Apply OpenCV to Process Raw Image Captured from Camera then Use Scikit-Learn to Perform Multiple Classification Task and Program It into GUI: Handwriting Letter Recognition

Team: Import Success

Zhu Haokai, Li Jingpeng, Xu Xichen, Zhong Zhengxi, Du Jiahe

South China University of Technology

Shien-Ming Wu School of Intelligent Engineering

Intelligent Manufacturing Engineering

Contents

I. Introduction

● background

Letter recognition is a technology that uses computers or other machines to recognize handwritten letters. This technology has emerged with the development of intelligence. As an example, there are some applications that can help teachers correct test papers, etc. The letter recognition technology proposed in this project requires a camera to capture video stream, and use trained classification models to determine the letters in the picture. Therefore, this project has demand for basic knowledge about machine vision and image process. And in order to make it more user-friendly, this project has designed a set of easy-to-understand human-computer interaction interfaces.

● Problem

In the process of implementing letter recognition technology, there are several issues that must be considered:

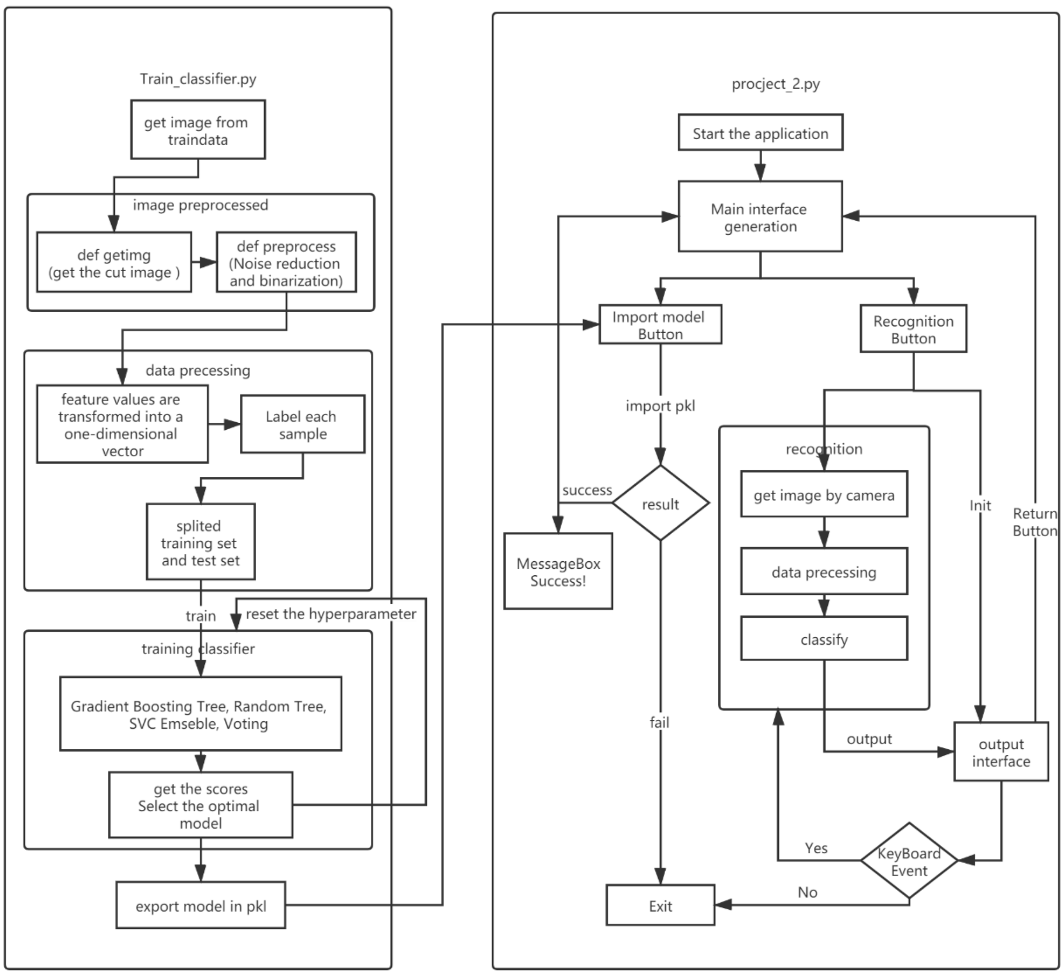
1. How to improve the performance of classifiers?
2. How to make the computer capture the recognized pictures correctly, if the dimension of the pictures is too high, how to deal with these pictures?
3. After capturing the picture, how to correctly locate the area containing the letter and extract the characteristic contour of the letter area?
4. In business considerations, how to design a good human-computer interaction interface to promote the promotion of programs and technologies?

