COL774 Assignment 4
Converting Handwritten equations to Latex Code

Vaibhav Seth - 2021MT10236
Harsh Vora - 2021CS10548

1 Non Competetive Part

1.1 BLEU Scores over Pre-trained

	BLEU Score
Validation Set : Handwritten Dataset	0.01
Testing Set : Synthetic Dataset	0.223
Validation Set : Synthetic Dataset	0.23

1.2 BLEU Scores over Fine Tuned

	BLEU Score
Validation Set : Handwritten Dataset	0.23
Testing Set : Synthetic Dataset	0.017
Validation Set : Synthetic Dataset	0.02

The model that we have used is a CNN-LSTM Encoder-Decoder architecture. The Architecture is as specified in the assignment statement. We have restricted the label sizes to 128 for pre-training the LSTM. There are around 2 per cent examples that have length greater than 128 so we can ignore them without losing much data. 0.5 teacher forcing ratio was used in all models. We used a batch size of 256 for training the model. All the models were trained on a NVIDIA A100 (40GB) GPU availble on the High Performance Computing servers at IIT Delhi.

2 Competitive Part

2.1 Architecture

We modified our original CNN-LSTM model to create our final model. We replaced the original Encoder CNN with ResNest-50[2] architecure. The idea was to use a pre-trained encoder with support for attention mechanism and we found ResNest-50 to be extremely useful. The second modification that we made was adding Attention to the LSTM Decoder as well. We used a version of Bahdanau Attention[1] to increase the predictive power of the model. We pass the output of the encoder layer as the initial hidden states of the LSTM. The input at each time step is given by concatenating the Encoder outputs, Embedding vector, and the attention vector. Here, we also make one change from the previous model in using a hidden dimension of 1000. The predivtive power of the model was significantly boosted. We went from 0.29 on the kaggle leaderboard to 0.56 by using the improved model. We would have liked to experiment with slightly different architectures by varying the hidden dimension size, or using multiple layered LSTM, but we couldn't due to a lack of time. After fine-tuning usign validation scores, we achieved a BLEU Score of **0.5919** on the public leaderbaord on Kaggle. Here also we restricted the sequence length during training to be 128 and used a batch size of 128.

