International Journal of Advanced Science and Research
ISSN: 2455-4227; Impact Factor: RJIF 5.12
Received: 02-06-2020; Accepted: 19-06-2020; Published: 27-06-2020
www.allsciencejournal.com
Volume 5; Issue 3; 2020; Page No. 10-14

Automation of data visualization from documents using deep learning

Madhumathi S[1], Gomathi R[2], Sathish Kumar G[3]
[1] Post Graduate in Computer Science and Engineering, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India
[2] Associate Professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India
[3] Assistant Professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

Abstract
Text Detection API can detect text in a variety of documents including financial reports, medical records and tax forms. The project is to automate the data extraction process by making it easy to add documents text detection and analysis to applications. It is based on highly scalable, deep learning technology to analyze it on a daily basis. The project makes powerful and accurate analysis available with a simple API. It extracts data quickly from millions of documents which can accelerate the decision making. For unstructured data, automation of data visualization from documents using deep learning helps to produce structured data for the Document Analysis API to extract text, forms and tables.

Keywords: text extraction, optical character recognition, binary threshold

1. Introduction
There are a lot of processes in industries that use manual work which is time and power consuming. Some of these works can be automated. One such work is explained and ways to automate is defined with descriptions. There are usually people who are hired to take a look at all the documents and reports that a company receives and manually take only the required data and save it in the company's database for further needs [1].
This takes a lot of time and manual power. There can also be same documents sent by different people and it can be saved again and again which is a waste of space [2].
Thus, the proposed system is to automate this method and to eliminate duplicate documents. The document is first pre-processed i.e checked for any tables that are present in it [3].
Most of the time tables contain the required information so in the proposed system they are detected. Then, the texts are highlighted using binary threshold [4].
The proposed system using Optical Character Recognition is to identify the texts in the tables. Optical Character Recognition **(**OCR**)** or sometimes simply text recognition is called the method of extraction of text from images. It is then stored in a CSV file format which is in an editable form and can be reviewed when needed later on. These files are compared for the similarity between the using Natural Language Processing in python. This way the similar documents can be eliminated. This method is implemented using AWS where the recognized texts will automatically save in the database in the format (.csv) in an editable form [5].

2. Literature Survey
Text detection deals with finding text area from input image, whereas recognition deals with converting obtained text into characters and words. Methods used for this purpose are categorized as stepwise methods and integrated methods. Stepwise methods have the separate stages of detection and recognition and they proceed with detection, classification, segmentation and recognition. Integrated methods have information sharing amongst detection and recognition stages and these methods aim at recognizing words from text available [6].
The studies various stages in process of text detection and recognition and analyses and compares different approaches used to undergo these stages in the experiment. It presents importance of every processing stage and advantages, disadvantages and applications of approaches used by various contributors to solve these problems. Various applications of text detection and recognition are also reviewed in this paper [7].
A four-stage method: (i) binarization based on local thresholding, (ii) tentative character component detection using gray-level difference, (iii) character recognition for calculation of similarities between the character candidates and the standard patterns stored in a database, and (iv) relaxation operation to update the similarities. They were able to extract and recognize characters, including multi-segment characters, under varying illuminating conditions, sizes, positions, and the fonts when dealing with scene text images, such as freight train, signboard, etc. However, binary segmentation is inappropriate for video documents, which can have several objects with different gray levels and high levels of noise and variations in illumination are considered. Furthermore, this approach places several restrictions related to text alignment, such as upright and not connected, as well as the color of the text (monochrome). Based on experiments involving 100jimages, their recall rate of text localization was 85.4% and the character recognition rate was 66.5% [8].
An application of CC-based method to the detection and

recognition of text on cargo containers, which can have uneven lighting conditions and characters with different sizes and shapes. Edge information is used for a coarse search prior to the CC generation. The difference between adjacent pixels is used for the determination the boundaries of potential characters after quantizing an input image. Local threshold values are then selected for each text candidate, based on the pixels on the boundaries. These potential characters are used to generate CCs with the same grey-level. Thereafter, several heuristics are used to filter out non-text components based on aspect ratio, contrast histogram, and run-length measurement. Despite their claims that the method could be effectively used in other domains, experimental results were only presented for cargo container images [9].

