

HANDWRITTEN DIGIT RECOGNITION ON SURVEY FOR EDUCATION

O.Ankhtuya¹, W. G. C. W. Kumara² and Ch. Enkhtuul³

¹Dept. of CSIE, National Central University, Taiwan. Email: ankhaa8@gmail.com

²Dept. of CSIE, National Central University, Taiwan. Email: chinthakawk@gmail.com

³Khumuun Tsogtsolbor High School, Zuunmod, Mongolia. Email: tuul_ch2005@yahoo.com

ABSTRACT

Pattern recognition is used in many fields for different purposes. Especially in education, teachers spent much time to check exams, quizzes and survey data of students' replies in paper-based materials during the impossible condition for students to use the computer. This paper presents recognition of handwritten numbers by using histogram of gradient features and support vector machine classification on survey data for education. First, we took photos that contain the numbers in order to insert data into computers easily. Then those images were passed through many preprocessing operations such as segmentation, binarization, normalization and location, to prepare the image for the recognition. This method was implemented in Matlab to identify handwritten samples of the single digits from 1 to 5. Histogram of Oriented Gradients features were employed in training and test data sets for different cell sizes.

Key words: Handwritten recognition, HOG features, SVMs, digit recognition

1. INTRODUCTION

Carrying out survey questionnaires for different purposes of studies is very important to learn about users views or the service provided. Online surveys are much useful in this regard. But still there is considerable number of people in the world with no access to internet which makes survey data collection from such groups is a difficult task. Instead we tried to have paper and pencil based questionnaire for the students in a remote area with no internet access. And their responses were collected as digital images. Due to the large number of questions in the questionnaire and the large number of students, manually processing this data is a very difficult task. Hence we tried to find a way to handle this automatically and our experience is presented in this paper.

Histogram of Oriented Gradients (HOG) features is a widely used technique in image processing for different kinds of object detection. As an example HOG features are used for human detection in [1].

K. Terasawa and Y. Tanaka presented a word spotting method based on line-segmentation, sliding window, continuous dynamic programming, and slit style HOG feature [2]. Their method is applicable regardless of what language is written in the manuscript because it does not require any language-dependent preprocess.

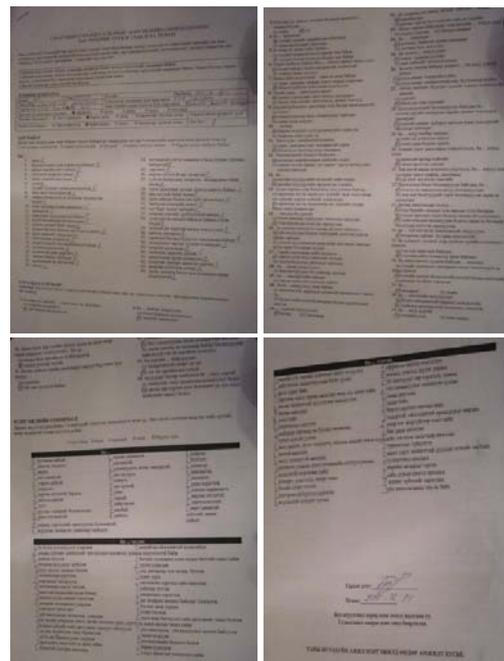


Figure 1: An example response of the questionnaire

M. B. Yilmaz *et al.* presented an offline signature verification system based on a signature's local histogram features [3]. The signature is divided into zones using both the Cartesian and polar coordinate systems and two different histogram features are calculated for each zone: histogram of oriented gradients (HOG) and histogram of local binary patterns (LBP). The classification is performed using Support Vector Machines

(SVMs), where two different approaches for training are investigated, namely global and user-dependent SVMs.

