Model

Text Recognition -> Image-to-text -> Tagging (NER)

Using yolov8/opencv, easyOCR, spacy

Text Recognition

OpenCV

The first task was to identify the speech bubbles. Initially I achieved this task by utilizing image processing techniques. The image is then converted to grayscale to simplify subsequent operations. To reduce noise, Gaussian blur is applied, followed by binary thresholding which highlights potential speech bubbles by converting the image into a high-contrast binary format. The code then identifies contours in the binary image, which represent the boundaries of the speech bubbles. Each contour is processed to determine its bounding rectangle, and those rectangles are filtered based on their size to identify genuine speech bubbles.

Yolov8

Later I realize that using model such as Yolov8 (a model used for detecting objects in an image/video). I finetuned it on a small dataset to recognize speech bubbles. However, since the dataset was pretty small (couldn’t find comic datasets relevant to the task) had around 100 images, the performance might be little bad but so far for the sample images it has performed decently well by identifying the speech bubbles. Further training can significantly improve its detection rate.

Yolov8 is pretty fast, OpenCV detection is little slow and produces a lot of noise. Finetuning Yolov8 on bigger dataset can improve our detection significantly.

Image-to-Text

easyOCR

Very popular image-to-text model, this model can both detect and produce image-to-text, the reason why we aren’t using it directly is that detection is slightly bad. So I try to focus different models for detecting. I have tried using tesseract for image-to-text and the results produced by it are bad, so I continued with easyOCR. One of my goal was to improve image-to-text quality so I tried finetuning it on comic text data, images of text obtained from comics. However, I found there was no improvement infact it degraded slightly. Hence, I used the English-g2 model provided by them.

Tagging(NER)

My initial goal was to use a LLM to tag characters, for that I wanted to use mistral-nemo model which could be accessed using ollama. The reason I wanted to use LLM is that the data that will be used would be fanfiction and if we have trained it on comics and other stories, it would be very easy to identify various characters, aliases, references and locations. However, using mistral-nemo through ollama took really long time to generate a response. So, I had to go with a different strategy. This time I went with a NER model with huge English vocabulary, spaCy. Using their pretrained model, I was able to identify most characters present in the sample comic images.

Future work

Given I have been using Yolov8 for detecting speech bubbles, we can extend this model to identify characters , i.e. tag people to images. In case we incorporate LLM to the present model, we will be able to combine all aliases of the person.

Few images of model functioning and findings (Scroll)

A screenshot of a computer screen

Description automatically generated

25 Epochs data (Yolov8 training, all data present in train folder)

A collage of a person's face

Description automatically generated

Yolo speech bubble detection

A screenshot of a computer code

Description automatically generated A screenshot of a computer program

Description automatically generated

Noise is more in opencv but yolo slight chance it might miss few, best approach would be run both