**Project Report**

**on**

**Text Extractor Using AWS**

**Submitted as Mini Project Report**

**FOR MINI PROJECT LAB(KCS-554)**

**Session 2022-23**

**In**

**Cloud Computing**

**By**

**Sachin Kumar Singh**

**(2000320130137)**

**Under the guidance of**

**MS. Meenakshi Vishnoi**

**ABES ENGINEERING COLLEGE, GHAZIABAD**

|  |  |  |
| --- | --- | --- |
| Description: C:\Users\es00952\Desktop\Logo.jpg | APJAK logo |  |

**AFFILIATED TO**

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW**

**(Formerly UPTU)**

**STUDENT’S DECLARATION**

I hereby declare that the work being presented in this report entitled “Text Extractor Using AWS” is an authentic record of our own work carried out under the supervision of Ms. Meenakshi Vishnoi.

The matter embodied in this report has not been submitted by us for the award of any other degree.

**Dated:** **Signature of students**

Sachin Kumar Singh(IT)

## This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

|  |  |
| --- | --- |
| Signature of HODMr. Amit SinhaInformation TechnologyDate............................ | Signature of SupervisorMs. Meenakshi VishnoiAssistant ProfessorInformation Technology |

## 

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. We owe special debt of gratitude to Ms. Meenkashi Vishnoi, Assistant Professor, Information Technology, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor (Dr.) Amit Sinha, Head, Department of Information Technology, ABESEC Ghaziabad for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Name :Sachin Kumar Singh

Roll No.:2000320130137

Signature:

**TABLE OF CONTENTS**

1. Abstract

2.      Introduction

3.      Background and Literature Survey

4.      Proposed work & Methodology

5.      Implementation & Results

6.      Conclusion and future work

7. References

8. Appendix

**ABSTRACT**

Nowadays everyone has mobile phones and people usually capture images of textual documents, handwritten papers, banners, packed food, prescriptions, bills and anything which is of their work. Sometimes they need to work on same data which is present in images and they write/type it manually which requires a lot of time. Also, there is huge demand of storing the data or information available in paper documents format into a digital format so that the data is searchable and editable.

Hence, there is need of software which is able to extract text from captured images. So we are making a project which can extract text from uploaded images. The uploaded image is analyzed using AWS Textractor of Amazon Web Services which is implemented using BOTO3 which convert the detected text into machine readable text. This AWS Textractor is a cloud-based service which allow developers to directly use the cloud services without actually knowing about it which is a feature of platform as a service (PaaS) of Cloud Computing. With this service we can built an application using machine learning concepts without prior knowledge of it, but by simply accessing this service through cloud.

The main advantage of this application is that it is free of cost as we are using all the services by remaining in a free-tier only. The objective of this project is to use AWS Textractor  to automatically extract text and data from scanned documents without any machine learning (ML) experience.

**INTRODUCTION**

Amazon Textract is a service that automatically extracts text, handwriting, and data from scanned documents. It goes beyond simple optical character recognition (OCR) to identify, understand, and extract data from forms and tables. Today, many companies manually extract data from scanned documents such as PDFs, images, tables, and forms, or through simple OCR software that requires manual configuration (which often must be updated when the form changes. You can quickly automate document processing and act on the information extracted, whether you’re automating loans processing or extracting information from invoices and receipts. Textractor can extract the data in minutes instead of hours or days.

In this project we are going to extract text from images by using AWS OCR API with the help of BOTO3 SDK. We can upload sample text images and can get the text what is written in the image. When we run the python script the GUI pops up. This GUI is made using tkinter library of python. Its prompts the user to upload the desired image. Once the image is uploaded it is passed on to the AWS Textractor via BOTO3 which is a python SDK. The image is processed and it’s text is extracted and shown as output on the terminal. The technologies used in this projects are Python , AWS Textractor and BOTO3.

**Literature Survey**

* [An Overview and Applications of Optical Character Recognition](https://www.researchgate.net/publication/267465115_An_Overview_and_Applications_of_Optical_Character_Recognition)

Author: Dr Yusuf Perwej

Abstract:

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic conversion of scanned or photographed images of typewritten or printed text into machine-encoded/computer-readable text.  The paper presents introduction, major research work and applications of Optical Character Recognition in various fields. At the first introduction of OCR will be discussed and then some points will be stressed on the major research works that have made a great impact in character recognition. And finally the most important applications of OCR will be covered and then conclusion.

