Extraction of handwritten and printed text from a form

Aayush Shah, Neel Doshi, Jainam Shah, Kunal Goel, Purva (Deepali Choudhari) Raut
Dwarkadas J. Sanghvi College of Engineering, Mumbai – 400056, India
Email: 08shahjainam@gmail.com, aayushshah2597@gmail.com, nldoshi414@gmail.com, goelkunal59@gmail.com, deepali.choudhari@djsce.ac.in

Abstract. A lot of offline information or request forms available at certain government firms and similar companies use text boxes to fill the information. The data from these manually filled forms has to be manually filled in the machine database requiring a person to type the data. The proposed system uses a blend of machine learning technologies to extract the text data from such forms and automatically convert it to text that can be processed and understood by the computer.

Keywords. ANN, CNN, Pytesseract, OpenCV2, Contours.

1. Introduction
There are times when instead of an online form, the customers need to be presented with an offline form that needs to be filled by hand. A lot of banks, government offices and agencies use handwritten forms to obtain details and queries from the customers. Nowadays, almost every company or office has employed a digital database along with their websites to store customer information and other details. Due to this, the employees have to manually type the data of the form and add it to the database for every form that is filled. This consumes a lot of time and efforts and is not an efficient way. The manual typing of the data can also result in a few errors that can go unnoticed and filled in the database. To overcome these drawbacks and to reduce the time and effort drastically, the paper proposes a system that can convert the text written in the text boxes in the forms to digital data that can be processed by a machine. This is possible due to the advancements in machine learning algorithms and techniques. Open Source Computer Vision Library (OpenCV2) is an open source computer vision and machine learning software library [1]. The library focuses on image processing, video capturing and analysis. Furthermore, Convolutional Neural Networks (CNN) are used to obtain the text data from handwritten text. Along with this, Pytesseract is used which is an Optical Character Recognition (OCR) Tool. Pytesseract is used to identify the text data from printed text through the scanned image of the form. The proposed system aims to reduce the time needed to convert handwritten text box forms into digital text and facilitate the society. The system will also facilitate further researchers and provide a starting point for further improvements.

2. Literature Survey
There are various techniques used for character recognition from images of handwritten or printed text as input. Neural Networks such as Convolutional Neural Networks have been applied to implement character recognition of handwritten text [2]. These different approaches have different requirements
for dataset corpus. This section compares some notable methods and approaches which have been used for character recognition and discusses their advantages as well as drawbacks. The inferences made from the review of these approaches were beneficial.

2.1. Handwritten Capital Letter Recognition based on OpenCV
Optical Character Recognition (OCR) has always been a largely researched topic. The ability to recognize handwritten characters on a piece of paper and transfer it to the computer in a format that can be processed and worked with by the computer. The main research of the paper is the recognition of capital English alphabet letters drawn with a connected input such as a mouse. The paper deals with mouse movement track reading and processing. The programming language used for development is C++ with Microsoft Visual C++ 2010[3]. The image needs to be pre-processed to indicate the characteristics of letters. Image binarization operation is used for this part. The grey part of the processed image is kept at 0 or 255 to make the black and white part of the image very prominent. The application needs a feature library during the design process to extract the feature value of every character written. Template matching is used for the recognition of the letters. This is the last step in the process.

- Advantages: Primitive design, simple code, versatility in implementation on any machine, inexpensive.
- Drawbacks: Does not recognise written letters, needs a connected input. Does not process text written on paper.

2.2. Handwritten character recognition using Deep-Learning
This paper presents a system for offline detection of handwritten characters using deep neural networks. In the system, handwriting recognition is based on image segmentation. OpenCV is used to perform Image processing and Tensorflow is used to train the Neural network. The system comprises an Android Application for the frontend and a Backend for the processing. The system uses Convolutional Neural Network (CNN) Model. The system proposed in the paper has an accuracy of 94% [2]. The images are first pre-processed, median filtering is used to remove the noise in the image. The image is then converted to greyscale to ensure uniformity in the images. The handwritten characters are separated from the background using thresholding. The image is then segmented into separate lines and each line is further segmented into words, which are then segmented into separate characters.

- Drawbacks: The system does not recognise cursive handwriting. The system only recognises English characters.

2.3. Automatic Classification of Handwritten and Printed Text in ICR Boxes
The paper talks about the significance of difference in the machine printed and handwritten text for recognition techniques. They have proposed a classifier which classifies text present inside ICR cells as handwritten or machine printed text. Based on the outcome of this classification, the images are given input into OCR or ICR engines for achieving optimal performance.

This process involves fundamental steps such as scanning the document, performing the pre-processing of the scanned document, and zone segmentation which contain ICR cells for individual components. The features are extracted from the components and the classification process is executed by the system.

