actually contains five elements: (x, y, W, H, C), among which the first four represent the size and position of the boundary box and the last value is the confidence degree. Using edge computing chip `Kendryte 210` to implement classification.

3.3.3 Histogram of Oriented Gradients and Machine Learning Classifier

Firstly, prepossess the images, and then calculate the gradient value of pixel points using formula below:

$$\begin{aligned} g &= \sqrt{g_x^2 + g_y^2} \\ \theta &= \arctan \frac{g_x}{g_y} \end{aligned} \tag{8}$$

After that, gradient histogram is formed. Then normalize the blocks, and finally HOG feature is collected.

The feature were collected 108 dimensions, Using **Maltlab** to testing multiple machine learning algorithm, the 99% accuracy algorithm **CART**(classification and regression tree) has been obtained.

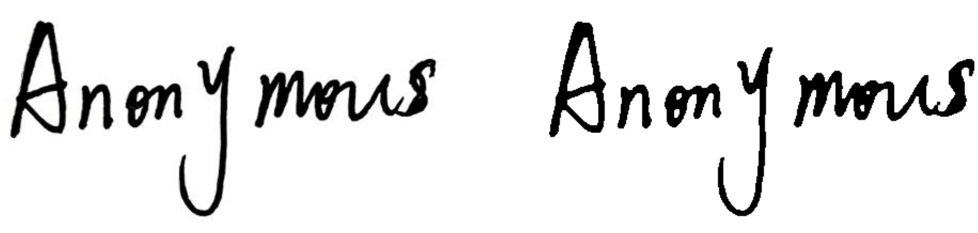
3.4 Testing

Take one signature of a person as the main body each time The test set include 20 pictures of the person, other five members' 6 picture each and 10 optional forged signatures. Two more signatures per person for the extra test set. We adjust the classifier parameters, draw the curve picture based on a total of 21 times tests.

4 Result and Discussion

+++++ subsectionApproaches results

4.0.1 Prepossessing



(a) before (b) after

Figure 3: The comparison of prepossessing

4.0.2 Gabor extraction result

(a)

(b)

