Report (Part 2)

Roll Number: 2022101027 Name: Debangan Mishra

Approach

The main approach for tackling part 2 was to generate synthetic data and train an object detection model on the basis of it. I have used YOLO v8 (Nano version), as it is a smaller version of a powerful object detection model. Ultralytics (the creators of YOLO) provide a library which makes it very easy to train and finetune their YOLO models. Hence, the primary task of this part was to be able to generate a dataset which matches the input distribution well enough. I did not want to create a model which is dependent on the number of digits in the image, as that would lead to a possible overfitting of the model. Hence, I created an synthetic polyMNIST dataset which contains 3, 4, 5 digits with 5%, 90%, 5% proabilities to match the given data's distribution. It is created by collating MNIST images randomly with small scaling. Note that the images are not scaled too small or too large, again, to match the input dataset distribution as the numbers are mostly the same size there. The images are placed at random locations in the 40 x 168 image.

An interesting thing observed was that this approach is highly sensitive to the dataset creation. I was initially creating the images without any overlap between the images. However, this led to a very bad accuracy (~33%) in the model as the input images have digits which are very close to each other (nearly touching at many instances), and this led to the synthetic data distribution not matching the input distribution leading to a poor performance. However, when I added an overlap with some random probability, the performance of the YOLOv8 nano dramatically increased to ~80%. Also note that Ultralytics requires the input data to be in a certain format but that was fairly simple as I was creating the data myself.

In conclusion, the approach that I used was training an object detection model on synthetic data carefully generated to match the input data distribution.

Side Note

I was also doing a Convolutional RNN approach where the convolutional encoders are followed by an RNN decoder to create a Seq2Seq approach which is typically used for tasks like OCR and Captcha Recognition as they involve identifying a sequence of images (similar to the task at hand). However, this model was not performing well despite great efforts. Even though the loss was declining, the predictions were always the same number repeated many times (for example, "1111...."). I decided to quit this approach and move to a transformer based object detector which are known to perform better.