# [A Detailed Analysis of Optical Character Recognition Technology](https://www.researchgate.net/publication/311851325_A_Detailed_Analysis_of_Optical_Character_Recognition_Technology)

Author: Karej Hamad

Abstract:

In many different fields, there is a high demand for storing information to a computer storage disk from the data available in

printed or handwritten documents or images to later re-utilize this information by means of computers. In this paper we

investigate OCR in four different ways. First we give a detailed overview of the challenges that might emerge in OCR stages. Second, we

review the general phases of an OCR system such as pre-processing, segmentation, normalization, feature extraction, classification and

post-processing. Then, we highlight developments and main applications and uses of OCR and finally, a brief OCR history are discussed.

Therefore, this discussion provides a very comprehensive review of the state-of-the-art of the field.

**BACKGROUND**

Text Extraction plays a major role in finding vital and valuable information. Text extraction involves detection, localization, tracking, binarization, extraction, enhancement and recognition of the text from the given image. These text characters are difficult to be detected and recognized due to their deviation of size, font, style, orientation, alignment, contrast, complex colored, textured background.

Due to rapid growth of available multimedia documents and growing requirement for information, identification, indexing and retrieval, many researches have been done on text extraction in images.

Several techniques have been developed for extracting the text from an image. The proposed methods were based on morphological operators, wavelet transform, artificial neural network, skeletonization operation, edge detection algorithm, histogram technique etc. All these techniques have their benefits and restrictions.

Text information extraction (TIE) from images is an open research area because of its unsolved challenges with respect to the heterogeneity in image types, mode of image capture, position of text and the clarity of text information.

Currently, the number of images captured using mobile phones is voluminous. The information from such images is capable of providing valuable input to the user as well to applications that depend on the image text information.

Text is the pipeline of human communication and images containing text can aid the semantic understanding of the image. Types of image text are explored along with an introduction to TIE and its applications.

Text detection is emphasized and an attempt to categorize the features used by text detection is made. With a brief discussion on the onset research works, the available datasets and performance metrics are listed out.

With the increasing popularity of portable camera devices and embedded visual processing, text extraction from natural scene images has become a key problem that is deemed to change our everyday lives via novel applications such as augmented reality. Text extraction from natural scene images algorithms is generally composed of the following three stages:

1. detection and localization,
2. text enhancement and segmentation and
3. optical character recognition (OCR).

The problem is challenging in nature due to variations in the font size and color, text alignment, illumination change and reflections.

**PROPOSED WORK AND METHODOLOGY**

**TECHNOLOGIES USED:**

**AWS TEXTRACT:**

[Amazon Textract](https://aws.amazon.com/textract/) is a fully managed machine learning service that automatically extracts text and data from scanned documents that goes beyond simple optical character recognition (OCR) to identify, understand, and extract data from forms and tables. Many companies today extract data from scanned documents, such as PDFs, tables and forms, through manual data entry (that is slow, expensive and prone to errors), or through simple OCR software that requires manual configuration which needs to be updated each time the form changes to be usable.

To overcome these manual processes, Textract uses machine learning to instantly read and process any type of document, accurately extracting text, forms, tables and, other data without the need for any manual effort or custom code.

Today, many companies manually extract data from scanned documents such as PDFs, images, tables, and forms, or through simple OCR software that requires manual configuration (which often must be updated when the form changes). To overcome these manual and expensive processes, Textract uses ML to read and process any type of document, accurately extracting text, handwriting, tables, and other data with no manual effort.. Textract can extract the data in minutes instead of hours or days. Additionally, you can add human reviews with Amazon Augmented AI to provide oversight of your models and check sensitive data.

**BOTO3:**

Boto3 is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2..

Boto (pronounced boh-toh) was named after the fresh water dolphin native to the Amazon river. The name was chosen by the author of the original Boto library, Mitch Garnaat, as a reference to the company.

