MINOR PROJECT

(B. Tech. 6th Semester)

On

OPTICAL CHARACTER RECOGNITION

Submitted By

Ayushi Poddar (Roll no. : 1504016)

Sabita Kumari (Roll no. : 1504068)

Kunal Kumar (Roll no. : 1504024)

Under the supervision of

Dr. Rakesh Ranjan

Assist. Professor ECE Department

Acknowledgment:

I would like to express my deepest gratitude and thanks to my supervisor, Prof. Rakesh Ranjan, who suggested the idea of the Optical Character Recognition and who gave me the needed information to start working on the project. Also, I would like to thank him for being supportive and for his guidance through this semester and for giving me the necessary advices to be able to realise this project. I am really grateful to his contribution. Moreover, I would like to thank him for his supervising methodology that made my tasks easier and motivated me through this period.

Finally, I would like to thank my parents, brothers, and all my friends for their support. They have supported me during the difficult moments and encouraged me in carrying out this project work.

Contents

ABSTRACT

In this growing word there is a growing demand for the software system to recognise characters in computer system when information is scanned through paper document as we all know that we have a lot of paper and document in printed format. These days there is a huge demand for storing these documents in machine or computer hard disk and further reuse the information by searching purpose.
To do this One can simply scan the document and store it in image form but to reuse this information is very difficult to search the content from these documents line by line and word by word. Computer is unable to recognise the text in the document. This concept of storing the information in machine encoded text and then searching and reading the content is called document processing and for this we need a software that is optical character recognition.
Optical character recognition, usually abbreviated to OCR, which can translate scanned images of handwritten, typewritten or printed text into machine-encoded text.

INTRODUCTION

Optical character recognition (also known as optical character reader,OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether 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, whether passport documents, invoices, bank statements, computerised receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitising printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs.[2] Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

There are two types of OCR:

1. Document recognition -Targets type written text, one glyph or character at a time.
2. Handwriting recognition -Targets type written text, one glyph or character at a time involves feature extraction.

OCR is about reading or recognition of machine printed characters like those found in a book. Whereas handwritten character recognition is, as the name specifies, about recognising characters written by a human hand.

The latter is more complex due to the wide variations in the style or manner in which humans write. It is easier to do OCR in the sense that optical characters appear almost the same and are easy to isolate from one another in a word. Handwritten characters on the other hand may not be so easy to tell apart.

With OCR you can use basic template matching schemes and achieve a very high accuracy level especially if you allow the system to tolerate minor distortions. Handwritten character recognition requires more sophisticated approaches such as SVMs or neural nets especially convolutional neural networks. Handwritten character recognition is normally based on multi-stage feature extraction schemes. Here in this project we are dealing with document recognition.

The ultimate objective of OCR is to simulate the human reading capability so the computer can read, understand, edit and do similar activities it does with the text.

Model of Optical Character Recognition:

1. Image Processing
2. Image Segmentation
3. Training and Testing
4. Recognition

IMAGE-PROCESSING

Pre-processing is a common name for operations with images at the lowest level of extraction-both input and output are intensity images.The aim of pre-processing is an improvement of the image data that surpasses unwanted distortions or enhances some image features important for further processing.

There are following types of pixel brightness transformation:Brightness transformation modify pixel brightness -the transformation depends on the properties of the pixel itself.

Brightness corrections.

Grayscale transformations.

Image thresholding

Thresholding is a non-linear operation that converts a grayscale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value.

It is however far more efficient to use the ImageThreshold operation which also provides several methods for finding the "optimal" threshold value for a given image. ImageThreshold provides the following methods for determining the threshold value.

1. Simple thresholding- If pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black).
2. Adaptive thresholding- In simple thresholding, we used a global value as threshold value. But it may not be good in all the conditions where image has different lighting conditions in different areas. In that case, we go for adaptive thresholding. In this, the algorithm calculate the threshold for a small regions of the image. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination..

SEGMENTATION

Segmentation partitions an image into distinct regions containing each pixels with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Meaningful segmentation is the first step from low-level image processing transforming a greyscale or colour image into one or more other images to high-level image description in terms of features, objects, and scenes. The success of image analysis depends on reliability of segmentation, but an accurate partitioning of an image is generally a very challenging problem.

Segmentation techniques are either contextual or non-contextual. The latter take no account of spatial relationships between features in an image and group pixels together on the basis of some global attribute, e.g. grey level or colour. Contextual techniques additionally exploit these relationships, e.g. group together pixels with similar grey levels and close spatial locations.

Maximal Stable Extremal Regions

MSER is a method for blob detection in images. The MSER algorithm extracts from an image a number of co-variant regions, called MSER. An MSER is a stable connected component of some grey-level sets of the image.

MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds. All the pixels below a given threshold are white and all those above or equal are black. If we are shown a sequence of thresholded images It with frame t corresponding to threshold t, we would see first a black image, then white spots corresponding to local intensity minima will appear then grow larger. These white spots will eventually merge, until the whole image is white. The set of all connected components in the sequence is the set of all extremal regions. Optionally, elliptical frames are attached to the MSERs by fitting ellipses to the regions. Those regions descriptors are kept as features The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary.

