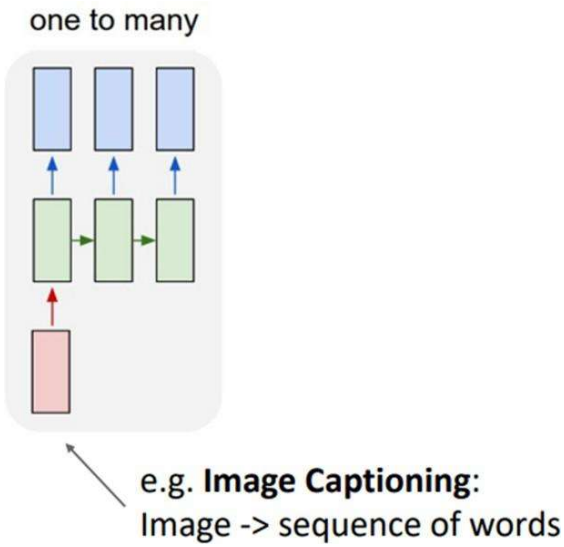
Project Report

Image Caption Generator using CNN- LSTM

Introduction:

This project is based upon image captioning using CNN-LSTM deep learning model. CNN-LSTM model is a more usable and efficient way which uses a hybrid mode that comes from CNN and LSTM. In case of image captioning this hybrid, model is used in two tire manners. The image captioning problem is formulated as a one-to-many prediction model sequence. For this purpose, the LSTM prediction model is used to understand the image. In practice the RGB image tensor is not processed since they are not accurately workable. Hence the normal Vanilla LSTM cannot be used for prediction.



The feature extraction of input image is done deep CNN architecture where the LSTM architecture is used for the final output of the caption.

The main motive for image captioning is to generate a proper readable text captions for humans from given input images. Generally neural network models are sub-categorized into two different parts.

a. Extraction of the features from image
b. Processing and modelling the language

Algorithms:

Since the image captioning uses hybrid deep learning, there are two learning algorithms involved i.e. Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) which falls under Recreational Neural Network (RNN).

i. Convolutional Neural Network (CNN)

CNN processes the data having the input shape similar to a three-dimensional matrix. CNN model has many layers including input layer, Convo Layer, Pooling Layer, fully connected layers, SoftMax, and Output layers. In this case it takes in an image as input, assigns importance (learnable weights and biases) to various aspects/objects in the image, and distinguishes between them. As a result, CNN's input layer is an image. A 3D matrix is used to display the image data. When a pooling layer is used, the size of the image is decreased after the convolution layer has been applied. Completely connected layers, which involve neurons, biases, and weights, are a type of connection layer that links one neuron in one layer to some other neuron in a different layer. Objects are categorized using a method employing the SoftMax layer, which is utilized for multi-classification.

ii. Long-Short Term Memory

LSTM models are generally used as sequence prediction models. It has the same work in our proposed architecture. LSTM has proved to be more efficient than other RNN models because of the short time memory period. Besides that, LSTM uses an overlook gate to determine the correct statistics and eliminate the rest in the processing of the input. In other words it grasps the required information from the processing of inputs as well as forget gate and also it does remove the non-required data.

Objective Functions:

For our proposed model we have used a cross entropy loss function and in the final stage, BLEU scores are used to measure the performance which outputs the result. With this model, Bleu-score is utilized. We can determine whether n-gram will produce captions for the given data utilizing BLEU Score. The BLEU Score ranges from 0 to 1. A proposed translation of the text is compared with one or more equating to determine its BLEU score, or Bilingual Evaluation

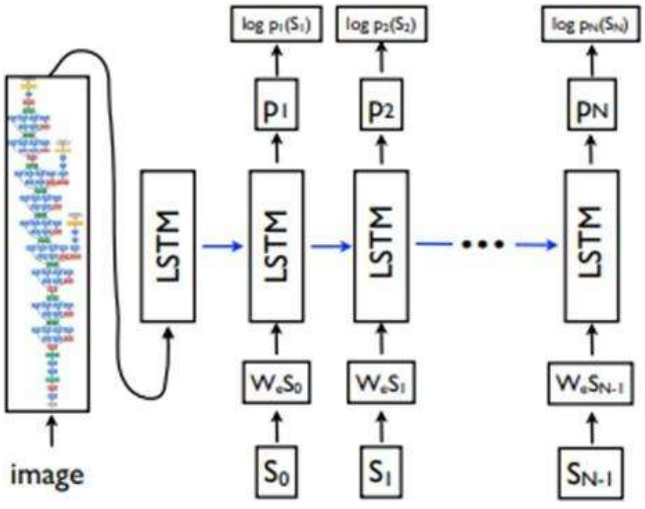
Flowchart of the working application:

The flowchart of the working application is as follows.

