NYC DDC TOWN +GOWN: NYC

MACHINE READING CODE

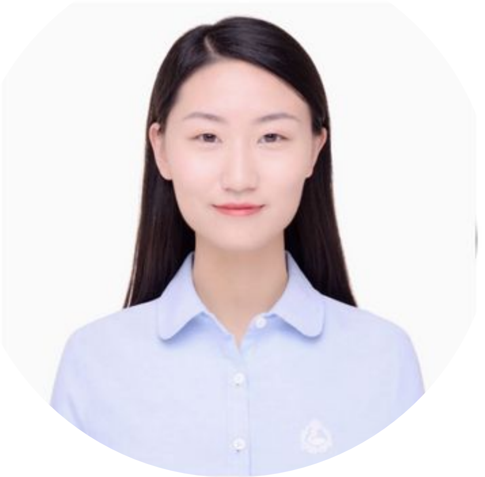
TEAM 26

Agenda

1	Our Team	7	Analysis
2	Problem Statement	8	Technical Workings
3	Our Journey	9	DEMO
4	Literature review	10	Comparative Analysis
5	Our Work	11	Pricing
6	Research Journey	12	Future Scope

MEET THE TEAM

Likhith Ravilla



Jiaxuan Xu

Xingying Li

Chuheng He

Preethi Bachu

Problem Statement

Problem:

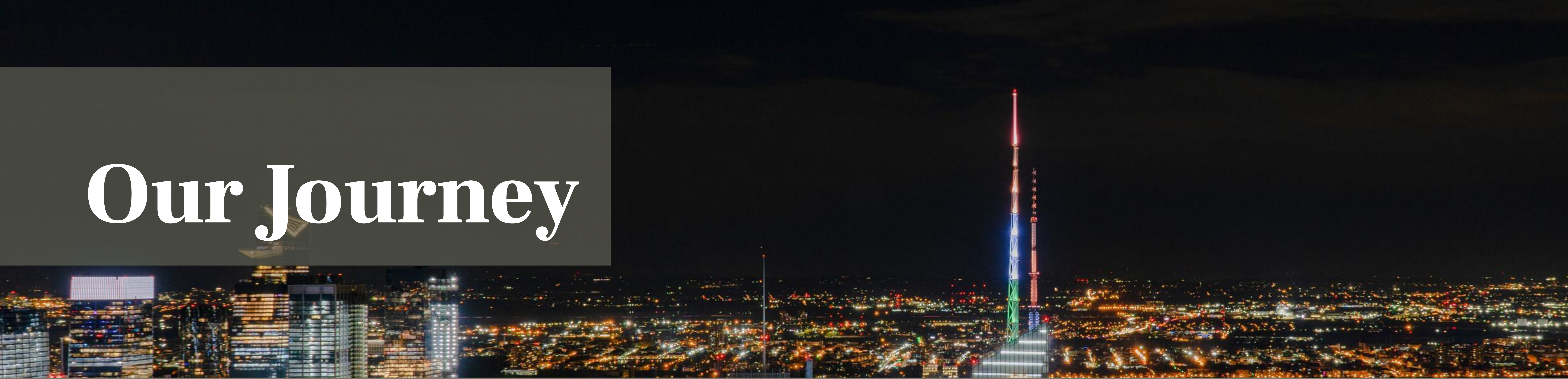
The CUSP capstone project faced challenges in extracting information from NYS DEC forms due to handwritten and printed content, requiring manual data input.

Solution:

To address this, a new project was initiated to develop a product using optical handwriting recognition techniques, enabling streamlined analysis and digitization of form information.

Impact:

This solution aims to improve efficiency, reduce errors, and enhance productivity when working with forms and archaic documents, empowering users to easily extract and analyze data.



