



OPTICAL CHARACTER RECOGNITION FOR ELECTRONIC INVOICES USING AWS SERVICES

Sameer. M. Patel

Student, B. Tech, Department of Information
Technology

K. J. Somaiya College of Engineering
Mumbai, Maharashtra, India

Sarvesh. S. Pai

Student, B. Tech, Department of Information
Technology

K. J. Somaiya College of Engineering
Mumbai, Maharashtra, India

Mittal. B. Jain

Student, B. Tech, Department of Information
Technology

K. J. Somaiya College of Engineering
Mumbai, Maharashtra, India

Vaibhav. P. Vasani

Professor, Department of Computer Science

K. J. Somaiya College of Engineering
Mumbai, Maharashtra, India

Abstract - Optical Character Recognition is basically the mechanical or electronic conversion of printed or handwritten text into machine understandable text. The complication of Optical Character Recognition in different conditions remains as relevant as it was in the past few years. At the present time of automation and innovations, Keyboarding remains the most common way of inputting or feeding data into computers. This is probably the most time consuming and labor-intensive operation in the industry. Automating the process of recognition of documents, credit cards, electronic invoices, and license plates of cars – all of this could help in saving time for analyzing and processing data. With the increased research and development of machine learning, the quality of text recognition is continuously growing better. Our paper is focused on providing a brief explanation of the different stages involved in the process of optical character recognition and through the proposed application; we aim to automate the process of extraction of important texts from electronic invoices. The main goal of the project is to develop a real time OCR web application with a micro service architecture, which would help in extracting necessary information from an invoice.

Keywords - Optical Character Recognition, Electronic Invoices, AWS Textract, AWS Lambda Functions, AWS Gateway;

I. INTRODUCTION

Machine oriented replication of human activities or functions, like reading, has been a dream. However, over the last few decades, machine reading has become a reality [1].

Optical Character Recognition has become one of the most promising applications of technology in the field of artificial intelligence and pattern recognition, which is used

to convert handwritten characters, letters or words into a digital format. It is more of a common and recognized method of extracting information and digitizing printed texts through which we are able to electronically edit, search, and store information more compactly and efficiently.

The process of Optical Character Recognition consists of various stages, which include Pre-processing, Classification, Segmentation, Detection, and Feature Extraction [2]. With proper research and by making efficient changes in various techniques used in each stage of the process, we can increase the accuracy of the algorithm.

This paper gives a brief overview of the various optical recognition techniques used in the process of Optical Character Recognition. Following it, the paper focuses on explaining the proposed system, which has a micro service architecture based on AWS, and explaining the different features of the proposed system. The proposed system is based on Python, NodeJS, ReactJS, and AWS micro services.

II. LITERATURE REVIEW

Lots of Research and work has been proposed in the field of Optical Character Recognition. Accuracy in the algorithm has always been the major factor when it comes to calculating the performance of the algorithm. This section focuses on the research and different techniques that people have done in the field of Optical Character Recognition.

It was only until the 20th Century, when a couple of patents were awarded for OCR devices, where one of them was patented to Gustav Tauschek in Germany in 1929 and another to Paul Handel in the United States in 1933 [3]. These ideas never materialized into actual machines at that time due to technological limitations, but they formed the basis of many OCR systems to come.

It was the year 1949 when Radio Corporation of America eventually started working on the computer-based OCR to

help blind people but as a matter of fact; their machine was too expensive and was not able to achieve the desired accuracy [3]. M. Sheppard, in 1951 played a vital role in the foundation of many modern OCR systems and named it GISMO. GISMO had the power of performing various functions as it was able to read all the musical symbols on a page, but it had its own limitations as well [3].

It was very clear that a standardized OCR font was required due to the problems and complexity faced during the development of an OCR system. This led to the formation of OCRA and OCRB along with the collaboration of ANSI and EMCA, which helped in improving the efficiency rate of an OCR. In 1978, Kurzweil Computer Products eventually started selling OCR as a commercial product, where LexisNexus were one of their first customers who brought the program to upload legal documents and papers onto its online databases. After two years, Kurzweil sold his company to Xerox, which had an interest in further improving and commercializing the paper to computer text conversion [8].