Contours:

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same colour or intensity. The contours are a useful tool for shape analysis and object detection and recognition.
For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.Since OpenCV 3.2, **findContours**() no longer modifies the source image but returns a modified image as the first of three return parameters.In OpenCV, finding contours is like finding white object from black background. So remember, object to be found should be white and background should be black.
To draw the contours, cv2.drawContours function is used. It can also be used to draw any shape provided you have its boundary points. Its first argument is source image, second argument is the contours which should be passed as a Python list, third argument is index of contours (useful when drawing individual contour. To draw all contours, pass -1) and remaining arguments are color, thickness etc.

MACHINE LEARNING ALGORITHM

K-NEAREST NEIGHBORS:

In pattern recognition , the **k-nearest neighbors algorithm (k-NN)** is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature-space. The output depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integers, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.
- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

k-NN is a type of instance-based learning, or lazy-learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.
Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor.
The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.The algorithm is not to be confused with k-means, another popular machine learning technique.
algorithm.
The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.
In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance). In the context of gene expression microarray data, for example, k-NN has also been employed with correlation coefficients such as Pearson and Spearman.Often, the classification accuracy of k-NN can

be improved significantly if the distance metric is learned with specialised algorithms such as Large Margin Nearest Neighbor or Neighbourhood components analysis.
A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the *k* nearest neighbors due to their large number. One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its *k* nearest neighbors. The class (or value, in regression problems) of each of the *k* nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation. For example, in a self-organising map (SOM), each node is a representative (a centre) of a cluster of similar points, regardless of their density in the original training data.

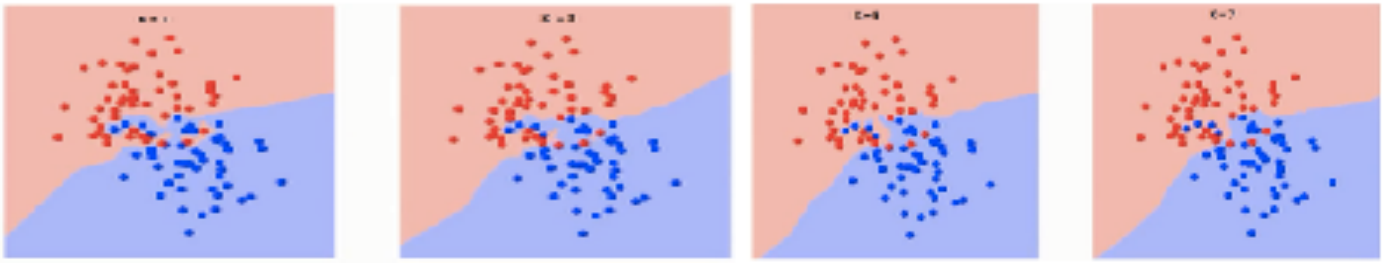


fig: Different boundaries with different K values

With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access on different K-value. Following is the curve for the training error rate with varying value of K :

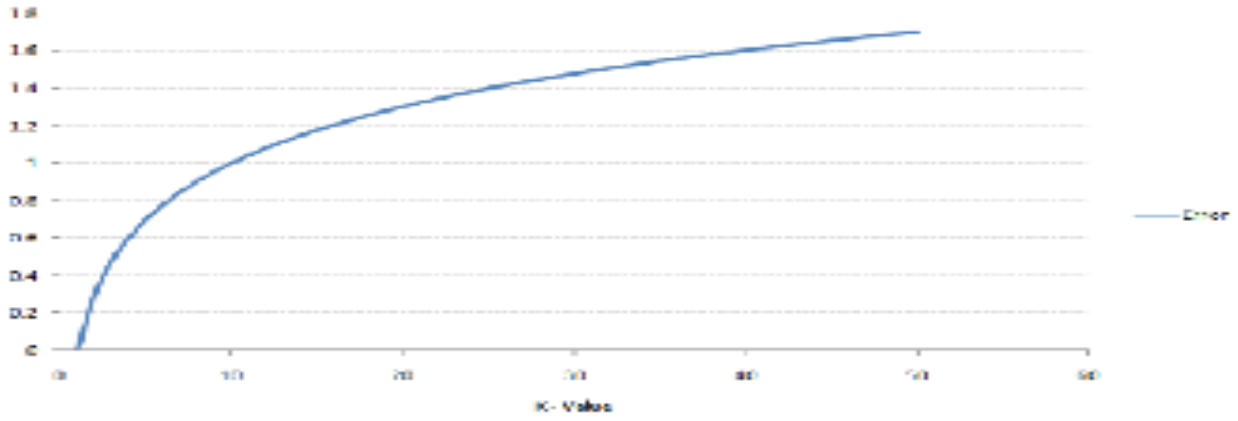
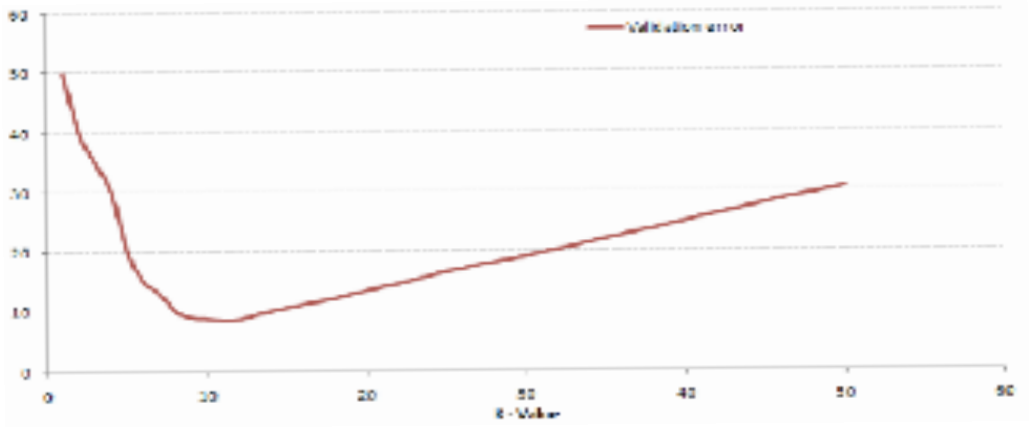