Another CC-based method, which uses colour reduction. They quantize the colour space using the peaks in a colour histogram in the RGB colour space. This is based on the assumption of that the text regions cluster together in this colour space and occupy a significant portion of an image. Each text component goes through a filtering stage using a number of heuristics, such as area, diameter, and spatial alignment as mentioned. The performance of this system was evaluated using CD images and book cover images [10].

Segments of an image using colour clustering in a colour histogram in the RGB space. Non-text components, such as long horizontal lines and image boundaries are eliminated. Then, horizontal text lines and text segments are extracted based on an iterative projection profile analysis. In the post-processing stage, these text segments are merged based on heuristics. Since several threshold values need to be determined empirically, this approach is not suitable as a general-purpose text localizer. Experiments were performed with 50 video images, including various character sizes and styles, and a localization rate of 87% was reported [11].

Cluster-based templates for filtering out non-character components for multi-segment characters to alleviate the difficulty in defining heuristics for filtering out non-text components. A similar approach was also reported on Cluster-based templates are used along with geometrical information, such as size, area, and alignment. They are constructed using a K- means clustering algorithm from actual text images [12].

The homogeneity of intensity of text regions in images. Pixels with similar grey levels are merged into a group. After removing significantly large regions by regarding them as background, text regions are sharpened by performing a region boundary analysis based on the grey level contrast. The candidate regions are then subjected to verification using size, area, fill factor, and contrast. Neighboring text regions are examined to extract any text strings. The average processing time was 1.7 seconds per frame on a 133 MHz Pentium processor and the miss rate ranged from 0.29% to 2.68% depending on the video stream [13].

3. Proposed Methodology

In this system, Automation of data visualization from documents using deep learning is a serviced that automatically detects and extracts data from scanned documents. First it detects the presence of all the tables in the document of image file. After the detection of tables, then it extracts the data that is present within the tables. Simple Optical Character Recognition (OCR) is used to identify the contents of fields in forms and information stored in tables. Without the use of any manual intervention, detection of key-value pairs in document images automatically so that the inherent context of the document can be retained. Usually the retrieved data cannot be stored but in this proposed system the data will be stored in an editable form (for eg: csv file) however it was in the table in the uploaded file. It also compares documents to check if they are duplicated so the same files need not be saved again which is space and time consuming.

3.1 Methodology

Connectivity is made by the implementation of process. It is done in a few steps to develop and to deploy the application to get the necessary output. They are given below.

3.1.1 Develop

Install the CLI
Setup the application services
Integrate with your App

3.1.2 Deploy

Connect your repository
Deploy the application
Process

3.1.3 Develop - Process

Install the CLI: First, everything needed must be installed in our system. The CLI is an important thing that must be installing using the commands in the Command Prompt. The latest python version must also be installed. The AWS version can be checked to confirm the installation process successfully.

Setup the application services: The application services should be setup. There should be the amplify API application. All the services that are necessary for the Automation of data visualization from documents using deep learning must be installed.

Integrate with your app: There must be connectivity between the Amazon AWS and the system. This connectivity is made possible by the coordinates, private key and the secret key.

3.2 Deploy

Connect your repository: The keys are created in our account in the AWS management console. These keys are then used to connect our system and the console using the passwords. So the console and the system to deploy will be connected.

Deploy the application: The application will be deployed by the program in python and the commands to get the input in the command prompt.

Process: As soon as the input image is given, the application will first detect any tables present in the image using the techniques. Then when the tables have been detected, the texts will be got using the OCR and then it will be saved in the format csv which is editable.

