Handwriting Recognition

Lupou Krisztián-Róbert
Technical University of
Cluj-Napoca
Email: Lupou.Ad.Krisztian@student.utcluj.ro

***Abstract*—Handwriting recognition is the process of conversion of handwritten text into machine readable form. The goal of this process is to detect and recognize characters from input image and converts it into American Standard Code for Information Interchange (ASCII) or other equivalent form.**

I. Introduction

The project aims to classify and convert handwritten names in order to obtain machine readable text. The biggest problem in handwritten character recognition is the wide variety of handwriting styles, which can be completely different for different writers. The proposed solution is trying to simplify the transcription of official documents that contains names to a digital editable version. The project will implement the pre-processing, segmentation, feature extraction and the classification of each image from the dataset.

II. Bibliographic Study

Rasika R. Janrao , Mr. D. D. Dighe et al. [4] presented a character recognition technique using Learning Vector Quantization and K-Nearest Neighbor Classifier. The presented article compared the two classifiers, and the result was favorable for the KNN classifier with 93.65% accuracy compared to 88.29% for LVQ. For this results, they used diagonal feature extraction structure for handwritten character recognition in which all individual characters resized into 90×60 pixel and divided in 54 equal zone of 10×10 pixel. Feature extraction is done diagonally means from pixel of each zone feature moving with their diagonal.

Namrata Dave et al. [3] presented a few segmentation methods for Handwritten Character Recognition. The proposed segmentation use 3 levels of segmentation: line segmentation, word segmentation and character segmentation. For line segmentation it was used the horizontal scanning of the image with pixels grouped into multiple regions from the entire image. For word and character segmentation it was used the vertical scanning of the image. Depending on the values of the pixels, they are grouped into multiple regions. The different region indicates different content in the image file. Subsequently the desired content can be extracted. Slant angle estimation is used to perform skew correction for the extracted word in heavy noise.

Aissa Boudjella, Brahim Belhouari Samir and Omar Kassem Khalil et al. [2] presented handwritten character recognition based on a multiple Fermat's Spiral. The Fermat's Spiral was used for feature extraction for characters and they obtained a 99% success rate using with Cluster K-Nearest Neighbor classifier.

Nawaf Hazim Barnouti, Mohammed Abomaali and Mohanad Hazim Nsaif Al-Mayyahi et al. [1] proposed the zoning technique for feature extraction. In this technique, the rectangle character images are divided into a number of overlapping or non-overlapping regions of predefined sizes. Then features are computed for each zone. The average pixel density was found by dividing the number of foreground pixels by the total number of pixels in each zone.

The dataset used for this project has 206799 first names and 207024 surnames in total. The data was divided into a training set (331059), testing set (41382), and validation set (41382) respectively. The images have different sizes and the names are written with capital letters. The library used for this project is OpenCV.

III. Method

The method of Handwriting Recognition projects includes pre-processing, segmentation, feature extraction and classification. The input images are scanned files that contains names of persons. The extraction of a character which is from the name of a person consists of pre-processing of the input image and the segmentation of the input image by using histograms.

A. Pre-processing

In the pre-processing stage, the input image is binarized. Binarization is the process of converting a pixel image to a binary image as shown in Figure 1.

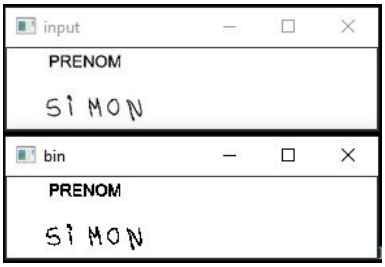

Fig. 1. A binarized image from dataset

B. Segmentation

Segmentation is among the most crucial and is an essential step in handwriting recognition. The biggest problem in this project is that the images from the dataset may have other characters that don't represent the name of the person (lines,

”NOM”, ”PRENOM”, ”:”, etc.).
For segmentation we have the following processes:

- line segmentation;
- character segmentation.

