International Journal of Engineering & Technology, 7 (4.5) (2018) 36-39

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Handwritten Optical Character Extraction and Recognition from Catalogue Sheets

Subramanian Shanmugavel [1, a], Jagadeesh Kannan [2, b], Arjun Vaithilingam Sudhakar [3, c]

[1]*School of Informatics, Computing and Engineering Indiana University Bloomington, Indiana 47408*
[2]*Professor - School of Computing Science & Engineering, VIT University, Chennai-India*
[3]*Project Engineer-Wipro Technologies, Chennai-India*
**Corresponding author E-mail: [a]shanmusu@umail.iu.edu ,[b]dr_rjk@hotmail.com ,[c]vsarjun412@gmail.com*

Abstract

The dataset consists of 20000 scanned catalogues of fossils and other artifacts compiled by the Geological Sciences Department. The images look like a scanned form filled with blue ink ball pen. The character extraction and identification is the first phase of the research and in the second phase we are planning to use the HMM model to extract the entire text from the form and store it in a digitized format. We used various image processing and computer vision techniques to extract characters from the 20000 handwritten catalogues. Techniques used for character extraction are Erode, MorphologyEx, Dilate, canny edge detection, find Counters, Counter Area etc. We used Histogram of Gradients (HOGs) to extract features from the character images and applied k-means and agglomerative clustering to perform unsupervised learning. This would allow us to prepare a labelled training dataset for the second phase. We also tried converting images from RGB to CMYK to improve k-means clustering performance. We also used thresholding to extract blue ink characters from the form after converting the image in HSV color format, but the background noise was significant, and results obtained were not promising. We are researching a more robust method to extract characters that doesn't deform the characters and takes alignment into consideration.

Keywords: *Character Extraction and Recognition: CNN: Erode: k-means: OCR*

1. Introduction

The Optical Character Recognition (OCR) field has seen many advancements in recent years. But when it comes to identifying handwritten characters from forms which also contains printed characters and lines, the methods fail to extract and classify characters. The forms contain printed characters along with vertical and horizontal lines between which the handwritten characters are present. Sometimes the handwritten characters are overlapped with the lines and not properly aligned in the form which leads to dis-ambiguity. We doubt a generic solution exist for such problem. In this paper we discuss the techniques tried to extract the information from such scanned forms images.

The research is being done with the help of Geological Sciences department. The department has compiled a collection of images by scanning the handwritten forms of fossils and other artifacts. The department wants to digitize the information which will allow them to create and maintain a searchable repository for the information. It will be easier for them to share the in-formation with other organizations. They can also set the standards in this area by leading the way and innovating methods for implementation.

The collection includes catalogues, tags for fossils and other artifacts, and other types of documents. For our part, we will be looking to digitize only the catalogues. Digitizing the catalogues will take place in two phases. In the first phase, we will extract the characters and look to cluster the extracted characters. This will allow us to manually label the characters which in-turn will provide us with training data for classification.

In the second phase we will iterate through the fields in the form, classify the data and extract out meaningful words using the HMM model. In this paper we will focus on the first phase of the research and describe the methods used to extract and cluster characters from the catalogues.Ayatullah Faruk Mollah , Nabamita Majumder , Subhadip Basu and Mita Nasipuri (July 2011) [1] This paper presents a complete Optical Character Recognition (OCR) system for camera captured image/graphics embedded textual documents for handheld devices. At first, text regions are extracted and skew corrected. Noman Islam, Zeeshan Islam, Nazia Noor (December 2016) [2] this paper defines the process of digitizing a document image into its constituent characters. Despite decades of intense research, developing OCR with capabilities comparable to that of human still remains an open challenge

