IEEE-CTSoC VIRTUAL INTERNSHIP REPORT

AUG-NOV 2024

R SANJANA

OCR tool that reads tables

PROJECT INSTRUCTORS:
Prof. Prabindh Sunderason & Prof. Pavitra Y.J

GOAL: An OCR tool utilizing Florence for reading and extracting text from tables.

OVERVIEW:

Florence-2 is a flexible model that can handle various tasks involving images and text by simply using text instructions. It can do things like generate image captions, detect objects, and understand how images are organized. Trained on a large dataset (FLD-5B) with billions of labels across millions of images, it is great at learning multiple tasks.

Florence-2 was used for OCR region recognition in table images, which helps it identify and extract structured text from tables efficiently. Among the models specified in the documentation ("microsoft/Florence-2-large · Hugging Face")microsoft/Florence-2-base-ft was used.

Instead of relying on rectangular bounding boxes, Florence 2 leverages quad boxes, which define regions using four corner points, improving its ability to interpret document structures and layouts.

Although Florence-2 performed well in recognizing standard tables from test cases, it faced challenges with images containing slight noise or poorly defined tables. In such cases, EasyOCR was incorporated

into the project to improve text recognition from the tables in those test cases.

EasyOCR is an open-source optical character recognition (OCR) tool designed to recognize text in images. It supports over 80 languages which includes key indian languages such as Hindi,Tamil, Kannada etc. It is built with deep learning models that offer high accuracy in extracting text from various types of images. EasyOCR is known for its simplicity and ease of use.

LEARNINGS:

1. Gained insights into the limitations and benefits of various models and frameworks, including PyTesseract, PaddleOCR, and EasyOCR.
2. While PaddleOCR (a framework from PaddlePaddle a deep learning model) initially appeared promising for table extraction and recognition, its compatibility issues with the latest versions led to frequent errors, highlighting the importance of framework stability in project development.
3. Learned about libraries such as:

   i)PIL (Python Imaging Library): Used for image processing tasks such as resizing, cropping, and enhancing images before feeding them into models.

   ii)Numpy: Used for data manipulation and handling arrays, particularly when working with image data and model inputs.

   iii)Transformers: Used for loading and utilizing pre-trained models, enabling seamless integration of models like Florence 2.

iv)EasyOCR: Applied for efficient text extraction from images, particularly useful for recognizing text in noisy or poorly defined tables.

4. Florence 2's resource-intensive architecture is better suited for GPU execution rather than CPU. The model's substantial computational requirements and high memory consumption can easily surpass the available RAM when running on a CPU. A Lot of memory crashes while execution were seen because of this.
5. Understanding Prompt-Based Methods: Prompt-based methods in Florence 2 involve providing text prompts or image-based cues that guide the model to perform specific tasks. These prompts help the model understand the context on what task should be performed.

   Unlike regular prompts given to any search engine these prompts aren't too precise. Which provided the model usage little more challenging to get the desired outcome.

Scope for Enhancement:

1. A model/Algorithm can be developed that automatically examines table structures, performs basic analysis, and creates summaries and visual reports. This would help identify important details, spot patterns, and give insights from large datasets, making data analysis faster and more automated.
2. Florence-2 has been trained on diverse datasets, including images from multiple languages, to perform OCR tasks. This

allows it to effectively recognize and extract text from a variety of language inputs, making it adaptable for multilingual text recognition across different scripts.

3. With some image processing concepts, can be used to also recover/restore and extract text from damaged images.
4. A potential future development could involve integrating Text-to-Speech (TTS) technology with the extracted data to make it accessible in audio formate.

REFERENCES

1. https://huggingface.co/microsoft/Florence-2-large
2. https://github.com/Sghosh1999/Computer-Vision-OCR-Florence2.
3. Bin Xiao, Haiping Wu, Weijian Xu, et al. "Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks." *arXiv preprint arXiv:2311.06242* (2023).https://doi.org/10.48550/arXiv.2311.06242
4. https://github.com/JaidedAI/EasyOCR
5. Bhatt, B. (n.d.). **EasyOCR Demo**. GitHub. https://github.com/bhattbhavesh91/easyocr-demo?tab=Apache-2.0-1-ov-file
6. PaddleOCR Documentation