­[[1]](#footnote-1)

Machine Learning Final Project Report

Walabon, Yen-Pin Hsu, Chengyin Liu, Team ANNoying ddd

*Abstract*—in this project we aim to create a robust character recognition by using machine learning method via state-of-the -art features. The characters that we are going to recognize is the Chinese animal zodiac hand writing sized with a random quarter being blocked. The two main feature extraction method is Histogram of Oriented Gradient (HOG) and Local binary patterns (LBP) and the main classifier we use is Support vector machine (SVM).

# INTRODUCTION

The hand-writing recognition has been studied for many decades. In this project, we are about to recognizing the hand-writing of the Chinese animal zodiac which is written by the students who took this class. In order to raise the difficulty of the recognition job. The original image was divided in to four equally segments, left-top, right-top, left-down, and right-down. For each hand-write image there is a quarter blocked randomly. Our job is to use the training images to train a model to recognize the testing images.

We divided the full task in several parts which will discuss in detail in the later pages. For the training part, there are four main steps, Data filtering, preprocessing, feature extraction, and training. For the preprocessing, first we filter the insignificant hand-writing images. The definition of insignificant data will be define in the METHOD section. By the filtering mechanism, we can get a more ideal training data set from the original training set since there is less noise in the data set. After we done selecting good training data, we do preprocessing to the selected data in order to make the appearance of the hand-writing image to be more uniform. For the location and size, we normalized the image by defining the bounding area if the Chinese character in image. Then we fit the longer edge of the bounded character into a pixel image and put it in the middle. Every character’s size and location is similar within the same class after the preprocessing procedure. Then to lower the intra-validation further, we apply thinning to lower the influence of the different styles of hand-writing.

The feature extraction method we select is Local Binary Patterns (LBP) and Histogram of Oriented Gradient (HOG). These two are both the state of the art method in computer vision and pattern recognition field. LBP is the particular case of the Texture Spectrum model proposed in 1990 and first described in 1994. It has since been found to be a powerful feature for texture classification. The LBP feature vector is calculated by the following steps. First we divide the image into cells, for each pixel in a cell, we compare the pixel to each of its 8 neighbors. When the center pixel’s intensity value is greater than the neighbor’s value. We give it a 1, else 0. This gives an 8 digit binary vector. Then we computer the histogram of the occurrence frequency of each kind of 8 digit binary vector over all the cells. Then we normalized the histogram for each cell and concatenate then as the LBP feature vector.

|  |
| --- |
|  |
| Fig. 1 Example of HOG descriptor on pedestrian detection (a) original image; (b) HOG image |

HOG are feature descriptors for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portion of given image. *Navneet Dalal* and *Bill Triggs* first described HOG in there June 2005 CVPR paper. In their work they focused their algorithm on the problem of pedestrian detection in static images. The essential thought behind the HOG descriptor is the local object appearance and shape within an image can be described by the distribution of intensity gradients. Since the local object appearance and shape is also two key points to recognizing the hand-writing of Chinese character. We select HOG as another feature extracting method. The implementation of HOG can be achieved by dividing the image into small connected regions so call cell. The combination of these histograms then represents the descriptor. The local histogram is normalized by calculation a measure of the intensity across a larger region of the image called block and normalize all cells within the block with this value. The detail for both features is shown in the METHOD section Feature Extraction part.

For the training section. We choose support vector machine (SVM) with linear, polynomial, and Radial basis function kernel. We get the parameters to train the model by doing cross validation.

The testing data will do the same preprocessing and feature extraction but not data filtering since we have to recognize every input images even if the input data is impossible to identify. After we translate the image into a feature vector, we put the feature vector to the trained model to finish the recognition job. The Result for different setting of preprocessing and feature extraction with different kernel of SVM will be shown in the Experiment section. Then we will give a conclusion about what we found and learned from this project on the conclusion section.

