The task I decided to tackle at SROIE 2019 competition website is to perform OCR on receipt images. The task description is to accurately recognize the text in a receipt image. The participants are required to offer a list of words recognized in the image. The task will be restricted to words comprising Latin characters and numbers only.

The dataset available on the competition website provides the coordinates for each text area and its transcription for each image. To train an accurate text recognition model, the images in its existing form could not be used for training. So using the coordinates for each text, I built a script that went through every ground truth file and cropped the text regions, and wrote a new ground truth file containing the path to that image and its transcription. Now in the process of the training process, file read times and image loading times are very important and to optimize it, lmdb data format is used. To convert our dataset into lmdb format, we use another script to convert it into an appropriate format for creating our data loading processes. Also because our data only consists of black text on a white background, I decided to train the model on single-channel images(grayscale).

For text recognition, I tested two different methods: CTC(Connectionist Temporal Classification) and attention. In OCR, using CTC for predicting labels is the de facto standard but recently, Attention has been making a lot of advancements in text-related tasks. So, I trained models with both Attention and CTC to compare their performance on the above dataset. I used ResNet and VGG as the feature extraction methods from the image and used 2 layers BiLSTM as the sequence modeling architecture. Comparing performances between using ResNet or VGG and CTC and Attention, many interesting details were noticed. The model using ResNet has better accuracy over the VGG network and trains much faster due to residual connections. Another point that was noticed that Attention models converge much faster than CTC models and have better robust performance. In the case of optimizers, the Adadelta optimizer which is generally used for OCR tasks did not converge well in this scenario and hence Adam optimizer was used. The loss function used for the attention model was CrossEntropy loss and for CTC models it was CTC loss.

On comparing my results with the test set, the best results I got were {"precision": 0.7781092244148692, "recall": 0.7404648680124224, "hmean": 0.758820457992491}. This could have been improved further by using more data augmentations like color jitter, hue saturation change, noising and denoising images, etc, using more hidden layers in BiLSTM, using a bigger feature extractor, training for more time(training was done on google colab).