International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 09 | Sep -2017 www.irjet.net p-ISSN: 2395-0072

Text Recognition Using Convolutional Neural Network: A Review

Rutuja Medhekar[1], Anushree Chopde[2], Sijin Saji[3], Tejesh Shelke[4], A.Jadhav[5]

[1234]*Department of Information Technology, JSPM's RSCOE, S. P. Pune University, Pune, India.*
[5] *Professor, Dept. of I.T Engineering, JSPM's RSCOE, Maharshtra, India.*

Abstract - *This paper helps to understand the methodology in converting the old printed documents into a soft copy .This will be an android app which will take image and convert it into soft copy using OCR (Optical Character Recognition) principles and concepts. It will be more beneficial as it an android app. This paper is a review of how our system will work and it also reflects error detection and correction techniques to make system more efficient. It also focuses on developing our own OCR engine to improve the existing one and remove the anomalies, if any. It is based on Artificial Neural Network and Nearest Neighbor concepts.*

Key Words: **Optical Character Recognition, Error detection and Correction, Artificial Neural Network, Nearest Neighbour.**

1. INTRODUCTION

To have the softcopies of Government documents, business officials, old newspapers, worn out documents which are in hard format is a very tedious job. In order to ease this process we have various OCR engines but at certain point even they have certain anomalies which complicate the process in turn. Various researches and advancement is going on to improve its functioning. We, therefore are making an attempt to develop our own OCR engine to ease out the process of text detection, recognition and correction and it will also make offline conversion of text possible.

This paper highlights the fact that how an android app will make it possible for every common man to ease the tedious process of scanning and converting the images into soft copy just on a click even while working offline. It focuses on comparative study and implementation of OCR concepts and principles using Artificial Neural Network and Nearest Neighbor concepts. It also clearly explains every processing stage in detail and efforts taken to overcome the drawback of existing mobile device OCR applications and limitations of mobile device capabilities so that it generates powerful app which processes things in reduced time and increased accuracy that recognizes almost every different font.

This paper clearly mentions the review of our system in following parts:

Section A -explains the processing stages that are carried out and also defines various concepts that are used in it.

Section B- Presents the applications of our system and also how efficiently the work will be extended in future so that it will be useful for every common man including physically challenged people.

2. METHODOLOGY

This part focuses on methods used in our system. This includes 1.Preprocessing Techniques, 2.Data Collection 3.Text detection 4.Character Segmentation and recognition 5.Spelling correction.

The following explains first how Neural Network concepts are applied:

2.1 Artificial Neural Network

An Artificial Neural Network (ANN) is a soft computing technique. This technique can be applied where we can have a large solution domain. The main purpose of ANN is as a classifier for different classification algorithms. This technique is inspired from the nervous system of human brain.

The proposed structured of the survey is a three layer architecture. First is the input layer which receives input from the segmented character images of standard size. Second layer is a hidden layer, this layer is use to train the neural network for specific font styles in case of character recognition process. The final layer is the output layer, this layer is used to generate Unicode values for different characters which itself worked as a matching criteria.

3. Step by Step Process

3.1 Data Collection

For the identification of text from an unseen image, firstly we need to strengthen our database which contains the definition of the character identified from the test to display the text contained by the image.

3.2 Pre-Processing

During this step pre-processing methods such as resizing and quantization are used. As the main difficulty in the image processing field is the size of the image which depends on

the image resolution means number of pixel contained by the image. The total number of pixels leads to the number of iterations performed to do the work to reduce the number of iteration we need to resize the image to some standard size like 256X256 or 512X512 etc. and also color quantization is required.

3.3 Segmentation

After performing the pre-processing steps the next step is extract the idea from the image that where text is present in the image this can be done by using Canny edge detector filter.

Algorithm for Segmentation process is :

Step1: Pick the topmost left pixel of the image as first pixel of first line and consider its coordinates x and y as pixel(0,0). Also set the line number L_number=0, initially.

Step2: Consider each pixel along the line by fixing the value of y component.
a. on the appearance of a black pixel, consider the value of y-coordinate as the line top (l_top).
b. In case of the absence of any black pixel in the line, increment the value of y-coordinate and try to search in the next line.

Step3: Consider each pixel along the line by fixing the value of y component.
a) if no black pixel appeared along the width then consider y-1 as line bottom (l_bottom).
b) In case of the presence of any black pixel in the line, increment the value of y-coordinate and tried to search in the next line.

Step5: start below the bottom of the last line found and repeat steps 1-4 to detect subsequent lines

Step6: When the last line of image reached then stop.