2. The Dataset

The dataset consists of 20000 catalogues of Geological Specimens from the Geological Department Blooming-ton and Indiana Geological survey. The scanned image consists of a table with headings as printed characters in light orange ink. The table also contains handwritten characters in blue ink. The goal of the research is to extract these blue ink characters and apply k-means

clustering which will allow us to inspect the extracted characters. We can further label these data in the clusters to use as a training dataset in the second phase of the research. The position of the blue ink characters in not fixed in the form which makes this task difficult. If the characters appeared at fixed position the problem would become simple and characters can be extracted by iterating through the image. As visible in the image, some of the written characters are very light and are not properly aligned. Each section has 6 rows, with each row holding up to 80 characters. The sections can be further divided into further subsections. But since the subsections are continuous we can process the entire subsection together. The catalogues have been written by the former member of the geological department over the course of 30 years. The catalogues are divided into 5 main sections. The main 4 sections contain information that needs to be digitized. The last section is only for remarks and need not be digitized. Goyal, Aditi, Kartikay Khandelwal and Piyush Keshri (2010) [3] this paper describes an OCR for printed Hindi text in Devanagari script, using Artificial Neural Network (ANN), which improves its efficiency. S. Iamsa-at and P. Horata,(December 2013)[4] This paper presents a framework for investigating and comparing the recognition ability of two classifiers: Deep-Learning Feedforward-Backpropagation Neural Network (DFBNN) and Extreme Learning Machine (ELM).
A. Chaudhuri et al.[5] This paper explains segmentation, feature extraction and classification . Chirag I Patel, Ripal Patel, Palak Patel(May 2011) [6] In this paper an attempt is made to recognize handwritten characters for English alphabets without feature extraction using multilayer Feed Forward neural network

FIG. 1: original.jpg and eroded.jpg

FIG. 2: original.jpg and dilated.jpg

One of the first problems we faced with the catalogues is the skewness of the image. As we proceed from left to right, image is skewed slightly up. This mean for a pixel in the border starting in the 80th row, the last pixel in the same border would be in the 65th row. This is a big problem as it means that when extracting a cell, we need to ensure to consider that each cell is slightly up-shifted than the previous cell in the same row.

FIG. 3: represents close transformation

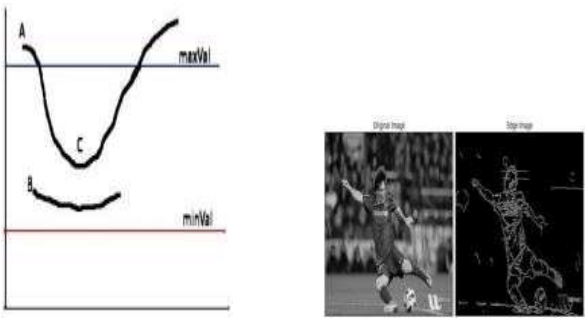

FIG. 4: Canny edge detector output

3. Image Preprocessing

We used the OpenCV (Open Source Computer Vision) library that was started by Intel and is currently maintained by Itseez. It also supports deep learning frame-works like TensorFlow and Caffe.

We'll discuss all the major pre-processing techniques that we used before the character extraction process.

Erode [7]: A kernel slides through the image and the original pixel is considered 1 if and only if all the pixels in the kernel are 1, else it is 0. Figure 2 shows original and the eroded images.

Dilate [7]: A kernel slides through the image and the original pixel is considered 1 if at least one of the pixels in the kernel is 1, else it is 0. Figure 2 shows the original and the dilated images.

Close [7]: Closing is basically Dilation followed by Erosion. This helps in closing small" holes" in the image. This has the effect of filling up the image. Figure 2 shows the original and the dilated images.

Canny Edge detector [8]: The Canny edge detection algorithm calculates the gradient at each pixel. If the gradient of the pixel is a above a max threshold, it is considered as a sure-edge pixel. If it is below the max threshold, above the min threshold and connected to a sure-edge, it is considered as a valid edge. If it is not connected to a sure edge, is discarded. Figure 3 and 4 shows the original image and the processed image with fine edges detected.

4. Extracting Characters [9]

We tried below mentioned character extraction methods for extracting characters from the processes images.

4.1 Finding Contours

Using the algorithm developed by Satoshi Suzuki, we can find contours in the image. This has been developed keeping shape analysis and object detection and recognition in mind. Refer figure 7.