● Solution

This project proposes a more complete technology that uses data processing and image recognition technology to realize real-time letter recognition. The steps are as follows:

1. How to get required data from raw pictures.
2. Use data to train multiple classifiers, and according to the results of the classifiers, design the optimal classifier， and save the classifier to .pkl file.
3. Design the graphical user interface, use the camera to capture the frame to be trained, and pass each frame to the classifier.
4. Get the classification result and send it to the GUI interface by the computer, waiting for the next step.

The flow chart of our project is as follows.



● Goal

The expected goal of the experiment is firstly that the program should have good robustness and can accurately process the video captured under different environments and light. In addition, the classifier also has a higher accuracy (up to 90% or more), which is to ensure the reliability of letter recognition, Finally, the interface should provide button for training model, image recognition and the previous page, result output bar close button and other basic functions. Through the above three requirements, this project will build a complete system.

II. Teamwork

Member	Contribution
Zhu Haokai	Searching relative information Code writing of classifiers
Li Jingpeng	Code writing of data preprocessing and classifier Code writing of image recognition
Xu Xichen	Code writing of image recognition Code writing of GUI design
Zhong Zhengxi	Searching relative information Code writing of image preprocessing
Du Jiahe	Searching relative information Give the first and second phases' presentation
All	Topic selection and data selection Proposal writing The production of the first and second phases' PPT Reporting writing

III. Methodology

- The data used in this project comes from Github. The specific link is https://github.com/realchoi/CharacterRecognition/tree/master/CharacterRecognition, which provides a rich training set of images. The blogger promised the legality of the data, and he used this training set to complete the graduation design.

- In addition, the Python compilation environment used in this project is Python3.7, and the machine vision part uses the OpenCV-Python interface provided by Intel company. The interface design uses the PyQt5 library provided by Anaconda, while the classifier algorithm mainly uses the functions provided by scikit-learn library, which is specified as follows:

library	**Description**
numpy	Provide the data type of array for Python to facilitate compatibility with the opencv interface, it also used to extract features as the matrix
pandas	Read the matrix formed by the image features for easy processing
cv2	Opencv for Python, which provides a large number of internal functions for the Python interface for image recognition preprocessing
Bagging Classifier	The library of the classifier using the bagging algorithm
GradientBoosting Classifier	The library of the classifier using the Gradient boost algorithm
SVC	Provide SVC as a basic classifier for classifiers of integrated algorithms
RandomForest Classifier	The library of the classifier using random forest algorithm
Voting Classifier	A classifier that "votes" to produce the final label based on the results of multiple basic classifiers
QtGUI	The library providing basic controls used in Qt GUI design
QtWidgets	The library providing some parts and the initialization of each interface
QtCore	The library providing some core functions of Qt and encapsulated libraries and some "syntactic sugar"
pickle	Use this library to save the trained classifier to avoid the problem of retraining every time open the application

- The preprocessing of data and pictures can be divided into the following two aspects: access train data and classifier training.

Accessing train data:

Since the data sets used are many pictures containing a single letter, which means that the training set cannot be directly used for training without preprocessing.

(1) Firstly, classify the data sets (pictures) in alphabetical order, save different letter pictures in different folders, and give letter their own labels.
(2) Secondly, convert the RGB image to grayscale and use Gaussian blur operator to remove noises. Then perform edge contour processing and Sobel operator noise reduction on the obtained image to get the final image.
(3) Extract the final binary image into a high-latitude matrix, in addition, use the correct compression algorithm to compress the dimensions of the image without compromising accuracy, and each final matrix contains each image's information.
(4) Use the obtained multiple matrices and labels to train the classifier.

