Poster/Banner Text Detection and Measurement API

Team Name: *Alpha Strike*

Team Members: Sushovan Bera & Trisha Maity

Table of Contents

Abstract

This project introduces a **RESTful API** designed for the automated detection and measurement of textual content in posters and banners. Utilizing **Tesseract OCR** for text extraction and **OpenCV** for image processing, the system enables users to upload an image via URL, from which it identifies text regions, extracts the text, and estimates the physical dimensions (in centimeters) of the detected text blocks based on pixel measurements.
The API returns a structured JSON response containing bounding box coordinates, confidence scores, extracted text, and approximate dimensions, making it suitable for applications such as advertisement auditing, digital archiving, print layout verification, and compliance checking.
Developed using **Python** and **Flask**, this tool demonstrates a practical approach to solving real-world problems in media analysis by combining powerful open-source libraries with a flexible web interface. While the solution assumes a standard DPI for measurement estimation, it provides a strong foundation for further improvements, including dynamic DPI detection and support for multilingual text.

Introduction

Posters and banners are vital mediums for communication in both digital and physical formats. Detecting whether an image contains poster-like text and accurately measuring its dimensions can assist in verifying visual content compliance, ad validations, or archiving. This project addresses this challenge by implementing an OCR-based Flask API that automates text detection and measurement from images.

With the growth of digital media and advertising, there is a growing need for systems that can identify and measure poster content automatically. Manual identification and measurement are time-consuming and error-prone. This project proposes an API that automates this process using OCR techniques combined with image preprocessing. The main goal is to accurately detect the text, provide confidence scores, and return the size of the image and the bounding box in real-world units.

Literature Survey

Optical Character Recognition (OCR)

•Tesseract OCR is an open-source engine that converts printed text in images into machine-readable format.
•It returns detailed metadata including confidence and text location.

Image Preprocessing Techniques

•Grayscale conversion simplifies image data.
•Otsu's thresholding method enhances contrast for better OCR performance.

Text Localization

•Techniques that identify the position of text help in isolating the content area.
•pytesseract.image_to_data helps localize each detected word.

DPI and Real-World Measurement

•DPI (dots per inch) is used to convert image pixel dimensions to centimeters.

RESTful APIs

•APIs provide programmatic access to image processing services, useful in modern web and mobile platforms

Methodology

Tool/Library	Purpose
Flask	Web framework used to build the REST API
Python	Programming language used for the entire project
OpenCV	Image processing (convert to grayscale, threshold)
Tesseract OCR	Recognizes and extracts text from images
pytesseract	Python wrapper for Tesseract OCR

Workflow

- **Input: Image URL (or file upload).**
- **Preprocessing**:
  - Convert to grayscale.
  - Apply adaptive thresholding (OTSU method).
- **OCR:**
  - Extract text data using pytesseract.image_to_data().
  - Identify bounding boxes for each word.
- **Calculations:**
  - Compute overall bounding box.
  - Estimate image size in centimeters using DPI.
  - Calculate average OCR confidence.
- **Output:**
  - JSON response containing:
    - containsPoster: True/False
    - boundingBox: x, y, width, height
    - dimensionsCm: estimated physical size
    - confidence: OCR accuracy

Sample JSON Output


```
{
"boundingBox": {
"height": 583,
"width": 1581,
"x": 1121,
"y": 1084    },
"confidence":
0.72,
"containsPoster":
true,
"dimensionsCm":
{       "height":
80.96,
"width": 107.95
}}
```

Dataset

This project does **not rely on a pre-annotated dataset**. Instead, it uses a **dynamic, user-driven approach**, where users provide **image URLs** of posters and banners. These images are fetched at runtime, processed, and analyzed for text detection. However, for testing and demonstration purposes, a **curated collection of poster/banner images** was gathered from publicly available sources

- The API is image-agnostic and works on any user-provided image via url.
- Sample test images include posters from Wikimedia Commons, advertisement banners, and flyers.
- No fixed dataset is required; it's a runtime detection tool.