A set of new features are introduced, which can be extracted from the processed images and an approach is proposed to find classification rules for identification between the two text types.

- Advantages: The algorithm accurately and efficiently classifies the type of text into machine printed or handwritten text with low computation complexity. It’s robust and can be applied to the majority of document types. It is less time consuming because of its simplicity.
- Future Scope: The recognition accuracy can be further increased by using features like stroke width and pixel distribution.
2.4. An Algorithmic Approach for Text Recognition from Printed/Typed Text Images

This paper proposes a text extraction algorithm from a document which has been scanned in text format, by taking help from Otsu’s algorithm for segmentation purposes and for skew direction, it uses the Hough transform method.

Scanned images of text are converted to character streams by the use of this approach, making them readable by machines. The proposed algorithm only deals with English alphabets (lowercase, uppercase) and numbers (0-9). Steps involved in the process includes image pre-processing, image segmentation conversion into individual characters and character recognition. During pre-processing binary format conversion is made of the scanned image, however most images contain background noise making skew removal and reduction in noise necessary. Hough transformation is used for detecting skew. Based on this the rotation angle is identified. After noise removal, various filters are applied to the image, an enhanced image is obtained by subtraction of the applied filters. Segmentation involves detecting each line and then processing it hence cropping individual characters from the line. The cropped characters are resized according to the size of the template and compared with them by the use of the correlation factor. The proposed algorithm has an average accuracy of 93% and when different font styles are used, average accuracy comes up to about 70%.

- Advantages: Despite rotation and scaling, the algorithm outputs a good result.
- Limitation: Recognition of only the font type Verdana of size 14.
- Future Scope: Improvement in the algorithm can be made, more types of font (sizes and fonts) can be used for system training.

3. Comparisons and Discussions

The four approaches have different recognition methods and each has different dataset requirements. Each dataset has different forms of inputs, both printed and handwritten text is present on the images used in the data corpus. From the above-mentioned papers, ICR Boxes has the best performance in recognizing both handwritten and printed text however the recognition accuracy can still be increased [4]. Handwritten character recognition using Deep-Learning is a good approach, except that it has not been tested to recognize handwritten text which is enclosed between boxes [2]. In addition, this system has used the NIST dataset which does not consist of handwritten numerical characters [2].

| Approach | Character Recognition Method | Dataset Used | Disadvantages | Output |
|----------|-------------------------------|--------------|---------------|--------|
| Tesseract OCR Engine | Adaptive recognition | Over 100 languages used | Does not recognise handwritten characters | Identifies text from printed bills |
| Algorithmic Approach for Text Recognition from Printed/Typed Text Images | Otsu’s algorithm for segmentation | Verdana font type | Recognises only Verdana font type, needs to be improved with more font types | Printed text of Verdana font |
| Automatic Classification of Handwritten and Printed Text in ICR Boxes | Classifier for OCR or ICR | 1663 scanned zones, 947 handwritten, 716 printed zones form | Recognition accuracy can be increased | Recognises printed and handwritten text |
| Handwritten character recognition using Deep-Learning | Convolutional Neural Network | NIST | Does not recognise cursive letters, not useful to extract form data characters enclosed in boxes | Recognises handwritten text from images |
4. Proposed Methodology
The main objective of the research is to develop a project which can extract handwritten information from the images of the forms. Forms such as NGO forms and government forms usually have boxes which have handwritten characters in them and entities such as First Name, Last Name, etc., are printed on the left of the box. The handwritten characters (value to the entity) are recognized with the help of Neural Networks. Both Artificial Neural Networks and Convolution Neural Networks are used for determination for accuracy of prediction. The printed text (entity) is determined by using Pytesseract. Figure 1 shows the proposed architecture.

5. Proposed Algorithm
The project targets to determine both handwritten characters (in boxes) and printed text. After extraction of both text entity-entity value pairs are made which are stored in the form of a dictionary. Figure 2 shows the proposed algorithm.
5.1. Image Pre-processing

Image pre-processing is used for improvement of image data that possesses unwanted distortions and has some data which does not have relevant data. Image pre-processing also helps in reduction of redundancies. Figure 3 shows the form which has been used for the research. Image pre-processing techniques used are:

- Normalisation - In this the range of pixel intensity is changed. Here the normalisation is used to range the pixel value either 0 or 1. The image of the form is sharpened and its further processed.
- Negative Image - Negation of an image is converting the light areas into dark and vice versa. After normalisation negation of image converts the pixel value of 1 to 0 and pixel value of 0 to 1. This step is necessary as it makes it easy for detection of boxes in the next step.

