INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume3, Issue3)

Available online at: www.ijarnd.com

Character Recognition

Aditya Karlekar

Student, Hitkarini College of Engineering and Technology, Jabalpur, Madhya Pradesh

ABSTRACT

The purpose of this research is to create a system that takes handwritten as well as printed English characters and numerals as input, process the character, train the k space nearest neighbor algorithm, recognize the pattern and modify the character to a beautified version of the input. This research is aimed at developing software, which will be helpful in recognizing characters of English language. This research is restricted to English characters and numerals only. It is also helpful in recognizing special characters. It can be further developed to recognize the characters of different languages. One of the primary means by which computers are endowed with humanlike abilities is through the use of Machine Learning. Machine Learning allows the system to be more efficient and accurate with successive iteration. Pattern recognition is perhaps the most common problem that can be solved using machine learning approach. The K space nearest neighbor algorithm is one of the most efficient machine learning algorithms that can be used for pattern recognition problem and works well with limited data for learning. The process followed consist of following steps: Image Input, Preprocessing, Segmentation, Feature Extraction, and Classification. A kNN trained for classification is designed to take input samples and classify them into groups or clusters. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting printed and handwritten characters and wishes to improve upon previous character recognition systems. The developed system performed as per the expectations successfully identifying multiple characters in a single row and worked well with handwritten characters as test data even when no handwritten character data was provided in training set.

Keyword: *Pre-processing, Segmentation, Feature Extraction, Classification, K-space Nearest Neighbor Algorithm.*

1. PURPOSE

The purpose of this research is to take English characters and numerals as input, process the character, train the k space nearest neighbor algorithm, to recognize the pattern and modify the character to a beautified version of the input. The proposed methodology uses some techniques to remove the background noise and features extraction to detect and classify the text. The proposed method comprises of 4 phases: Pre-processing; Segmentation; Feature Extraction; Classification and Recognition. One of the primary means by which computers are endowed with humanlike abilities is through the use of Machine Learning. Machine Learning allows the system to be more efficient and accurate with successive iteration [1].

Figure 1 shows the flow diagram of the process involved.

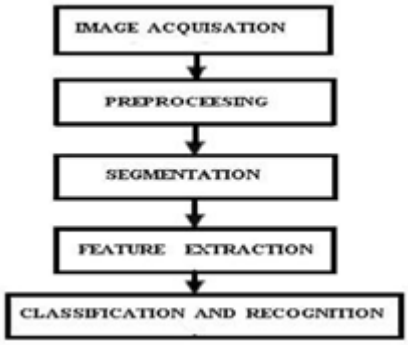


Figure 1: Flow chart of the process

2. PREPROCESSING

The pre-processing is a series of operations performed on the scanned input image. It essentially enhances the image rendering it suitable for segmentation. The role of pre-processing is to segment the interesting pattern from the background. Noise filtering and smoothing are performed in this step. The pre-processing also defines a compact representation of the pattern. Binarization process converts a gray scale image into a binary image. Pre-processing of the sample image involves few steps that are mentioned as follows:

2.1 Grey-scaling of RGB image

Grey-scaling of an image is a process by which an RGB image is converted into a black and white image. This process is important for Binarization as after grey-scaling of the image, only shades of grey remain in the image, binarization of such image is efficient

2.2 Binarization

Binarization of an image converts it into an image which only has pure black and pure white pixel values in it. Basically, during binarization of a grey-scale image, pixels with intensity lower than half of the full intensity value gets a zero value converting them into black ones. And the remaining pixels get a full intensity value converting it into white pixels.

2.3 Inversion

Inversion is a process in which each pixel of the image gets a colour which is the inverted colour of the previous one. This process is the most important one because any character on a sample image can only be extracted efficiently if it contains only one colour which is distinct from the background colour. Note that it is only required if the objects we have to identify if of darker intensity on a lighter background.

Figure 2 shows Image Processing steps: RGB, Grey Scale, Binarization, and Inversion.