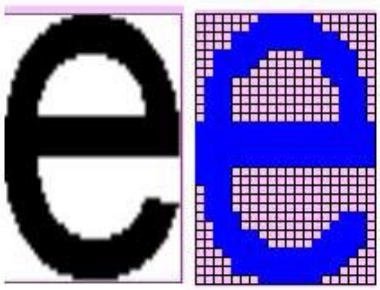

Fig. 1 Segmentation results

NEURAL NETWORK OCR ARCHITECTURE:

This is a classification approach. This works on the concept that, for every component, two consecutive nearest neighbours are found by the application of Euclidean distance. The different parameters like dimensions, distance and the alignment of component and their neighbours are compared.

It tells about network size and configuration setups are explained. The size and initial estimates of the accuracy are based on the Matlab simulations.

A. Feedforward Neural Network –

A feedforward neural network consists of a high number of neurons, were each neuron consists of a nonlinear activation function and followed by addition. There is no data transfer between neurons in the same layer; neurons only send data to neurons in the next layer. The feedforward neural network is widely used due to its relatively simple structure and effectiveness in classification problems.

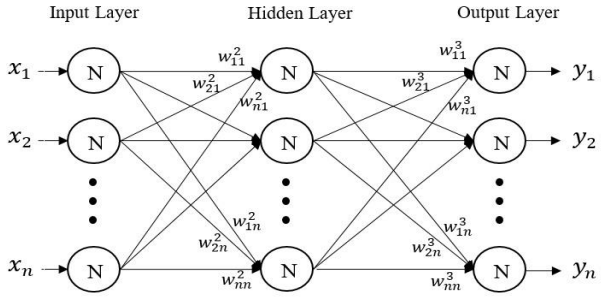

Fig. 2.Generic structure of a 3-layer feedforward network

B. Network Setting –

Trained by a supervised learning algorithm, the feedforward neural network is a data-driven classifier. The learning algorithm used in this work is the scaled conjugate gradient, which is suitable for optimization problems with a lot of parameters such as the feedforward neural network configuration used here.

In this work, the test images were not retouched to remove noise, and rather the objective was to implement a network that is capable of classifying images in the presence of noise, shadow, and other environmental imperfection.

E.g. Fig. 2 shows samples that were used for training and testing the network. The used images are all from the resized and segmented images of Ontario license plates. Therefore, the size of the input layer of the network was set based on the size of the input image. A resolution of 21_9 was used for each image. Therefore, the input layer was set to 189. The output layer was used to generate a 5-bit binary number as an indicator for various characters, in this case, alphabet and numbers. Using Matlab, a test program was developed to evaluate various network configurations. In the setup, the number of nodes in the second layer was set variable. The

range of the number of nodes for the second layer was set from 100 to 200 with a step size of 10. From this result, 160 were determined to serve as the most suitable number of nodes for the network's second layer.

Therefore a network size of 189 160 36 was configured for this application. When the optimum size for the feedforward network was determined, the network was trained and tested in Matlab. The dataset was divided into training data and testing set. The test data was only used to verify network's performance after the training was finished. In other words, the network has not seen the testing data during the training phase. The network was able to classify the noisy and imperfect images in 99:1% of times.

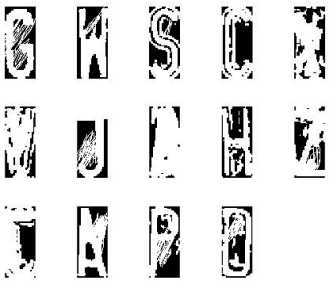

Fig.3 Unclassified data

C. NEURAL NETWORK-BASED OCR IMPLEMENTATION –

After the structure and size of the network had been determined, the next task was to implement the hardware realization.

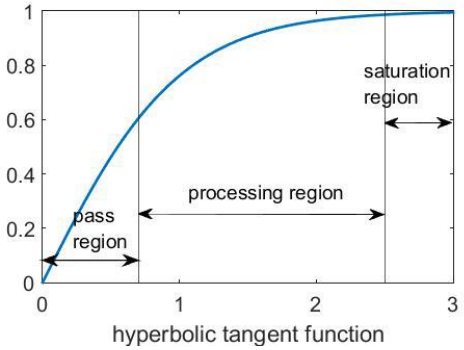

Fig Different regions defined in hyperbolic tangent function

Saturation Region: In the saturation region, the function is close to its maximum value of 1. In this region, the output of the neuron varies within a small range. Therefore, if the boundary of saturation region is set properly, the function in this region can be expressed as a constant value f(x) = C, while keeping the error within the acceptable range.

Processing Region: The remaining part of the function is the processing region, which is the most non-linear part of the function. In order to approximate the activation function in this region, it is divided into smaller sub ranges. Two issues should be considered here, the number of required sub ranges, and the value representing each section. In order to determine the approximate function, the maximum allowable error between the approximation and ideal function (_) is of significance. By increasing the number of sub ranges, the approximation error can be decreased, but at the same time, it increases the area and complexity of the design.