Our Journey

Getting to know each other and NYC DDC

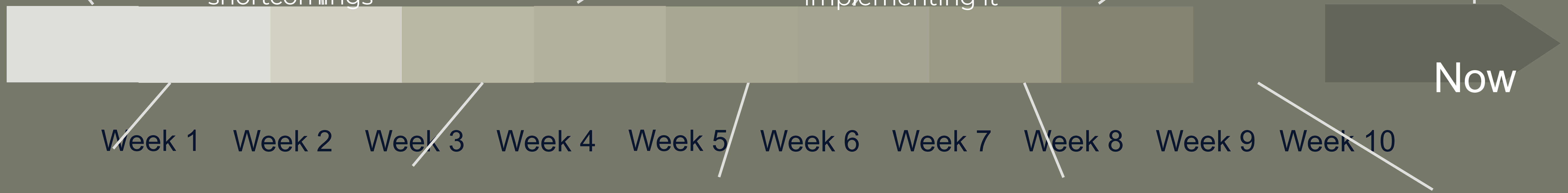
Conducting further research and understanding their shortcomings

Assessed the Scope and Complexity of the Project

Worked on understanding the API and implementing it

Created a sample demo using the API

We're ready with our final product



Reviewing the CUSP project and understanding their technologies

Looking for other prospective technologies

Came across Google Vision and Cloud

Started extracting the required features and fine tuning the

Worked on building the front end and clearing the backend

Literature Review

An overview of the CUSP project

CUSP students visualized the flow of construction and demolition waste (CDW) in Long Island and New York City to support policymaking for local CDW recycling and reuse.

They transformed regulatory reports into a structured format and merged them into a machine-readable dataset.

A user-friendly spatial visualization tool was created for interactive exploration by non-technical users.

Challenges arose in data handling due to OCR software's difficulty in reading forms, resulting in misrepresenting repeated information and using aliases for data fields.

Manual data entry became necessary in the later stages of the project

Our Work

We developed a machine reading website using GCP + Vision API

Users can upload their scanned PDF forms, and the website generates a CSV file with all the extracted text data

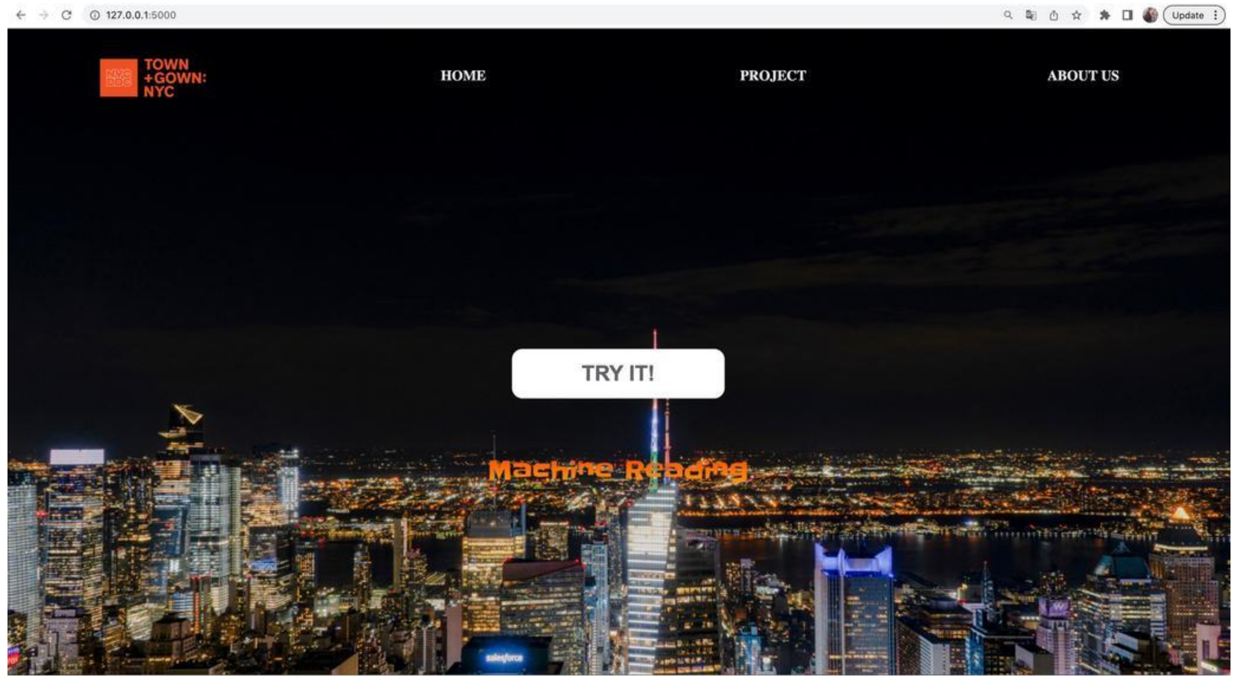
The website also showcases our research and all the other elements involved