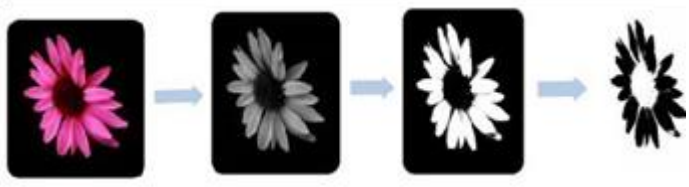


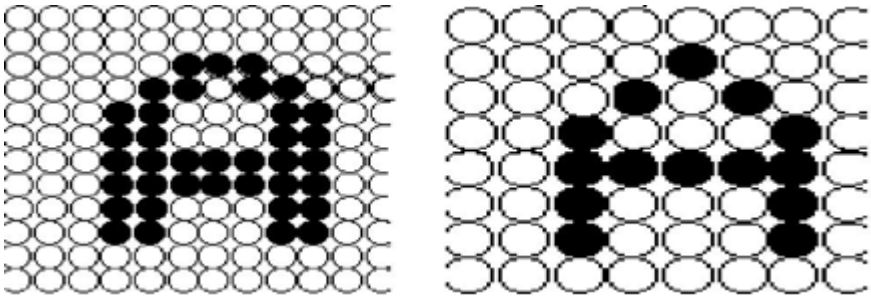
Figure 2: RGB => Grey Scale => Binarization => Inversion

3. SEGMENTATION

In the segmentation stage, an image of the sequence of characters is decomposed into sub-images of individual character. The pre-processed input image is segmented into isolated characters by assigning a number to each character using a labelling process. This labelling provides information about a number of characters in the image. Each individual character is uniformly resized into pixels.

3.1 Normalization:

After extracting the character we need to normalize the size of the characters. There are large variations in the sizes of each Character hence we need a method to normalize the size. Figure 3 shows the image before and after normalization.



Original Image Normalized Image

Figure 3

4. FEATURE EXTRACTION

Features of a character depict the morphological and spatial characteristics in the image. Feature extraction is a method of extracting features of characters from the sample image.

4.1 Structural Feature Extraction:

This is a primitive method of feature extraction which extracts morphological features of a character from image matrix. It takes into account the edges, curvature, regions, etc. This method extracts the features of the way character are written on image matrix. The following are essentials of Structural Feature Extraction:

Indexing and Labelling

This is a process by which distinct characters in an image are indexed and labeled in an image. Thus helps in classification of characters in the image and makes feature extraction of characters simple.

Boxing and Cropping

This is a process of creating a boundary around the characters identified in an image. This helps by making cropping of characters easier. After boxing, the characters are cropped out for storing them as input variables for recognition.

Reshaping and Resizing

Reshaping is done to change the dimensions of the acquired character in the desired shape. Resizing is done to reduce the size of characters to a particular minimum level.

5. CLASSIFICATION

5.1 kNN (k space nearest neighbor algorithm)

In pattern recognition, the k-nearest neighbors' algorithm (k-NN) is a non-parametric method used for classification and regression [2]. 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 integer, 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, it can be useful 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 shortcoming 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.

6. RESULTS

6.1 Data Set

Figure 4 shows the data set used for training.

