Project Name: Photo Caption Generation

The model architecture is as below. The model has three parts.

1. Part 1: CNN which is used for image feature extraction. This is a 16-layer VGG model excluding the last prediction layer. Hence for feature extraction only the first 15 layers are used. The output is a feature matrix (1x4096) for each of the input images.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
Total params:		134,260,544
Trainable params:		134,260,544
Non-trainable params:		0

2. Part 2: RNN which is used for caption generation.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 34, 256)	1940224	input_2[0][0]
dropout_1 (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 256)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 256)	1048832	dropout_1[0][0]
lstm_1 (LSTM)	(None, 256)	525312	dropout_2[0][0]
add_1 (Add)	(None, 256)	0	dense_1[0][0], lstm_1[0][0]
dense_2 (Dense)	(None, 256)	65792	add_1[0][0]
dense_3 (Dense)	(None, 7579)	1947803	dense_2[0][0]
Total params:		5,527,963	
Trainable params:		5,527,963	
Non-trainable params:		0	

3. Part 3: Part 1 and Part 2 are merged to make a final prediction i.e. to generate a caption for any new image.

Model Architecture Diagram:

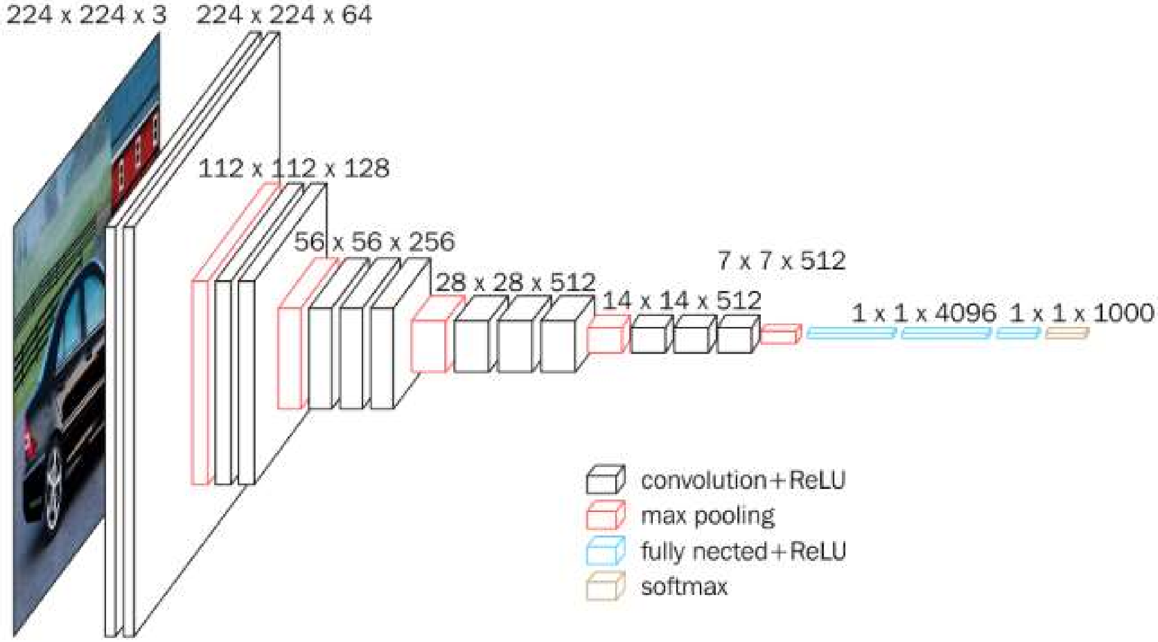


Figure 1: CNN

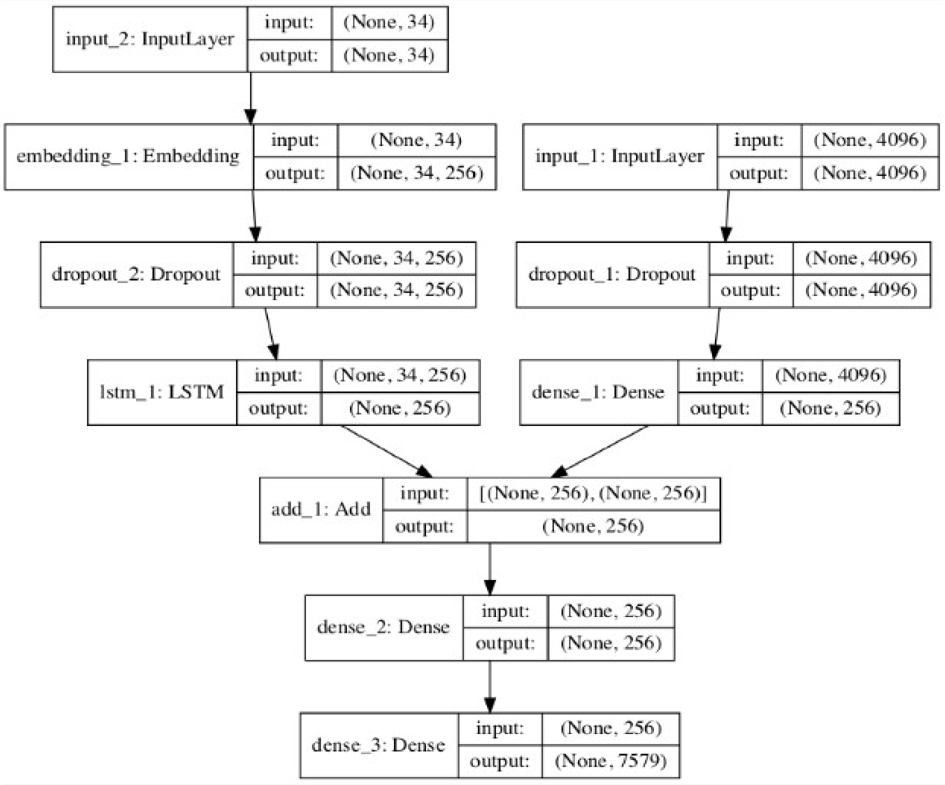
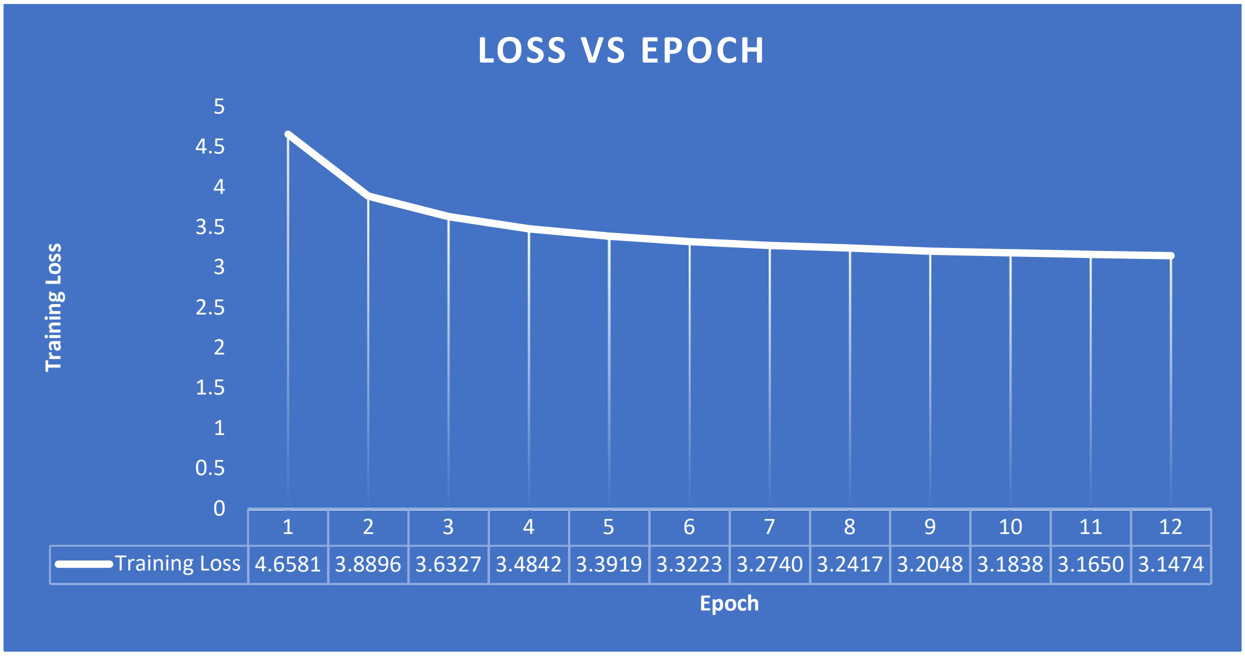