- **Selection and training of classifiers**

  In order to pursue higher accuracy, the method adopted in this project is to train multiple classifiers and design the optimal classifier based on the obtained classification results. The types of classifiers used include but are not limited to generated classifiers of random forest, decision tree and other algorithms, and Bagging classifier using support vector classification (SVC) as sub-classifier is also trained. The priority of choosing classifiers are accuracy, f1-score, predicting time, respectively. Finally, the pickle library will be used to package the classifier, so that the optimal classifier can be directly called after training, avoiding the problem of retraining every time the project is opened.

- **Capture of the image to be recognized**

  When the camera is turned on, the user can manually select the video frame to be recognized through keyboard events. Once the corresponding video frame is captured, the program will immediately locate the area with large color difference in the video frame, and intercept a complete rectangular area as the image to be recognized according to the outer contour of the area. After preprocessing the image, the trained classifier will obtain the type of the letter, which is regarded as output shown.

IV. Algorithm

● Image recognition and its preprocessing

In this task, we use the functions in openCV library for image preprocessing.

- GaussianBlur( ): Gaussian filter is used to smooth the image. This function convolutes the source image with the specified Gaussian kernel.
- cvtColor( ): It is used to convert different color spaces in the image. We convert true color RGB image to gray image (COLOR_ RGB2GRAY)
- Canny( ): Edge detection of input image.
- Sobel( ): It can smooth the noise and provide more accurate edge direction information, but the accuracy of edge location is not high enough.
- convertScaleAbs( ): Change image format from CV_ 16S goes back to the original uint8 form.
- addWeighted( ): Overlap the image. When Sobel operator is used to denoise, the X direction and Y direction are divided into two parts. After processing, they need to be overlapped.
- adaptiveThreshold( ): to determine the binarization threshold of the pixel position according to the pixel value distribution of the neighborhood block of the pixel
- getStructuringElement( ): Get a custom kernel
- morphologyEx( ): Open or close operation of the image. We use open operation to remove the noise.
- resize( ): Force image size change. Finally, we size the image to 30 * 30