Figure 4: Training Data

6.2 Outputs

Figure 5.1 and 5.2 shows output for file alphabets.png

Figure 5.1 Image alphabets.png

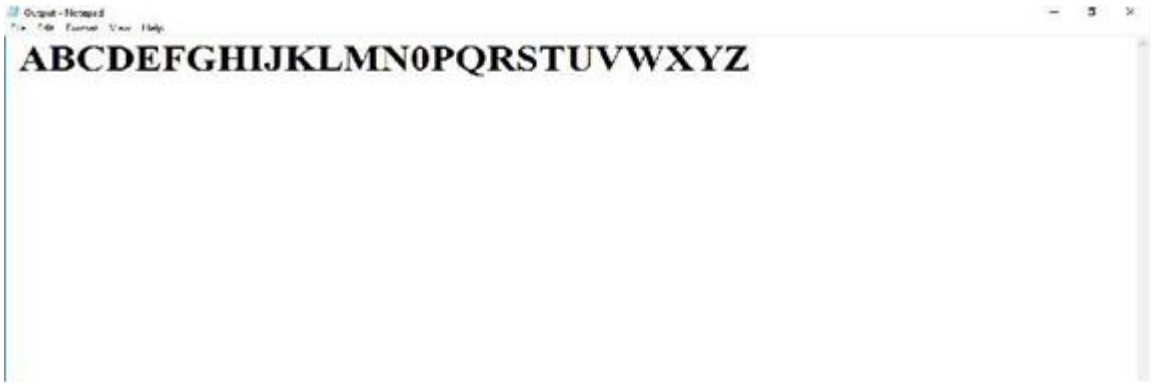


Figure 5.2 Output produced

Figure 6.1 and 6.2 shows output for numbers.png

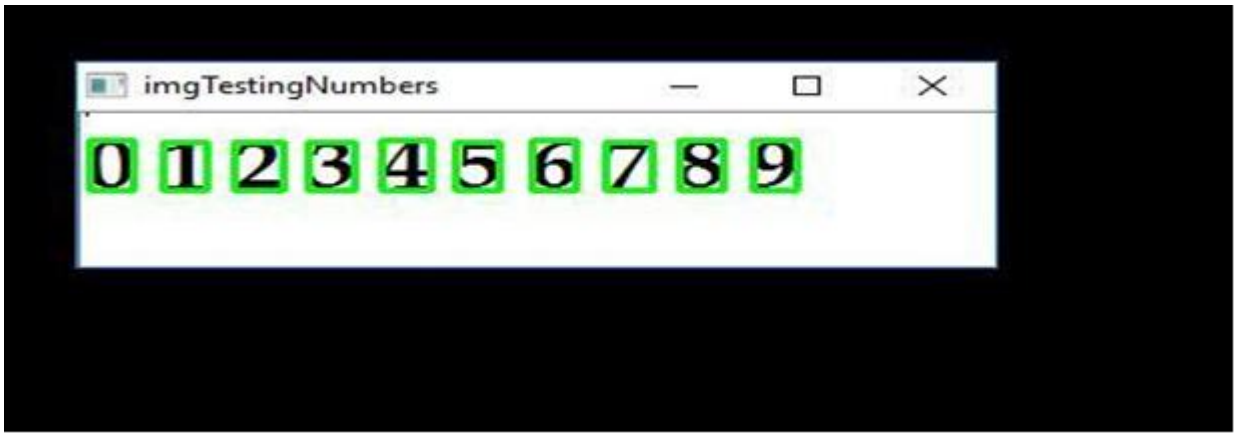


Figure 6.1 Image numbers.png

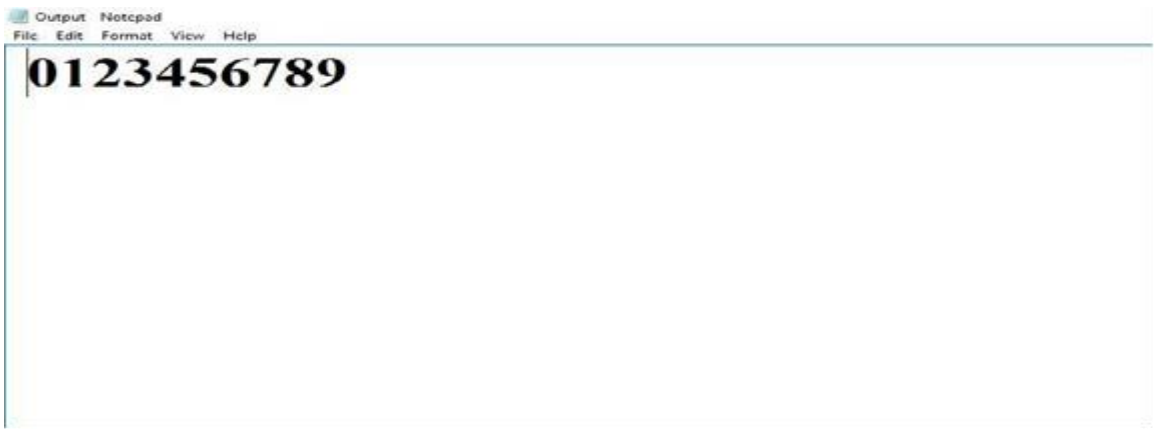


Figure 6.2 Output produced

Figure 7.1 and 7.2 shows output for small_letter.png

Figure 7.1 Image small_letters.png

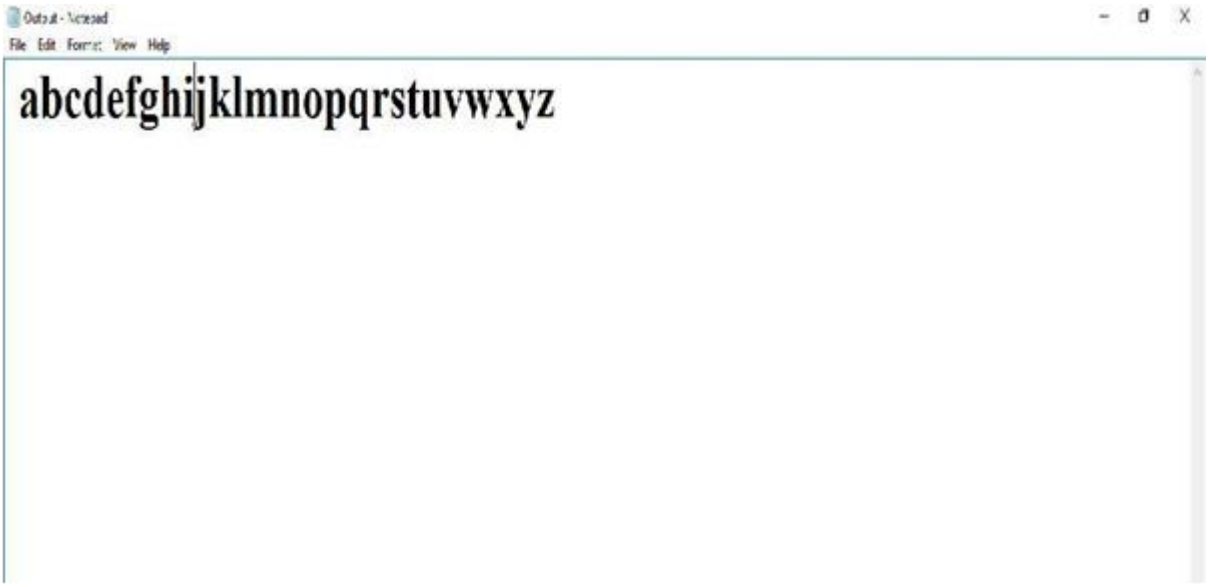


Figure 7.2 Output produced

Figure 8.1 and 8.2 show output for cat.png

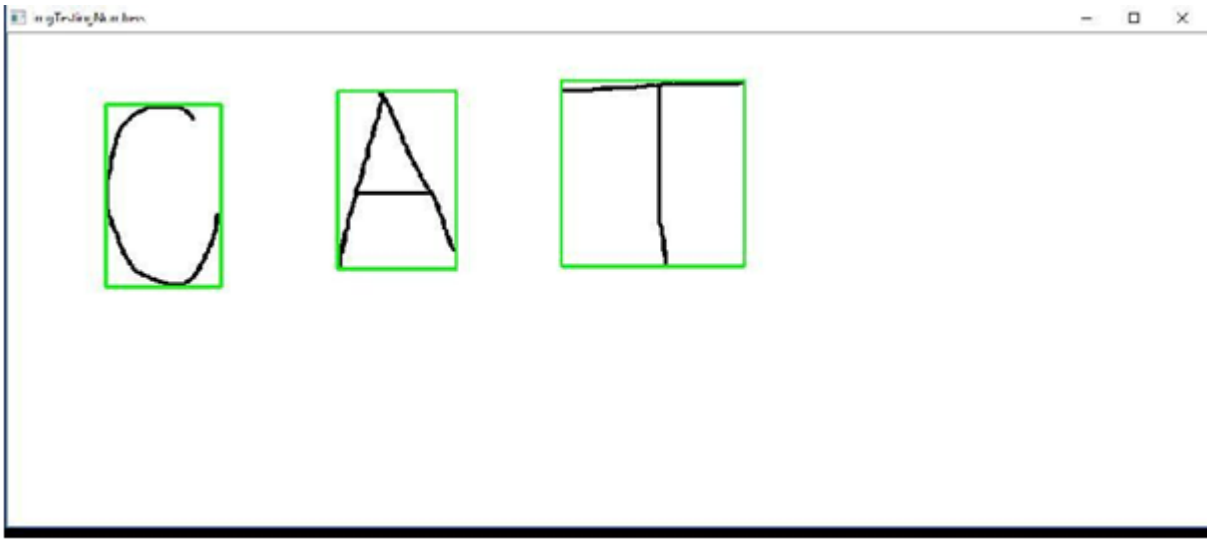


Figure 8.1 Image Cat.png

Figure 8.2 Output Produced

7. CONCLUSION

From the developed system we concluded that effectiveness of the system depends upon feature extraction. The normalization techniques play an important role in the overall efficiency of the system. The success of the system also depends vastly on the ability of the k-space nearest neighbor algorithm to effectively recognize patterns. The methodology has produced good results for images containing written and printed text written in different styles, different size, and alignment with the varying background. To make the system more accurate, we can use more than one type of classification technique and also varied training sets to properly train the system. kNN proves to be effective when the small dataset is used for training, however, the efficiency of algorithm diminishes when more features are added or/and training data volume is increased.

The program code for this system can be found at: https://github.com/akarr24/Character-Recognition

8. ACKNOWLEDGEMENT

The author would like to thank Professor Amit Chandanan, Head of Department, IT, Hitkarini College of Engineering and Technology for his guidance and support.

9. REFERENCES

[1] Machine Learning, Mitchell Tom McGrawHill, 1997
[2]"An introduction to kernel and nearest-neighbor nonparametric regression" Altman, N.S. (1992).