We use the AWS SDK for Python (Boto3) to create, configure, and manage AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). The SDK provides an object-oriented API as well as low-level access to AWS services.

**PYTHON:**

Python is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Its design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with the use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule).

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Type_system#DYNAMIC) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm),including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). It is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)) and first released it in 1991 as Python 0.9.0.[[36]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-36) Python 2.0 was released in 2000 and introduced new features such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) garbage collection, [reference counting](https://en.wikipedia.org/wiki/Reference_counting), and [Unicode](https://en.wikipedia.org/wiki/Unicode) support. Python 3.0, released in 2008, was a major revision that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

**IMPLEMENTATION AND RESULTS**

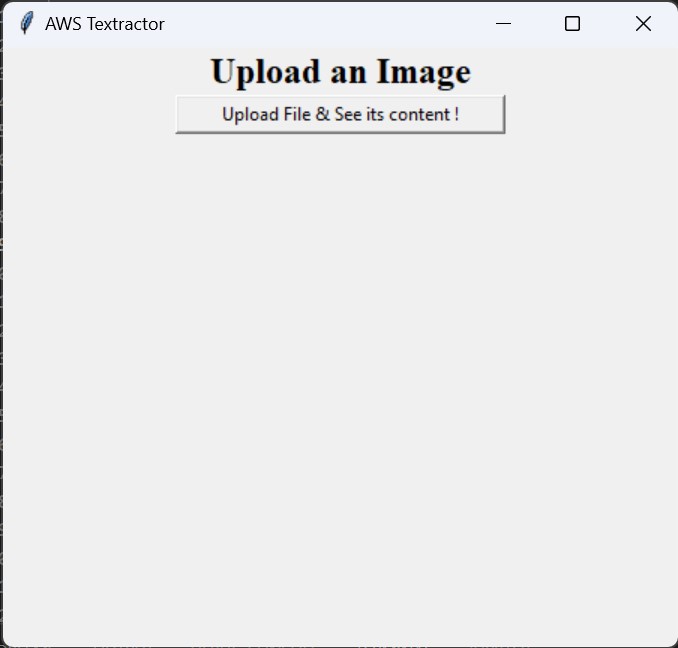
This project is implemented using Python ,BOTO3 and AWS textract. The python script is written to build a GUI .So as the python script is made to run,the GUI pops up.This GUI is made using tkinter library. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task. We just import the tkinter module and create the GUI main window.We can also addd widgets to it.

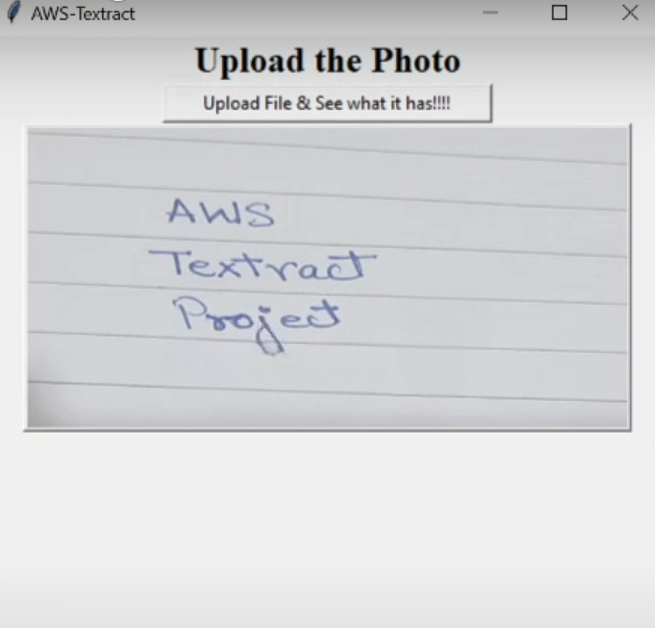
The GUI will ask the user to upload the image to converted to text.As the user uploads the image the python scripts connects to the AWS textract via BOTO3. Boto3 is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2.

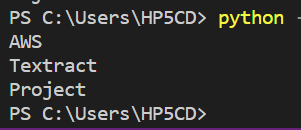
After receiving the image file the AWS Textract converts it to text form. Amazon Textract is a machine learning service that automatically extracts text, handwriting, and data from scanned documents. It goes beyond simple optical character recognition (OCR) to identify, understand, and extract data from forms and tables.