3.3 Binary Threshold

The simplest methods of binding convert each pixel into an image with a black pixel if the image size is less than a fixed T, or a white pixel if the image's size is larger than the default. In the example image on the right, this result in a black tree gives a complete black and white snow which

becomes completely white. It is a method in OpenCV in which a pixel values share with respect to a given limit value. By counting, each pixel value comparison takes place and that is compared to the numerical value. If the pixel value is smaller than the limit, it 0 can be set; otherwise, maximum value can be set (usually 255). Bone casting is a very popular technique for dividing limbs, used to classify something that is considered to be front and back. The threshold is a sum of two regions on both sides that is below the limit or beyond the limit. In Computer Vision, Sensitive images are used for the performance of the method of divination. The conversion of image to the color space happens in the beginning

3.4 Table Detection

Table detection is a very important step in many text analysis applications as the use of tables to present important information to the reader in an orderly manner. It's a difficult problem due to the varied entries and coding of tables. Investigators propose several strategies for table detection based on document analysis. Most of these techniques fail to perform in general because they are based on mechanical engineering features that are not capable of structural change. The proposed method applies high accuracy to document images with various properties including documents, research papers and magazines.

Table identification of a particular method where parallel or vertical text and lines are applied. Hierarchical characterization of physical particles of model-based top-down approach for table analysis happens. The table area can be characterized to horizontal lines, columns, horizontal space and vertical space. Label methodology of individual cells by elementary cell characterization is performed in order to properly collect the underlying niche or overlapping cells of the logical units. These raw labels match the table model, which means that the table can be extracted with relational information. Text areas are identified using the bottom-up approach and the detected characters are classified into words and then into phrases. Depending on the threshold on the horizontal and vertical run length, the lines are obtained. In the same area, text blocks are compared to the alignment of these identified lines. Furthermore, complete understanding the structure of the table can be done using horizontal and vertical projection profiles, the algorithm attempts to add missing horizontal and vertical columns. Different reports, documents or images might have tables which are in different formats. So, the method to detect tables has to be very accurate. If training data is used then we need to give a lot of input data in order to identify all the tables in a specified document with full accuracy where it also takes a lot of time to check and process the input documents. A new table format of a trail method is followed then it will not be detected so it has to be customed manually. As there are these issues while using datasets, this method of using the horizontal lines, vertical lines and the spaces seems to be much more accurate and faster.

3.5 Text Extraction

Text extraction makes it easy to collect data from documents and forms quickly and accurately. It automatically detects the layout of the document and key elements on the page, understands the data relationships in any embedded forms or tables, and captures everything intact with its context. The data you collect can be instantly used in the app or stored in the database. Automation of data visualization from documents using deep learning's pre-trained machine learning models do not need to write code for data extraction. Because models have already been trained on millions of documents from many industries, including invoices, receipts, contracts, tax documents, sales orders, registration forms, benefit applications, insurance claims and policy documents. There is no longer the need to maintain code for every document or form you receive, or worry about how the page layouts will change over time. Textrack preserves the composition of data stored in tables during extraction. This can be helpful for documents with structured data such as financial statements or medical records, which contain column names in the top row of the table and then rows of individual entries. All data collected is returned with bounding box coordinates. Coordinates make up a polygon frame that contains every single identifiable data, such as a single word, a line, or a table. It helps to audit where a word or number in the source document comes from. This helps guide the user in document search systems that return scans of the original documents as a result of the search.

The extraction of texts from identified tables is done using the Optical Character Recognition (OCR). These are the two basic ways that can generate a rank list of candidate characters. Matrix Matching Comparing an image to a glyph stored on a pixel-by-pixel basis; it is also called "pattern matching", "pattern recognition" or "image correlation". This depends on the input glyph being separated from the rest of the image and the stored glyph in the same font and size. This technique works best with typewritten text and does not work well with new fonts. This is a technique that early physical photocell-based OCR implemented rather than directly. Feature extraction glyphs are decomposed into "properties" such as lines, closed loops, line direction and line intersections. The extraction features reduce the size of the representation and make the detection process computationally efficient. These features are compared to an abstract vector-like representation of a character, which can be reduced to one or more glyph prototypes. The most common methods of feature detection in computer vision apply to this type of OCR, which is commonly found in "Intelligent" handwriting recognition and most modern OCR software. The nearest neighbour classifications, such as the k-nearest neighbour algorithm, are used to compare the image properties with the stored glyph properties and to select the closest match. Automation of data visualization from documents using deep learning uses neural networks that are trained to identify entire lines of text without focusing on single letters.