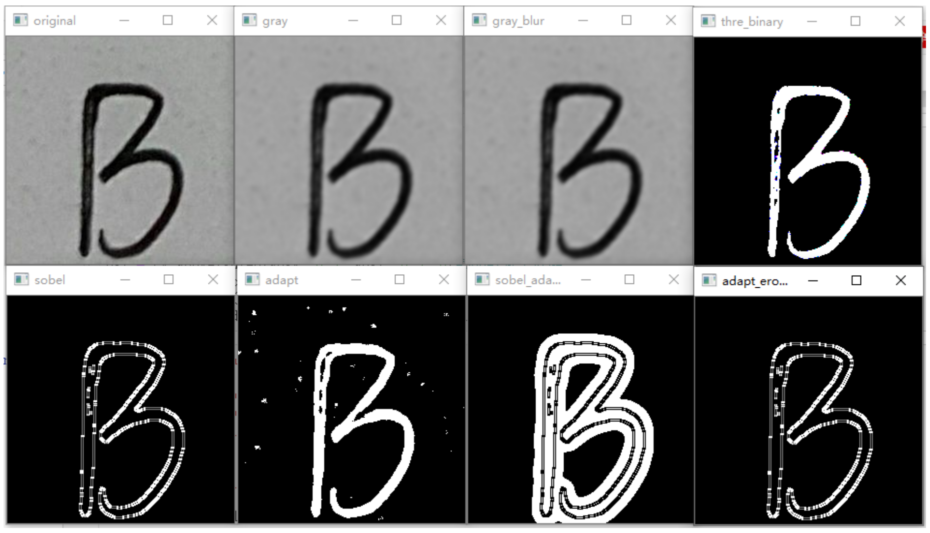
● Algorithms used in classification

- Random Forest: RF uses cart decision tree as weak learner. On the basis of using decision tree, RF improves the establishment of decision tree. RF selects n samples from random samples returned from sample set N, and then selects K features from all features to generate a single random decision tree, which further enhances the generalization ability of the model.
- MLP: Multi-Layer Perceptron classifier with 100x100 sized hidden layer, the activation function logistic and use 0.0002 as the regularization parameter.
- SVC_Bagging: For support vector machines classifier (SVC), we select two kernels to training the model. The first is “RBF” that the reason is this kernel is infinitely differentiable and covariance function has mean square derivatives of all orders, and are thus very smooth. The second is “Linearly”, because the feature matrix of image is sparse which is easily to linearly divided.
- Gradient Boosting: The basic idea of boosting is to make each round of base learners pay more attention to the samples of the previous round of learning errors in a certain way. In gradient boosting, the negative gradient is taken as the measurement index of the previous round of learning errors, and the errors made in the previous round are corrected by fitting the negative gradient in the next round of learning.
- Voting: Soft Voting Classifier, the average value of the probability of a certain category of all model prediction samples is taken as the standard, and the corresponding type with the highest probability is the final prediction result.

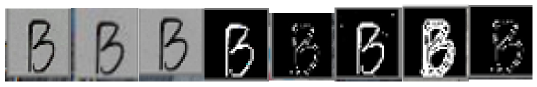
V. Experiment Results

● Image preprocessing

Different methods and their combinations of image preprocessing are used to get better features of the training set. The following figure shows an example of the effect of preprocessing. The methods used are gray transform, gaussian blur, simple binary threshold, Sobel filter, adaptive threshold, combination of Sobel and adaptive threshold, combination of Sobel and adaptive threshold and corrosion, respectively.

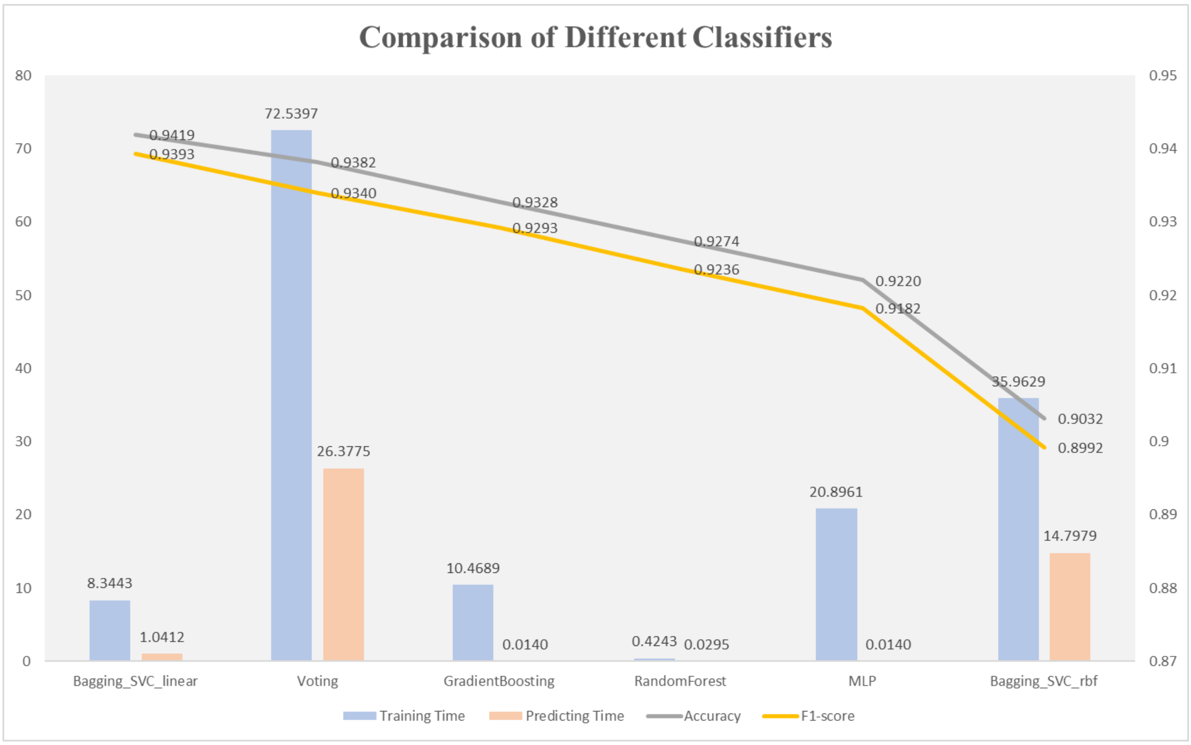


After image operations, the picture is resized to be 30 by 30 pixels, as shown in the following figure.