![Figure 3. An Image of the form.](image)

5.2. Detection of boxes

Step 1: Detection of horizontal and vertical lines and applying morphological operations to extract the handwritten text from the boxes it is necessary to detect each box in the image of the form. Detection of the box is done by Contours in the OpenCV library of python. Kernel length is decided. Similarly kernel length is decided for detection of vertical and horizontal lines.

- Vertical Kernel Length - It is used to determine all the vertical lines in the form with given kernel length.
- Horizontal Kernel Length - It is used to determine all the horizontal lines in the form given kernel length.
Morphological operations are performed for detection of the lines. After the detection of the vertical lines in the image the image with vertical line is saved. Similarly, the image with detection of horizontal lines is stored. Both the images are eroded so that the final images contain clear images of horizontal and vertical lines. Now as to form a box both the images of detection are summed up to give an output of an image with only boxes. The morphological operation performed with erosion makes the image of boxes clear with elimination of any noises. The output image is stored and further used by contours. Figure 6 displays the final image of the boxes needed to determine the contours.

Figure 4. Vertical line detection in the form.

Figure 5. Horizontal line detection in the form.

Figure 6. The final detection of boxes in the form.
Step 2: Using contours to detect individual cells of boxes:
Contours an OpenCV2 function is used for analysis of shape, finding the size of the object and detection of the object. The image generated at the end of the morphological operation is considered as the baseline for detection of such boxes. Thus, the contour detects similar objects in the original form. The smaller boxes in which the handwritten characters are present are then detected by defining the height and width of the smaller boxes which are detected from the bigger box outside. The particular small box with character is given to the Deep Learning Model for prediction of individual handwritten characters.

5.3. Deep Learning Model for prediction of handwritten characters
Deep Learning is an Artificial Intelligence function that uses neural networks to imitate the human working. In the project deep learning model is used to predict the character present in the small box which is given to it by a function at the end of the contours detection. For better accuracy and results both CNN and Artificial Neural Network (ANN) are used.

Data Set: The dataset used is EMNIST dataset which is an extended version of NIST data set. The dataset consists of images of size 28x28 pixels and consists of 47 classes of characters which includes upper- and lower-case characters and numerical digits for zero to nine [5]. These data values are converted into an array which represents the respective image and is then given to the model to train.

• Training dataset: It consists of 112800 rows and 785 columns. Each cell in the dataset represents the pixel value of an image of either an alphabet or digit. The first column of the dataset represents the class value and the rest 784 columns forms an image of 28x28 pixels.
• Testing dataset: It consists of 18800 rows and 785 columns. Similar to training dataset each value determines the pixel value of the image and the first column represents the class of the image. 784 columns represent pixel values of 28x28 pixels.

5.3.1. Artificial Neural Network. An Artificial Neuron Network (ANN) [6] is also known as a Neural Network. The structure and functions of these biological neural networks makes it a computational model. It receives, processes and transmits information similar to an artificial human nervous system [6].

The 3 types of layers in a neural network are as follows:
• Input Layer: This layer is responsible for communicating with the external environment and also represents the condition for which we are training the neural network. Its job is to deal with all the inputs only [6]. The inputs then get transferred to the hidden layers. Every input neuron has an influence over the output.
• Hidden Layer: The hidden layer is an intermediate layer found between the input layer and the output layer and which has an activation function applied on it. It is a collection of neurons which processes the inputs [6] obtained by its previous layer and thereby extract the required features. There are multiple hidden layers present in a single Neural Network.
• Output Layer: This layer collects and transmits the information according to the way the neural network has been designed. The output layers pattern is directly traced back to the input layer. To determine the no. of neurons in this layer, it should be directly related to the type of work that the neural network is performing.
5.3.2. Convolution Neural Network. A Convolutional Neural Network (ConvNet/CNN) \cite{7} is a Deep Learning algorithm which takes an image as an input and assigns weights to multiple objects in an image and also differentiates those aspects from one another. It successfully captures the Spatial and Temporal dependencies in an image with the help of relevant filters \cite{7}. The ConvNet reduces the images into such a form which is easier to process and thereby preventing loss of any features that are vital for getting a good accuracy and prediction.

- **Convolution Layer**: It is the Kernel layer which extracts the high-level features from the input image. ConvNets are not limited to only one Convolutional Layer. The first layer captures the Low-Level features such as edges, colour, gradient orientation, etc \cite{7}. As more layers are added, the architecture adapts to features of High-level thereby providing a network which has a complete understanding of images in the dataset.

- **Pooling Layer**: The Pooling Layer works similar to the Convolutional Layer and is responsible for reducing the spatial size of the Convoluted Feature as well as the computational power which is required to process the data through dimensionality reduction \cite{7}.

