**PROJECT REPORT**

**TEXT EXTRACTION FROM IMAGES**

Table of Contents

**Introduction3**

**Literature Review4**

**Block Diagram7**

**Algorithm8**

Training8

Testing9

**Result10**

Training11

Testing12

**Conclusion and Future Reference13**

**References13**

**Introduction**

Text extraction from images is basically extracting the relevant text data from image. Due to the rapid evolution in technology there has been digitization in all areas. Lots of resources are available in electronic medium. Many existing paper based collections, scanned documents, pictures, book covers, magazines, pamphlets, educational, business cards etc. are converted into images.

[6] Several studies on text extraction from images and video have been suggested in the past. These studies can be categorized into two types: texture-based approaches and region-based approaches [7]. Region-based algorithms are based on the properties that distinguish text from background, such as color, edges, intensity, etc. These methods are bottom-up approaches that detect small candidate regions and then group them into text regions

Image Text is the text information embedded or written in image of different form. Image text can be found in captured images, scanned documents, magazines, newspapers, posters etc. These image texts are highly available nowadays and they are very important in representing, describing and transferring information which help peoples in communication, solving problems, availability, creation of new types of jobs, cost effectiveness, productivity, globalization and cultural gap etc. The information from these image documents would give higher efficiency and ease of access if it is converted to text form. The process by which Image Text converted into plain text that computer can recognize its ASCII character is Text Extraction. The information from image documents should be converted into text in order to get efficient use and access of it like archiving or reporting that are used in different image based applications such as office works.

Some of these images (containing text) impose many challenging research issues in text extraction and recognition. Extraction of text from images have many useful applications in detection of vehicle license plate , document analysis , analysis of article with tables, keyword based image search, identification of parts in industrial automation , content based retrieval, object identification, name plates, street signs, document retrieving, , page segmentation, text based video indexing address block location etc. Due to the evolution of the multimedia documents and growing requirement, studies show a great amount of interest in efficient extraction of text. Intensive research projects are performed for text extraction in images.

Although Text Extraction is worked on a lot, there are several obstacles that every researcher faces while extracting text from images or videos. Usually these obstacles are due to the various properties of an image; like different size, color, font and alignments or often also due to complexity of background and bad quality of the image. It is due to these obstacles that even today text extraction remains unsolved (100% accuracy).

**Literature Review**

A large number of approaches have been proposed during the course of period for extracting the text from the images. The existing work on text extraction from images can be classified according to different criteria. This paper classifies methods according to the different types of image, analyzes those algorithms and discusses them.

P. Nagabhushan [1] proposed an efficient approach to detect and extract the text in complex background color document images. The proposed method uses an algorithm called the canny edge detector algorithm to detect the edges. Once the edges are obtained, the dilation operation is performed on them. It was found that it created holes in most of the connected components. These correspond to character strings. Connected components without holes were eliminated. Other nontext components were eliminated by computing and analyzing the standard deviation of each connected component. There was an unsupervised local thresholding which was devised to perform foreground segmentation in detected text regions. At the end the noisy text regions were identified and were processed to further enhance the quality of retrieved foreground.

Thai [2] describes an approach for novel text extraction from graphical document images. The proposed method uses Morphological Component Analysis (MCA) algorithm. This algorithm describes an advancement of sparse representation framework. It shows two appropriately chosen discriminative over complete dictionaries. Two discriminative dictionaries were based on undecimated wavelet transform and curvelet transform.

Angadi [3] proposed an automated method to detect and extract text prior to further image analysis. The proposed methodology uses Discrete Cosine Transform (DCT) based high pass filter to remove and prevent the dissemination of the constant background. The processed image was divided into 50x50 block and then the texture feature matrix was computed. A discriminant function which was anew was used to classify text blocks. The detected text blocks were merged back together to obtain new text like regions. Finally, the refinement phase was a post processing step used to improve the detection accuracy. This phase used to cover small portions of missed text present in adjacent undetected blocks and unprocessed regions. The proposed method comprises of 5 phases; Background removal/suppression in the DCT domain, texture features computation on every 50x50 block and obtaining a feature matrix D, Classification of blocks, merging of text blocks to detect text regions, and refinement of text regions. This methodology had been conducted on 100 indoor and outdoor low resolution natural scene images containing text of different size, font, and alignment with complex backgrounds containing Kannada text and English text. The approach also detected nonlinear text regions and can be extended for text extraction from the images of other languages with little modifications.

Thanh-Ha do [4] proposed an approach to extract text regions from graphical documents. In this method, first empirically two sequences are constructed of learned dictionaries for the text and graphical parts respectively. Then sparse representations are computed for all different sizes and non overlapped document patches in these learned dictionaries. Each patch can be classified based on these representations. It is classified into the text or graphic category by comparing its reconstruction errors. Same-sized patches in one category are then merged together to define the corresponding text or graphic layers which are combined to create a final text/graphic layer. Finally, in a post processing step, text regions are further filtered out by using some learned thresholds.