Solution

Google Cloud Platform

+

Cloud Vision API



Machine Reading Website

Research Journey

Our journey from developing OCR algorithms to GCP vision API

Build a Handwritten Text Recognition (HTR) System using TensorFlow

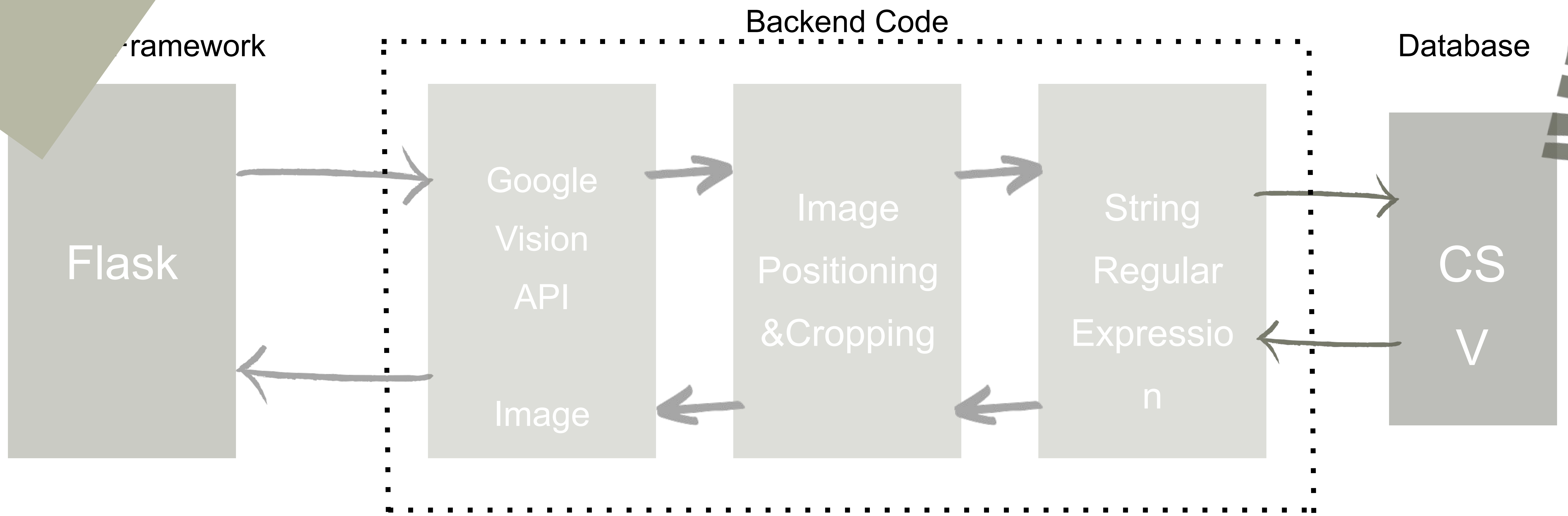
Utilize Google Cloud Platform + Vision API to work on Optical Character Recognition (OCR)

Google Vision API vs OCR

Criteria	Vision API	Building an OCR Model
Time and Resources	Requires less time and resources as pre-trained models and APIs are available	Requires significant investment of time and resources for data collection, annotation, model development, and ongoing maintenance
Accuracy	Highly accurate and reliable due to pre-trained models and large and diverse datasets	Potentially higher accuracy but requires significant expertise and ongoing optimization
Customization	Limited customization but can meet the needs of most use cases	Greater customization and control over OCR process, but may not be feasible for certain use cases
Cost	Typically more cost-effective with lower upfront costs and predictable ongoing costs	Expensive due to the need for a team of experts in machine learning and OCR
Maintainence	The provider handles ongoing maintenance and updates	Requires ongoing maintenance and optimization to ensure continued accuracy and performance
Data Variety	Can handle a wide variety of image formats and languages	Requires diverse training data that reflects the range of variability present in the data to be processed