- Importing the required libraries.
- Establishing the GPU memory's training-related settings
- Bringing in collection of images and their corresponding captions
- Plotting these few dataset photos and related captions
- Improving captions for more refined analysis
- Clearing the captions so they can be used later
- Extracting features
- Plotting related pictures out from collection
- Tokenizing the captions for further processing
- Preprocessing the captions and photos in accordance with the model's specifications
- Building the LSTM model
- Training the LSTM model
- Plotting the cross-entropy loss value
- Generating captions
- Utilize BLEU scores to measure the performance

Network Architecture:

Our proposed model uses an architecture which is relatable to encoder-decoder model architecture which basically uses an encoded vector as input and outputs valid/proper captions. This type of architecture further can be related to both natural language processing and computer vision. The encoder element entails utilizing the CNN model to create an encoded vector of the input image that needs a caption. Feature extraction from of the image is primarily accomplished in this.



The proposed architecture can be divided into three parts, features extractions, tokenization, and prediction model.

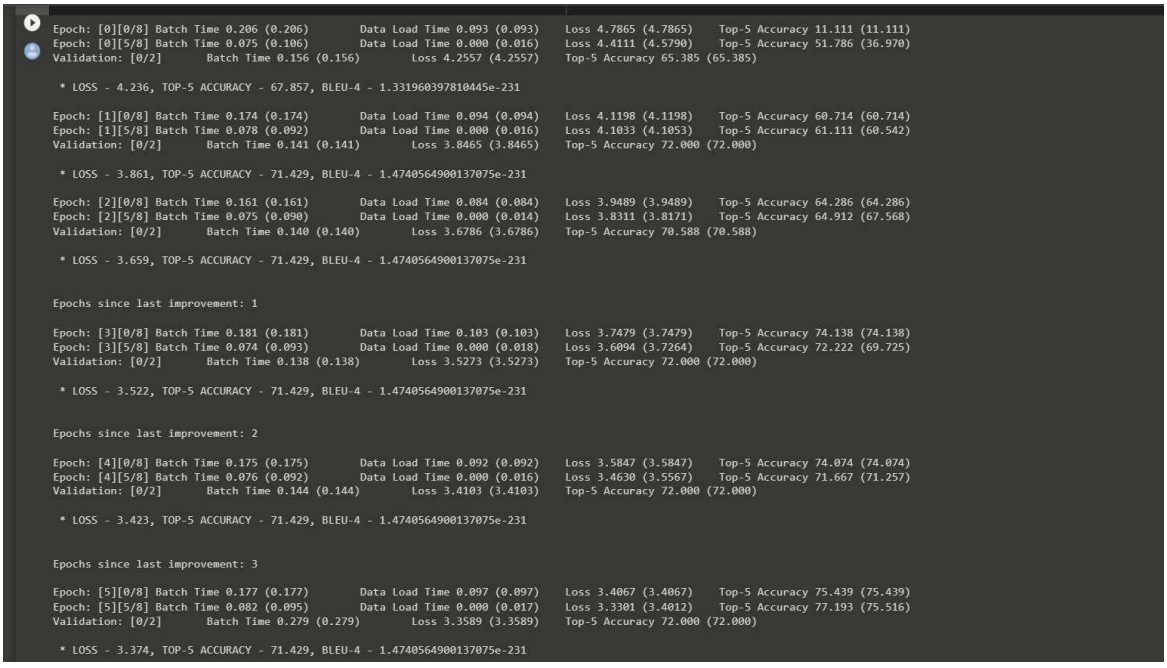
Features Extraction: This is the preprocessing stage in which the input image is fed as the input to CNN model which outputs an encoded vector. Vector features, also referred to as embeddings, are produced. The CNN model captures features from the original images, which are then compressed into lower feature vectors that are RNN-compatible. Consequently, it is referred known as the encoder.

Tokenization: RNN, which was given the feature vectors by CNN, is the system's next stage and decodes them. Thus, the captions are produced while the word order is forecasted.