Results

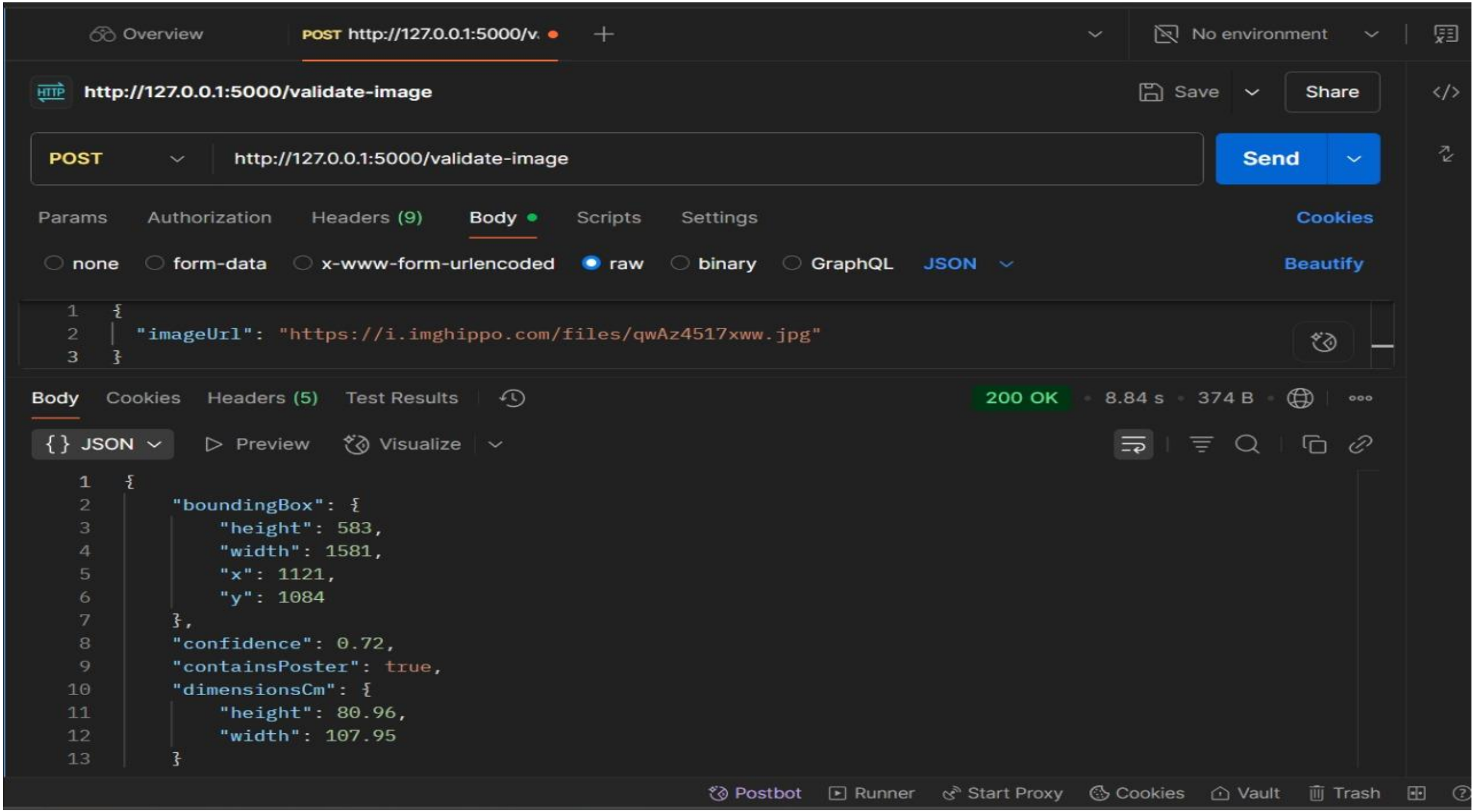
- Accurately detects poster/banner text from various images.
- Produces bounding boxes with average OCR confidence > 70%.
- Handles both remote image URLs and uploaded files.
- Gives physical size approximation using standard DPI (96).

Input:

{ "imageUrl": "https://i.imghippo.com/files/qwAz4517xww.jpg"}

Output:

{ "boundingBox": { "height": 583, "width": 1581, "x": 1121, "y": 1084 }, "confidence": 0.72, "containsPoster": true, "dimensionsCm": { "height": 80.96, "width": 107.95 }}


Overview
POST http://127.0.0.1:5000/v.
No environment
http://127.0.0.1:5000/validate-image
Save
Share
POST
http://127.0.0.1:5000/validate-image
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"imageUrl": "https://i.imghippo.com/files/qwAz4517xww.jpg"
}
Body
Cookies
Headers (5)
Test Results
200 OK
8.84 s
374 B
{} JSON
Preview
Visualize
{
"boundingBox": {
"height": 583,
"width": 1581,
"x": 1121,
"y": 1084
},
"confidence": 0.72,
"containsPoster": true,
"dimensionsCm": {
"height": 80.96,
"width": 107.95
}
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Challenges Faced

- Handling broken or inaccessible image URLs
- Tesseract OCR failing on low-resolution or blurry images
- Converting pixel-based measurements accurately without DPI metadata
- Network errors when fetching images

Conclusion

The **Poster/Banner Text Detection and Measurement API** successfully demonstrates an automated solution for detecting text within visual media and estimating its physical dimensions. By leveraging open-source technologies such as **Tesseract OCR** and **OpenCV**, the system is capable of processing poster images in real time, extracting relevant textual information, and providing meaningful measurement data in a structured JSON format.

The key achievement of this project lies in its **simplicity**, **scalability**, and **applicability** across multiple domains such as advertising analysis, compliance verification, and digital archiving. The API's ability to convert pixel-based text bounding boxes to centimeter measurements allows for approximate physical size estimation, which can be crucial for layout assessments in printed media.

While the system performs well in most scenarios, it also highlights common challenges in OCR-based systems—particularly when dealing with image noise, variable resolutions, or complex font styles. The current approach assumes a fixed DPI for conversion, which introduces a degree of approximation but maintains a balance between accuracy and ease of use.

In summary, this project delivers a **functional, user-friendly, and modular API** for poster text detection and measurement, laying the groundwork for more advanced enhancements in future iterations.

References

- R. Smith, "An Overview of the Tesseract OCR Engine", ICDAR 2007
- N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE, 1979
- L. Neumann and J. Matas, "Real-Time Scene Text Localization and Recognition", CVPR 2012
- Flask Documentation: https://flask.palletsprojects.com/
- Tesseract OCR: https://github.com/tesseract-ocr/tesseract
- OpenCV Library: https://opencv.org/

Thank you