NYU
TANDON SCHOOL OF ENGINEERING
Polytechnic Institute

Full Stack Architecture



Framework

Flask

Backend Code

Google Vision API

Image

Image Positioning &Cropping

String Regular Expression

Database

CSV

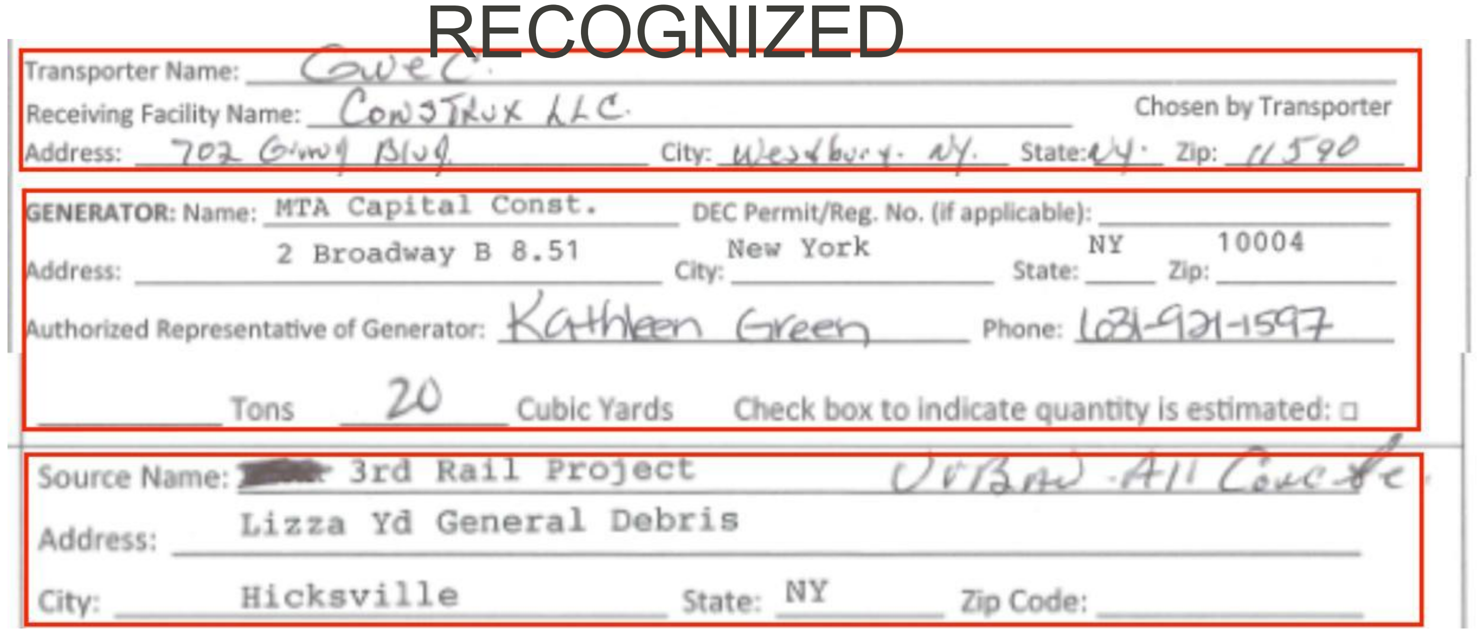
Image Positioning & Cropping

DEFINED TWO FUNCTIONS

RAW SCANNED PDF FILE

1A-002_AB_Oil_c
dd.2021....wtd.pdf

IMAGE WITH REGIONS NEED TO BE RECOGNIZED



Transporter Name: GweC.
Receiving Facility Name: ConsTrux LLC. Chosen by Transporter
Address: 702 [illegible] Blvd City: Westbury. NY. State: NY. Zip: 11590

GENERATOR: Name: MTA Capital Const. DEC Permit/Reg. No. (if applicable):
Address: 2 Broadway B 8.51 City: New York State: NY Zip: 10004
Authorized Representative of Generator: Kathleen Green Phone: 631-921-1597
Tons 20 Cubic Yards Check box to indicate quantity is estimated: □