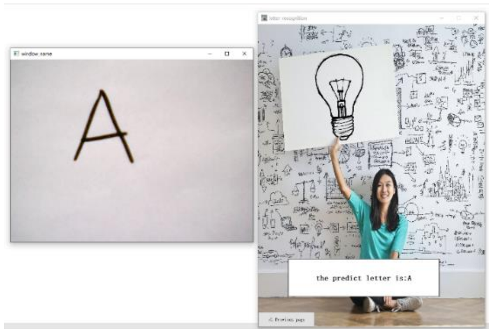
● Classifier training

The trained classifier models and its performance are shown in the following figure.

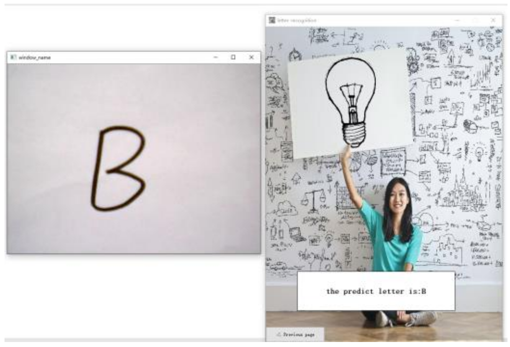


● **Hand-writing letters recognition**

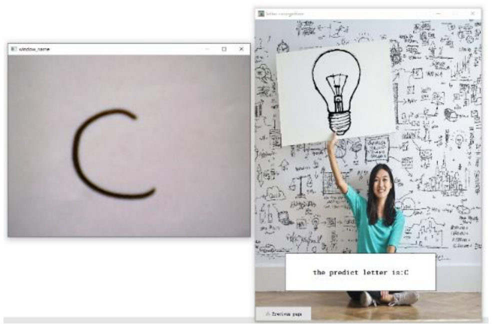
The following pictures are the screen shots when running our program. The left shows the real time recognition and the right shows the result. After some tests the letter recognition is very robust.



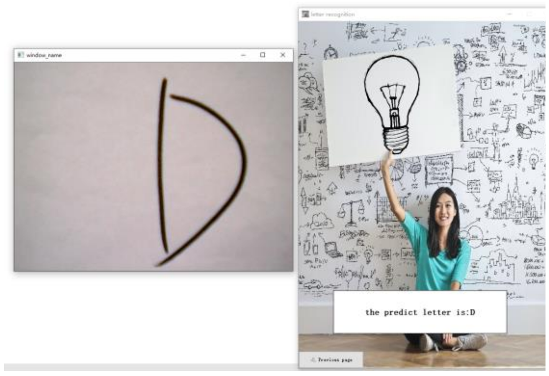
the predict letter is:A



the predict letter is:B



the predict letter is:C



the predict letter is:D

VI. Discussion

● Problem-solving

■ Number of features and training speed

Too high picture pixels will result in too many sample features, resulting in slower training speed. Appropriate dimensionality reduction can increase the training speed while ensuring accuracy, and it is also conducive to the preservation of data in csv files.

■ Eliminate the effect of different letter positions

When a simple processed picture is put into the classifier, the position features of the letters in the picture will be considered, and the positions of the letters in the training set are concentrated in the middle. In the actual process of using a camera to recognize letters, it is difficult to ensure that the positions of the letters are uniform in the center, so there will be misjudgment of classification caused by inconsistent sample distribution. Therefore, it is necessary to consider how to avoid the influence of letter position, which can be solved by locating the letter in the picture. By locating the letters of the picture, the area of the picture that contains the letters can be intercepted. Using this area as a sample can remove the influence of the position of the letters in the initial picture on the accuracy.

■ Uneven shooting brightness

The general binarization process is only applicable to the situation that the brightness of the picture is consistent, and the inconsistent brightness will cause the effective information in the picture to be removed or more interference information is added. In the actual situation, the camera cannot guarantee to shoot at the same brightness of each position, which leads to other light and dark features caused by illumination in addition to the letters, which will affect the training of the classifier. At this time, you need to choose adaptive binarization, because adaptive binarization can adjust the threshold according to different light intensity.