Ranjini [5] talks about English text extraction from blob in comic image. Detecting text and extracting text from comic images helps to preserve the text and formatting during the conversion process and it provides very fine quality of text from the printed document. Initially, a pre-processing is done on the image. In the pre-processing step the RGB images are converted into a binary image by applying the threshold values between 0 to 1.Then the image is subjected to balloon detection. CCL algorithm is applied to the noise removed RGB images for detecting the connected components in the image which helps to detect the connected components in the image often it is used for Balloon detection. Text blob extraction is performed on them which are used to identify text blobs from non text blobs, To avoid the false detection and to reduce the complexity the text blobs need to be identified exactly. The identification is done, based on the features of blob size. Finally, the text is recognized and extracted by using the optical character recognition method.

These are just a few examples of the work of previous researchers in this field. All this research laid the foundation to more modern ways of extraction of text using MSER (Maximally Stable Extermal Regions) and OCR (Optical Character Recougnition).

During my experiments with text extraction, I at first used a combination of MSER, Stroke Width Detector and OCR to solve this problem. However since these methods are based on feature extraction, the algorithm only worked for a certain type of images and would make gross mistakes if any other image was inputted. I hence moved to deep learning to find a solution to this problem.

For this paper, I used transfer learning to fit a dataset of handwritten characters to the well-known VGG-16 neural network to make the system recognize characters in an image.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

VGG-16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. For this specific problem, I altered the last 3 layers of VGG-16 network that classify the data to make them classify the characters instead. With this altered network, a training model was made that can now be used to recognize characters.

**Block Diagram**

BLOCK 1: An image containing a character is inputted into the algorithm.

BLOCK 2: The image is processed to fit the training model based on VGG-16. The input size of the VGG-16 model is 3x224x224. Hence, the image is converted into size 224x224 and is then converted from Greyscale to RGB (if needed).

BLOCK 3: The image is then passed through the 43 layers of the altered VGG-16 model to process the images and extract its features so that the system can understand if it fits the description of being a character.

BLOCK 4: After being passed through the neural network, the system finally gives the output and predicts what character was present in the image.

**Algorithm:**

**Training:**

At first, we must load the training dataset from the folder. After this, resize all images to fit the VGG-16 model requirements (3x224x224). Now that the dataset is ready, we must prepare the neural network. After loading the VGG-16 network, we must change the last 3 layers from the 43 layers. These layers are responsible for classifying the image. VGG-16 is trained to classify 1000 different images; hence we need to remove the last 3 classification layers and add our own layers to classify our specific dataset. Now that the setup is done, we start training the model. Training the model can take time depending on the system specifications as VGG-16 is a massive neural network consisting of 43 complex layers. Once the training process is complete, we need to save the model so that we can use it again later to extract text.

**Testing:**

Preparing the dataset for testing the images is the same as that of training\*. Once this is done, we load the model we made during training. This model is then used to classify the images and make predictions of each character. After this is done for all images, we form a confusion matrix of all the outputs and hence calculate the average accuracy of the model.

Note\* – The dataset used for testing should be distinct. If images used while training the model are used for testing also, we won’t get accurate results as the model already knows what the image is before even classifying it.

**Result:**

In this experiment, a dataset of handwritten character images was used, with each character having 6k-70k sample images as training set and another 200-2000 sample images as test set. Due to hardware limitations, we used 4000 images each for training and 10/100 images each for testing. The model was trained to detect 38 characters i.e. a-z (both capital and small), 0-9 and a few special characters like ‘#’, ‘$’, ‘&’ and ‘@’.

Example:



Fig 1. – Variations of ‘#’ dataset

C:\Users\Nishant Nadkarni\AppData\Local\Microsoft\Windows\INetCache\Content.Word\0.jpg



Fig 2. – Variations of ‘2’ dataset



Fig 3. – Variations of ‘a’ dataset\*



Fig 4. – Variations of ‘A’ dataset\*

Note\*- Both ‘A’ and ‘a’ are a single dataset. Hence, the system reads both of them as ‘A’. This is common for all letters.