Prediction Model: This is the final stage of the application. Here the vectors are decoded, and the final output is being generated using the BLEU score.

Results:

Specific to our model we have used Cross entropy as loss function and the data set is trained on the loss. The final caption is generated using the BLEU score. The dataset is split into 90% training and 10% test and 20 mini batches are used. We have used transfer learning to finetune the dataset and model evaluation. Following are the results generated.



```
Epochs since last improvement: 4

Epoch: [6][0/8] Batch Time 0.169 (0.169)        Data Load Time 0.091 (0.091)    Loss 3.3249 (3.3249)    Top-5 Accuracy 76.364 (76.364)
Epoch: [6][5/8] Batch Time 0.076 (0.094)        Data Load Time 0.000 (0.017)    Loss 3.3653 (3.3751)    Top-5 Accuracy 73.684 (74.627)
Validation: [0/2]       Batch Time 0.143 (0.143)        Loss 3.3501 (3.3501)    Top-5 Accuracy 68.627 (68.627)

 * LOSS - 3.295, TOP-5 ACCURACY - 71.429, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 5

Epoch: [7][0/8] Batch Time 0.179 (0.179)        Data Load Time 0.103 (0.103)    Loss 3.3612 (3.3612)    Top-5 Accuracy 71.930 (71.930)
Epoch: [7][5/8] Batch Time 0.074 (0.093)        Data Load Time 0.000 (0.018)    Loss 3.1249 (3.2777)    Top-5 Accuracy 82.143 (76.205)
Validation: [0/2]       Batch Time 0.139 (0.139)        Loss 3.2490 (3.2490)    Top-5 Accuracy 72.000 (72.000)

 * LOSS - 3.253, TOP-5 ACCURACY - 71.429, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 6

Epoch: [8][0/8] Batch Time 0.201 (0.201)        Data Load Time 0.114 (0.114)    Loss 3.3466 (3.3466)    Top-5 Accuracy 69.091 (69.091)
Epoch: [8][5/8] Batch Time 0.075 (0.096)        Data Load Time 0.000 (0.019)    Loss 3.3768 (3.1951)    Top-5 Accuracy 77.358 (76.716)
Validation: [0/2]       Batch Time 0.142 (0.142)        Loss 3.2402 (3.2402)    Top-5 Accuracy 76.471 (76.471)

 * LOSS - 3.244, TOP-5 ACCURACY - 75.000, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 7

Epoch: [9][0/8] Batch Time 0.182 (0.182)        Data Load Time 0.095 (0.095)    Loss 3.2462 (3.2462)    Top-5 Accuracy 73.585 (73.585)
Epoch: [9][5/8] Batch Time 0.076 (0.094)        Data Load Time 0.000 (0.016)    Loss 3.1336 (3.2113)    Top-5 Accuracy 74.576 (74.096)
Validation: [0/2]       Batch Time 0.204 (0.204)        Loss 3.2225 (3.2225)    Top-5 Accuracy 71.154 (71.154)

 * LOSS - 3.194, TOP-5 ACCURACY - 73.214, BLEU-4 - 1.4740564900137075e-231
```

```
Epoch: [15][0/8]        Batch Time 0.175 (0.175)        Data Load Time 0.093 (0.093)    Loss 3.0575 (3.0575)    Top-5 Accuracy 80.702 (80.702)
Epoch: [15][5/8]        Batch Time 0.075 (0.097)        Data Load Time 0.000 (0.016)    Loss 2.8035 (2.9413)    Top-5 Accuracy 84.211 (79.822)
Validation: [0/2]       Batch Time 0.142 (0.142)        Loss 3.1214 (3.1214)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.082, TOP-5 ACCURACY - 85.714, BLEU-4 - 4.707681607020777e-155


Epochs since last improvement: 1

Epoch: [16][0/8]        Batch Time 0.193 (0.193)        Data Load Time 0.103 (0.103)    Loss 2.8837 (2.8837)    Top-5 Accuracy 78.571 (78.571)
Epoch: [16][5/8]        Batch Time 0.075 (0.096)        Data Load Time 0.000 (0.018)    Loss 2.8665 (2.9350)    Top-5 Accuracy 85.185 (78.806)
Validation: [0/2]       Batch Time 0.142 (0.142)        Loss 3.0621 (3.0621)    Top-5 Accuracy 86.275 (86.275)

 * LOSS - 3.061, TOP-5 ACCURACY - 85.714, BLEU-4 - 4.799135604588869e-155


Epochs since last improvement: 2

Epoch: [17][0/8]        Batch Time 0.177 (0.177)        Data Load Time 0.088 (0.088)    Loss 3.0518 (3.0518)    Top-5 Accuracy 75.000 (75.000)
Epoch: [17][5/8]        Batch Time 0.077 (0.093)        Data Load Time 0.000 (0.015)    Loss 3.0464 (2.9173)    Top-5 Accuracy 71.698 (78.078)
Validation: [0/2]       Batch Time 0.142 (0.142)        Loss 3.0826 (3.0826)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.053, TOP-5 ACCURACY - 85.714, BLEU-4 - 4.707681607020777e-155


Epochs since last improvement: 3

