**Documentation for Bounding Box Merging Project**

**Overview**

This project focuses on developing a comprehensive system that utilizes Optical Character Recognition (OCR) and clustering algorithms to automatically identify and merge closely related bounding boxes extracted from images of Excel sheets. The primary goal is to streamline the process of data extraction from structured documents, enhancing the efficiency of data processing tasks.

**Approach**

The project is divided into two main components:

1. OCR-based Data Extraction: Utilizes Tesseract OCR to read images and extract bounding box coordinates and save it to two csv files.

2. Unsupervised Model for Merging: Employs the DBSCAN clustering algorithm to group and merge closely related bounding boxes based on their spatial proximity.

**Algorithms Used**

- Tesseract OCR: For extracting text and bounding box coordinates from images.

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Used for clustering bounding boxes. Chosen for its ability to handle noise and find arbitrarily shaped clusters.

- StandardScaler: From scikit-learn, used for standardizing features before applying DBSCAN.

-Evaluation Metrics: Davies-Bouldin Score and Calinski-Harabasz Score.

**Input/Output Formats**

- Input: A ZIP file containing images of an Excel sheet with two columns of bounding box coordinates. Each image represents a portion of the full Excel sheet.

- Output: A CSV file containing the merged bounding box coordinates, indicating which bounding boxes from the input images have been combined.

**Instructions for Running the System on New Datasets**

**Setup**

Clone the github repository

* git clone https://github.com/saimaharana4/bounding\_box\_merger.git

Head towards the project repository

* cd bounding\_box\_merger

Activate virtual env

* venv\Scripts\activate

Install dependencies

* pip install -r requirements.txt

**Running the OCR-based Data Extraction**

1. Place your ZIP file containing the images in “bounding\_box\_merger\data\raw” directory.

2. Update the `zip\_file\_path` variable in the script to point to your ZIP file.

3. Run the OCR script to extract bounding box coordinates and save them into CSV files.

```python

python src/data\_extraction.py

```

**Running the Data Science Model for Merging**

1. Ensure the CSV files generated from the OCR step are in the “D:\bounding\_box\_merger\data\processed\csv” location.

2. Adjust `list\_a\_csv\_path` and `list\_b\_csv\_path` in the clustering and merging script to point to these files.

3. Run the clustering and merging script to process the extracted bounding boxes and generate the merged bounding boxes.

```python

python src/clustering\_and\_merging.py

```

**Visualization and Evaluation**

- For visualization, adjust the `image\_path` and `boxes\_csv\_path` in the `visualization.py` script to point to an image and its corresponding CSV file of bounding boxes. Run the script to see the overlaid bounding boxes.

- For evaluation, We have use Calinski-Harabasz Score and Davies-Bouldin Score in our notebook/ to get the best result.

**Note:**

- Ensure Tesseract OCR is correctly installed and configured on your system, including setting the `pytesseract.pytesseract.tesseract\_cmd` variable to the path where Tesseract is installed.