In recent years, despite having a massive advancement and improvement in the accuracy of Optical character recognition, the ability to understand contents by an OCR Algorithm is far below that of humans. Optical Character Recognition has witnessed tremendous growth and research and it is estimated that this research will reach new heights in the coming years [4]. Therefore, it was the year 2019, when Amazon announced the general availability of Amazon Textract, which is a fully managed AWS service that uses machine learning to automatically extract information and insights from documents. AWS Textract was way beyond the normal and simple Optical Character Recognition to identify texts and contents from tables and forms in a document. This service also supports different multiple image formats like scans, PDFs, and photos [7].

scanner may be used, it is necessary to select a scanner with a great sensing tool and transport mechanism.

B. Image Pre-Processing

This phase basically refers to the process of improving the quality of the images. This can be done either by sharpening the image pixels so that all the information in the image is clearly visible. This is one of the most important phases in the process of Optical Character Recognition, which will help in increasing the accuracy of our application. The different techniques involved in improving the quality of the images include Noise Removal, Skew Removal, Thinning, and different morphological operations.

C. Image Segmentation

This phase basically helps in extracting the single characters from an image, which is basically then sent to the recognition system. Advanced character segmentation techniques are required when the image has noise in the background. It is used to separate image into its constituent characters. Segmentation can be divided into Implicit Segmentation and Explicit Segmentation.

D. Feature Extraction

This phase basically helps in the extraction of characters based on their features, which are extracted from the high-quality images and are then observed with the help of inter-class variations and only those characters are selected which are efficiently computable. Different geometrical features like corner points, loops and curves are extracted from the given image.

E. Character Classification

This phase basically helps in the arrangement of the segmented characters into different classes and categories. Based on the result, they can be divided into two categories:

1. Structural Pattern Classification:
 - a. Based on the structure of the image, the features are classified.
2. Statistical Pattern Classification:
 - a. Based on the probabilistic models, the features are classified.

F. Image Post-Processing

This phase basically helps in the post processing of the result extracted from the above steps. After performing the above steps, the result obtained is not accurate in nature. Hence, we need to apply certain techniques like Natural Language Processing to remove errors from the result. This step helps in improving the overall accuracy of the results obtained or produced by the OCR engine.

Different methods or approaches like Contextual approaches, dictionary-based approaches or multiple classifiers are used to get better and accurate results.

III. STEPS INVOLVED IN OCR

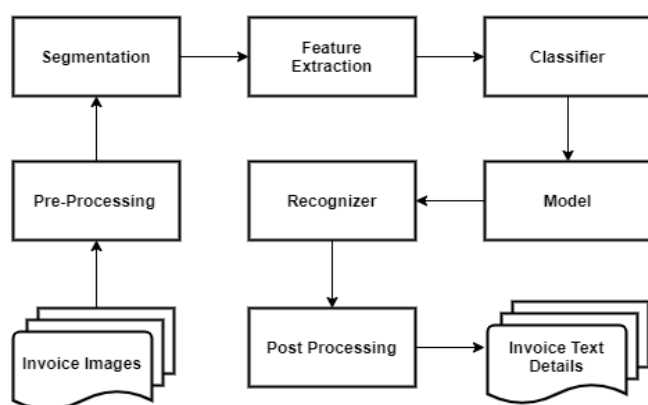


Fig. 1. Steps Involved in OCR

A. Image Acquisition

This phase basically refers to the process of collection of images for conversion to printed format from different sources. In order to capture an image, a digital camera or a

IV. METHODOLOGY

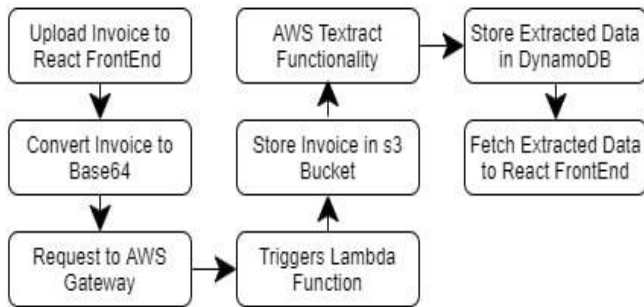


Fig. 2. System Flow

Our Companies and organizations across different industries have a huge number of physical invoices that need to be processed. It is very difficult to extract necessary information from a scanned document when it contains forms, paragraphs, tables etc.