Source Name: [redacted] 3rd Rail Project Urban All Concrete
Address: Lizza Yd General Debris
City: Hicksville State: NY Zip Code:

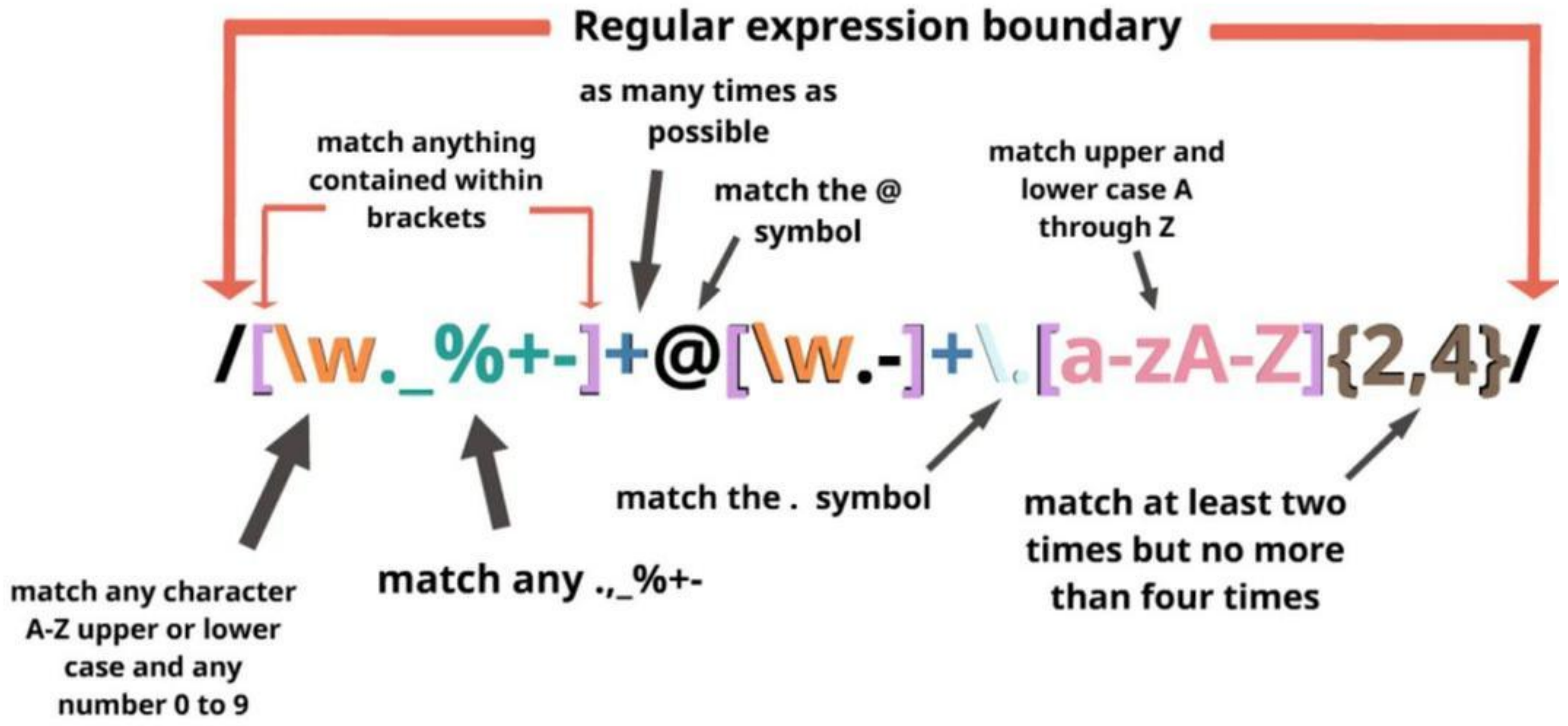
pdf2png():