■ **Effective feature extraction**

After the image is compressed after gray-scale conversion, binarization and edge detection, the number of effective features obtained is relatively small. Therefore, it is necessary to increase the number of features. After using the feature multiplication algorithm to increase the number of features, the recognition rate does not change significantly, and even leads to a decrease in accuracy. This may be because the algorithm itself only multiplies the original feature columns, but in the project, a sample is a picture, and the feature is determined by the value and position of the pixel of the picture. Most of these feature values are 0, which leads to the reduction of effective features after multiplication. Therefore, on the basis of the original image, we use edge detection and binarization twice, and add morphological transformation to obtain a picture with more letter outline features than simple binarization and edge detection. The accuracy rate is greatly improved.

● Classifiers selection

- Random forest and SVM have good effects on most data sets, so they can be used for training data first, and the accuracy obtained can be compared with other classifiers to determine the effect of the classifier.

- Each letter has different characteristics. Some letters whose characteristics are not obvious or are similar with those of other letters are easy to be misidentified. Random forest does not have the function of classifying these difficult-to-recognize characteristics, so it is necessary Use the Gradient Boosting classifier to train the classifier for difficult-to-recognize features by applying weights to improve model accuracy.

● Comparison of different classifiers

- ■ We use random forest, voting classifier, Gradient Boosting classifier and bagging classifier based on SVM for ensemble learning.

- ■ From the perspective of training time, the Voting classifier is the longest. This is because the Voting classifier integrates other classifiers. The vector machine itself calculates the training time is long but the bagging classifier only consists of 10 base SVM estimators. Random forest training takes the shortest time, because the base classifier decision tree of random forest requires shorter training time. Gradient boosting use gradient boosting in the process of training the decision tree in the later part and it takes time to calculate, so it will take longer than random forest training.

- ■ Voting classifier also has relatively high F1score and Accuracy, which shows that its base classifier has relatively high accuracy and uniqueness. However, the Voting classifier training time is relatively long, which is not conducive to debugging, and the local classifier saved by the Voting classifier occupies a large memory. Random forest and bagging classifier base on SVM and not only have short training time, they also have good performance on Accuracy and F1score, and they take up less memory. Gradient boosting classifier takes resources and time to calculate and both of gradient boosting classifier and random forest do not have high generalization compared with SVM. Comprehensive time, space costs, generalization and robust characteristic, bagging SVM classifier is the most suitable for this project.

● The applicability of the project

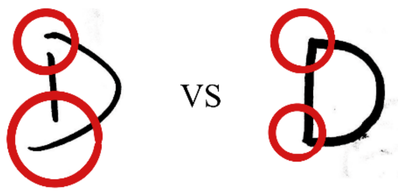
The topic of English letters recognition focused on in this project can be generalized to character recognition. With corresponding dataset, single hand-writing characters like Roman numeral or Greek letters can be recognized in a similar way.

Additionally, the methodology applied in this project is the fundamental of identifying a series of characters, for instance, English vocabulary and license number. The preliminary idea is to divide the area of each characters in the image, and processing them with the method designed in this project, then output the combination of these characters.

● Limitation

■ Insufficient of data samples

The number of total samples used is about 1240, which is not so satisfying for building models for a classification problem. Since the hand-writing styles are vary from person to person, and the difference of the connection of the strokes between training set and test set will also lead to mistakes in recognition.



Therefore, the shape of letters which different from the one of training samples will have higher possibility to be misclassified. If more samples are provided, the accuracy of recognizing from the image read from camera may be higher.

- Sensitive to noise

  The program is sensitive to noise, the specific performance is when there are creases on the paper the accuracy will decrease. Although the extra points and fine lines on the paper can be used to reduce the noise by opening and closing operations during data preprocessing, the creases of the paper are similar with the width of the outline of letters. If they are processed away, the features of the letters that need to be recognized It may also be eliminated. Therefore, the difference between features extracted from train and test samples will interference the detection of the letter and cannot predict what letter it is.