Companies and organizations have been addressing these problems with manual keyboarding or custom designed OCR technology, which eventually takes a lot of work force and time. This process requires templates for extraction and custom workflows. Even after extracting the text or content from a document, companies want to extract meaningful and necessary insights from these invoices for their end users. This leads to building a complex NLP model because training such a model would require large amounts of training data and resources.

This paper proposes a real time OCR Web application for extracting necessary information from electronic invoices using AWS Microservices. The various steps involved in the process is:

1. The user can upload an Invoice or a bill from which he/she needs to extract information in the form of PNGs/JPEGs.
2. The Invoice is then converted into Base64 format before it gets pushed into the S3 Bucket for further processing.
3. Then the functionality of AWS Textract service extracts information from the Invoice in the form of a JSON object.
4. Then using Regular Expressions used in the Lambda Function, we extract only the useful information and text from the JSON Object and send it back to the server.
5. Then using this information, the React server automatically fills up the invoice using the information, which was extracted from the uploaded Invoice.

V. IMPLEMENTATION

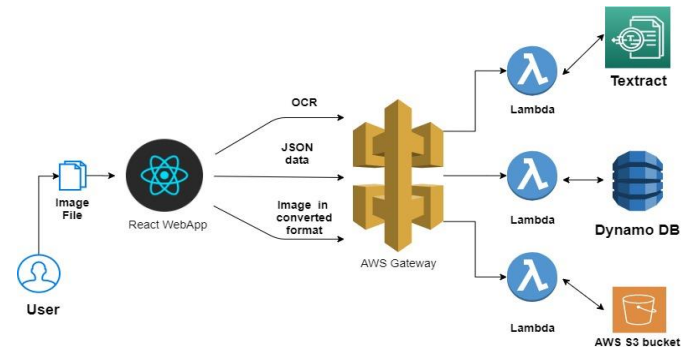


Fig. 3. System Architecture

This system, which is basically a real time OCR based React application, will be issuing three API calls which are going to be asynchronous in nature. The first API call will basically send our invoice image in base64 format to a restful API, and then the webapp server will asynchronously make a call to the OCR engine, which is basically the AWS Textract services. The third asynchronous API call will make sure that we send data at the end to AWS DynamoDB.

The architecture also contains an AWS API Gateway, which is basically an entrance, or a gateway through which all the requests and calls will be sent, and the gateway is going to be based on a restful API. The gateway will be connected to different Lambda functions, which will process the corresponding request from the client, so the AWS Gateway is basically an AWS Managed Service, which will basically automate common activities, such as patch management, monitoring, security, change requests, backup services, and also provide full-lifecycle services to provision, run, and also support our infrastructure.

Amazon S3 will be playing a big role in serving our application. On the backend side, the entry point of all the different requests will be the API Gateway, so it will be the API Gateway's responsibility.

Now, based on the URL that the client hits, it will redirect the client's request to one of the micro services, which will trigger a lambda function. Now the first lambda function is based on NodeJS, which will basically be an image uploader and will be used to upload invoices to Amazon S3. Our project will also be using NodeJS to insert and read data from DynamoDB. The second lambda function, which will be used to interact with the DynamoDB, is based on Python and will be in conjunction with AWS Textract to build the Optical Character Recognition functionality in our application.

VI. RESULTS AND DISCUSSIONS

A sample of invoices comprising of various types of bills was selected and uploaded into the S3 bucket, which gave the AWS Textract access to the sample of invoices. The AWS Textract Model then performs various object detection operations to create bounding boxes around the segmented text present in the invoice.