Epoch: [18][0/8]        Batch Time 0.176 (0.176)        Data Load Time 0.099 (0.099)    Loss 2.8702 (2.8702)    Top-5 Accuracy 80.000 (80.000)
Epoch: [18][5/8]        Batch Time 0.077 (0.093)        Data Load Time 0.000 (0.017)    Loss 2.7761 (2.9312)    Top-5 Accuracy 81.818 (76.596)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 3.0436 (3.0436)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.003, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.180198600618692e-78

Epoch: [19][0/8]        Batch Time 0.158 (0.158)        Data Load Time 0.082 (0.082)    Loss 2.6387 (2.6387)    Top-5 Accuracy 89.474 (89.474)
Epoch: [19][5/8]        Batch Time 0.075 (0.097)        Data Load Time 0.000 (0.015)    Loss 2.8580 (2.8789)    Top-5 Accuracy 76.471 (77.644)
Validation: [0/2]       Batch Time 0.139 (0.139)        Loss 3.0261 (3.0261)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 3.014, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.180198600618692e-78


Epochs since last improvement: 1

Epoch: [20][0/8]        Batch Time 0.169 (0.169)        Data Load Time 0.092 (0.092)    Loss 2.9413 (2.9413)    Top-5 Accuracy 77.586 (77.586)
Epoch: [20][5/8]        Batch Time 0.079 (0.093)        Data Load Time 0.008 (0.017)    Loss 2.6866 (2.8694)    Top-5 Accuracy 85.455 (77.812)
Validation: [0/2]       Batch Time 0.139 (0.139)        Loss 3.0345 (3.0345)    Top-5 Accuracy 84.314 (84.314)
```

```
DECAYING learning rate.
The new learning rate is 0.000320

Epoch: [10][0/8]        Batch Time 0.182 (0.182)        Data Load Time 0.105 (0.105)    Loss 3.2448 (3.2448)    Top-5 Accuracy 69.643 (69.643)
Epoch: [10][5/8]        Batch Time 0.075 (0.093)        Data Load Time 0.000 (0.019)    Loss 3.2076 (3.1399)    Top-5 Accuracy 74.545 (74.925)
Validation: [0/2]       Batch Time 0.148 (0.148)        Loss 3.1881 (3.1881)    Top-5 Accuracy 78.000 (78.000)

 * LOSS - 3.180, TOP-5 ACCURACY - 76.786, BLEU-4 - 1.4584558130772648e-231


Epochs since last improvement: 9

Epoch: [11][0/8]        Batch Time 0.163 (0.163)        Data Load Time 0.083 (0.083)    Loss 3.1642 (3.1642)    Top-5 Accuracy 77.358 (77.358)
Epoch: [11][5/8]        Batch Time 0.076 (0.091)        Data Load Time 0.000 (0.014)    Loss 3.0887 (3.0369)    Top-5 Accuracy 77.778 (78.788)
Validation: [0/2]       Batch Time 0.150 (0.150)        Loss 3.1780 (3.1780)    Top-5 Accuracy 78.846 (78.846)

 * LOSS - 3.157, TOP-5 ACCURACY - 80.357, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 10

Epoch: [12][0/8]        Batch Time 0.171 (0.171)        Data Load Time 0.095 (0.095)    Loss 3.1728 (3.1728)    Top-5 Accuracy 76.596 (76.596)
Epoch: [12][5/8]        Batch Time 0.077 (0.092)        Data Load Time 0.000 (0.016)    Loss 3.4569 (3.1179)    Top-5 Accuracy 65.455 (75.602)
Validation: [0/2]       Batch Time 0.318 (0.318)        Loss 3.1827 (3.1827)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.134, TOP-5 ACCURACY - 85.714, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 11

Epoch: [13][0/8]        Batch Time 0.364 (0.364)        Data Load Time 0.167 (0.167)    Loss 3.0801 (3.0801)    Top-5 Accuracy 78.431 (78.431)
Epoch: [13][5/8]        Batch Time 0.084 (0.137)        Data Load Time 0.000 (0.030)    Loss 3.1463 (3.0180)    Top-5 Accuracy 73.214 (75.893)
Validation: [0/2]       Batch Time 0.138 (0.138)        Loss 3.1581 (3.1581)    Top-5 Accuracy 82.353 (82.353)

 * LOSS - 3.123, TOP-5 ACCURACY - 82.143, BLEU-4 - 1.4740564900137075e-231


Epochs since last improvement: 12

Epoch: [14][0/8]        Batch Time 0.178 (0.178)        Data Load Time 0.099 (0.099)    Loss 2.9875 (2.9875)    Top-5 Accuracy 76.000 (76.000)
Epoch: [14][5/8]        Batch Time 0.080 (0.093)        Data Load Time 0.000 (0.017)    Loss 3.0834 (3.0563)    Top-5 Accuracy 75.862 (75.380)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 3.1368 (3.1368)    Top-5 Accuracy 82.353 (82.353)

 * LOSS - 3.099, TOP-5 ACCURACY - 82.143, BLEU-4 - 4.799135604588869e-155
```