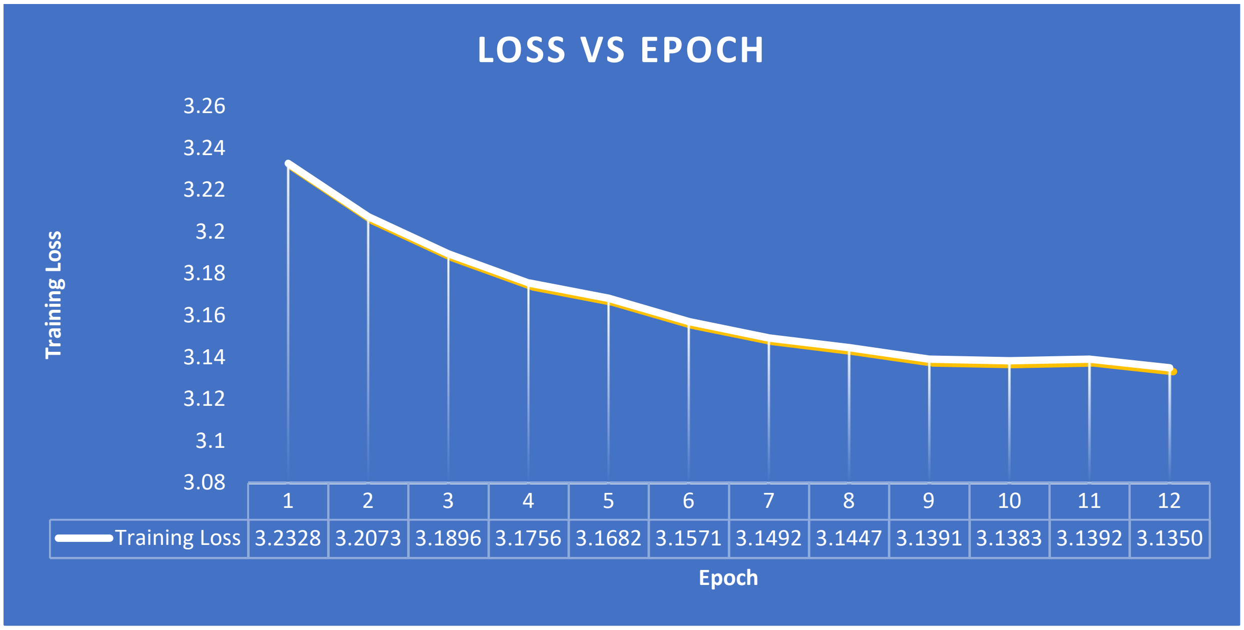
Figure 2: Caption Model with Image feature input