Figure 4: The comparison of prepossessing

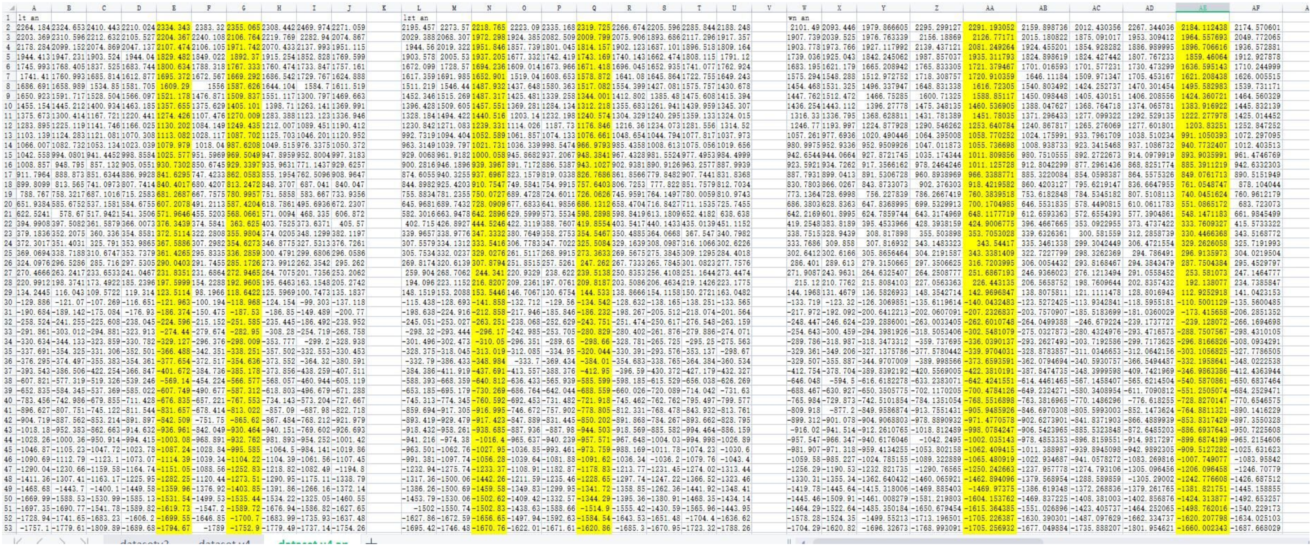


Figure 5: Gabor filter result 32 dimension features

Doing the verification of algorithm based on first person signature, value closer to 1 means the signature have more likelihood signed by:

```
1t   1 测试一:0.99997 测试二:0.99989
lzt 0 测试一:4.5874e-06 测试二:0.00092746
wn  0 测试一:0.00023358 测试二:0.02872
wyx 0 测试一:4.0857e-06 测试二:9.2691e-06
wzc 0 测试一:2.1901e-05 测试二:1.9469e-05
yyb 0 测试一:5.867e-05 测试二:0.00013955
```

Figure 6: Testing result

4.0.3 K210 running result

After training, the loss graph can be obtained:

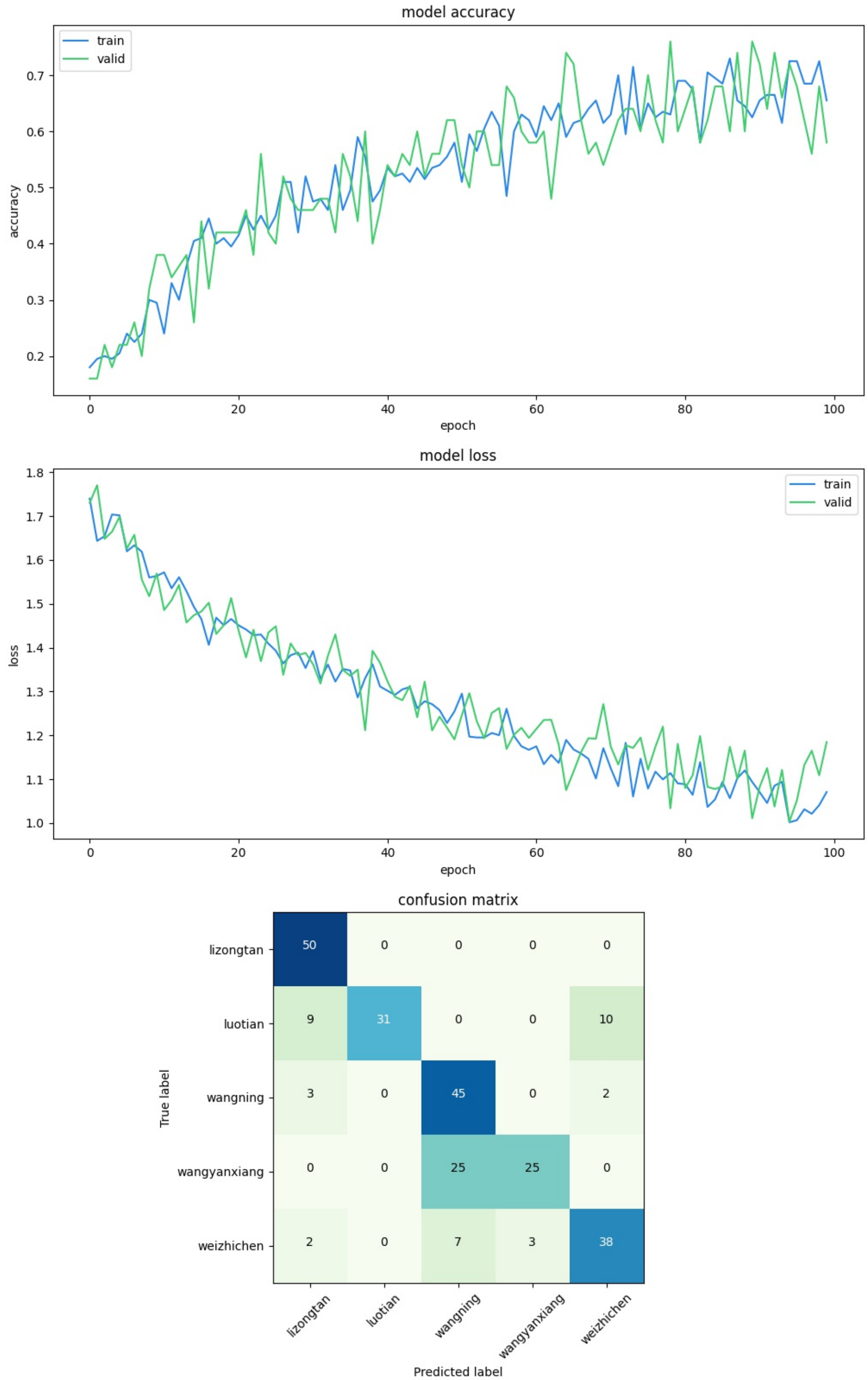


Figure 7: loss matrix for training

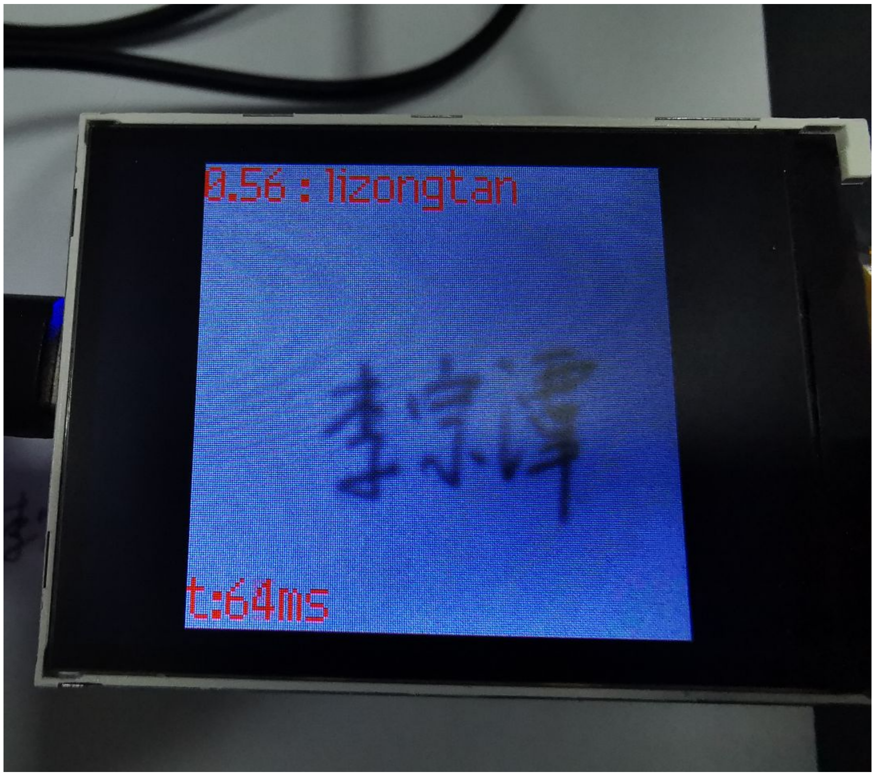


Figure 8: The Embedded system K210 running picture

4.0.4 HOG extraction result

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
10	0.2404	0.2029	0.2095	0.2029	0.2083	0.2078	0.2078	0.2527	0.2053	0.2601	0.2013	0.2029	0.2535	0.2055	0.2078	0.2128	0.2843	0.2578	0.1567	0.1848
11	0.0769	0.0747	0.0703	0.0747	0.0729	0.0692	0.0692	0.0410	0.0727	0.0828	0.0771	0.0747	0.0269	0.0714	0.0732	0.0813	0.1013	0.0226	0.0104	0.0137
12	0.2308	0.2241	0.2109	0.2241	0.2187	0.2075	0.2075	0.1229	0.2181	0.2484	0.2313	0.2241	0.0808	0.2142	0.2197	0.2306	0.2843	0.0677	0.0312	0.0410
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0.2404	0.2426	0.2415	0.2426	0.2404	0.2413	0.2413	0.2388	0.2407	0.2601	0.2408	0.2426	0.1486	0.2405	0.2414	0.2306	0.2843	0.2414	0.1359	0.1618
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0.2404	0.2122	0.2232	0.2122	0.2179	0.2273	0.2273	0.1516	0.2208	0.2573	0.2182	0.2122	0.0668	0.2190	0.2091	0.2289	0.2843	0.1846	0.0912	0.1186
17	0.0833	0.0707	0.0744	0.0707	0.0726	0.0758	0.0758	0.0505	0.0736	0.0858	0.0727	0.0707	0.0223	0.0730	0.0697	0.0763	0.1531	0.0615	0.0304	0.0395
18	0.2404	0.2029	0.2095	0.2029	0.2083	0.2078	0.2078	0.2527	0.2053	0.2601	0.2013	0.2029	0.2535	0.2055	0.2078	0.2128	0.2843	0.2578	0.1567	0.1848
19	0.2171	0.2051	0.2065	0.2051	0.2055	0.2059	0.2059	0.2527	0.2091	0.1459	0.2034	0.2051	0.3062	0.2076	0.2090	0.2212	0.0754	0.2912	0.3333	0.2772
20	0.0431	0.0715	0.0695	0.0715	0.0729	0.0727	0.0727	0.0392	0.0694	0.0253	0.0747	0.0715	0.0369	0.0744	0.0702	0.0668	0.0246	0.0376	0.0489	0.0430
21	0.1293	0.2146	0.2086	0.2146	0.2186	0.2180	0.2180	0.1176	0.2081	0.0760	0.2240	0.2146	0.1106	0.2231	0.2105	0.2005	0.0739	0.1127	0.1467	0.1291
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0.2404	0.2426	0.2415	0.2426	0.2404	0.2413	0.2413	0.2527	0.2407	0.1612	0.2408	0.2426	0.2408	0.2405	0.2414	0.2306	0.1539	0.2407	0.2747	0.2729
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0.1438	0.2309	0.2292	0.2309	0.2233	0.2142	0.2142	0.2059	0.2161	0.1078	0.2238	0.2309	0.1170	0.2202	0.2206	0.2289	0.0892	0.1343	0.1182	0.1480