For line segmentation, this projects used horizontal histogram. The horizontal histogram has the values for the number of black pixels for each line of the image.

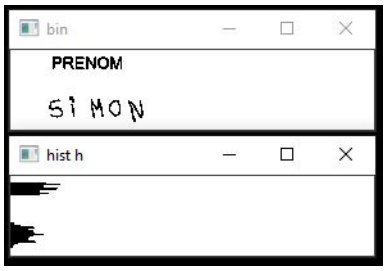

Fig. 2. Horizontal histogram for an image from dataset

In order to extract the correct line, you have to find the biggest segment. From what I observed the lines that contains ”NOM”, ”PRENOM” or other charactes are usually shorter than the lines that contains names. The result is the longest group of lines with at least 3 black pixels per line.

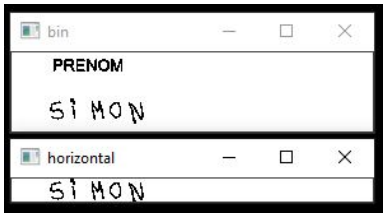

Fig. 3. The result for line segmentation for an image from dataset

For character segmentation, this projects used vertical histogram, but applied on the previous result for the line segmentation. The horizontal histogram has the values for the number of black pixels for each column of the image.

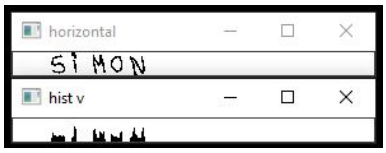

Fig. 4. Vertical histogram for the line segment

In order to extract the correct character, you have to find characters that there are not from ”NOM”, ”PRENOM”, ”:” and other characters. From what I observed the words ”NOM” and ”PRENOM”, the characters in this words are closer to each other and the ”:” charcher is after these words, closer to them. The written names in these images don’t start from the first column, so the characters that starts from the first column are excluded. The written names in the dataset usually have white columns between them, more then ”NOM” and ”PRENOM” and the length of a group of columns that represent a letter is not too long and not a single column. The result for an image is a set of images that represent the identified letters in the image. In this project each image of a letter is resized to 20x20.

C. Feature extraction

In this stage, the features of the characters that are essential for classifying them at recognition stage are extracted. This is an important stage as its successful operation improves the recognition rate and reduces the misclassification. I identified 3 features that could be extracted from a character:

- Zoning;
- Horizontal histogram;
- Vertical histogram.

Zoning is a well-known technique used in character recognition. In this technique, the rectangle character images are divided into a number of overlapping or non-overlapping regions (zones) of predefined sizes [1]. The size of a region for this project is 4x4 and has to be a divisor of the zoned image size. For each zone the number of black pixels is counted and saved in an array. The total number of zones if each image has his width equal to his height is equal to:

$$(zoned\_image\_size/zone\_size)^2$$

where zoned_image_size could be his width or his height, same for zone_size.

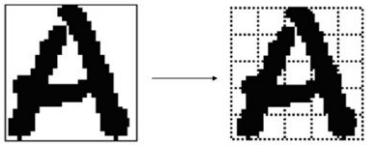

Fig. 5. Zoning technique for feature extraction

Projection histograms (horizontal and vertical) count the number of pixels in each column and row of a character image. Projection histograms can separate characters such as ”m” and ”n”. The results are saved in two different arrays, one for horizontal and one for vertical.