4.2 Approximating Contours [10]

After detecting the contours, we approximated the contours. This soothes out the contours, by reducing the number of vertices. The reduction is based on epsilon. A low epsilon returns an image with contours close to the original image, while a high epsilon approximates the image to closest shape. The approximated Contour is shown in Figure 5

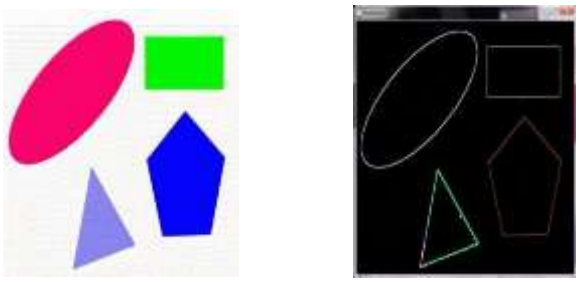

FIG. 5: Find Contours using OpenCV Approximated Contou

FIG. 6: poor images

4.3. Boxing the Letters

Once we have our contours, we calculate the corner pixels. Using these, we can get the max width and height of our characters. Then we can extract the characters based on these values.

FIG. 7: Approximated Contour

5. Inspecting Characters

We used clustering technique to cluster the extracted characters images. This is to inspect the images falling in a single cluster. All the images of a single character should fall within same cluster. This also helped us to label the images for training a HMM model in phase 2 of the research.

Upon inspecting the characters before clustering we realized that the characters have been extracted with thick edges. This is not a problem for certain characters like A, but for characters like O, this is a bigger problem as shown in figure 5 and 7. Most of the characters are extracted well.3.4.1. Figure captions

Figures must be numbered using Arabic numerals. Figure captions must be in 8 pt. Regular font. Captions of a single line (e.g. Figure 2) must be centred whereas multi-line captions must be justified Captions with figure numbers must be placed after their associated figures

6. Clustering Characters [11]

To cluster the characters, we need to extract features from the images. The first feature we used are Histogram of Oriented Gradient or HOGs. A HOG is a vector used to identify the edge of objects in images. To calculate HOGs, first we calculate the x-gradient (left image) and y-gradient (centre image). We convert these to polar notations which gives us magnitudes (right image) and angles. After we have the magnitudes, we bin the values to get the HOGs foreach image. A sample HOG transformation for a image is shown in figure 8.
HOGs have been used for classifying handwritten characters in Hindi language by students from Stan-ford. However, since the dataset was not readily avail-able, they had to generate the training and testing samples themselves. Since then, attempts have been made to replicate their experiments with other languages such as Thai and Marathi.

After extracting the features, we tried unsupervised clustering technique such as k-means, hierarchical clustering etc. Our end goal for the research was to have clusters for the different characters which we can manually label and then use in the second phase of the research, digitizing the information using Hidden Markov Models. The labelled dataset can be used as emission probabilities, along with initial and transition probabilities calculated from corpus provided by Geological Sciences.

FIG. 8: good images

FIG. 9: HOG Visualization

6.1 Unsupervised Clustering

Based on our k-means clustering technique on the HOG values, we are not getting very good results. Part of the reason maybe the fact that we had to make our characters thicker to be able identify them using findContours(). This would distort the characters. The second reason could be that our HOGs calculation is not rotation invariant.

We also tried Agglomerative clustering, but the results were not satisfactory. This is another indicator that our HOG features are not good descriptors for clustering.

We tried using k-means clustering to cluster the pixels. This would have allowed us to have 3 clusters; 1) Background of the form, 2) Printed characters and grids, and 3) ink of the handwritten character.

However, the ink used is blue and is very dark. Hence it is not possible to distinguish the ink pixels from the printed pixels based on the colour. Therefore k-means is not able to cluster together the ink pixels with a high degree of success as seen from figure We even tried converting the image from RGB to CMYK. CMYK has the advantage of being friendlier for colour segmentation than RGB and is often used to differentiate pixels based on the pixel intensities. But even so, using k-means has high error.