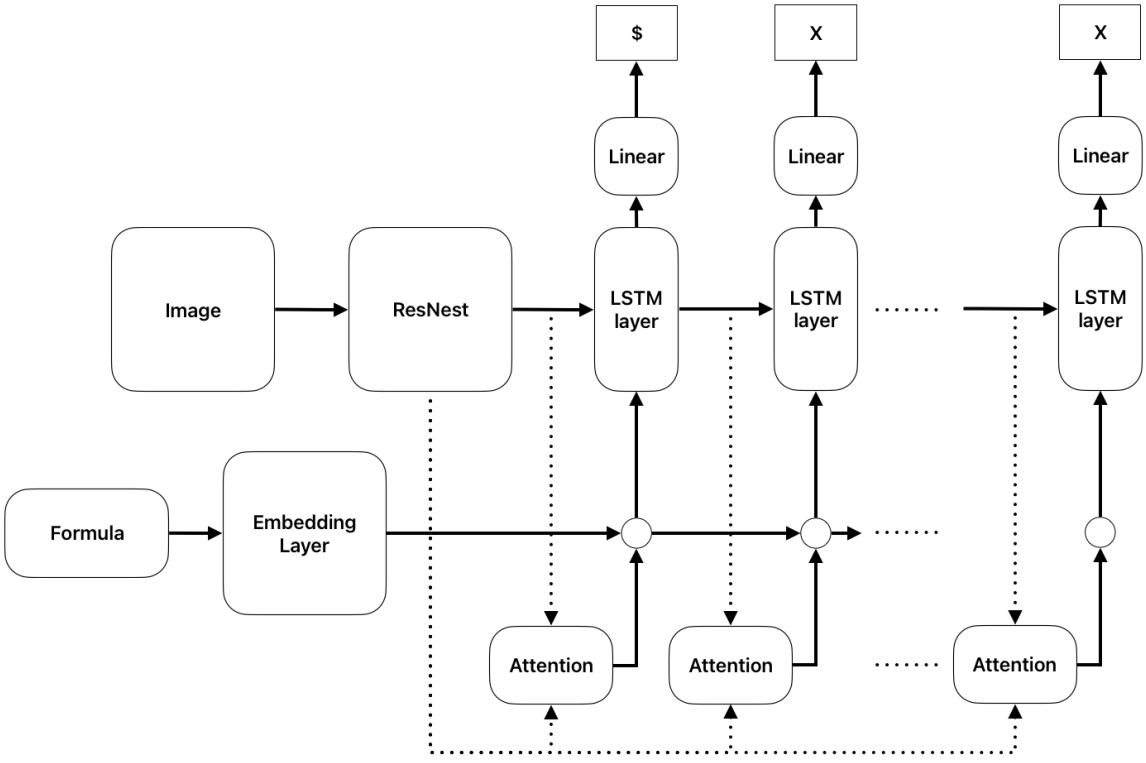


Figure 1: Diagram of the LatexResNest Model

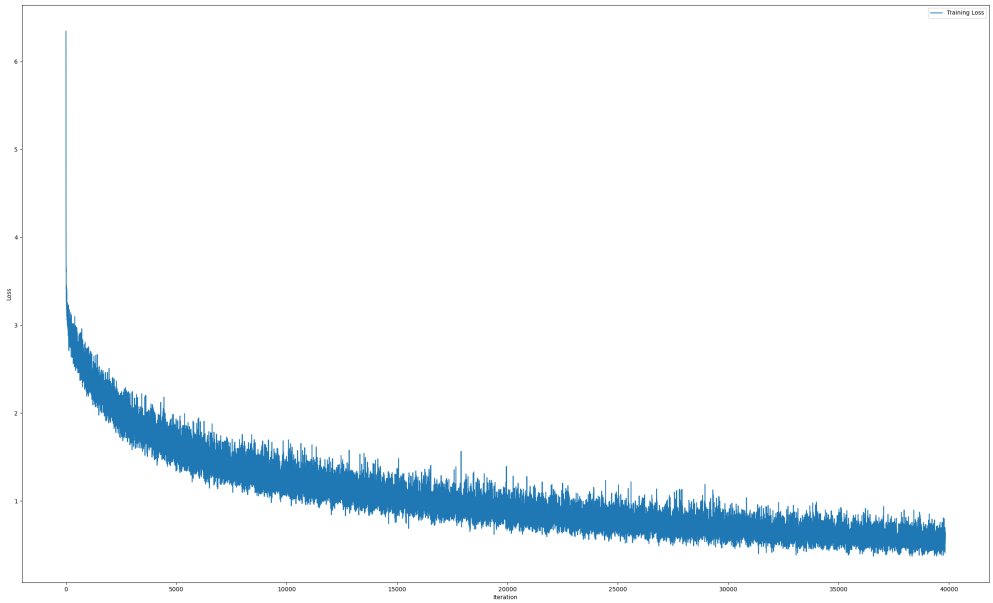