The output is shown on the IDE terminal word by word, each on a different line .We can change this display by changing the blocktype.This is how are our project was implemented.

**RESULTS:**







**CONCLUSION AND FUTURE WORK**

While making this project, we came to know about various AWS services and their concepts. Amazon Web Services offer many products for storage, databases , compute , containers , developer tools, machine learning etc. Many of these products are also free tier eligible. We learnt about these services during our AWS course and its modules and labs. One such product is textract that we have used in our project. AWS Textract extracts texts from the images and give it as output.

We also learnt about BOTO3 and its functionalities. BOTO3 is an AWS software developer kit for python. It allows our python code to get connected to AWS and avail its services. Boto3 makes it easy to integrate our Python application, library, or script with AWS services including Amazon S3, Amazon EC2, Amazon DynamoDB, and more. In our project , BOTO3 acts as an intermediate between the python script and AWS textract.

We got hands on experience of building GUI using Tkinter module of python and integrating it with AWS Services.

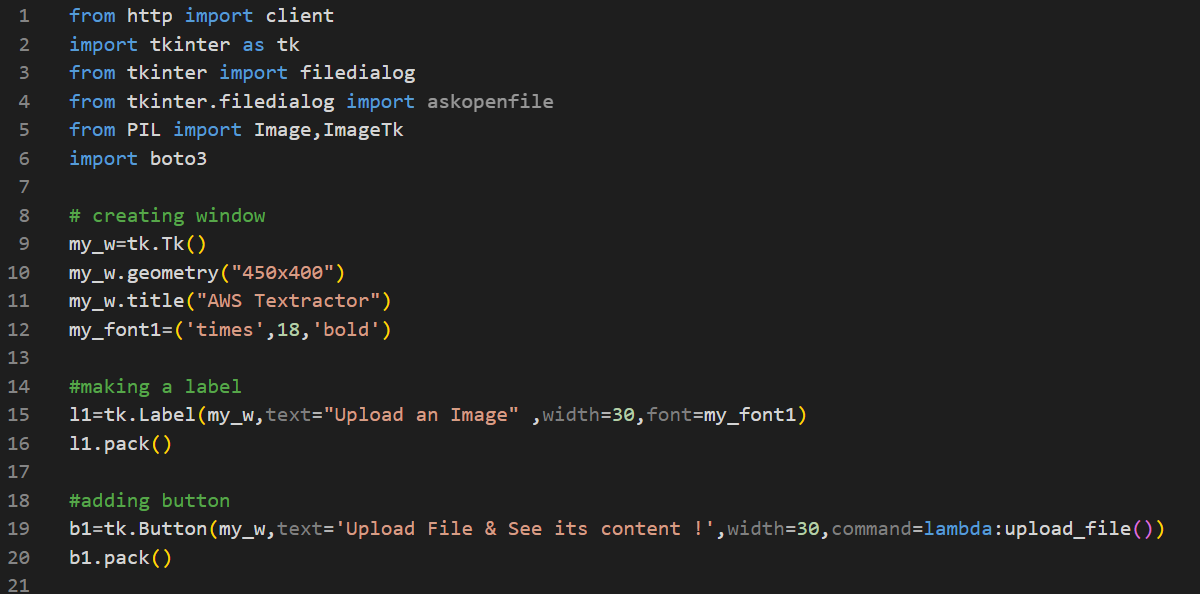
In future we can add other features to this project. These will include a conversion feature of text to pdf format. Another one would be scanning tables, charts along with text. This will render our project with multiple functionalities. We also have work more towards its stability and speed.