Model Training (CPU Only): Dropout probability=0.5



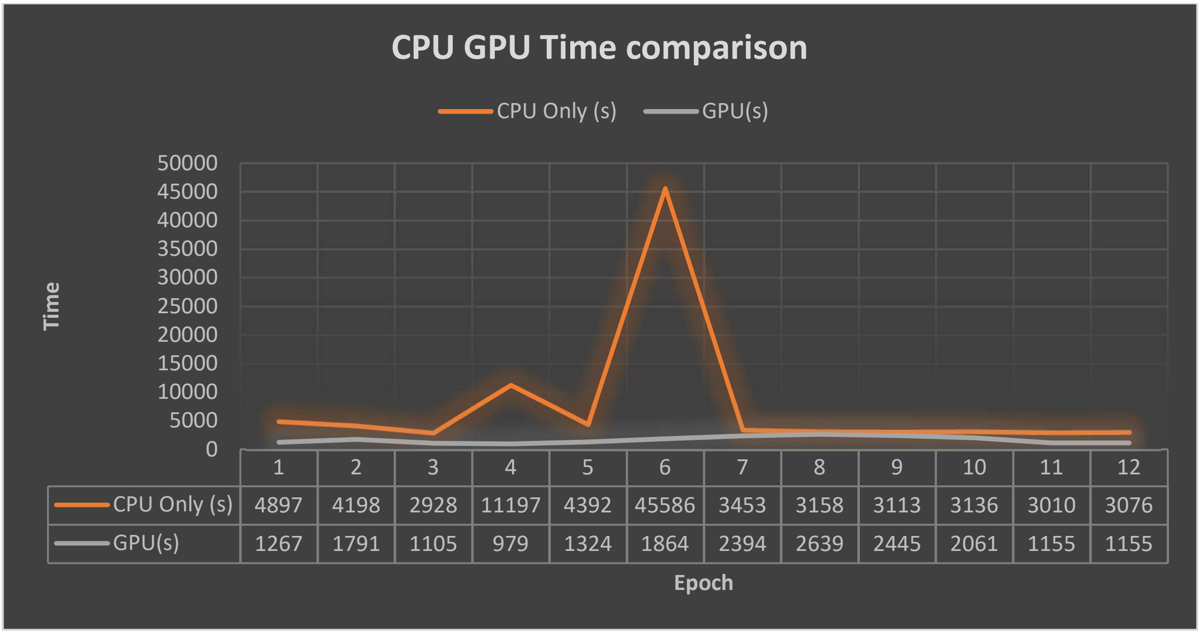
	1	2	3	4	5	6	7	8	9	10	11	12
Training Loss	4.6581	3.8896	3.6327	3.4842	3.3919	3.3223	3.2740	3.2417	3.2048	3.1838	3.1650	3.1474

Model Training (CPU+ GPU): Dropout probability=0.5



	1	2	3	4	5	6	7	8	9	10	11	12
Training Loss	3.2328	3.2073	3.1896	3.1756	3.1682	3.1571	3.1492	3.1447	3.1391	3.1383	3.1392	3.1350