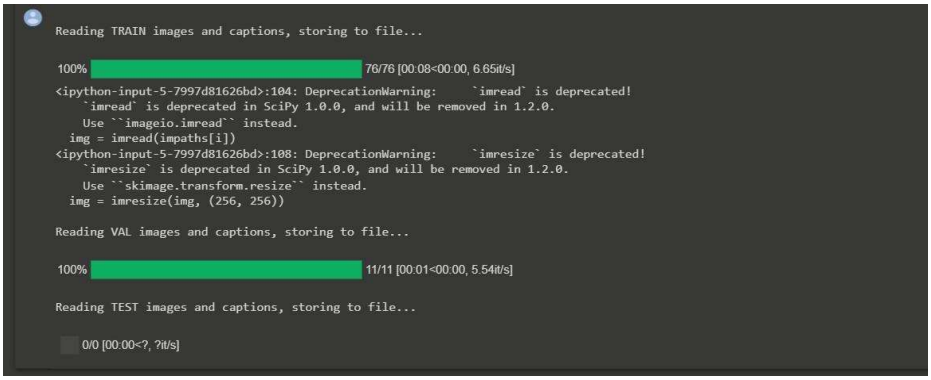
```
Epochs since last improvement: 3

Epoch: [18][0/8]        Batch Time 0.176 (0.176)        Data Load Time 0.099 (0.099)    Loss 2.8702 (2.8702)    Top-5 Accuracy 80.000 (80.000)
Epoch: [18][5/8]        Batch Time 0.077 (0.093)        Data Load Time 0.000 (0.017)    Loss 2.7761 (2.9312)    Top-5 Accuracy 81.818 (76.596)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 3.0436 (3.0436)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.003, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.180198600618692e-78

Epoch: [19][0/8]        Batch Time 0.158 (0.158)        Data Load Time 0.082 (0.082)    Loss 2.6387 (2.6387)    Top-5 Accuracy 89.474 (89.474)
Epoch: [19][5/8]        Batch Time 0.075 (0.097)        Data Load Time 0.000 (0.015)    Loss 2.8580 (2.8789)    Top-5 Accuracy 76.471 (77.644)
Validation: [0/2]       Batch Time 0.139 (0.139)        Loss 3.0261 (3.0261)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 3.014, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.180198600618692e-78


Epochs since last improvement: 1

Epoch: [20][0/8]        Batch Time 0.169 (0.169)        Data Load Time 0.092 (0.092)    Loss 2.9413 (2.9413)    Top-5 Accuracy 77.586 (77.586)
Epoch: [20][5/8]        Batch Time 0.079 (0.093)        Data Load Time 0.008 (0.017)    Loss 2.6866 (2.8694)    Top-5 Accuracy 85.455 (77.812)
Validation: [0/2]       Batch Time 0.139 (0.139)        Loss 3.0345 (3.0345)    Top-5 Accuracy 84.314 (84.314)

 * LOSS - 3.003, TOP-5 ACCURACY - 85.714, BLEU-4 - 4.799135604588869e-155


Epochs since last improvement: 2

Epoch: [21][0/8]        Batch Time 0.160 (0.160)        Data Load Time 0.084 (0.084)    Loss 2.9720 (2.9720)    Top-5 Accuracy 72.549 (72.549)
Epoch: [21][5/8]        Batch Time 0.075 (0.089)        Data Load Time 0.000 (0.014)    Loss 2.8796 (2.7912)    Top-5 Accuracy 74.074 (79.104)
Validation: [0/2]       Batch Time 0.138 (0.138)        Loss 2.9967 (2.9967)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 3.008, TOP-5 ACCURACY - 85.714, BLEU-4 - 5.7592918561109494e-155


Epochs since last improvement: 3

Epoch: [22][0/8]        Batch Time 0.254 (0.254)        Data Load Time 0.166 (0.166)    Loss 2.7921 (2.7921)    Top-5 Accuracy 76.786 (76.786)
Epoch: [22][5/8]        Batch Time 0.074 (0.105)        Data Load Time 0.000 (0.028)    Loss 2.9389 (2.7893)    Top-5 Accuracy 75.926 (79.697)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 2.9670 (2.9670)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 2.964, TOP-5 ACCURACY - 85.714, BLEU-4 - 5.810039418610042e-155


Epochs since last improvement: 4
```

```
Epochs since last improvement: 2

Epoch: [21][0/8]        Batch Time 0.160 (0.160)        Data Load Time 0.084 (0.084)    Loss 2.9720 (2.9720)    Top-5 Accuracy 72.549 (72.549)