2. METHODOLOGY

A survey was carried out in Mongolia using a four-page questionnaire. Digital images were captured of all the responses from students. An example image for one response is shown in Figure 1. As the images were captured using a regular digital camera and using free hand, captured images first need to be corrected as taken from the same camera position. User is asked to select four points as top left corner of the thick black line, top right corner of the thick black line, left of the date line, and right of the date line (Figure 2 left). Original coordinates of aforementioned four points are (49, 56), (361, 56), (224, 317), and (176, 317) in pixels. Using the original coordinates and the user marked coordinates spatial transformation structure is then found. This provides the transformation matrix as shown in the following example.

$$\begin{bmatrix} 1.15, & 0.14, & 4.35 \times 10^{-4} \\ 0.05, & 1.17, & 3.60 \times 10^{-4} \\ -37.72, & -94.88, & 1 \end{bmatrix}$$

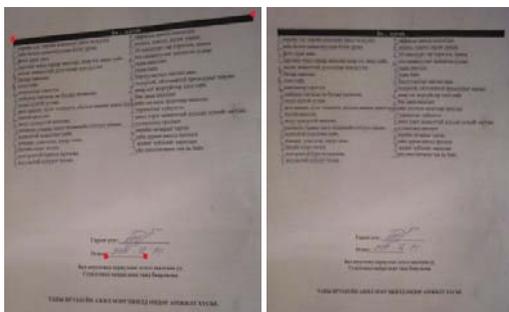


Figure 2: An example camera correction

Applying this matrix on the original image camera corrected image can be found as shown in Figure 2 right.

Then user asked to mark four points again as marked in Figure 3. Points are bottom left of the thick black line, right of the last answer line, left of the first answer in the second column of answers, and right of the last answer line in the second column. Then answer sections of first column and second column is cropped. There are 19 questions in the left column and 17 questions in the right column. Then the separated column are equally divided and cropped to separate each answer of the response. These separated individual answers are then resized to a 16*16

pixel images. An example set of separated answers are shown in Figure 3.



Figure 3: An example separated answers

As can be seen in Figure 3 cropped and separated answers are not yet readable. Several image processing steps are then applied in order to sharpen the answers in the image as follows. First images are converted to grayscale. Then the grayscale images are adjusted for the gray values and an image dependent threshold is found. Finally based on that image dependent threshold grayscale image is converted to a black and white image. Figure 4 shows the preprocessing results for the example image set shown in Figure 3.



Figure 4: Preprocessing results for the images in Figure 3

These detected preprocessed hand written answers were used as the test data set. As the training data set, a set of images available in Matlab was used [4]. Figure 5 shows the images available for 2 digit in the training dataset.

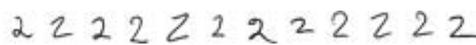


Figure 5: Images for 0 digit available in Matlab

Then HOG (Histogram of Oriented Gradients) features for training and test data sets are found for different cell sizes. Figure 6 shows HOG features found for 3 different cell sizes.

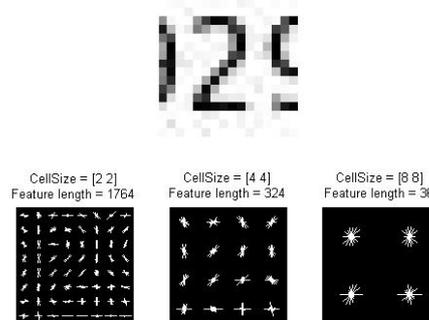


Figure 6: HOG features at different cell sizes [4]

Then a support vector machine (SVM) classifier is trained using the svmtrain command available

in Matlab [4]. Then HOG features are found for the test data set, and use the SVM built in previous step to find the matching numbers.

3. RESULTS

Figure 7 shows an example training and test images before and after preprocessing. Top row shows the training example while bottom row shows the test example. Left column shows the original figure before the default preprocessing. Since our own preprocessing step is used we directly fed the preprocessed image. Right column shows preprocessing results.