3.4 Error Detection and Correction

OCR accuracy has affected by poor image quality (e.g., scanning resolution, noise) and any mismatch between the instances on which the character image classifier was trained and the rendering of the characters in the printed document (e.g., font, size, spacing).There are chances of producing different error distribution by our OCR engine as it depends on the language and the image quality of analyzed collection. These errors can be categorized according to the following types, listed in the order they occur during the OCR process:

- Word detection – which fails to detect text in the image, commonly caused by poor image quality or text mixed with graphics.
- Word segmentation – failing to bound an individual word correctly, due to wrong interwork space detection, generally due to different text alignments and spacing.
- Character segmentation – which fails to bound single characters in a segmented word. This is very frequent for cursive or connected alphabets, such as printed Arabic or handwritten Latin-alphabet languages. It may also occur due to an analog process (e.g., printing and scanning speckles) that might pause the connected components.
- Character recognition – failing to identify the correct character for a bounded character image.

ERROR CORRECTION METHODOLOGY:

The OCR error model is vital in suggesting and evaluating candidates. At the heart of the error model is a candidate generation for correction, based on a confusion matrix giving conditional probabilities of character edits. The possible error corrections include the error types listed below, except word detection problems as the algorithm has no input image to correct. The latter has to be addressed with image preprocessing or detection robustness.

The error correction methodology comprises three stages:

1) Generating correction candidates – The original word is expanded using confusion matrix and a dictionary lookup, forming all together a correction-candidates vector.
2) Feature extraction – These features are extracted for each word in the vector.
3) Word classification – A two-stage classification process, where the first stage ranks the correction candidates according to their correctness probability at this position, while the second selects the highly possible word between the original word and the highest-ranked candidate.

Training Data:

This is the type of data that will resemble the test data.

OCR Text Tokenizer:

The text is tokenized using standard space delimiter tokenizer so as to structure the text for enabling correction at a word level. It parses word recognition strength produced by the OCR engine, forming a first level feature extraction.

Correction-Candidate Generator:

This module is designed in order to generate correction candidates for a tokenized word in accordance with an observed OCR error model. This error model supports the correction of erroneous character segmentation and recognition, as well as word segmentation.

a) Character segmentation and recognition errors
b) Word segmentation errors

Candidate Ranker:

The ranker's role is to produce an ordered word vector of correction candidates, calculating a score for each correction candidate, which correlates with how probable a correction is at a specific position. Every candidate is scored individually from all others in the word vector; then this candidate is compared to all the other correction-candidates. This hardly considers the original OCR output, as it has different features and will be considered in a secondary stage. From the preliminary stage, the input vector was cleaned from all its non-dictionary words.

Correction Decider:

The correction decider is a classifier that decides whether a replacement should be made of the OCR word with its highest ranked correction-candidate.

1. **Feature Extractor***:* The decision is thus calculated by a trained regression model using following correction pair features as input:
   - ✓ Confidence
   - ✓ Term frequency
   - ✓ Proportional dictionary features
2. **Decision Decider:** This decision is made by a model trained on the total transcribed corpus of correction pairs. Pairs with erroneous OCR word and correct candidate were marked with a positive output, indicating that these cases are suitable for replacements.

4. APPLICATIONS

Our system therefore has following applications:

1. Data Entry for business documents, e.g. check, passport, invoice, bank statement and receipt.
2. Extracting business card information into a contact list.
3. Make electronic images of printed documents searchable, e.g. Google Books.

Apart from these there are various applications which overcome the drawback of existing applications.

4. CONCLUSIONS

Performance of the current system is observed by taking into consideration the variations in number of iterations and the variations in the number of characters. The proposed system produce 98.89 % recognition rate for three different fonts by considering up to 90 characters at a time.

REFERENCES

[1] International Journal of Machine Learning and Computing, Vol. 2, No. 3, June 2012.

[2] 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)

[3] 2016 12th IAPR Workshop on Document Analysis Systems.

[4] https://blogs.dropbox.com/tech/2017/04/creating-a-modern-ocr-pipeline-using-computer-vision-and-deep-learning/.

[5] 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 www.iosrjournals.org.

[6] https://en.wikipedia.org/wiki/Optical_character_recognition.

[7] http://ieeexplore.ieee.org/document/6836618/.

[8] M UsmanRaza, et al., "Text Extraction Using Artificial Neural Networks", in Networked Computing and Advanced Information Management (NCM) 7th International Conference,,Gyeongju, North Gyeongsang, 2011, pp. 134 - 137.

[9] Fonseca, J.M., et al., "Optical Character Recognition Using Automatically Generated Fuzzy Classifiers", in Eighth International Conference on Fuzzy Systems and Knowledge Discovery, Shanghai, 2011, pp. 448 - 452