OCR Performance Analysis

PyTesseract OCR

PyTesseract was tested on document images containing both printed and handwritten text. The goal was to check how well it could read text, give confidence scores, and correctly locate text using bounding boxes.

The results showed that PyTesseract **struggled with most of the images**. In several cases, it **did not return any readable text** at all. When text was detected, the **output was often incorrect** and **did not match the actual content** of the image. **Confidence scores were not generated for any image**, indicating that the model was not confident about the detected text.

Bounding box analysis also showed poor performance. Many boxes were **misplaced, fragmented, or drawn over non-text regions**. Overall, PyTesseract was found to be unsuitable for handling handwritten or low-quality document images, as it relies heavily on rule-based and heuristic methods.

EasyOCR

EasyOCR was then evaluated on the same set of images to compare its performance with PyTesseract. EasyOCR **showed significantly better results**, especially in detecting and localizing printed text. **Bounding boxes were mostly accurate and well-aligned, and the recognized text for printed regions was reliable**.

For handwritten text, EasyOCR performed better than PyTesseract but **still faced some challenges**. In some cases, **spaces between letters** in a handwritten word were incorrectly interpreted as word boundaries, **causing a single word to be split into multiple words**.

Certain limitations were also observed with **text orientation and style**. **Rotated or skewed text was not detected properly**, with some words being missed or merged with nearby text. **Artistic or decorative fonts, including three-dimensional styled letters, were not detected** by the model. Bounding boxes remained accurate in most cases but failed in scenarios involving rotated or stylized text.

Need for Fine-Tuning EasyOCR

Although EasyOCR performed better than PyTesseract, the observed errors indicate that the **pre-trained model is not fully optimized for all document styles**. Fine-tuning the EasyOCR recognition model on task-specific data can help improve performance. **By training the model**

on images containing handwritten text, rotated text, and similar font styles, the OCR system can learn to better recognize such patterns.

Fine-tuning would mainly improve character recognition accuracy, spacing between letters, and robustness to variations in handwriting and orientation. This makes EasyOCR a suitable candidate for customization and further improvement, unlike PyTesseract, which does not support effective model training.