Optical Character Recognition using KNN on Custom Image Dataset

Tapan Kumar Hazra
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
tapankumar.hazra@iemcal.com

Dhirendra pratap singh
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
dhirendrapratapsingh398@gmail.com

Nikunj Daga
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
dodmst@gmail.com

***Abstract*— The aim is to develop an efficient method which uses a custom image to train the classifier. This OCR extract distinct features from the input image for classifying its contents as characters specifically letters and digits. Input to the system is digital images containing the patterns to be classified. The analysis and recognition of the patterns in images are becoming more complex, yet easy with advances in technological knowledge. Therefore it is proposed to develop sophisticated strategies of pattern analysis to cope with these difficulties. The present work involves application of pattern recognition using KNN to recognize handwritten or printed text.**

Keywords—K-nearest neighbor, optical character recognition, custom image dataset, handwritten character recognition

I. INTRODUCTION

These days there is a huge demand in storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process [3]. Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in image.

[1] Optical Character Recognition (OCR) conveys the meaning of recognized English (or any other language whichever is used in training dataset) characters as well as digits that maybe images of handwritten text, or may be just computer text fonts of various types.

[2] It is an application software which analyses and processes an image document to recognize efficiently the characters present within it. The image document can be a handwritten or printed text, PDF document or a scanned photo. It translates images into recognizable machine encoded editable text. It recognizes only those characters for which the system has been trained for using specific classification algorithm[6].

[3] Optical Character Recognition/Reader (OCR) is conversion of images of typed, handwritten, scanned or printed text into machine-encoded text by computer. The input can be from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast). It is widely used as a form of information entry from printed paper data records, passport documents, invoices, bank statements, computerized receipts, government application forms and many such kind [1].

Same is applied to test images to extract its features for comparison and classification.

II. ALGORITHM FOR PRESENT WORK

The accuracy and diversity of the method lies in the training data set that is used for training the OCR system to acquire the recognizer ability and to perform the same. The positive aspect of present work is that any custom image can be used for the purpose of training the classifier.

A. The Algorithmic steps are:

1. The training image having the set of characters in different format is read and processed. Training image is converted into a grey-scale, blurred, threshold and flattened image so that it is easier for the system to understand the image features and differentiate between the objects(characters) and unwanted background
2. From the contours with data, features are extracted for each character and stored in a numpy array. Then this array and an array containing labels are combined and stored in a text file.
3. The second module is for training and testing. The saved text files are loaded. Then a KNN classifier object is created using cv2, which is trained using the text file.
4. Now the image to be tested is read and again the image is converted and processed to extract the features from the contours with data.
5. Then the contours with valid data is checked and separately stored in a list.
6. These contours are marked with green rectangles on the testing image and shown.
7. These green marked rectangles are cropped, resized threshold and flattened.

978-1-5386-2215-5/17/$31.00 ©2017 IEEE

8. Then using find nearest function KNN object the character label with most matching features are obtained and displayed as string.

4 After converting an image into a gray-scale image, it is analyzed and then the difference in spacing and pixel spaces determine the text and its limits so that it may analyze the characters separately without any hindrance to the adaptability that has been found in this recognizer.

5 KNN algorithm is used because of its higher accuracy over non-linear multiclass problems.

6 While performing testing of our application, it has been noted the OCR developed is very flexible even for untrained data. The training dataset is an image file saved in suitable (.png or other) format so that it can be used to train the classifier. The model can be scaled for any local language, just by changing training image file and labels in the code.

The image then helps in establishing certain parameters as to which would be used to determine the difference between several characters with some similarities amongst those, like the similarity in 'm' and 'n'. The difference in between such similar characters can only be known by using pre-requisite knowledge of such characters which is why the training set needs to possess enough examples of each such characters so that it may easily differentiate between them.

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. ADVATAGES OF KNN

Before KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects [4]:

- Ease to interpret output
- Calculation time
- Predictive Power

Comparison of KNN with other classification algorithms:

TABLE I. Comparison of KNN with other

	Logistic Regression	CART	Random Forest	KNN
1. Ease to interpret output	2	3	1	3
2. Calculation time	3	2	1	3
3. Predictive Power	2	2	3	2

KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy interpretation and low computation time.