Epoch: [21][5/8]        Batch Time 0.075 (0.089)        Data Load Time 0.000 (0.014)    Loss 2.8796 (2.7912)    Top-5 Accuracy 74.074 (79.104)
Validation: [0/2]       Batch Time 0.138 (0.138)        Loss 2.9967 (2.9967)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 3.008, TOP-5 ACCURACY - 85.714, BLEU-4 - 5.7592918561109494e-155


Epochs since last improvement: 3

Epoch: [22][0/8]        Batch Time 0.254 (0.254)        Data Load Time 0.166 (0.166)    Loss 2.7921 (2.7921)    Top-5 Accuracy 76.786 (76.786)
Epoch: [22][5/8]        Batch Time 0.074 (0.105)        Data Load Time 0.000 (0.028)    Loss 2.9389 (2.7893)    Top-5 Accuracy 75.926 (79.697)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 2.9670 (2.9670)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 2.964, TOP-5 ACCURACY - 85.714, BLEU-4 - 5.810039418610042e-155


Epochs since last improvement: 4

Epoch: [23][0/8]        Batch Time 0.186 (0.186)        Data Load Time 0.105 (0.105)    Loss 2.8290 (2.8290)    Top-5 Accuracy 78.182 (78.182)
Epoch: [23][5/8]        Batch Time 0.075 (0.096)        Data Load Time 0.000 (0.019)    Loss 2.8029 (2.8063)    Top-5 Accuracy 71.930 (77.515)
Validation: [0/2]       Batch Time 0.147 (0.147)        Loss 2.9764 (2.9764)    Top-5 Accuracy 86.000 (86.000)

 * LOSS - 2.953, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.382164041261909e-78

Epoch: [24][0/8]        Batch Time 0.170 (0.170)        Data Load Time 0.091 (0.091)    Loss 2.6718 (2.6718)    Top-5 Accuracy 79.630 (79.630)
Epoch: [24][5/8]        Batch Time 0.075 (0.091)        Data Load Time 0.000 (0.016)    Loss 2.7901 (2.7069)    Top-5 Accuracy 82.143 (81.737)
Validation: [0/2]       Batch Time 0.147 (0.147)        Loss 2.9034 (2.9034)    Top-5 Accuracy 86.538 (86.538)

 * LOSS - 2.952, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.4011469287201905e-78

Epoch: [25][0/8]        Batch Time 0.176 (0.176)        Data Load Time 0.100 (0.100)    Loss 2.7442 (2.7442)    Top-5 Accuracy 81.481 (81.481)
Epoch: [25][5/8]        Batch Time 0.075 (0.093)        Data Load Time 0.000 (0.017)    Loss 2.7750 (2.7070)    Top-5 Accuracy 79.661 (80.000)
Validation: [0/2]       Batch Time 0.137 (0.137)        Loss 2.9054 (2.9054)    Top-5 Accuracy 86.275 (86.275)

 * LOSS - 2.901, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.5585106520843143e-78

Epoch: [26][0/8]        Batch Time 0.182 (0.182)        Data Load Time 0.104 (0.104)    Loss 2.9731 (2.9731)    Top-5 Accuracy 69.811 (69.811)
Epoch: [26][5/8]        Batch Time 0.073 (0.094)        Data Load Time 0.000 (0.019)    Loss 2.6355 (2.7641)    Top-5 Accuracy 85.714 (80.428)
Validation: [0/2]       Batch Time 0.140 (0.140)        Loss 2.8555 (2.8555)    Top-5 Accuracy 88.235 (88.235)

 * LOSS - 2.903, TOP-5 ACCURACY - 85.714, BLEU-4 - 2.5585106520843143e-78
```