- **Fully Connected Layer (FC Layer)**: In this layer, the matrix is flattened into vectors and is provided to a fully connected layer.

![Figure 7. ANN Layer Architecture.](image)
5.4. Pytesseract
Python-tesseract is an OCR tool for python which recognizes as well as reads the text embedded in images. Python-tesseract is a wrapper for Google’s Tesseract-OCR Engine. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

6. Results
Dataset is trained with both ANN and CNN models to predict the handwritten characters. Both the models have given desirable results.

6.1. ANN Evaluation
ANN model is built with an input layer followed by a dropout layer, hidden layer with Relu as an activation function, and dropout layer, and again a hidden layer with Softmax as an activation function. Finally, the output layer which is built with Adam optimisation algorithm, Sparse Categorical Cross Entropy as loss function and metrics used is Accuracy which is used for determination of correctness of prediction. The figure 9 represents the graph epoch versus the training and validation accuracy. It can be seen that validation accuracy has the peak at 12th epoch and training accuracy has the peak at 27th epoch. The overall accuracy of the model is 85.56%.

Figure 8. CNN Layer Architecture

Figure 9. Epochs vs Accuracy (ANN)
6.2. **CNN Evaluation**

CNN model is built with input layer of convolution 2D layer with kernel size of 3x3 and Relu as an activation function and input dimension of 28x28 pixels followed by Max pooling layer with pooling size of 2x2 and strides of 2, Convolution 2D layer with kernel size of 3x3 and Relu as an activation function, Max pooling layer with pooling size of 1x1 and strides of 1, dropout layer with threshold probability of 0.5 and finally a flattening layer. This complete CNN is connected with two dense layers of ANN with activation functions of Relu and Softmax respectively. Adam optimization algorithm is used for optimization, Sparse Categorical Cross Entropy as loss function and metric for validation used is spares categorical accuracy. In figure 10 which represents the graph epoch versus the training and validation accuracy. It can be seen that validation accuracy has the peak at 12th epoch and training accuracy has the peak at 27th epoch. The overall accuracy of the model is 83.36%.

![Figure 10. Epochs vs Accuracy (CNN)](image)

6.3. **Prediction on the form**

As ANN has a better prediction of 85.56% it is used for extraction of information from the form. After the processing of the image of the form and finding the contours each large box is broken down into the smaller boxes of size 28x28 pixels which is given to the model as the input and the prediction is done of individual characters.
6.4. *Pytesseract prediction*

This python library is used for detecting printed text from the form that is to extract the text of the entity. This text is stored as the string in the list.
**Roll No.**

**First Name:**

**Middle Name:**

**Last Name:**

**D.O.B.:**

**Age:**

**Adhar No.:**

**Mob No.:**

**Address:**

**Pin code:**

**City:**

**PAN No.**

![Output of Pytesseract](image)

Figure 12. Output of Pytesseract

7. **Conclusion**

The project has successfully identified the written text in the boxes of the form using ANN. Each character is predicted, and a string is formed of particular value which is stored in the list form. The research also has a key point in not predicting the empty box that is the box which is empty with no character written in it. The output of the Pytesseract is then merged with its corresponding values of the ANN prediction and corresponding list is formed which contains the pair of entities and its value. Which can be saved to the database directly without manual work.

**References**

[1] OpenCV Documentation(https://docs.opencv.org/master/d1/dfb/intro.html).

[2] Rohan Vaidya, Darshan Trivedi, Sagar Satra, Prof. Mrunalini Pimpale. Handwritten Character Recognition Using Deep Learning, Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018).

[3] Yuncong Lu. Handwritten Capital Letter Recognition based on OpenCV, MATEC Web of Conferences 277, 02030 (2019), JCMME 2018.

[4] Abhishek Jindal, Mohd Amir, Automatic Classification of Handwritten and Printed Text in ICR Boxes 2014 IEEE International Advance Computing Conference (IACC) 978-1-4799-2572-8/14/$31.00 c 2014 IEEE (https://doi.org/10.1051/matecconf/201927702030).

[5] Gregory Cohen, Saeed Afshar, Jonathan Tapson, André van Schaik.EMNIST: an extension of MNIST to handwritten letters , arXiv:1702.05373 ,[cs.CV].
[6] Artificial Neuron Networks (Basics) | Introduction to Neural Networks | by Shubh Saxena.
[7] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.
[8] Neha Agrawal, Arashdeep Kaur, An Algorithmic Approach for Text Recognition from Printed/Typed Text Images 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) 978-1-5386-1719-9/18/$31.00 © 2018 IEEE.