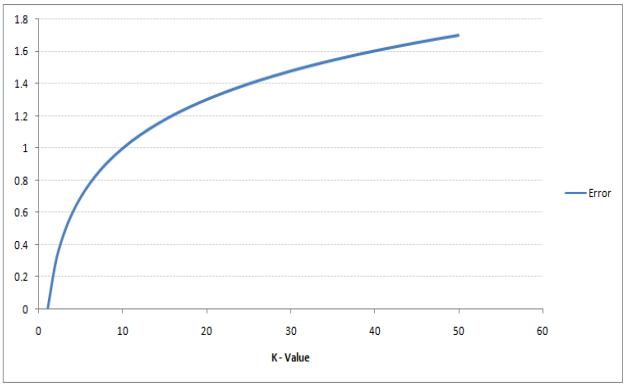

Fig. 1. Parameter K vs. Error rate

7 As it is clearly observed from Fig. 1., the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

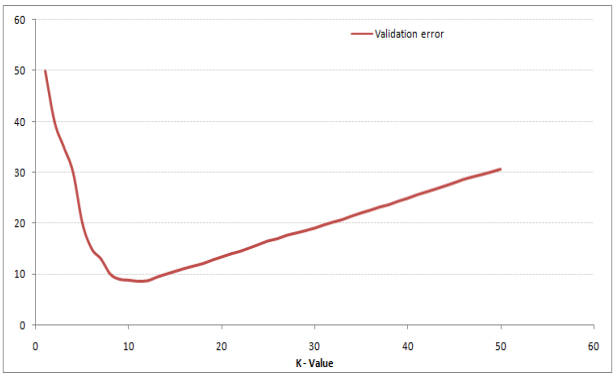

Fig. 2. Validation of error curve

This is the Validation of the error curve with varying value of K is shown in Fig. 1. At K=1, we were overfitting the boundaries. Thus, error rate initially decreases and reaches a minimal. After the minimal point, it then increases with increasing K. To get the optimal value of K, we can segregate the training and validation from the initial dataset. Now plotting the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

One of the major advantages of our OCR is it can be used in CCTV surveillance of an area by helping decode the writings found in the background in images.

All text and graphic files are kept separated until the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads; the template will do that without extra intervention.

IV. Results and Discussion

After successful implementation of the model, the same is tested on various test cases. Input, output and interpretation of results are given below:

A. Input

The custom Image dataset looks something like the figure below where all the different font styles of English Alphabet and digits are given in the custom training image dataset. So that when the application is given an input image to recognize the intelligence present in it the classifier processes and matches the features of input pattern with the features of training characters.

This training image is the distinguishing key of our work. Generally for training the classifier standard datasets are used that are available online from official sources like Artificial character dataset, Chars74k dataset, MINIST dataset[1], Gisette dataset etc., which constraints the application but in our application we can train the classifier by any custom image having characters of any language and hence can be used in language translation. Fig. 3. A shows a sample custom image dataset and Fig. 2 shows corresponding processed training dataset.

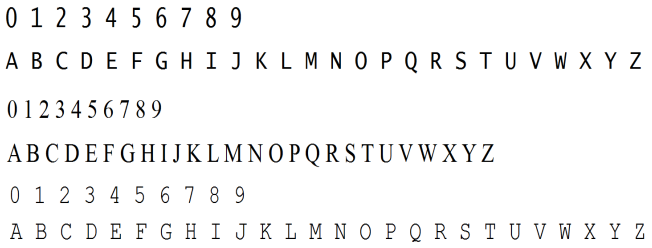

Fig. 3. Custom image dataset

8 The grayscale input of the training image dataset is as shown below

Fig. 4. Processed training dataset

B. Output

The output using different test cases where characters are recognized are shown in Fig. 5.

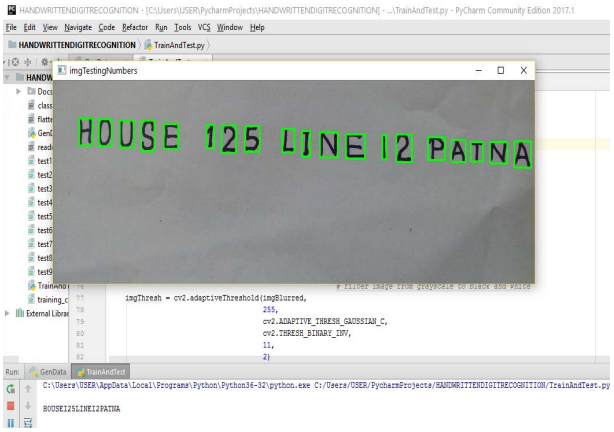

Fig. 5. Sample output after handwritten character recognition

9 In the Fig. 5, we see that the OCR has successfully identified the characters (digits and alphabets) from the image of a handwritten paper e.g. address of person in Fig. 4, and typed digits in Fig. 6.

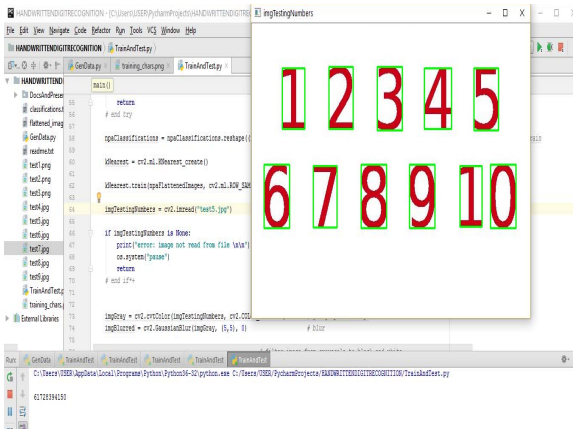

Fig. 6. Sample output after typeset character recognition

10 In the Fig. 5. And Fig. 6., we see that the OCR has successfully identified the digits from the image in order from right to left but in Fig. 7., we see that the OCR recognizes only English alphabets for which it has been trained and doesn't recognizes the Hindi characters for which it has not been trained.

