Custom-Object Character Recognition(OCR) on AWS (Google Drive/ Cloud Storage)

Build a Custom OCR by combining YOLO and Tesseract, to read the specific contents of a Lab Report and convert it into an editable file. Use YOLO_V3 to trained on the personal dataset. Then the coordinates of the detected objects are passed for cropping the detected objects and storing them in another list. This list is passed through the Tesseract to get the desired output.

Model

- You can train a custom YOLO_V3 model using your custom dataset.
- Make a folder named model and put the weights file inside it.

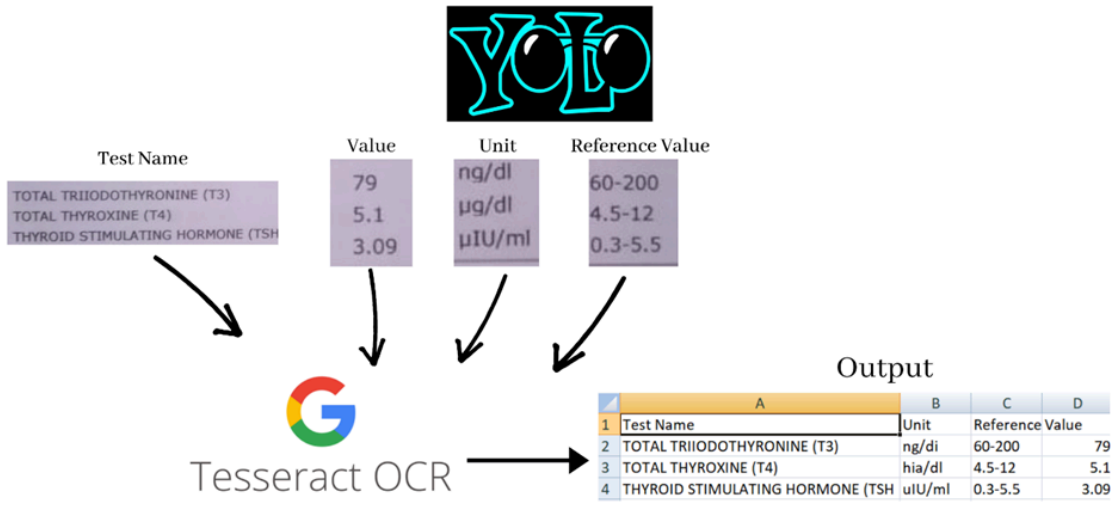
Data

- Link for the dataset https://drive.google.com/drive/folders/1uAc8xE6AS8YCb65iNMFyEfB5dzAWsgFN?usp=sharing

Dependencies

- Install Tesseract OCR Engine in the system
- Install Pytesseract library pip install pytessercat
- Install OpenCV pip opencv

Workflow



Getting Started

python Custom_OCR.py --image <yourimage.jpg>

Step by Step-by-step workflow

This section will highlight the steps I took to implement the Custom-OCR with YOLOv3 and potential areas to be worked on further.

This will show the step-by-step workflow on the following original image.

Detected regions

The first step of the process is taking the bounding box coordinates from YOLOv3 and simply taking the region within the bounds of the box. As this image is super small, we use cv2.resize() to blow the image up 3x its original size.

Then we convert the image to grayscale and apply a small Gaussian blur to smooth it out.

The image is then thresholded to white text with a black background and has Otsu's method also applied. This white text on a black background helps to find the contours of the image.

Then we apply a bitwise_not mask to flip the image to black text on a white background which Tesseract is more accurate with.

The preprocessed images are then passed over to Tesseract and the output is saved as a csv file.

Here's a breakdown of project tasks to do in Amazon SageMaker and S3 for your OCR project:

1. Set Up the Environment

- **Task 1.1:Set Up the Environment (on Google Colab)**
    - Mount Google Drive to store your dataset, model weights, and output files
    - Define folder structure in Drive: datasets/, models/, results.
- **Task 1.2: Set Up SageMaker Notebook Instance**
    - Launch a SageMaker notebook instance.
    - Install necessary dependencies: Tesseract, PyTesseract, OpenCV, and YOLOv3.
    - Ensure that the notebook instance has access to the S3 bucket.

2. Data Preparation

- **Task 2.1: Upload Dataset to colab**
    - Download and upload the dataset to Google Drive. -
    - Resizing images, creating YOLO annotations. - Store preprocessed data in Drive.

- **Task 2.2: Prepare the Dataset**
    - Download and upload the dataset to Google Drive. -
    - Preprocess dataset: resizing images, creating YOLO annotations. -
    - Store preprocessed data in Drive.

3. Model Training

- **Task 3.1: Train YOLOv3 Model**
    - Train YOLOv3 model using Colab GPU runtime.
    - Save trained weights to 'models' folder in Drive.
    - Validate the model using a subset of data.
    - Upload validation results (images with bounding boxes) to Drive.
- **Task 3.2: Model Validation**
    - Run inference on test images.
    - Extract coordinates and crop detected objects.
    - Preprocess: resize, grayscale, blur, threshold.
    - Pass preprocessed images to Tesseract.
    - Save output as CSV in Drive.

Custom-Object Character Recognition(OCR) on AWS

Build a Custom OCR by combining YOLO and Tesseract, to read the specific contents of a Lab Report and convert it into an editable file. Use YOLO_V3 to trained on the personal dataset. Then the coordinates of the detected objects are passed for cropping the detected objects and storing them in another list. This list is passed through the Tesseract to get the desired output.

Model

- You can train a custom YOLO_V3 model using your custom dataset.
- Make a folder named model and put the weights file inside it.

Data

- Validate the trained model using a subset of the data.
- Upload validation results, including images with bounding boxes, to the colab.

4. Inference and Post-Processing

- Run inference on test images.
- Extract coordinates and crop detected objects.
- Preprocess: resize, grayscale, blur, threshold.
- Pass preprocessed images to Tesseract.

- Save output as CSV in Drive

5. Evaluation and Optimization

- Compare OCR output with ground truth to evaluate accuracy.
- Fine-tune model, optimize preprocessing, and improve Tesseract output.

6. Automation and Deployment

- Compare OCR output with ground truth to evaluate accuracy.
- Fine-tune model, optimize preprocessing, and improve Tesseract output.

7. Documentation and Reporting

- Document the full pipeline.
- Save final reports in Drive.

Guidelines for Cost Management in Cloud Service:

Cost Management on Google Colab: - Google Colab is free for basic use, with free access to GPU and limited storage. - Upgrading to Colab Pro/Pro+ offers better resources (~$10-$50/month). - Google Drive: First 15 GB free

Total Estimated Cost

- **Estimated Cost: $0 (for basic users)**

This estimate assumes minimal usage and the smallest possible resources to complete the task for just two images. The cost will scale up if you train larger models, process more images, or use higher-tier instances.

Project Submission: 10th Nov, 2025