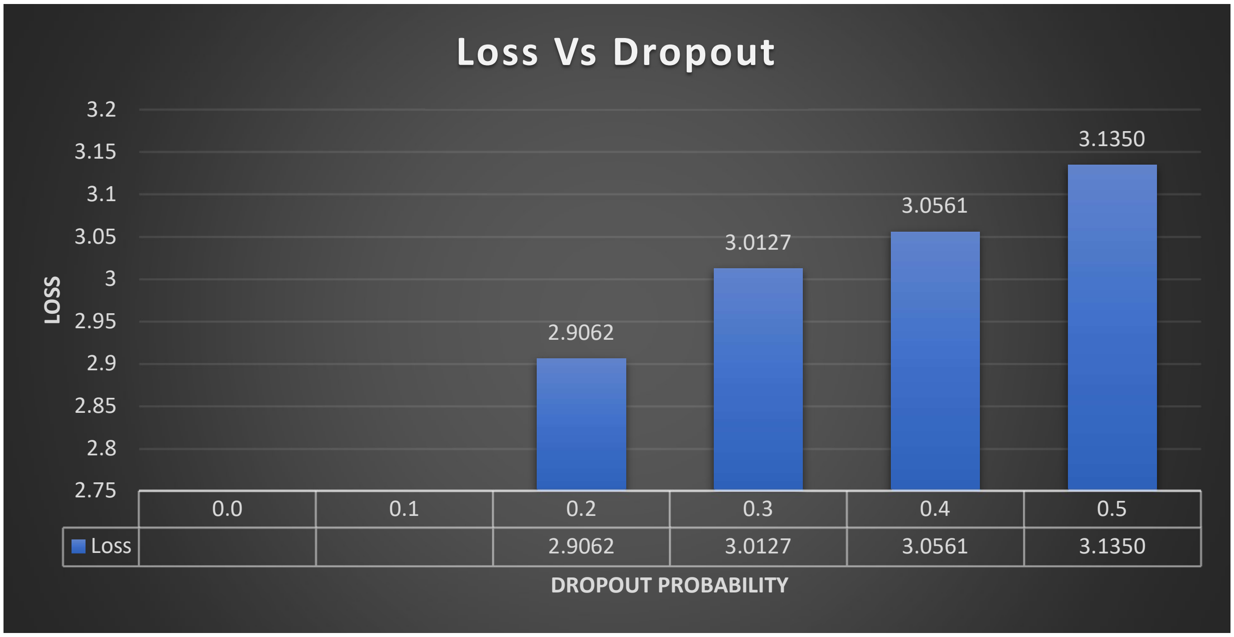
The execution time of 12 epochs drops from 25.6 hours (CPU only) to 5.6 hours (with GPU) which is a 4.57 times improvement. Below graph shows execution time over the epochs for CPU (only) and GPU.



	1	2	3	4	5	6	7	8	9	10	11	12
CPU Only (s)	4897	4198	2928	11197	4392	45586	3453	3158	3113	3136	3010	3076
GPU(s)	1267	1791	1105	979	1324	1864	2394	2639	2445	2061	1155	1155

Loss variation with Dropout Probability (CPU+ GPU):

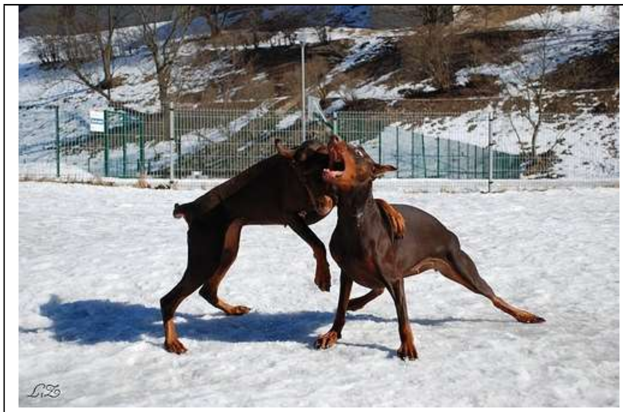
As expected the training loss is found to be increasing with increase in dropout probability.



	0.0	0.1	0.2	0.3	0.4	0.5
■ Loss			2.9062	3.0127	3.0561	3.1350

Test Results:

The prediction is done using model from epoch 11[model_10.h5]. We observed the Loss to stabilize after 11th epoch hence used the same model.

 Dropout Probability: 0.5	**True Captions:** 1. the dogs are in the snow in front of fence 2. the dogs play on the snow 3. two brown dogs playfully fight in the snow 4. two brown dogs wrestle in the snow 5. two dogs playing in the snow **Predicted Caption:** dog is running through the snow **BLEU Score:** BLEU-1: 0.520361 BLEU-2: 0.264667 BLEU-3: 0.179006 BLEU-4: 0.077796
Dropout Probability: 0.2	**Predicted Caption:** dog is jumping over hurdle **BLEU Score:** BLEU-1: 0.515793 BLEU-2: 0.272046 BLEU-3: 0.182864 BLEU-4: 0.082574

Details to execute the code:

Please access the below path to execute and verify the code. The required permission is provided to the folder. The input files and models generated are present in the folder.

/home/students/kollive/Project