**Training:**

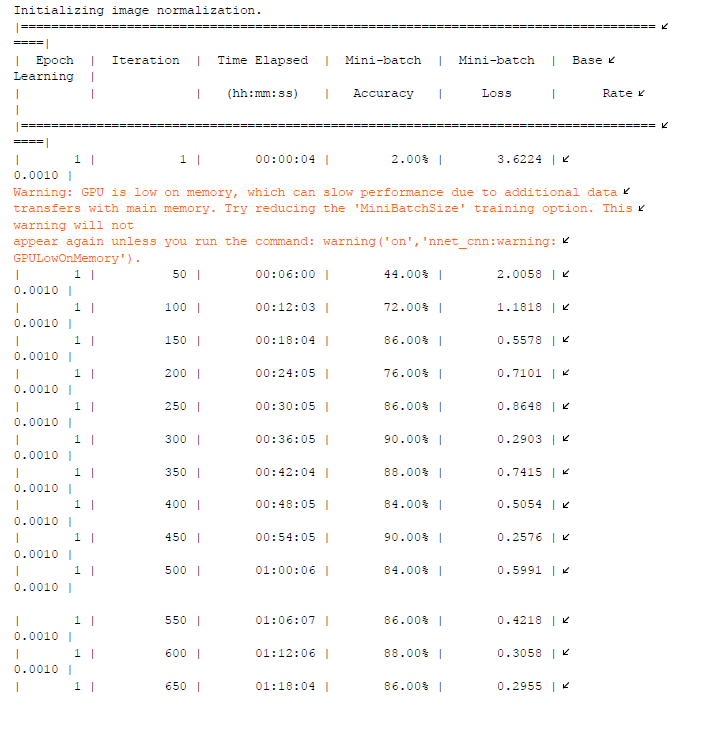


Fig 5.

Figure 5 shows the first few iterations of the training data. There were total 15,200 iterations and the model took 30 hrs 14 mins and 1 second (1 Epoch) to run them.

The model was made with a training rate of 0.001 and a batch size of 50.

**Testing:**

Case 1:



Fig 6. Testing using 10 images per character

Case 2:

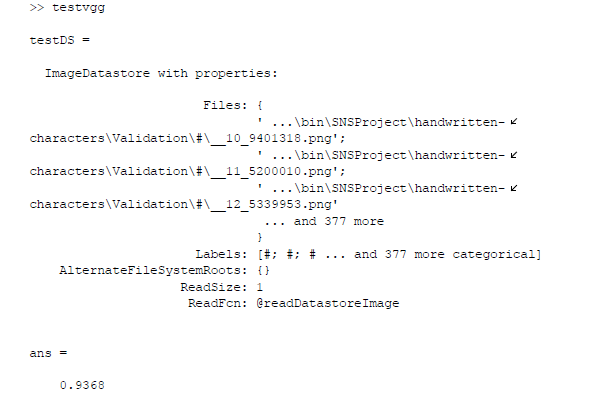


Fig 7. Testing done using 100 images per character

**Conclusion and Future Recommendation**

Both methods of text extraction (Method 1-MSER/OCR and method 2-deep learning) work for their own specific inputs. While the first method works only for specific type of images, the second method works for almost any handwritten text with ~92% accuracy. Since method 2 covers a wider area of images and does not need manual entry of data (features) as in MSER/OCR, it is better. However deep learning requires a system with a powerful GPU and CPU to train the model and hence has severe hardware limitations. No such limitations exist in method 1. However deep learning is also very flexible and can give higher accuracy if the model is allowed to train for longer. Using deep learning, one can also train the previously trained model with a new dataset and train the model further to classify printed text. Also, the higher system needs are just needed to train the model, once the model is trained, it can be used to extract text even on low end systems. Also, this algorithm could be further upgraded to detect a group of text using image segmentation and OCR first and then using this algorithm to extract individual character.

**References**

[1] Nirmala Shivananda and P. Nagabhushan,” Separation of Foreground Text from Complex Background in Color Document Images”, IEEE Seventh International Conference on Advances in Pattern Recognition, 2009.

[2] Thai V. Hoang , S. Tabbone(2010),“Text Extraction From Graphical Document Images Using Sparse Representation”in Proc. Das, pp 143–150.

[3] S. A. Angadi and M. M. Kodabagi,” Text Region Extraction from Low Resolution Natural Scene Images using Texture Features”,IEEE 2010.

[4] Thanh-Ha Do, Salvatore Tabbone, Oriol Ramos-Terrades,” Text/graphic separation using a sparse representation with multi-learned dictionaries”, 21st International Conference on Pattern Recognition (ICPR 2012)November 11- 15, 2012. Tsukuba, Japan.

[5] Dr.M.Sundaresan and S.Ranjini,” Text Extraction from Digital English Comic Image Using Two Blobs Extraction Method”,IEEE Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME) March 21-23, 2012.

[6] Najwa-Maria Chidiac, Pascal Damien, Charles Yaacoub “A Robust Algorithm for Text Extraction from Images”. 2016 39th International Conference on Telecommunications and Signal Processing (TSP) 27-29 June 2016.

[7] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images

and video: A survey,” Pattern Recognit., vol. 37, no. 5, pp. 977–997, 2004.