**REFERENCES**

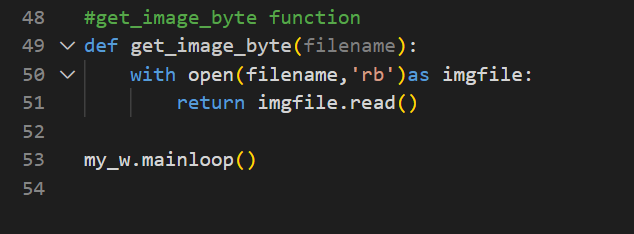
* [**https://aws.amazon.com/textract**](https://aws.amazon.com/textract)
* [**https://aws.amazon.com/sdk-for-python**](https://aws.amazon.com/sdk-for-python)
* [**https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html**](https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html)
* [**https://boto3.amazonaws.com/v1/documentation/api/latest/index.html**](https://boto3.amazonaws.com/v1/documentation/api/latest/index.html)
* [**https://www.codecademy.com/catalog/language/python**](https://www.codecademy.com/catalog/language/python)
* [**https://www.tutorialspoint.com/python/index.htm**](https://www.tutorialspoint.com/python/index.htm)
* [**https://www.geeksforgeeks.org/python-gui-tkinter/**](https://www.geeksforgeeks.org/python-gui-tkinter/)
* [**https://www.javatpoint.com/python-tkinter**](https://www.javatpoint.com/python-tkinter)

**APPENDIX**

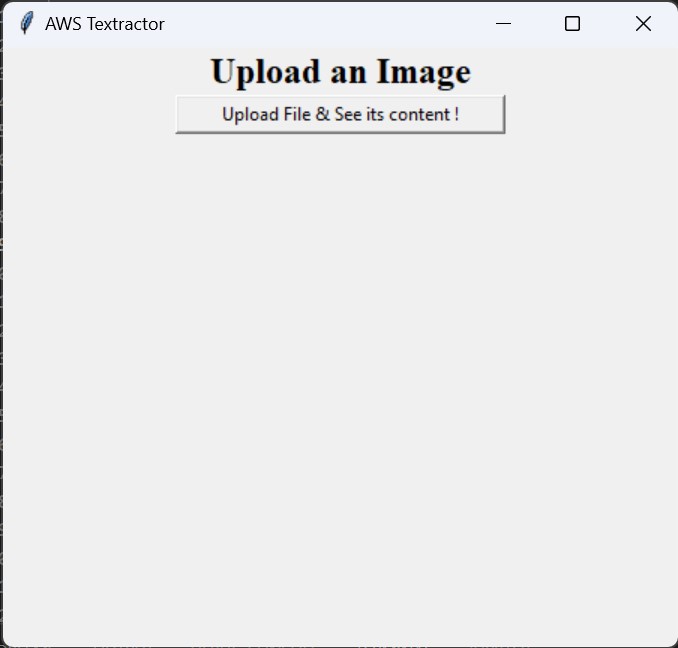
**CODE SNIPPETS:**

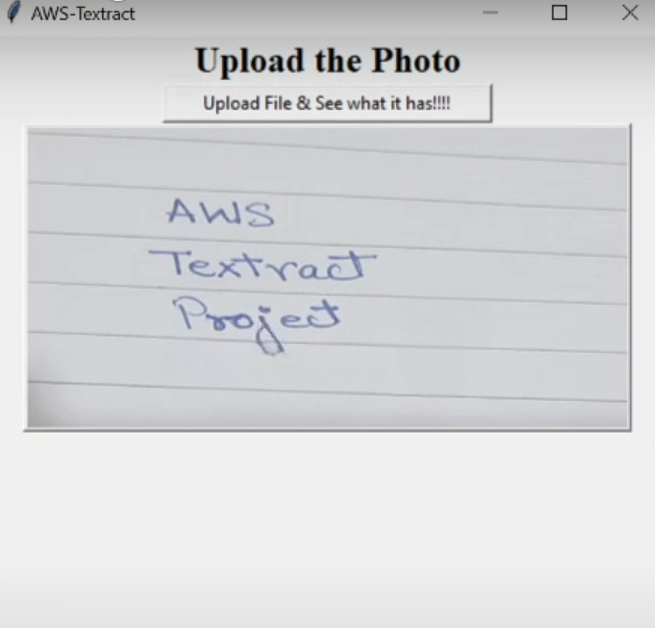
****

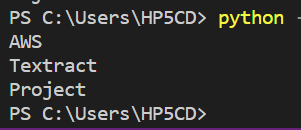
****

****

**RESULTS:**

****

****

****

**CERTIFICATES:**

****