**Assignment 5 – Text Identification using OpenCV, Tesseract (OCR), and Deep Neural Network**

**Name:** Shubham Gautam Dahane  
**Roll No:** 382015  
**PRN:** 22310312

**Problem Statement**

Implement a text detection and recognition system using OpenCV, Tesseract OCR, and a deep neural network. The system should detect regions containing text in images and recognize the textual content. The EAST (Efficient and Accurate Scene Text detector) deep neural network is used for text detection, and Tesseract OCR is applied for recognition.

**Objectives**

* To apply deep learning for text detection using the **EAST model**.
* To perform text recognition using **Tesseract OCR**.
* To preprocess and evaluate text datasets (ICDAR 2015).
* To calculate **accuracy metrics (WER, CER)** for performance evaluation.
* To visualize detection bounding boxes and recognized text.

**Requirements**

* **Operating System:** Windows/Linux/MacOS
* **Python Version:** 3.x
* **Tools:** Jupyter Notebook / Google Colab
* **Hardware:** CPU (GPU recommended for faster processing)
* **Libraries Used:**
  + OpenCV
  + NumPy
  + Matplotlib
  + Pytesseract
  + Imutils
  + Jiwer (for WER & CER calculation)
  + HuggingFace Datasets

**Theory**

Text detection and recognition is a two-step process:

1. **Text Detection (EAST DNN Model)**
   * EAST (Efficient and Accurate Scene Text Detector) is a deep neural network trained to localize text regions in images.
   * It outputs confidence scores and geometrical data for bounding box construction.
   * Non-Maxima Suppression (NMS) is applied to refine overlapping boxes.
2. **Text Recognition (Tesseract OCR)**
   * OCR (Optical Character Recognition) extracts text from detected regions.
   * Tesseract uses **LSTM-based sequence recognition** for character-level recognition.
   * Different Page Segmentation Modes (PSM) allow flexible recognition.

**Evaluation Metrics:**

* **Word Error Rate (WER):** Fraction of words incorrectly predicted.
* **Character Error Rate (CER):** Fraction of character-level errors.

**Methodology**

1. **Dataset Acquisition**
   * Used **ICDAR 2015 OCR dataset** from HuggingFace.
   * Contains scene text images with corresponding ground truth annotations.
2. **Preprocessing**
   * Images converted from RGB → BGR for OpenCV.
   * Resized to **320×320** (multiple of 32) for EAST network.
3. **Text Detection**
   * Loaded frozen\_east\_text\_detection.pb pretrained model.
   * Extracted bounding boxes via **decode\_predictions()**.
   * Applied **Non-Maxima Suppression (NMS)** to reduce redundant boxes.
4. **Text Recognition**
   * For each bounding box region, applied **Tesseract OCR**.
   * Extracted recognized text with configurations (--oem 1, --psm 6/7).
5. **Evaluation**
   * Compared predicted text against ground truth using **WER** and **CER** from jiwer.
   * Sample visualizations of bounding boxes and predictions.
6. **Visualization**
   * Used Matplotlib to plot images with bounding boxes and recognized text.

**Graphs and Visualizations**

1. **Bounding Box Detection & Recognition**



1. **Evaluation Metrics**
   * **Word Error Rate (WER)**: Shows word-level prediction accuracy.
   * **Character Error Rate (CER)**: Shows fine-grained character accuracy.

**Advantages**

* End-to-end pipeline: Detection (EAST) + Recognition (Tesseract).
* Works on **real-world scene text images**.
* Scalable to large datasets (ICDAR, COCO-Text).
* Robust evaluation with WER & CER.

**Limitations**

* OCR results degrade with **blurred, rotated, or low-contrast text**.
* EAST requires **fixed input size (multiples of 32)**.
* Bounding box precision affects recognition quality.
* Real-time performance slower on **CPU-only systems**.

**Applications**

* **Document digitization** (scanned papers, invoices).
* **License plate recognition**.
* **Scene text recognition** (signboards, street images).
* **Assistive technology** for visually impaired users.

**Working / Algorithm**

**Step 1:** Import libraries (OpenCV, Pytesseract, NumPy, Jiwer).  
**Step 2:** Load image dataset (ICDAR 2015).  
**Step 3:** Preprocess image (resize, blob creation).  
**Step 4:** Apply EAST model → extract bounding boxes.  
**Step 5:** Apply Non-Maxima Suppression.  
**Step 6:** Extract regions of interest (ROIs).  
**Step 7:** Perform OCR with Tesseract.  
**Step 8:** Compare with ground truth → calculate WER, CER.  
**Step 9:** Visualize predictions with bounding boxes.

**Conclusion**

The implemented text identification system successfully detected and recognized text from scene images.  
Using **EAST for detection** and **Tesseract for OCR**, the model achieved reasonable performance on the **ICDAR 2015 dataset**, with **WER and CER providing accuracy benchmarks**.  
Visualization confirmed correct detection in most cases.

This project demonstrates the effectiveness of combining **deep neural networks and OCR** for robust text identification in real-world scenarios.