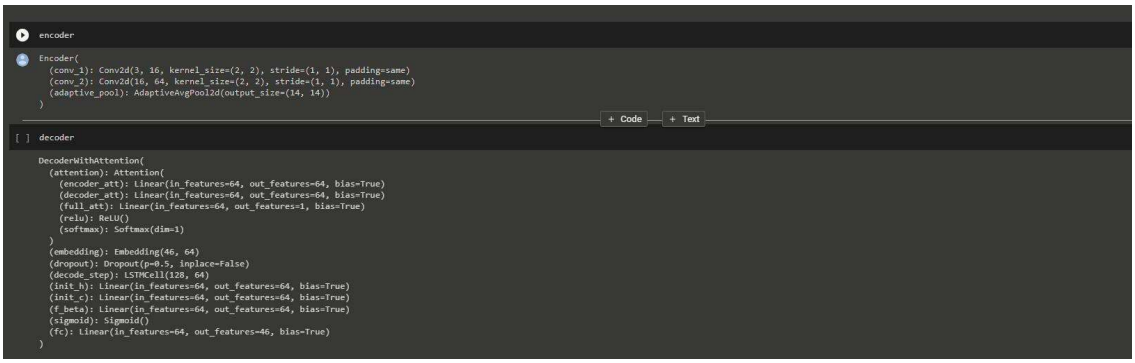
```
Epochs since last improvement: 15

Epoch: [41][0/8]        Batch Time 0.166 (0.166)        Data Load Time 0.080 (0.080)    Loss 2.3567 (2.3567)    Top-5 Accuracy 87.273 (87.273)
Epoch: [41][5/8]        Batch Time 0.075 (0.091)        Data Load Time 0.000 (0.014)    Loss 2.3115 (2.3761)    Top-5 Accuracy 83.929 (85.030)
Validation: [0/2]       Batch Time 0.133 (0.133)        Loss 2.7494 (2.7494)    Top-5 Accuracy 86.275 (86.275)

 * LOSS - 2.788, TOP-5 ACCURACY - 83.929, BLEU-4 - 2.498268189708147Se-78


Epochs since last improvement: 16


DECAYING learning rate.
The new learning rate is 0.000205

Epoch: [42][0/8]        Batch Time 0.166 (0.166)        Data Load Time 0.089 (0.089)    Loss 2.4804 (2.4804)    Top-5 Accuracy 85.714 (85.714)
Epoch: [42][5/8]        Batch Time 0.075 (0.090)        Data Load Time 0.000 (0.015)    Loss 2.1651 (2.4202)    Top-5 Accuracy 92.593 (84.985)
Validation: [0/2]       Batch Time 0.137 (0.137)        Loss 2.7840 (2.7840)    Top-5 Accuracy 82.353 (82.353)

 * LOSS - 2.763, TOP-5 ACCURACY - 83.929, BLEU-4 - 2.5389037617273243e-78


Epochs since last improvement: 17

Epoch: [43][0/8]        Batch Time 0.164 (0.164)        Data Load Time 0.086 (0.086)    Loss 2.3761 (2.3761)    Top-5 Accuracy 83.929 (83.929)
Epoch: [43][5/8]        Batch Time 0.076 (0.092)        Data Load Time 0.000 (0.015)    Loss 2.3977 (2.3830)    Top-5 Accuracy 87.755 (86.747)
Validation: [0/2]       Batch Time 0.135 (0.135)        Loss 2.7916 (2.7916)    Top-5 Accuracy 82.353 (82.353)

 * LOSS - 2.775, TOP-5 ACCURACY - 83.929, BLEU-4 - 2.5389037617273243e-78


Epochs since last improvement: 18

Epoch: [44][0/8]        Batch Time 0.153 (0.153)        Data Load Time 0.076 (0.076)    Loss 2.1717 (2.1717)    Top-5 Accuracy 91.071 (91.071)
Epoch: [44][5/8]        Batch Time 0.074 (0.090)        Data Load Time 0.000 (0.014)    Loss 2.4125 (2.3772)    Top-5 Accuracy 81.481 (83.832)
Validation: [0/2]       Batch Time 0.134 (0.134)        Loss 2.7686 (2.7686)    Top-5 Accuracy 84.000 (84.000)

 * LOSS - 2.747, TOP-5 ACCURACY - 83.929, BLEU-4 - 2.5585106520843143e-78


Epochs since last improvement: 19

Epoch: [45][0/8]        Batch Time 0.173 (0.173)        Data Load Time 0.086 (0.086)    Loss 2.4158 (2.4158)    Top-5 Accuracy 88.333 (88.333)
Epoch: [45][5/8]        Batch Time 0.080 (0.098)        Data Load Time 0.000 (0.017)    Loss 2.2192 (2.3263)    Top-5 Accuracy 85.455 (85.928)
Validation: [0/2]       Batch Time 0.136 (0.136)        Loss 2.7425 (2.7425)    Top-5 Accuracy 86.275 (86.275)
```