Figure 2: Training Loss for Pre-Training

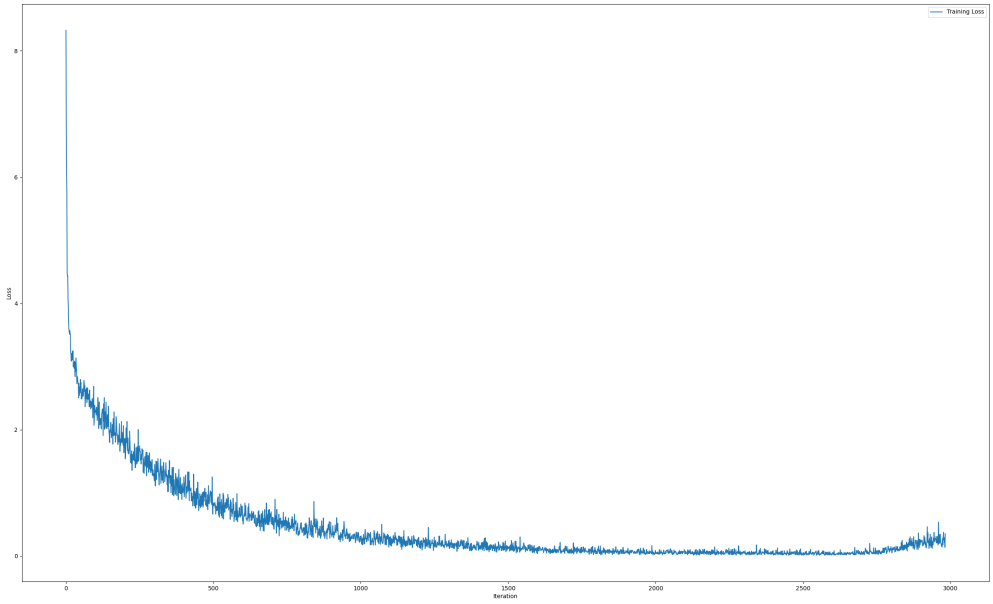
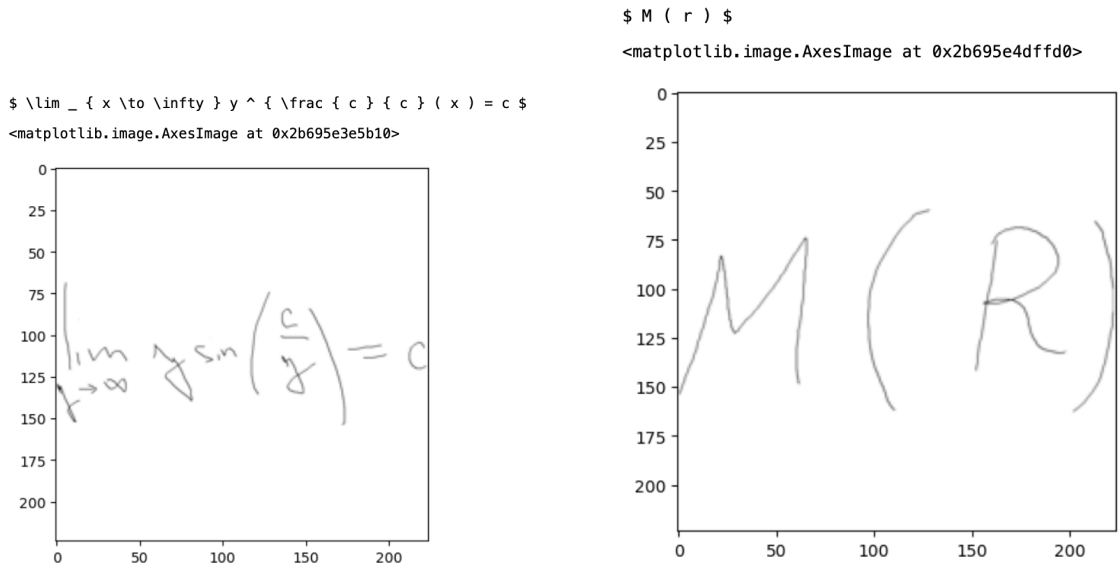