Deep learning for OCR is Deep Neural Networks (DNN), a learning technique for learning. This popularity is mainly achieved by both the DNN identifying text region and the letters simultaneously. Deep neural networks or convolutional neural networks (CNN) are essentially multi-layered learning and feature processing neural networks. Each neuron (node) in each layer is fed with information sent from the nodes connected to it. The processing mechanism (transfer function) then determines the extent to which the processed information is sent to the currently connected nodes. The structure of the network, that is, the way neurons and membranes are connected, plays a fundamental role. The role of determination in DNNs is that

they can make the structure different. Similar to the human visual system, different neurons and processing layers are more sensitive to the different properties of objects. The edges of objects are more sharply represented by a set of neurons, while others are more sensitive to colour gradients. Researchers exploit this diversity to build sophisticated structures in which neurons and membranes are propagated back and forth between them before producing a result.

3.6 Saving as a CSV File

After the texts are recognized from the tables, it is then saved in the CSV file format. CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or Open Office Calc.CSV stands for "comma-separated values". Its data fields are most often separated, or delimited, by a comma. It is a common format for data interchange as it is simple, compact and ubiquitous. It will open into Excel with a double click and nearly all databases have a tool to allow import from CSV. It is also readily parseable with simple code. When the required values are extracted and stored in this format it is like tables and is also in a editable form so it can be reviewed when needed and is much more advantageous.

4. Results and Discussion

4.1 Comparison to Find the Similarity Between Documents

The measure of similarity can be qualitative or quantitative. In the proposed system it is done using natural language processing. Qualitatively, it is evaluated against subjective criteria such as theme, sentiment, and overall meaning. The numerical parameters such as quantitative, document length, number of keywords, common words, etc. are compared. This process takes place in two stages. Vectorization involves converting documents into a vector of numbers. The following are some popular numbers (measurements): TF (Term Frequency), IDF (Inverse Document Frequency) and TF * IDF. Calculating the cosine similarity between a document vectors is a distance calculation. As we know, the cosine of the same vectors (dot product) is 1, asymmetric / perpendicular to 0, so the dot product of two vector-documents is some value between 0 and 1, which is a measure of the similarity between them.

Documents must be pre-processed first. There are many things that are not core to the text analysis exercise, such as finding similarity. So, they pre-process their words by turning them lowercase and eliminating stop words such as the why, the must. Each text should then be classified as a vector. Each verse has some common and some unusual words compared to each other. To calculate all possibilities, a word set containing words from both documents is formed. Then calculate the number of times a word has occurred using the frequency count method. Then one has to find out the significance of each word. Once the words in the text are vectorized, the similarity score between them is nothing but the distance between them. Thus the similarity between documents can be calculated.

The existing system uses Tensor flow where the functions are predefined and the object identification is made. The existing methodology wants many manual work to be done when the excel sheet is created and the data is to be loaded in it for the data extraction and the purpose of that particular data in future. The proposed system is about Optical Character Recognition data in the table on a document can be extracted with the process of automatic checking and the manual work of checking is reduced where the man power is also reduced. Artificial Intelligence plays a vital role in training the data and the tables can be identified separately and extraction is done. Separate identification of table and extraction are also possible in the proposed methodology where the existing fails to do for the separate table.

Fig 1: Before binary threshold

Fig 2: After binary threshold

Thus when a document or an image file is being uploaded it will be first converted like this so that there will not be any errors while the table is being detected or none of the tables will be skipped due to any background texts or colors that are present in the file.