fig: curve for training error rate with varying value of K

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.



Algorithm performed

The image was first loaded and scanned .Image processing was done so as to enhance the brightness of the image and make the image more clear so that we get the clear character after segmentation. We changed the image to grey scale and then thresholded it so as to separate the background from foreground and get the region of our interest. In our algorithm we used adaptive thresholding as this was more efficient than simple thresholding.

The thresholded image was then segmented using contour method and each segment was extracted out and fed into the machine learning algorithm. Here we have used K-NN as the learning algorithm.This compares the segmented character with the dataset with which it had been trained and finds out the nearest neighbour (character with which it have least Euclidean distance)and returns that character as the prediction.This process is done for all the segments and every character of the image is thus predicted. This was the testing part.

In training part we generated a dataset of various characters and stored it in its floating form which acted as labels for the character.

In testing the characters are compared with every characters of dataset and returns the character.

Reason for error

In our work we are getting some error due to the improper segmentation of the image. While using MSER every connecting lines are detected, even the lines which does not lie in the region of our interest.The inner portion of closed characters are also been detected and a character is being predicted for that region.While using contour if the adjacent character has same intensity the it will assume it as single contour and thus the characters are not predicted properly.

Results:

fig: Original image

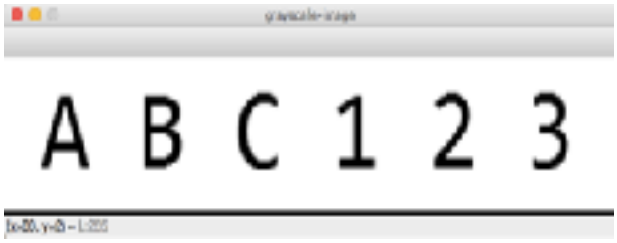


fig: grayscale image

fig: Thresholded Image

fig: Segmented image

fig: Bounding rectangles

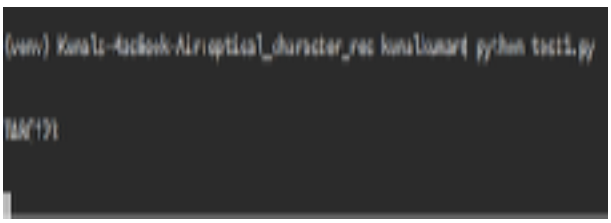


fig: Output

Related Works

A number of researches have been proposed over the years for character recognition.A noble idea was conceptualised way back in 1999 by (Sural and Das). The ability of a software or application like this one needs the ability to apply pattern recognition, pattern interpretation and learning. This sought the use of multi-layer perception model which is a feed forward model. They used the model to develop an OCR(Optical Character Recognition) for an Indian Language named Bengali. The model that was used belong to a class of artificial neural network. The advantage of using perception model is its efficiency of learning. It can be used to train and recognise any kind of data set than initially defined. Moreover, their approach left a great scope of developing OCR that target any language other than Bengali.Another work comes from Deepayan Sarkar from University of Wisconsin. He implemented OCR as an add on package for a MATLAB-like programming environment called R.

Although results were not that good, they were not bad either suggesting this technique is not flawed.More training data would improve robustness and accuracy. Moreover, he pointed out that the speed may need to be improved using C code.The authors have divided each character into a number of predetermined rectangular zones and extracted a 13-element vector comprising of the pixel values in those zones. A neural network classifier has been used to recognise the 26 alphabets of English language.

References:

1. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
2. https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html
3. https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html
4. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clusterin
5. https://en.wikipedia.org/wiki/Optical_character_recognition
6. https://www.slideshare.net/apashishp/optical-character-recognition-ieee-paper-study
7. https://www.encyclopedia.com/science-and-technology/computers-and-electrical-engineering/computers-and-computing/optical
8. http://opencv-python-tutroals.readthedocs.io/
9. https://docs.opencv.org/trunk/d3/d28/classcv_1_1MSER.html
10. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. II (Mar – Apr. 2015), PP 22-26