# Feature

|  |
| --- |
|  |
| Fig. 2 Character recognition system flow |

In this project, we try to recognize hand-writing Chinese characters by a sequence of functions. To apply learning method, the extraction of feature is necessary to get the meaningful information from training data. Nevertheless, selecting “good” training data is also a critical issue to build a good model. Combining the issues above, we constructed the system as Fig. 2. In each stage, different kind of method can be applied to get the result.

|  |
| --- |
|  |
| Fig. 3 Preprocessing: (a) original image; (b) normalized, dilated image; (c) thinned image |

* 1. Data filtering

No matter how good a method is, performance will be limited if we give bad training samples. Therefore, a filtering mechanism should be defined to filter the bad ones within the whole training sample set.

In this project, the target data is Chinese character. Most of the Chinese character is similar to rectangle, which means that the aspect ratio should not be too large or too small. Moreover, the data which is too small is also considered as bad data.

By this filtering mechanism, we can get a more ideal training dataset from the original training sample.

* 1. Preprocessing

After filtering the training sample, we still have something to do before extracting the feature from the data.

The goal of this method is the recognition of hand-writing characters. These characters can be written by different people, which results in great intra-class variation. Therefore, a normalization in order to make the appearance to be more uniform is necessary.

The variation can be classified into several kinds of situation: the location of the character within the image, the size of the character, the writing style of different writer…etc.

For the location and size, we do the normalization by defining the bounding area of the character in image. Then, we fit the longer edge of the bounded character into a pixel image, and put it in the middle. After this procedure, every character’s size and location should be similar within the same class. For the writing style, we use thinning to lower the influence of the different people’s writing.

After the normalization, we use basic image morphology to make the image more uniform. We use the combination of erosion and dilation to get the result to the next step.

* 1. Feature Extraction

To train the model, feature extraction that is able to represent the data characteristic is necessary. Therefore, we use several kinds of method to extract feature from the data, including some well-known method that is commonly applied in some detection algorithm.

* + 1. Two Dimension Histogram (TDH)

Each character has its own appearance. At the same time, the density of different part of a character will be different.

|  |  |  |
| --- | --- | --- |
|  |  |  |

First, we turn the gray scale image into binary image by formula (1) for every point within the image. stands for the intensity value in the original image, and stands for the new intensity value.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Then, we calculate the histogram by formula (2) in two directions, which means the x-axis in horizontal and y-axis in vertical. By connecting two histogram sequences, we can get the -dimensional feature vector.

* + 1. Local Binary Pattern

Local Binary Pattern (LBP) is a feature that is frequently used in detection. LBP is composed of a sequence of histogram, which is decided by the pattern distribution within a unit area.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

To apply LBP, we first divide the image into several blocks, for example, blocks. Then, every block will be a basic unit which gives a histogram. We calculate the pattern pixel by pixel within the block. Every pattern is decided by considering the 8 neighboring pixels. is a point of block, and is the neighboring point of . As formula (3) and (4), we can get the corresponding pattern .

However, since there are 8 neighbors for each point, there will be different combination of patterns. Since that most of the pattern should be smooth, we define the pattern that have less than 2 0-1 or 1-0 transition as *Uniform Pattern*. By considering uniform pattern, we can reduce the dimension from 256 to 59.

|  |  |  |
| --- | --- | --- |
|  |  |  |

After we define the bin index by the pattern, we vote by every pixel’s pattern to get the 59-bin histogram. stands for the histogram bin value of pattern . Therefore, we can get , that is the histogram of .

At last, we concatenate all the histograms of every block to get a -dimensional feature vector.

* + 1. Histogram of Oriented Gradient

Histogram of Oriented Pattern (HOG) is another feature that has outstanding detection performance. HOG is composed of a sequence of histogram. HOG represents the data by considering each pixel’s intensity gradient, which can somehow emphasize the edge information within an image.

|  |
| --- |
|  |
| Fig. 4 HOG block structure: blue: entire image; red: block; green: cell |

Similar to LBP, target image is divided into blocks before applying the HOG algorithm. With a slightly difference, the block is considered overlapped in HOG. Each block is divided into cells. Cell is the basic unit. Therefore, take the image for example, if we set the block as , the cell will be . Thus, the stride is 20 pixels (Fig. 4.) Finally, there will be 25 blocks in total.