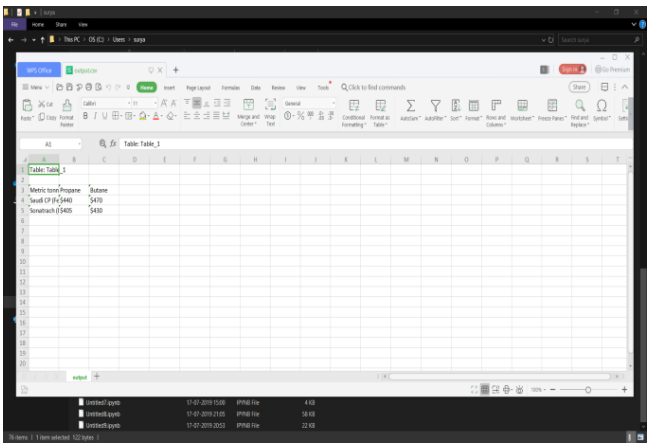

Fig 3: Text Extraction

5. Conclusion

Many industries hire people to go through all the documents and reports that are being sent in different formats from different organizations. Then they manually read and store the required data in tables for future use. This consumes a

lot of time and manual power. The proposed system automates this particular process. It takes the document or file as input, detects the tables, extracts the data and automatically stores it. The tables can be present in different formats which can all be detected using this technique. The proposed system uses the extraction of texts from images by using OCR from AWS which is more efficient and quicker. It eliminates similar documents using Natural Language Processing in python which saves space. The proposed system can become a very important automation tool as we do not need any manual labor in searching and entering the data and hence use this application.

6. References

1. Jain AK, Yu B. Automatic Text Location in Images and Video Frames," Pattern Recognition. 1998; 31(12):2055-2076.
2. Yao C, Zhang X, Bai X, Liu W, Ma Y, Tu Z, *et al*. Detecting Texts of Arbitrary Orientations in Natural Images," in Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2012, 1083-1090.
3. Koo H, Kim DH. Scene Text Detection via Connected Component Clustering and Non-text Filtering," IEEE Trans. Image Processing, 2013, 22(6).
4. Lee J, Lee P, Lee S, Yuille A, Koch C. AdaBoost for Text Detection in Natural Scene," in Proc. IEEE Int'l Conf. Document Analysis and Recognition, 2011, 429-434.
5. Elagouni K, Garcia C, Sbillot P. A Comprehensive Neural-Based Approach for Text Recognition in Videos using Natural Language Processing," in Proc. ACM Conf. Multimedia Retrieval, 2011.
6. Wang K, Belongie S. Word Spotting in the Wild", in Proc. European Conference on Computer Vision, 2010, 591- 604.
7. Neumann L, Matas J. On Combining Multiple Segmentations in Scene Text Recognition," in Proc. IEEE Int'l Conf. Document Analysis and Recognition, 2013, 523- 527.
8. Shivakumara P, Phan TQ, Tan CL. A Laplacian Approach to Multi- Oriented Text Detection in Video," IEEE Trans. Pattern Analysis and Machine Intelligence. 2011; 33(2):412-419.
9. Qixiang Ye, David Doermann. "Text Detection and Recognition in Imagery: A Survey" IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.
10. Hanif SM, Prevost L, Negri PA. A Cascade Detector for Text Detection in Natural Scene Images," in Proc. IEEE Int'l Conf. Pattern Recognition, 2008, 1-4.
11. Keivan Kianmehr, Reda Alhajj. Effectiveness of Machine Learning Techniques for Automated Identification of Calling Communities, 12th International Conference Information Visualisation, 2008.
12. Xiaodong Huang, Kehua Liu, Lishang Zhu. Auto scene text detection based on edge and color features in IEEE, 2012 International Conference on Systems and Informatics (ICSAI), 2012.
13. Palaiahnakote Shivakumara, Trung Quy Phan, Chew Lim Tan A Robust Wavelet Transform Based Technique for Video Text Detection, 10th International Conference on Document Analysis and Recognition, 2009.