The resultant segmented object's details are then received back in the form of a JSON object, which comprises the extracted text and its confidence value. The confidence score

assigned by the OCR engine for each field answer had a value of over 98% for various sample invoices.

With the help of regular expressions, we extract only the necessary information that is required from the JSON object. This information is then used to automatically fill the billing form. The confidence scores assigned for each of the test images of all alphabets, numbers and symbols has been graphically represented in Fig.8., Fig.9., Fig.10., Fig.11., Fig.12.

The comparison of the confidence scores was done with respect to other models like PaddleOCR [9] and EasyOCR [10]. It was found out that AWS Textract was giving a consistently high confidence score ranging above 95% whereas other models showed comparatively less confidence while recognizing certain characters. Furthermore, AWS Textract showed promising results in recognizing symbols as compared to the other two models.

```

OCRBody ▾ Object ⓘ
  ▾ body: Array(3)
    0: "$1004.6"
    1: "INV2028"
    2: "20/12/2020"
    length: 3
    ▶ __proto__: Array(0)
  statusCode: 200
  ▶ __proto__: Object
    
```

Fig. 6. Values Extracted from the Invoice

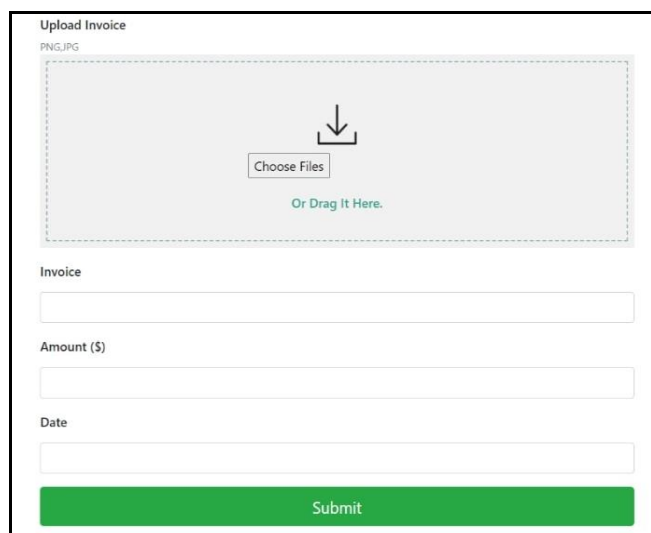


Fig. 4. React Interface

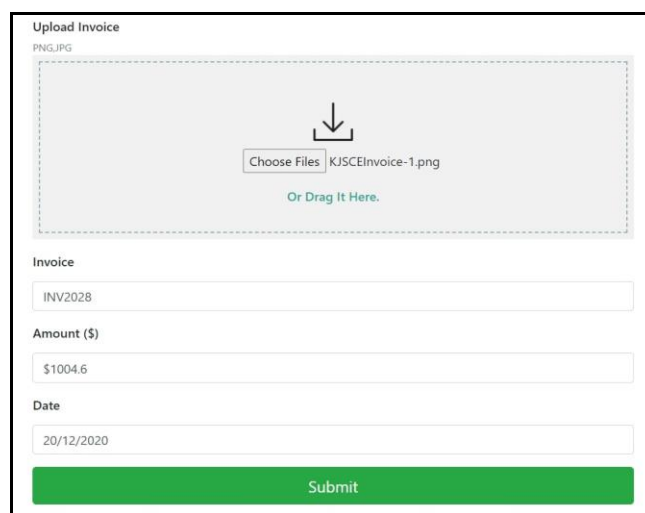


Fig. 7. Recognized values displayed after extraction

KJSCE		Invoice			
		Date	20/12/2020		
		Invoice #	INV2028		
		Acct. No.	SM1008		
Bill To	Ms. Jain	Ship To	Mr. Patel, Jeddah		
Item Name	Quantity	Units	Description	Unit Price	Amount
Paints	3		Asian Paints Color	200	600
Bulb	2		Phillips 7W	180.80	361.60
Subtotal					961.60
Shipping					25.00
GST					18.00
Total					1004.6
Amount Due					\$1004.6