To calculate the gradient information, we define a fixed number of directions to be examined. For each cell, we calculate the gradient distribution by dividing the direction into 9, which means there will be 9 bins for the histogram. Then, concatenate the 4 histograms to form a feature vector for a block. Finally, concatenate all the feature vectors of blocks, we get the final feature vector. For the parameters above, we get a -dimensional feature vector.

# Classifier

After extracting the feature from training data, we have tried several kinds of classifier to determine which is suitable for hand-writing recognition, including some popular models which are commonly applied in these classification problem.

1. Decision Tree

A decision tree is a flowchart-like structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes). A path from root to leaf represents classification rules.

It is simple to understand and interpret. People are able to understand decision tree models after a brief explanation. Its worst, best and expected values can be determined for different scenarios. And it can be combined with other decision techniques.

But for data including categorical variables with different number of levels, information gain in decision trees are biased in favor of those attributes with more levels.

We use the C4.5 algorithm to implement the classification of Chinese characters. But it does not have a good performance since each dimension of the HOG feature does not have meaningful information.

1. AdaBoost

We tried AdaBoost algorithm to improve the performance because it can be used in conjunction with many other learning algorithms. Adaptive Boosting is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. It is sensitive to noisy data and outliers. So in this hand-writing problem, it is so easy to be affected duel to the not enough training data.

AdaBoost generates and calls a new weak classifier in each of a series of rounds. For each call, a distribution of weights is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification.

Unfortunately, the number of iterations and the base algorithm for AdaBoost are hard to choose. And decision stump may not be a good base algorithm in this problem.

1. Support Vector Machines

SVM constructs a set of hyperplanes in a high or infinite-dimesional space. A good separation by these hyperplanes can be used for classification. The mappings used by SVM schemes are designed to ensure that dot products may computed easily in terms of the variables in the original space defined as kernel function. The selection of the kernel function and the parameters we used for the chosen kernel function is the main issue that we are going to discuss.

In the support vector machine part we use the famous library called LIBSVM which is developed by Chih-Ching Chang and Chih-Jen Lin. We test three different kinds of kernels, linear kernel, polynomial kernel and radial basis kernel. The linear kernel is defined as u^'\*v, the polynomial kernel is defined as [(γ\*u^'\*v+coef0)]^degree and the radial basis kernel is defined as e^(-γ\*[|u-v|]^2 ). We use linear kernel svm model as our rudimentary test. Once we done feature extraction for the training set. We constructed the linear svm model and fed the testing data to it. If the accuracy is greater than 50% we will try the polynomial kernel and the radial basis kernel further. The parameter we use for them is by doing cross validation via tool called grid which is also included in the library. In the experiments we can tell that the RBF kernel usually gives the best accuracy out of the three kernels.

1. K-Nearest Neighbors

The *k-*NN algorithm is among the simplest of all machine learning algorithms. It is a non-parametric method used for classification and regression.

Because there will be some much similar hand-writing of the Chinese animal zodiac in the training data and testing data, we choose *k-*NN to compare the results with SVM.

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

We choose different number of k to observe whether the result will be better with larger k in our experiment, but it seems not help.

# Experimental Results

We do a lot of experiments to find the best parameters for our LBP and HOG features, as well as the classifiers which will be affected by the parameters.

Table

|  |  |  |
| --- | --- | --- |
|  |  | SVM |
| Linear kernel |
| LBP | - | 65.559% |
| Dilation | 68.782% |
| HOG | - | 87.638% |
| Dilation | 88.281% |

Table 1 compares the accuracy rate of LBP and HOG and the results of them with dilation. It is easy to see that dilation will improve the performance since it add some noise to the training data.

Table

|  |  |  |
| --- | --- | --- |
|  | SVM | |
| Linear kernel | RBF kernel |
| Raw data | 32.348% | 44.464% |
| LBP | 68.782% | 58.637% |
| HOG  120120 | 88.281% | 93.684% |
| HOG  6060 | - | 92.447% |
| HOG  4040 | - | 91.682% |

Table 2 shows the performance of different SVM kernel with different kinds of feature. Linear kernel is better than RBF kernel when using LBP feature. But RBF kernel with the 120\*120 HOG feature has the best performance in our experiment.

Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Decision Tree J48 | *k-*NN | | |
| *k* =1 | *k* =2 | *k* =3 |
| HOG  120120 | 59.082% | 90.527% | 89.095% | 89.127% |

Table 3 shows the results of Decision Tree and *k-*NN with HOG feature. Decision Tree does not perform well in this problem. *k-*NN has a much better than Decision Tree, but slightly worse than RBF-SVM. And we find that the value of *k* almost do not affect the accuracy rate

Above are the results of our experiment for the first part of the testing data. Most of the best parameters are selected by cross validation.

# Discussion & Conclusion

We have proposed a Chinese character recognition system by using HOG feature and RBF kernel of SVM learning method.

In the preprocessing stage, we have tried several kinds of method, including location and size normalization, dilation and erosion, and thinning. The final image applying all the methods above seems to have a great result. However, image applying thinning method does not have a better performance. After analyzing with the combined feature, we think that it’s because that the thinned image contains the skeleton of the character, but skips the texture-like information. Though the skeleton should be good enough, or even better to describe a character, the feature we selected cannot fit this characteristic well.

In the feature extracting stage, we have tried TDH, LBP, and HOG. Two Dimension Histogram has a serious drawback that it can only consider the pixel density, which might not be precise enough to describe the shape or contour of a character. However, these are the most important part to recognize a character. Besides, the pixel-by-pixel way the count and vote the histogram also leads to that the result is greatly influenced by the uncertainty of hand-writing. LBP has a much better performance comparing to TDH. However, LBP is originally used for describing textural information, which is quite different from what we are focusing in this project. LBP can do quite good in object recognition, but difficult to have an outstanding performance when it comes to character recognition. HOG describes the gradient information, which is outstanding in object and pedestrian detection. Different from LBP, HOG involves the contour information, which is clear when the detecting object has clear contour. Therefore, HOG has the best performance out of three in our experiment.

In the classification stage, we have tried Decision Tree, AdaBoost, SVM, and *k-*NN. For SVM, we also tried three kind of kernel: linear, polynomial and radial basis function. For AdaBoost, we did not get the result successfully. Though we know that it is powerful, the performance is too sensitive to some significant dimensions of HOG feature. So it is difficult to adjust to get an ideal result. This results in serious overfitting situation. For SVM, we use LIBSVM as the training toolkit. When we use linear kernel, most of the feature set can achieve a good performance as long as it is reasonable. When it comes to RBF kernel, it has the best performance.

However, what has to be concerned about when using RBF kernel is the overfitting problem. When the training data amount is not large enough, this might become a serious problem. In this project, we can say that the training data and testing data is actually from the same set of data. Therefore, as long as we did not try to fit the training part, this should not be a problem. For *k-*NN, unlike SVM, the input testing data will try to find few nearest data, which means it will try to fit the data which is actually exist. This might avoid some problem that is caused by the different source of training and testing data, RBF-SVM might make the data to fit a point that is similar to others, but does not actually same to anyone of them. Fortunately, the data for this project does not have this kind of situation. RBF-SVM still outperforms *k-*NN.

Combining the ideas above, this is our final method: for preprocessing, we do location normalization, size normalization, and dilation, results in an image; for feature extraction, we use HOG feature with block size and cell size ; for classification, we use SVM with RBF kernel, with c=8.0, g=0.03125. This method can get an accuracy rate 95.0521% for the second part of the testing data. However, this method performs better when the training data and testing data have a same source. If we change the data source to the data we collected from internet, many factors need to be considered and this method may not have good performance.

References

1. Walabon is with the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, ROC, (e-mail: r01922107@ csie.ntu.edu.tw).

   Yen-Pin Hsu is with the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, ROC, (e-mail: r01922124@ csie.ntu.edu.tw).

   Chengyin Liu is with the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, ROC, (e-mail: r01944040@ csie.ntu.edu.tw). [↑](#footnote-ref-1)