- Image quality

  Our samples are pictures with white paper and black characters. When the light is very weak, the chromatic aberration between the handwriting letter and the background is very small, that is, the difference between their pixel values is very small, and there will be extra edges detected by the model. At the same time, if the background and handwriting colors are changed to other colors, the accuracy of the classifier may be reduced, and the parameters of the function need to be adjusted to adapt to different situations.

● Further improvement

- Noise processing

  In the expected result, we intend to apply it to the answer entry on the answer sheet because there is a rectangular box on the answer sheet.

Therefore, our initial idea is to use rectangle recognition algorithm in OpenCV to locate the rectangular frame after the regular preprocessing of the read image, and extract the image in the rectangular frame as a sample for classification. But in the process of recognition, we found that the rectangle will affect our recognition, because the edge of the rectangle will also be placed in the image, adding noise to the image, thus affecting the recognition accuracy.

The actual image can be reduced by locating the rectangular frame, that is, reducing the pixel length and width of the picture, so that the rectangular frame is not entered into the image. However, in actual shooting, there is a certain angle between the rectangular frame on the paper and the captured picture, making the actual rectangular frame appear as a parallelogram in the picture, and the OpenCV rectangular frame cannot be completely matched. Therefore, the compression of the positioned picture needs not to affect integral letter recognition, that is, to find a balance between letter recognition and removing the influence of the rectangular frame on the paper.

■ Semi-Supervised Learning

Considering that our sample size is small, with only about 300 samples per letter, and the diversity of samples is insufficient, we hope to use semi-supervised learning to optimize our model. The semi-supervised learning process does not rely on external consultation and interaction, and automatically uses the distribution information contained in the unlabeled samples, that is, the training set contains both labeled sample data and unlabeled sample data. This method can also obtain more samples without wasting manpower and material resources. In our project, the original training set is labeled samples, and every time the image camera reads the recognized will be added as an unlabeled sample to the total training set, thus realizing real-time tracking. We hope to enrich our training set through semi-supervised learning, so as to improve the generalization ability of the model and obtain higher accuracy.

VII. Innovation

- Data acquisition

  The methods of data acquisition: Based on the camera and OpenCV library, the handwritten letter image is preprocessed to obtain the required feature matrix as a test set.

- Images feature processing

  The acquisition of training data is not directly obtained from the existing database with numerical data, but a numerical matrix obtained through a series of images feature processing from the real image data, which is used as the sample of the classifier.

- Novel noise reduction algorithm.

  Simple gray scale conversion, Gaussian filtering, binarization and canny edge detection have fewer effective features of letters in the picture, and the classification effect is poor. On the basis of the original image processing, we use a further sobel algorithm and adaptive binarization to obtain information with more effective features, which is manifested in the fact that more contours of letters are added to the image, thereby effectively improving the classification The accuracy of the device.

- Ensembles

  The classifier does not use a single classifier, but uses multiple ensembles, and uses grid search to optimize the hyperparameters of the model, so that the model has a better classification effect for the classification task performed by the project.

- Hardware interface

  We provide a hardware interface for the classifier, that is, we combine the classifier with the camera, and write an interface to recognize the letters contained in the incoming image from the camera. On the one hand, we combine software and hardware; on the other hand, we have successfully used our own data.

- Multi-classification problem

  The classification problem studied is not a simple two-classification problem, but a multi-classification problem with strong extensibility, that is, it can be applied to more data types than the item label.

VIII. Conclusion

This project is a real time hand writing letter recognition based on machine learning algorithms and computer vision with a graphical user interface which improves its utility. 1240 raw images containing letters A B C and D are used as training set. After preprocess of image and improvement of classifiers, a robust classifier was trained.

We achieved real time different writing styles letters recognition in a short time with high accuracy. However, we have not applied it into specific areas to solve more general problem, such as examining multiple choices automatically and natural language processing. If more time is given, deeper research would have been done and more interesting applications would have been designed.

During the project, we have learned how to do simple image process, how to get the required training data, train and improve classifiers, apply machine vision knowledge in a project, design a GUI. More importantly, we have learned how to improvement a project to make it practical and how to apply it into real problem solving.