Mini - Network Formulation:

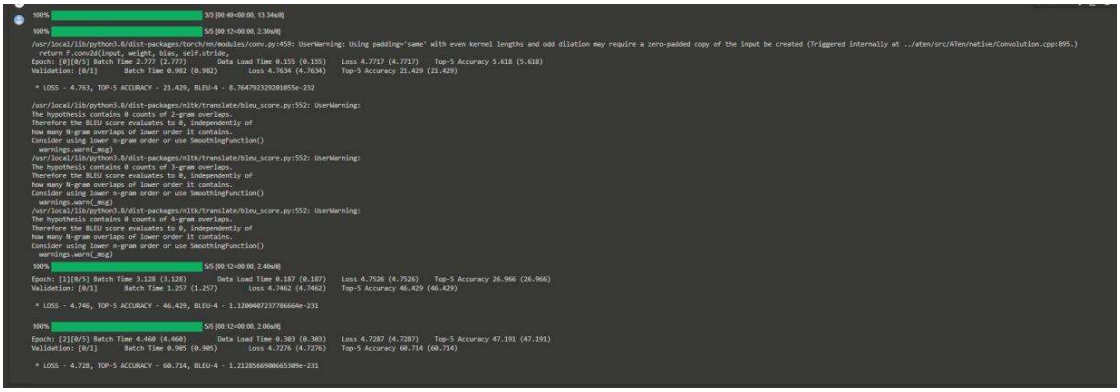
The mini network model is initialized using encoder-decoder and the model is trained by categorical cross entropy loss. At first the data preprocessing is done and then the training loop is executed. The training loop consists of encoder and decoder.



The complete encoder-decoder architecture mini network is formulated which runs according to the flow diagram shown above.



The evaluation of the mini network is using bleu scores after the data is trained.



Project Contributions:

Abhinaya Movva:

- Coding, Hyperparameter tuning, analyzing results

Harsha Kolla:

- Dataset, Json, Documentation

Discussion & Conclusion:

Automatically demonstrating what is included in an image or photograph has always been a topic of study in artificial intelligence. CNN and RNN-LSTM models are used in our implementation of the caption generator. The suggested model is based on multi-label classification and employs a CNN-RNN technique to produce captions, with CNN acting as an encoder and RNN model LSTM acting as a decoder. It integrates contemporary research in computer vision and machine learning. We used BLEU scores to assess the effectiveness of both the stated models. One could distinguish between good and terrible produced captions using the ratings. The main uses of this paradigm involve assistants, picture indexing, social networks, accessibility for persons with visual impairments, suggestions in editing software, and many more. While comprehensive image labelling methods that can achieve large labels for virtually each image have not yet been developed, deep learning-based image labelling systems have made substantial advancements in recent times. While comprehensive image labelling methods which can achieve large labels for virtually each image have not yet been developed, deep learning-based image labelling imaging systems have made substantial advancements in recent times. This endeavor will benefit individuals as an outcome.