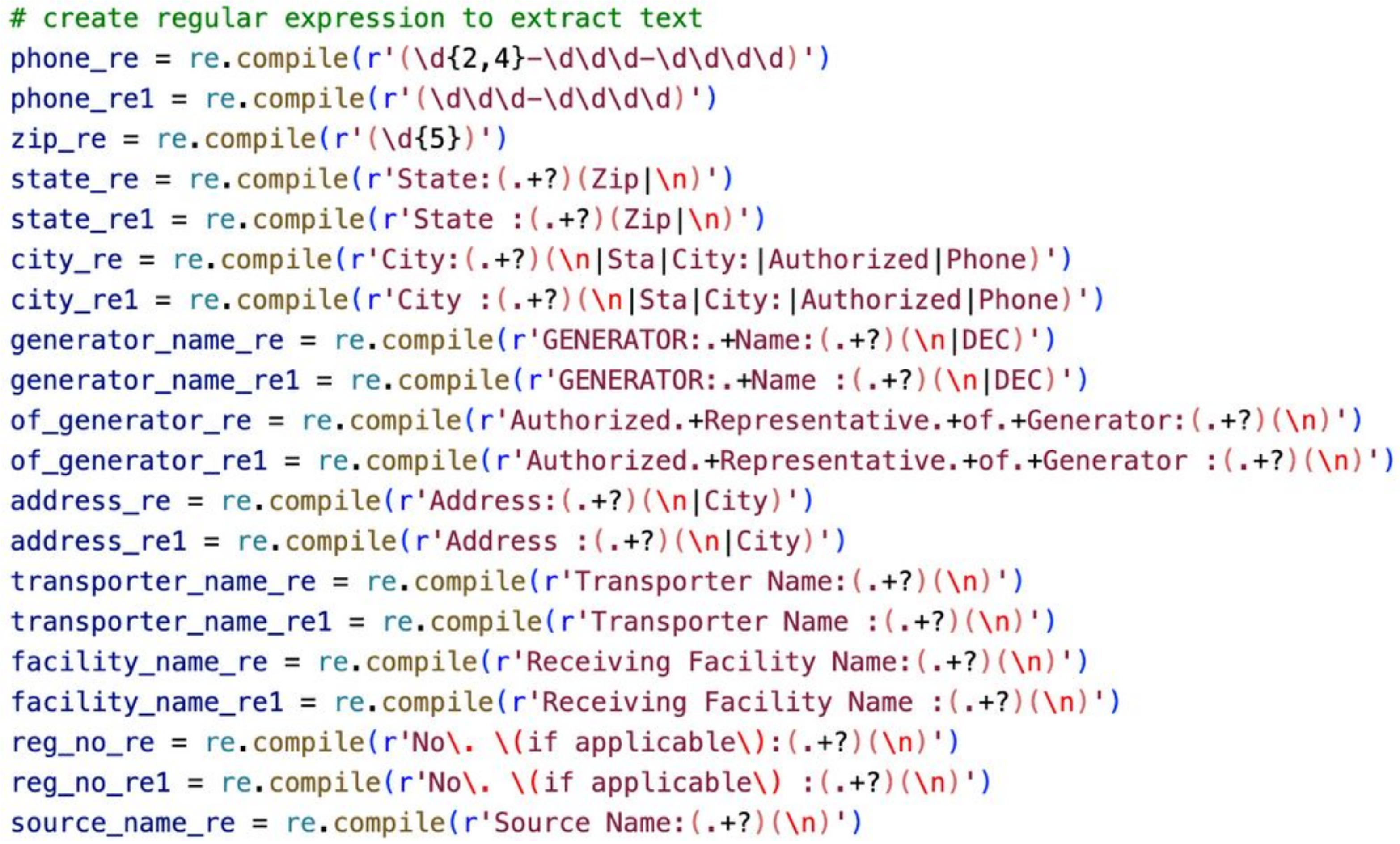
Used to extract images from a PDF, converts each page of a given PDF file into a PNG image file and saves it to a specified location.

crop():

Crops the image to three different regions of interest specified by the bounding box parameter to remove any unnecessary borders and returns the path of the cropped image file

