COL774: Assignment 4 Report

Aniruddha Chatterjee 2021MT10896

Pramsu Shrivastava 2021EE10140

Non-Competitive Part

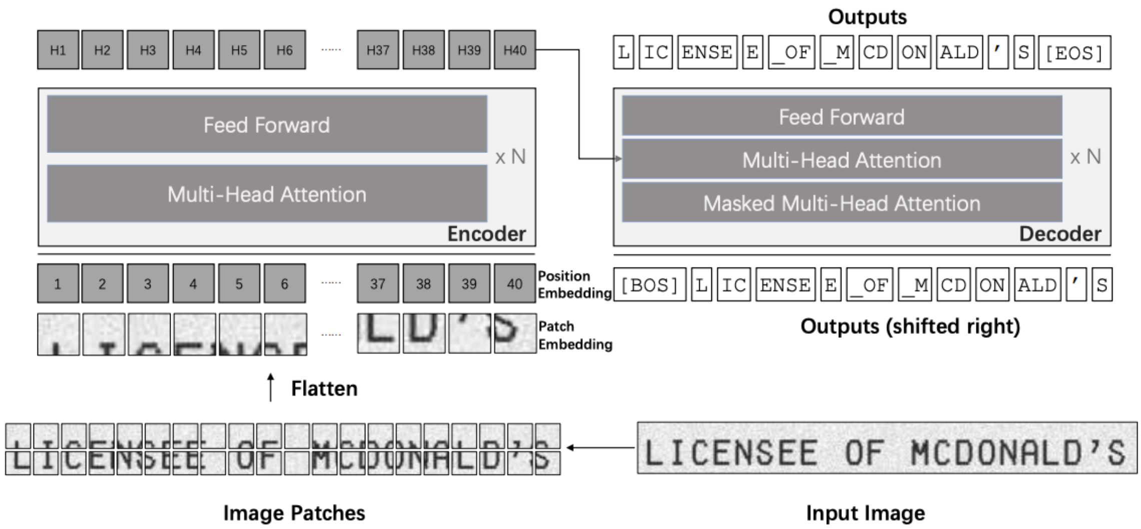
We create an Encoder-Decoder model in our EncDec class, according to the architecture given in the document.

- For utilizing the context vectors obtained from encoding the image data, we concatenate it into the decoder embedding, which is then passed to the LSTM.
- For training the model, we used cross entropy loss on our prediction tensor stack with the target tensor formed from the formula sequence.
- Within the resource and time constraints, we were able to run our model for 3 epochs through the synthetic train dataset.
- Using PyTorch’s DataLoader, we convert the images into their tensors one batch at a time, rather than dumping all 75000 into arrays. This saves computational load.
- Our model has been saved externally after each epoch, we will be loading this pre-trained model for generating CSVs.

BLEU Scores:

1. Synthetic Test Score: 0.003047054334382426
2. Synthetic Validation Score: 0.00878743926072268
3. Handwritten Validation Score: 0.00460340116158603

Competitive Part



In this part, we are going to fine-tune a pre-trained TrOCR model on the Handwriting Data training data.

TrOCR is a Transformer-Based Optical Character Recognition.

TrOCR is an instance of this, as it has an encoder-decoder architecture, with the weights of the encoder initialized from a pre-trained BEiT(Bidirectional Encoder representation from Image Transformers) and the weights of the decoder initialized from a pre-trained RoBERTa(Robustly Optimized BERT Pre-training Approach). The weights of the cross-attention layer were randomly initialized before the authors pre-trained the model further on millions of (partially synthetic) annotated images of handwritten text.

We fine-tuned the model using native PyTorch. Due to constrained time, we were able to train it for three epochs only.

We obtained a BLEU score of 0.25656 on the private dataset on the Kaggle competition.

We used the information from this repo in order to train our model - https://github.com/NielsRogge/Transformers-Tutorials/tree/master/TrOCR

The trained model can be found at this link (saved using torch.save) - https://drive.google.com/file/d/1-25FFNiWPoWlVINLA5mH8MyfvwGjGJTl/view?usp=sharing

References:-

TrOCR paper: https://arxiv.org/abs/2109.10282

TrOCR documentation: https://huggingface.co/transformers/master/model_doc/trocr.html