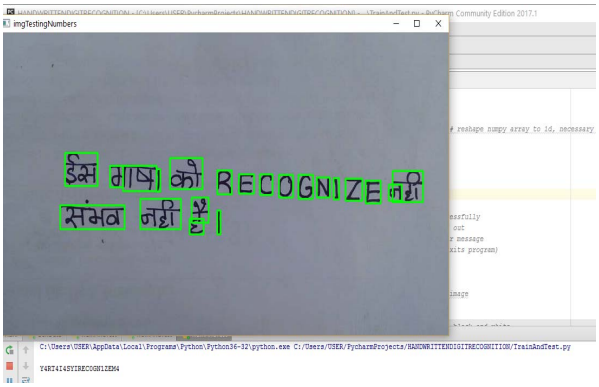

Fig. 7. Sample output for partial recognition

C. *Why python is chosen as development tool*

The following motivates to select python as development tool:

1. Very simple to understand and use
2. Presence of Third-Party Modules (Python Package Index)
3. Extensive Support Libraries - Python provides a large standard library with lacks of useful functions most suitable for developing application software
4. Python language is developed under an OSI-approved open source license, which makes it free to use and distribute, including for commercial purpose.
5. The size of the code is reduced
6. Python has built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, Python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

V. Application Areas

1. It can be used for digitizing printed texts so that it can be Electronically edited, searched, stored more compactly, displayed on-line, and used in Machine Processes such as Machine Translation, Text-to-Speech, Key Data and Text Mining
2. Postal services use OCR to read addresses form letter envelopes
3. It is widely used for information entry from passport documents, invoices, bank statements, voter ID forms etc.
4. Automated number plate recognition of vehicles
5. To create everyday database backup from newspapers
6. Intelligent transportation systems(ITS) deployments to identify the vehicle numbers from camera images and automatic complain registration against that vehicle number which can save a lot of human effort an time

VI. Accuracy Enhancement

OCR accuracy can be enhanced if the output is constrained by a *lexicon* – a list of words that are allowed to occur in a document. This might be, for example, all the words in the English language, or a more technical lexicon for a specific field. This technique can be problematic if the document contains words not in the lexicon, like proper nouns.

Another method of accuracy Enhancement include the integration of local language used in a region to accurately understand the writings of locals.

VII. Assumption Made

Before using KNN, let us see some of the assumptions made for KNN.

KNN assumes that the data is in a *feature space*. More exactly, the data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance – that need not necessarily be Euclidean distance, although it is the one that is commonly used [5].

Each of the training data consists of a set of vectors and class label associated with each vector. In the simplest case, it will be either + or – (for positive or negative classes). But KNN, can work equally well with arbitrary number of classes.

We are also given a single number "k". This number decides how many neighbours (where neighbours is defined based on the distance metric) influence the classification. This is usually an odd number if the number of classes is 2. If k=1, then the algorithm is simply called the nearest neighbour algorithm.

A principle favorable position of the KNN algorithm is that it works well with multi-modal2 classes in light of the fact that its decision is depend on a small neighborhood of similar target. Subsequently, regardless of the fact that the target class is multi-modal, the algorithm can in any case lead to great precision.

VIII. Limitations

- It utilizes every feature similarly in computing a part of processing for similitude. This can prompt to classification errors, particularly when there is just a small subset of features that are helpful for classification.
- Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results
- Computation cost is quite high because we need to compute distance of each query instance of all training samples

References

[1] Said Kassim katungya, Xuewen Ding and Juma Joram Mashenene 'Automatic Recognition of Handwritten Digits Using Multi-Layer Sigmoid Neural Network'.

[2] R.O.Duda, P.E.Hart and D.G.Stork, 'Pattern Classification', Johy Wiley, 2002.

[3] Text Recognition from Images: A Review Pratik Madhukar Manwatkar, Dr. Kavita R. Singh Department of Computer Technology, YCCE, Nagpur, (M.S.), 441 110, India.

[4] K Nearest Neighbour Classification over Encrypted Relational Data Gadekar R.R.1 Bhosale R.S.2 1ME Student 2Assistant Professor 1,2Department of Information Technology Engineering 1,2AVCOE, Sangamner, Maharashtra,India.

[5] A Survey of Classification Methods and its Applications International Journal of Computer Applications (0975 – 8887) Volume 53– No.16, September 2012 9 Geetika, PhD Scholar, ITM University, Gurgaon.

[6] Tapan kumar Hazra, Rajdeep Sarkar, Ankit Kumar, "Handwritten English Character Recognition Using logistic Regression and Nueral Network, " www.ijsr.net,Vol 5 Issue 6 , pp. 750-754, June 2016. DOI: http://dx.doi.org/10.21275/v5i6.NOV164228