26	0.0479	0.0770	0.0764	0.0770	0.0744	0.0714	0.0714	0.0686	0.0720	0.0359	0.0746	0.0770	0.0390	0.0734	0.0735	0.0763	0.0297	0.0448	0.0394	0.0493
27	0.2171	0.2051	0.2065	0.2051	0.2055	0.2059	0.2059	0.2527	0.2091	0.1459	0.2034	0.2051	0.3062	0.2076	0.2090	0.2212	0.0754	0.2912	0.3333	0.2772
28	0.2404	0.2021	0.2102	0.2021	0.2096	0.2061	0.2061	0.2527	0.2114	0.2601	0.2057	0.2021	0.3264	0.2054	0.2034	0.1937	0.2843	0.2912	0.3333	0.3368
29	0.0747	0.0738	0.0696	0.0738	0.0703	0.0775	0.0775	0.0420	0.0725	0.0561	0.0741	0.0738	0.0976	0.0737	0.0736	0.0501	0.1150	0.0787	0.1082	0.1118
30	0.2242	0.2215	0.2087	0.2215	0.2109	0.2326	0.2326	0.1260	0.2176	0.1682	0.2224	0.2215	0.2929	0.2211	0.2209	0.1502	0.2843	0.2360	0.3247	0.3354
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0.2404	0.2426	0.2415	0.2426	0.2404	0.2413	0.2413	0.2527	0.2407	0.2601	0.2408	0.2426	0.3264	0.2405	0.2414	0.2306	0.2843	0.2912	0.3333	0.3368
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0.2312	0.2165	0.2234	0.2165	0.2195	0.2095	0.2095	0.1687	0.2202	0.2601	0.2127	0.2165	0.3092	0.2173	0.2166	0.2040	0.2843	0.2912	0.3190	0.3368
35	0.0771	0.0722	0.0745	0.0722	0.0732	0.0698	0.0698	0.0562	0.0734	0.0875	0.0709	0.0722	0.1031	0.0724	0.0722	0.0680	0.1417	0.1123	0.1063	0.1409
36	0.2404	0.2021	0.2102	0.2021	0.2096	0.2061	0.2061	0.2527	0.2114	0.2601	0.2057	0.2021	0.3264	0.2054	0.2034	0.1937	0.2843	0.2912	0.3333	0.3368
37	0.1906	0.2033	0.2023	0.2033	0.2067	0.2082	0.2082	0.2544	0.2096	0.1472	0.2039	0.2033	0.1800	0.2055	0.2088	0.2107	0.0603	0.1929	0.1850	0.1623
38	0.0394	0.0743	0.0688	0.0743	0.0716	0.0731	0.0731	0.0440	0.0680	0.0333	0.0745	0.0743	0.0275	0.0748	0.0713	0.0619	0.0213	0.0316	0.0278	0.0279
39	0.1181	0.2228	0.2064	0.2228	0.2149	0.2192	0.2192	0.1321	0.2040	0.0998	0.2236	0.2228	0.0825	0.2244	0.2138	0.1856	0.0640	0.0948	0.0835	0.0838
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0.2371	0.2430	0.2411	0.2430	0.2405	0.2408	0.2408	0.2544	0.2424	0.1891	0.2430	0.2430	0.1589	0.2391	0.2393	0.2423	0.1380	0.1589	0.1450	0.1627