String Regular Expression & Matching


Regular expression boundary
as many times as possible
match anything contained within brackets
match the @ symbol
match upper and lower case A through Z
/[\w._%+-]+@[\w.-]+\.[a-zA-Z]{2,4}/
match the . symbol
match at least two times but no more than four times
match any character A-Z upper or lower case and any number 0 to 9
match any .,_%+-



```
# create regular expression to extract text
phone_re = re.compile(r'(\d{2,4}-\d\d\d-\d\d\d\d)')
phone_re1 = re.compile(r'(\d\d\d-\d\d\d\d)')
zip_re = re.compile(r'(\d{5})')
state_re = re.compile(r'State:(.+?)(Zip|\n)')
state_re1 = re.compile(r'State :(.+?)(Zip|\n)')
city_re = re.compile(r'City:(.+?)(\n|Sta|City:|Authorized|Phone)')
city_re1 = re.compile(r'City :(.+?)(\n|Sta|City:|Authorized|Phone)')
generator_name_re = re.compile(r'GENERATOR:.+Name:(.+?)(\n|DEC)')
generator_name_re1 = re.compile(r'GENERATOR:.+Name :(.+?)(\n|DEC)')
of_generator_re = re.compile(r'Authorized.+Representative.+of.+Generator:(.+?)(\n)')
of_generator_re1 = re.compile(r'Authorized.+Representative.+of.+Generator :(.+?)(\n)')
address_re = re.compile(r'Address:(.+?)(\n|City)')
address_re1 = re.compile(r'Address :(.+?)(\n|City)')
transporter_name_re = re.compile(r'Transporter Name:(.+?)(\n)')
transporter_name_re1 = re.compile(r'Transporter Name :(.+?)(\n)')
facility_name_re = re.compile(r'Receiving Facility Name:(.+?)(\n)')
facility_name_re1 = re.compile(r'Receiving Facility Name :(.+?)(\n)')
reg_no_re = re.compile(r'No\. \(if applicable\):(.+?)(\n)')
reg_no_re1 = re.compile(r'No\. \(if applicable\) :(.+?)(\n)')
source_name_re = re.compile(r'Source Name:(.+?)(\n)')
```

handler_pdf() function:

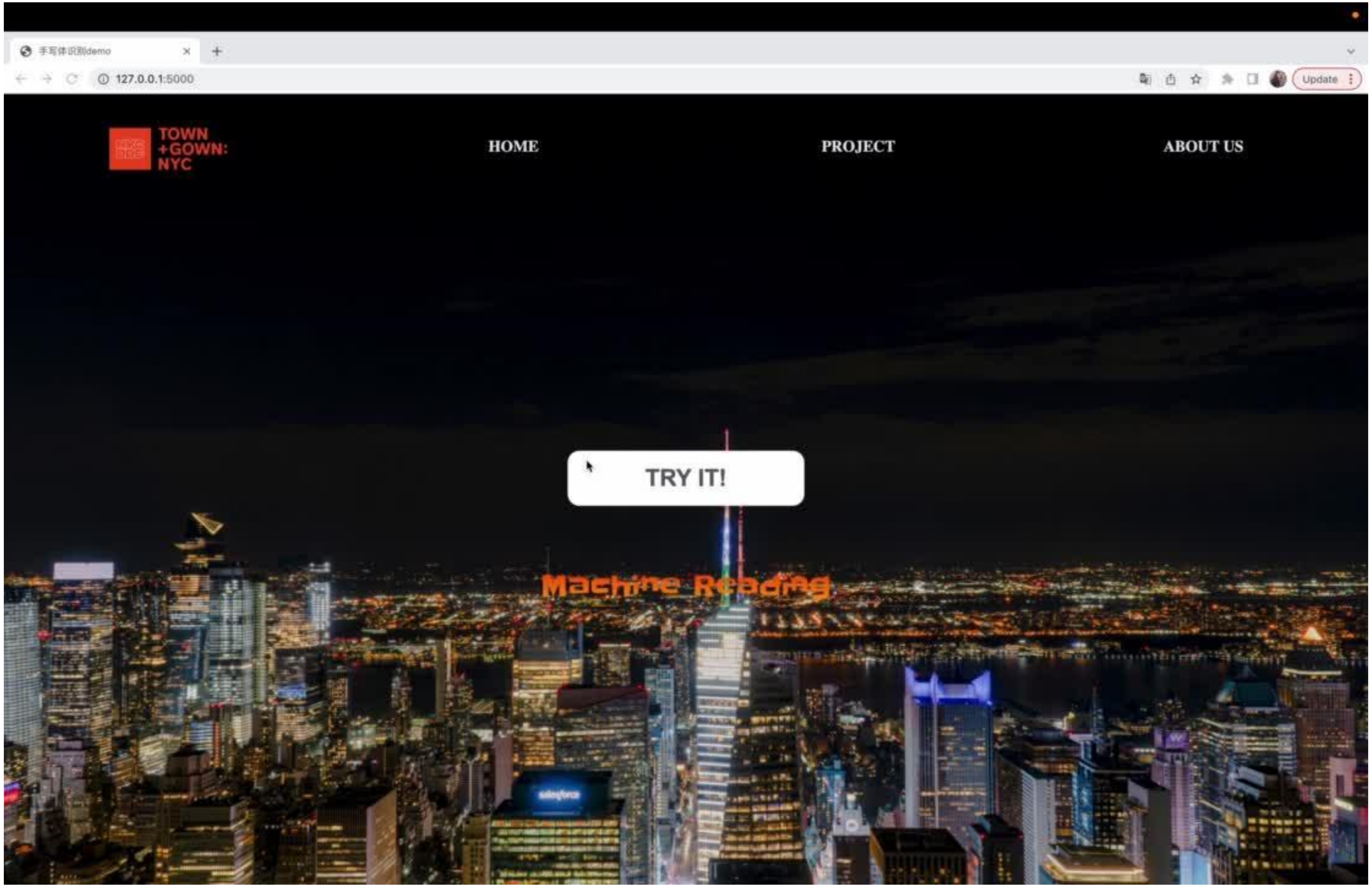
- Uses the PyPDF2 library to read the PDF file specified in the input path.
- Then it passes the content of the file to find_all_fields_v3()
- Returns the resulting dictionary.