Figure 3: Training Loss for Fine-tuning over 68 epoch model



(a) Max depth = 15

(b) Max depth = 25

Figure 4: Example predictions with images-1

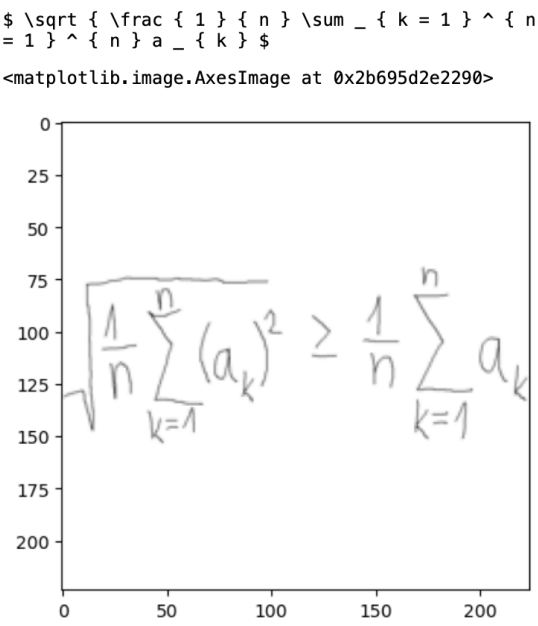


Figure 5: Example predictions with images-2

```
$ \sqrt { 1 + \sqrt { 2 + \sqrt { 3 + \sqrt 4 } } } $
```

```
<matplotlib.image.AxesImage at 0x2b695e7e0a10>
```

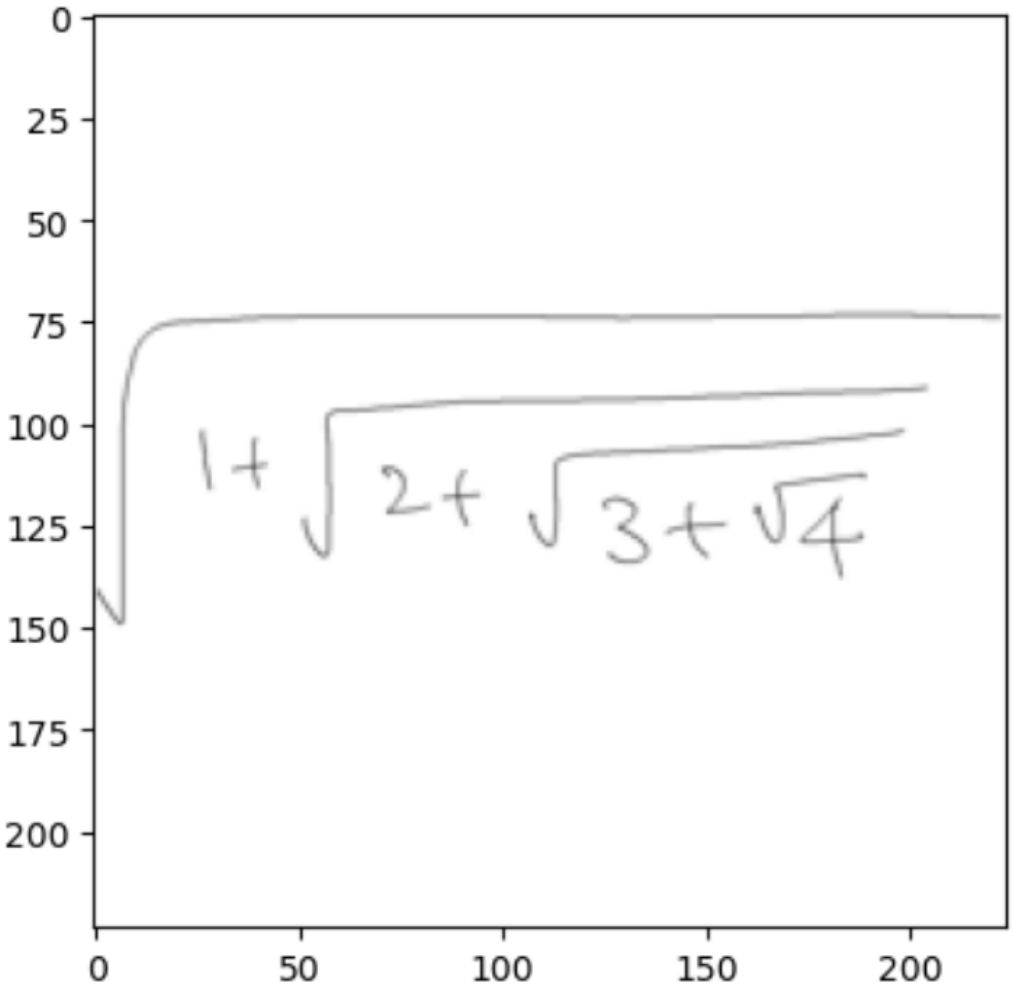


Figure 6: Example predictions with images-3

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[2] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks, 2020.