D. Classification

K-Nearest Neighbor (KNN) can be considered the simplest classifier because it does not build a model for the training set. The decision is made based on the closest K neighbors in the training set. The training dataset is defined in the form of feature matrix (X) that contains all the features for all images found to have the same number of letter as the length of the correct name for each image which is saved in a csv file for the training dataset. Each line contains all the 3 feature arrays (horizontal histogram, vertical histogram and zones) combined together. The size of feature matrix is n x d, where n is the total number of found letters and d is 65 (20 for the two histograms and 25 for zones). Another structure used was the label vector (y) which contains the class (ASCII code number) for each letters, values extracted from the training csv file. The size of label vector is n x 1. The label vector and the feature matrix

were saved in an csv file for a faster evaluation of the classifier and a faster testing on multiple input images.
For an unknown test instance x the features are extracted and then the distance is calculated:

$$d_i = dist(x, X_i)$$

The size of d is n x 2, where n is the total number of found letters, the first column is the calculated distance and the second column is the class of the letter calculated in relation to x. After this d is sorted in increasing order. The votes for the classes are counted for the first K distances and the class with the most votes is the result for the test instance x.
For the evaluation of the performance of the classifier the confusion matrix was calculated. The confusion matrix for a labeled dataset can be defined as a matrix containing in each cell M(i,j) the number of instances classified by the classifier into class i while having true class j. The ideal classifier would assign all instances to their correct class and would have large entries on the diagonal of the confusion matrix M(i,i). In general, the values show which classes are confused with each other and can help to improve the classifier performance by identifying specific features.
The accuracy for the classifier on a labeled test set is defined as the percentage of correctly classified instances. Is the division of the number of correctly classified instances by the total number of letters found in the test set.

IV. Evaluation and results

First test was done on an input image from the test dataset. The application loads the label vector and the feature matrix, then let you select the input image. The feature matrix has values for 1548106 letters.

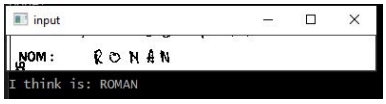

Fig. 6. A correct prediction for an input image

For many test images some letters have been mismatched. In the figure bellow, ”E” was confused with ”L”. Sometimes letters with thin lines or with split lines are easily confused.

Fig. 7. The confusion of ”E” with ”L”

Many results are influenced by the multiple styles of writing. Some writers have letters that could look like other letters. In the figure bellow the character ”J” was confused with ”T”, probably because ”J” has a top that look like the top of the letter ”T”. Another confusion is for ”N” with ”R”, probably because the second vertical line of ”N” is curved.

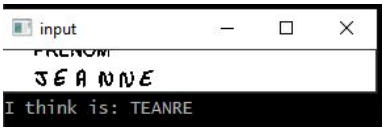

Fig. 8. The confusions of ”J” with ”T” and ”N” with ”R”

The evaluation of the classifier was done with the help of 100 test images. The classification of a bigger number of test images could take a long time, because the KNN classifier calculates 1548106 distances for each letter found in the test images. In this evaluation 492 letters were identified from 100 test images and 206 letters were identified correctly. The accuracy of the classifier is 41.86%. A confusion matrix was generated. For example, from this matrix it can be observed that character ”E” is confused with ”L” for 10 letters.

V. Conclusion

In this paper, a character recognition system has been proposed. The proposed system is based on image pre-processing, characters segmentation, feature extraction, and classification process, with a character segmentation made to work with the most cases of images from the training dataset and using features like horizontal histogram, vertical histogram and zoning. The achieved results was not was not the most accurate for many test instances, but at the same time other achieved the expected result. Future improvements for this project could be a better characters segmentation, more stages of pre-processing and increasing the number of images in the training set.

References

[1] Nawaf Hazim Barnouti, Mohammed Abomaali, and Mohanad Hazim Nsaif Al-Mayyahi. An efficient character recognition technique using k nearest neighbor classifier. *International Journal of Engineering Technology*, 7(4), 2018.

[2] Aissa Boudjella, Brahim Belhouari Samir, and Omar Kassem Khalil. Handwritten character recognition based on a multiple fermat’s spiral. *Advanced Materials Research*, 774–776, 2013.

[3] Namrata Dave. Segmentation methods for hand written character recognition. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(4), 2015.

[4] Rasika R. Janrao and Mr. D. D. Dighe. Handwritten english character recognition using lvq and knn. *International Journal of Engineering, Sciences Research Technology*, 5(8), 2016.