→

find_all_fields_v3() :

- A regular expression-based function
- extracts relevant fields information from a string input
- e.g: phone numbers, zip codes, state names, city names, addresses, facility names, generator names, and more from the input string.

DEMO


手写体识别demo
127.0.0.1:5000
Update
TOWN
+GOWN:
NYC
HOME
PROJECT
ABOUT US
TRY IT!
Machine Reading

Google vision vs Competitors

Technology comparison	Accuracy	Customization	Language Support	Maintenance	Integration
Vision API	Highly accurate and reliable due to pre-trained models and largest diverse datasets	Limited customization but is the most suitable for use case with the feature to training also available	Over 50 languages	Ongoing maintenance and updates are handled by providers	Easily integrates with other Google Cloud services, as well as third-party applications through APIs and SDKs
Amazon Rekognition	Provides facial analysis and text detection, but with more emphasis on facial recognition	Offers customization options through custom models and user feedback	Limited	Ongoing maintenance and updates are handled by providers	Offers integration with other Amazon Web Services, as well as third-party applications
IBM Watson Visual Recognition	The product is accurate but also depends on the training	Extensive customization to a point there has to be a dedicated resource to oversee this process	Over 20 languages	Ongoing maintenance and updates are handled by providers	Offers integration with other IBM Cloud services, as well as third-party applications
Microsoft Azure Computer Vision	accurate but requires significant expertise of the platform	Offers some customization options and control over the OCR process	Over 60 languages	Ongoing maintenance and updates are handled by providers	Offers integration with other Microsoft Azure services, as well as third-party applications through APIs and SDKs

Pricing Model

FEATURE	UPTO 1000 UNITS/MONTH	1001 - 5000000 UNITS/MONTH	> 5000001 UNITS/MONTH
Label Detection	Free	$1.50	$1.00
Text Detection	Free	$1.50	$0.60
Document Text Detection	Free	$1.50	$0.60
Safe Search Detection	Free	Free with Label Detection, or $1.50	Free with Label Detection, or $1.50
Facial Detection	Free	$1.50	$0.60
Facial Celebrity Detection	Free	$1.50	$0.60
Landmark Detection	Free	$1.50	$0.60

Future Scope

1

- Make this platform available for other NYC departments to help them digitize archaic documents.

2

- Create a credential usage system to increase the efficient usage of credits and allow multiple users to access our platform.

3

- Create a database and storage solution on the cloud to make this an end-to-end product and also enable data analytics to extract insights from the data

Thank you