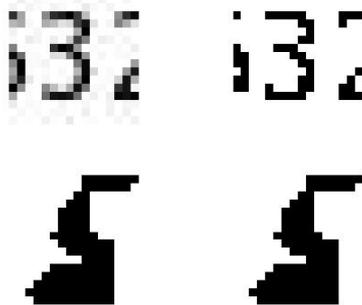


Figure 7: Before and after preprocessing

Table 1 shows the SVM classifier results. We have received the same values for each digit. If digits were recognized correctly higher values should be shown for the correct number.

Table 1: SVM results.

Digit	svm				
	1	2	3	4	5
1	4	0	0	2	0
2	2	1	1	0	0
3	0	0	2	0	0
4	0	0	0	3	0
5	0	0	0	0	0

There can be several reasons for these deviations of the results as follows.

1. Image data sets used for the training and test data sets are different. This results in very different HOG features.
2. Features of the test data set is very different comparing to the training dataset. Better methods of preprocessing must be found in order to extract the correct number from the image.

Following solutions are suggested to resolve the above mentioned problems as follows.

1. It is suggested to use the same datasets to create both training and test datasets.
2. A set of numbers should be first classified

manually for each number to be use as the training dataset.

3. Further preprocessing experiments must be carried out in order to find the best preprocessing method for the images used in this paper. As an example Figure 8 shows results of two different preprocessing methods and the histograms of the images.

Figure 8 shows two different preprocessing results for two images. First row shows the original images. Second row shows the histograms of each image in grayscale. Third and fourth rows shows results of two different preprocessing steps. First one considers the pixels with the value in between 40 around the average pixel value of whole image as the character. Number given above third row is the average value of the image. Second method uses Matlab command gray thresh to find the threshold value and the pixels with the difference with the threshold in less than 60 are considered as the character.

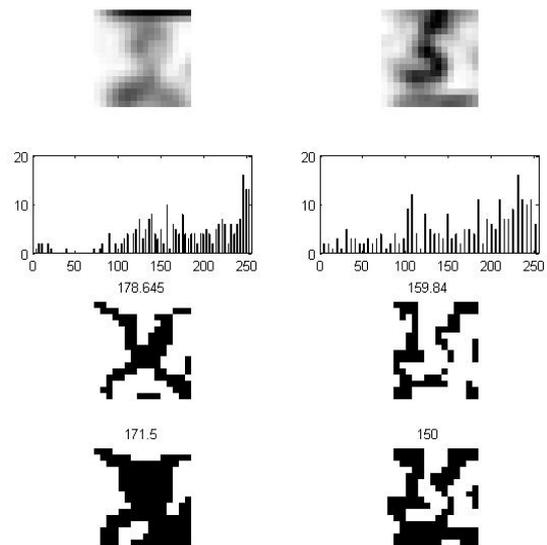


Figure 8: Different preprocessing methods

4. CONCLUSION

A work on handwritten character recognition for questionnaire responses using digital image processing was presented. Images of the questionnaire response was first camera corrected, then the relevant hand written characters were separated and preprocessed. Then HOG features were found for both training and test image sets. Finally a SVM based classifier was used to find the matching numbers for the detected letters. Results of the experiment cannot be used for the automatic detection yet since

the recognition rate is still not in the required level. Authors plan to use the same images as training and test datasets, improve the preprocessing step, and to automate character separation, as future works.

5. REFERENCES

- [1] N. Dalal and B. Triggs, “*Histograms of oriented gradients for human detection,*” in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886-893, 2005.
- [2] K. Terasawa and Y. Tanaka, “*Slit style HOG feature for document image word spotting,*” in: IEEE 10th International Conference on Document Analysis and Recognition, pp. 116-120, 2009.
- [3] M. B. Yilmaz, B. Yanikoglu, C. Tirkaz, and A. Kholmatov, “*Offline signature verification using classifier combination of HOG and LBP features,*” in: IEEE International Joint Conference on Biometrics, pp. 1-7, 2011.
- [4] <http://www.mathworks.com/machine-learning/examples.html?file=/products/demos/machine-learning/digit-classification-using-hog-features/digit-classification-using-hog-features.html>.