README: How to Run the Code

This project demonstrates the process of organizing data, converting image formats, and training a YOLOv5 model. Please follow the steps below to execute the code on your machine.

Setup Instructions

1. **Upload Data to Google Drive:**
   - Place the **Images** folder containing `.HEIC` images and the **Labels** folder with corresponding `.txt` files into your Google Drive.

Ensure the folders are stored in the following directory:
`/content/drive/My Drive/Colab Notebooks/CV/`

The directory should look like:
`/content/drive/My Drive/Colab Notebooks/CV/Images/`
`/content/drive/My Drive/Colab Notebooks/CV/Labels/`

2. **Place the YAML Configuration File:**

Add the provided `.yaml` configuration file to the following directory in your Drive:
bash
Copy code
`/content/yolov5/data/yolov5_config.yaml`

3. **Modify the File Paths:**
   - Update the file paths in the script to match your Drive's structure if it's different from the default mentioned above.
4. **Running the Code:**
   - Execute the script in Google Colab.
   - **Note:** The dataset is moved into training, validation, and test folders during execution. After the script runs, these folders will no longer exist in their original location. If you need to rerun the script, re-upload the **Images** and **Labels** folders to their original location.
5. **Dependencies:**

○ The script uses YOLOv5, which will be cloned from GitHub and requires Python libraries like `torch`, `Pillow`, and `pillow_heif`. Ensure these dependencies are installed during execution.

Steps to Run

1. **Clone YOLOv5:**
   ○ YOLOv5 will be cloned automatically during the execution. The script installs all required dependencies.
2. **Data Preprocessing:**
   ○ The `.HEIC` images will be converted to `.JPEG` format and organized into `train`, `val`, and `test` folders.
3. **Training the Model:**
   ○ The YOLOv5 model will be trained using the data split and configuration provided in the `.yaml` file.
4. **Evaluation:**
   ○ Once training completes, the script evaluates the model and prints performance metrics like precision, recall, and mAP.