Figure 9: part of feature on HOG extraction

4.1 Metric result

Approaches	*FAR*	*FRR*	*AER*	Accuracy
Gabor+BP	31.4%	3.3%	18.5%	81.5%
Yolo V2	0.0%	1.5%	1.4%	98.5%
HOG+CART	10.0%	6.7%	8.0%	92.0%

Table 1: Performance Table

4.2 Discussion

From the test, Yolo V2 get the best performance, the hog is in the middle, and the Gabor arrange the last in the performance ranking list. The methods all have disadvantages, the `K210` will be affected by the light condition, so its performance is unstable, also the Gabor performance will change greatly aiming for different people's signatures. The paper metrics only show the verification accuracy of the signatures from different angles, as the kind of result is what public focus. However, in the authentic situations, other parameters are also important. The verification speed is a significant part, the result shows the average speed of k210 is 62ms per signature as the system and environment has been set greatly, the average speed of hog method is about 0.1s per signature which is in the normal horizons. However, the average speed of Gabor method is up to 2s per signature, which may be too low-efficient for users.

5 Conclusion

In this paper, we proposed a useful offline signature verification system. The two main contributions of this paper are a new offline verification method using a color image and histogram, and comparison of the *EER* for each classifier. The performance of the proposed system was evaluated by conducting experiments using two our own data set . We empirically confirmed the effectiveness of the proposed method. Additionally, the results indicate that the BP neural network with Gabor filter method get the best results in most circumstances. In the future, we wish to utilize large amount of signature data to train our model. In this way, we are capable to make a standard to choose the proper feature extraction method to verify the different kind signatures. Also, we wish to optimize the algorithms for better performances.

References

[1] K. Matsuda, W. Ohyama, and T. Wakabayashi, "Multilingual-signature verification by verifier fusion using random forests," in *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2017, pp. 941–946.

[2] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verification—literature review," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–8.

[3] M. Ferrer, J. Alonso, and C. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 993–997, 2005.

[4] X. ran Li and F. zhen Hao, "Online handwritten signature verification based on siamese and bigru," pp. 91–95, 2020.

[5] M. K. Kalera, S. Srihari, and A. Xu, "Offline signature verification and identification using distance statistics," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 07, pp. 1339–1360, 2004.

[6] K. Zhi-hui, "A research on signatureverification based on 2d gabortransform and neural network," *Journal of Hebei Software Institute*, no. 4, 2016.

[7] J. Qing-yun, "Method of off-line signature recognition based on improved lpp and ecoc-svms," *Computer And Modernization*, vol. 278, no. 10, pp. 78–82, 2018.

[8] B. Shekar and R. Bharathi, "Eigen-signature: A robust and an efficient offline signature verification algorithm," in *2011 international conference on recent trends in information technology (ICRTIT)*. IEEE, 2011, pp. 134–138.

[9] M. Xiaoqin and S. Qingbing, "Handwritten signature verification algorithm based on lbp and deep learning," *CHINESE JOURNAL OF QUANTUM ELECTRONICS*, vol. 34, no. 001, pp. 23–31, 2017.

[10] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Computer Vision and Pattern Recognition*, 2017.

Appendices

A GitHub Repository

The code are all on the github project: https://github.com/Lazer2077/Signature-Vertification-Project/

B Labor Divisions

Li.ZT(Leader): Arrange labor for every members on group, writing the `Python` scripts for data prepossessing and report, help everybody's working.
Luo.T: Data set making. Testing
Wang.YX: Done the Approaches 1: Gabor transform using matlab. Testing
Wang.N:Done the Approaches 2: Perform Yolo V2 on `K210` and making ppt
Wei.ZC:Done the Approaches 3: HOG feature extraction and training CART algorithm.
Yang.YB:Writing the report,testing Yolo V2 on `K210`