Hand Writing Recognition

Laura SOMETHING, Evangelos Karountzos

***Abstract*—Off-line hand writing recognition has been an issue between researches for a long time. Various techniques have been introduce and implemented most of which are performing well for the most part. In cases however that the characters have no clear borders within the word itself, things can get more complicated. In this paper we present such a case as well as our approach to solve the problem and achieve translation of the characters provided to ASCII.**

***Index Terms*—Pure awesomeness**

I. Introduction

HANDWRITING stuff.

In section II we present the process of cleaning our textual data. In section III we show how we split the words into characters whilst in section IV we show the features we extracted from the characters to be used in the classification process. In section V we present our machine learning approach and in section VI we present our results for both the characters and the words. Finally in section **??** and VIII we present the discussion as well as our conclusions respectively.

II. Preprocessing of the data

For the handwriting to be effective we had to bring the data into a clean state. To this end we used a pipeline which is described below in figure II. As the figure explains the first step is to normalize the luminosity of the image. This step is essential for the thresholding to work since in some parts of dataset the ink has worn off the pages. By normalizing the luminosity we eliminate the need to adjust threshold values in later steps.

The next step is to binarize the image and extract the clean text from it. For the binarization we tested the Otsu approach and global threshold and we settled with the latter as it gave better results **??**. To eliminate the what is left from the noise we applied opening by reconstruction. This has a result to miss some punctuation characters which however did not prove crucial to our learning process. In figure **??** we can see the final result of the preprocessing of the data.

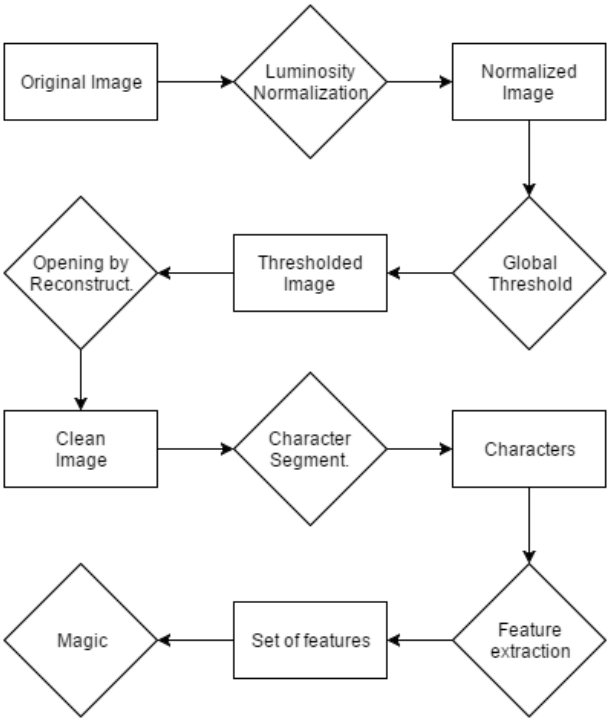

Fig. 1. Overview of our pipeline.

III. Character segmentation

Nothing to see here.

IV. Feature extraction

While we considered using various features for the character recognition such as crossing, projection histograms and celled projections [1][2]. From all the above the celled projections report the best results (7% and 10% better than crossings and projections respectively) [1].

Celled projections are extracted as follows. First the area of the character is split into regions, vertically, horizontally or mixed. In figure IV we see an example of vertical projection. Then for each of the regions we acquire the projections of the pixels on the left border of the region. This way binary vectors are created which then we concatenate.

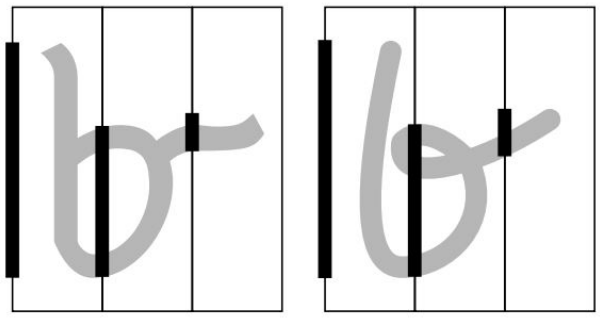

Fig. 2. In the above picture we see that even if the characters are differently written the projection feature will still extract consistent vectors.

V. Machine Learning

Nothing to see here.

VI. Results

Nothing to see here.

VII. Discussion

Nothing to see here.

VIII. Conclusion

Nothing to see here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] Hossain, M.Z., Amin, M.A. and Yan, H., 2012. Rapid feature extraction for optical character recognition. arXiv preprint arXiv:1206.0238.
[2] Trier, . D., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. Pattern recognition, 29(4), 641-662.
[3] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.