Fig. 5. Feeding Invoice to the System

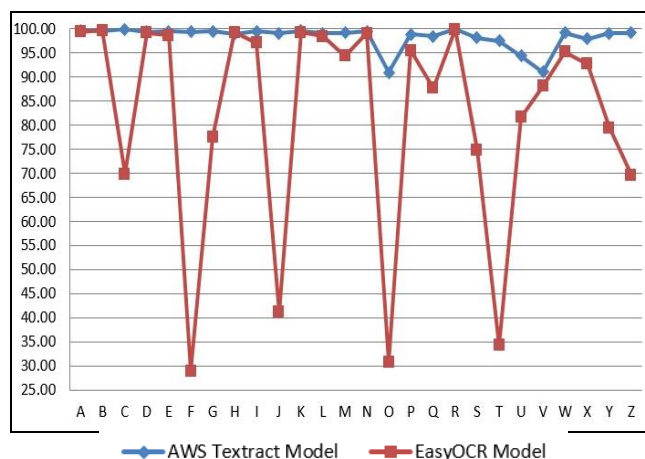


Fig. 8. Comparison of Confidence Scores of Alphabets

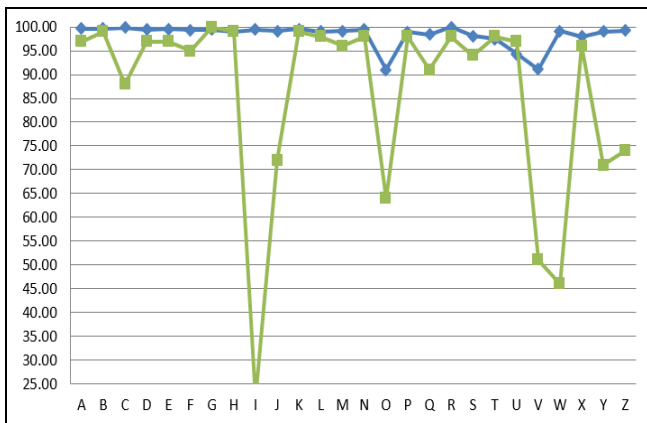


Fig. 9. Comparison of Confidence Scores of Alphabets

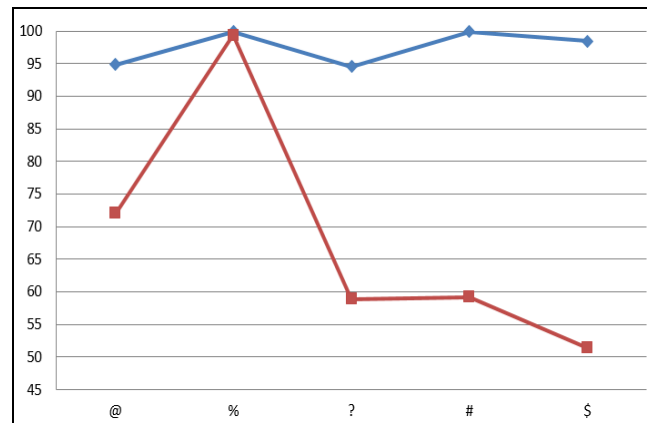


Fig. 12. Comparison of Confidence Scores of Symbols

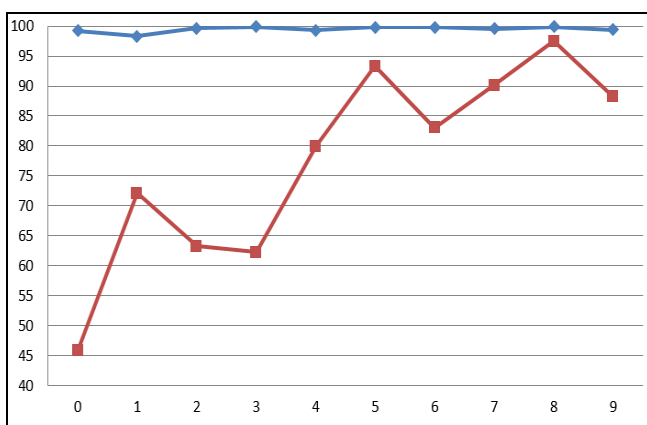


Fig. 10. Comparison of Confidence Scores of Numbers

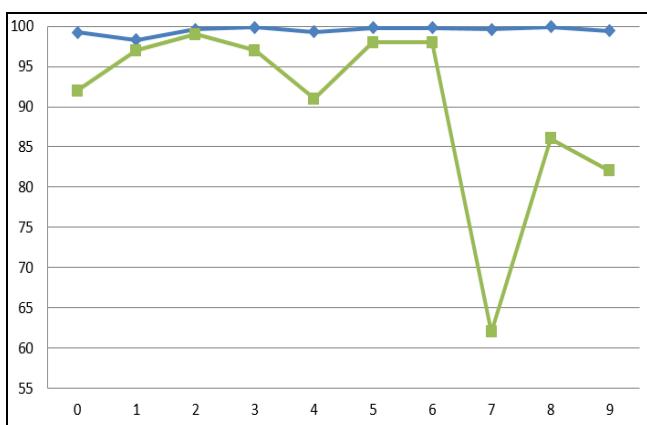


Fig. 11. Comparison of Confidence Scores of Numbers

VII. CONCLUSION AND FUTURE SCOPE

This paper not only gives a brief overview of all the different techniques involved in the process of Optical Character Recognition but the main aim of this paper is to showcase a fully-fledged real time OCR web application for Invoices. The recognition accuracy of the prototype implemented was found to be quite promising. In fact, during the implementation of this prototype, it was very clear that the pre-processing of the Invoices was a very crucial factor in increasing the accuracy of the model. We had to make sure that all images were properly aligned before they were fed into the system. It was found that the OCR engine did not recognize images with vertically aligned texts whereas those with horizontal alignment were recognized accurately. In addition, characters below the height of 15 pixels remained undetected.

The future scope of this project would be to allow the user to directly capture the invoices from his/her device camera for recognition. The future version will also include automated pre-processing of the invoices before they are sent to the OCR engine. Currently the system only allows PNGs and JPEGs as the accepted input format, but we aim to include PDF format as well. To improve accessibility, we plan to roll out a mobile app version that will incorporate all these features.