6.2 Convolutional Neural Network

We also implemented a Convolutional Neural Network to classify the extracted character images into one of the 36 classes. We used Tensor Flow and keras, a python library on top of tensor flow for building our CNN architecture as shown in figure 11. The number of parameters that are being trained in the network are 261,280 as shown in the Figure 10 network summary.

After manually clustering and labelling a training data set, we were able to achieve close to 96% accuracy. However, considering that the training set is so small, only a few thousand characters, we cannot rely solely on this metric, as promising as it looks.

7. Conclusion

Considering that HOGs reduce an image into a vector with only 9 bins and the fact that our extracted characters are not perfectly extracted, we may have to re-consider the use of HOGs as our features descriptor.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 13, 13, 32)	0
conv2d_2 (Conv2D)	(None, 11, 11, 32)	9248
max_pooling2d_2 (MaxPooling2	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 300)	240300
dense_2 (Dense)	(None, 36)	10836

Total params: 261,280
Trainable params: 261,280
Non-trainable params: 0

FIG. 10: CNN Summary

If we still want to use HOGs, then we need to find a more robust method to extract characters, one that does not deform the characters. Also having extracted characters rotated to be aligned will also be important. CNNs also provides a good way forward and is one we intend to develop further. We will look to implement unsupervised learning or semi-supervised one- shot learning.

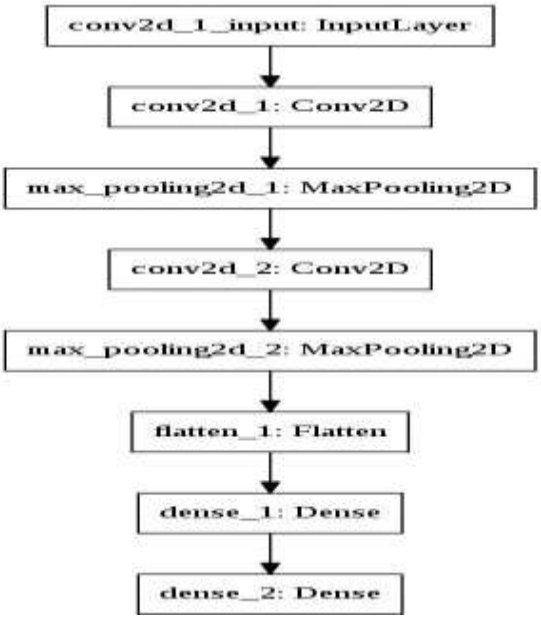

FIG. 11: CNN Architecture

References

[1] Ayatullah Faruk Mollah , Nabamita Majumder , Subhadip Basu and Mita Nasipuri "Design of an Optical Character Recognition System for Camerabased Handheld Devices"-IJCSI
[2] Noman Islam, Zeeshan Islam, Nazia Noor," A Survey on Optical Character Recognition System"-JICE
[3] Goyal, Aditi, Kartikay Khandelwal and Piyush Keshri. Optical Character Recognition for Handwritten Hindi.(2010).
[4] S. Iamsa-at and P. Horata, "Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Net-work,"(2013)
[5] A. Chaudhuri et al., Optical Character Recognition Systems for Different Languages with Soft Computing, Studies in Fuzziness and Soft Computing 352, DOI 10.1007/978-3-319-50252-6_2
[6] Chirag I Patel, Ripal Patel, Palak Patel Handwritten Character Recognition using Neural Network International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011
[7] https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosiondilata-tion/erosiondilatation.html
[8] https://docs.opencv.org/3.1.0/da/d22/tutorialpycanny.html
[9] http://opencvexamples.blogspot.com/2013/09/find-contour.html
[10] https://docs.opencv.org/3.1.0/dd/d49/tutorialpycontourfeatures.html
[11] https://www.learnopencv.com/histogram-of-oriented-gradientsMcMahon GT, Gomes HE, Hohne SH, Hu TM, Levine BA & Conlin PR (2005), Web-based care management in patients with poorly controlled diabetes. *Diabetes Care* 28, 1624–1629.