VIII. REFERENCES

- [1] Shyam G. Dafe, Shubham S. Chavhan (2018). Optical Character Recognition Using Image Processing. *International Research Journal of Engineering and Technology*, 5(3), 962-964.
- [2] Muna Ahmed Awel, Ali Imam Abidi (2019). Review on Optical Character Recognition. *International Research Journal of Engineering and Technology*, 6(6), 3666-3669.
- [3] Sushant Chandra, Saurav Sisodia, Preeti Gupta (2020). Optical Character Recognition - A Review.



International Research Journal of Engineering and Technology, 7(4), 3037-3041.

- [4] A. M. Sabu and A. S. Das, "A Survey on various Optical Character Recognition Techniques," 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, 2018, pp. 152-155, doi: 10.1109/ICEDSS.2018.8544323.
- [5] Shubhangi Singh, Pranjal Sakargayan, Ajitesh Singh (2018). Photo Optical Character Recognition Model. International Research Journal of Engineering and Technology, 5(11), 1696-1699
- [6] Prasanta Pratim Bairagi (2018). Optical Character Recognition for Hindi. International Research Journal of Engineering and Technology, 5(5), 3968-3973
- [7] H. Lin and C. Hsu, "Optical character recognition with fast training neural network," 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, 2016, pp. 1458-1461, doi: 10.1109/ICIT.2016.7474973
- [8] Perwej, Dr. Yusuf & Hannan, Shaikh & Asif, Ali & Mane, Arjun. (2014). An Overview and Applications of Optical Character Recognition. International Journal of Advance Research In Science And Engineering (IJARSE). Vol. 3. Pages 261- 274.