# Image Captioning Generator

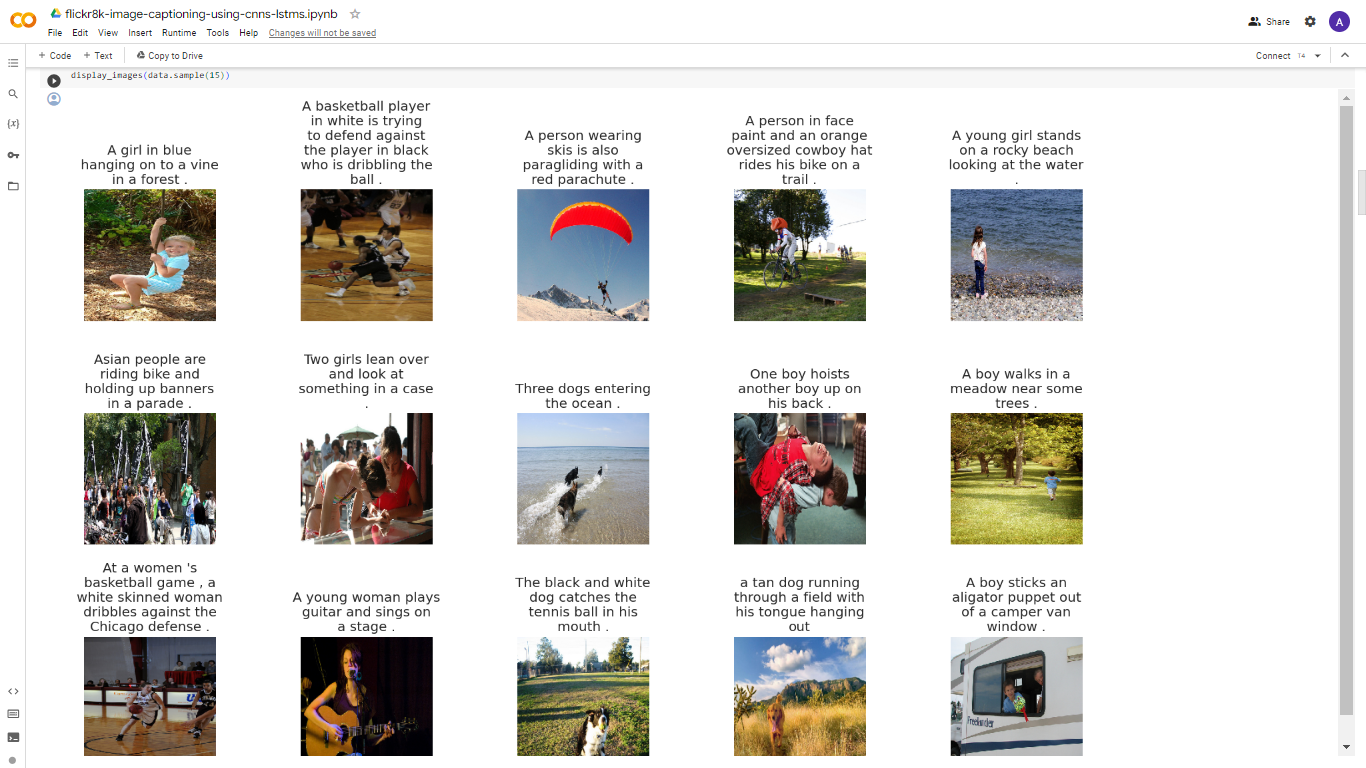
# What is Image Captioning? :

* Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions.
* This task lies at the intersection of computer vision and natural language processing. Most image captioning systems use an encoder-decoder framework, where an input image is encoded into an intermediate representation of the information in the image, and then decoded into a descriptive text sequence.

1. CNNs + RNNs (LSTMs):

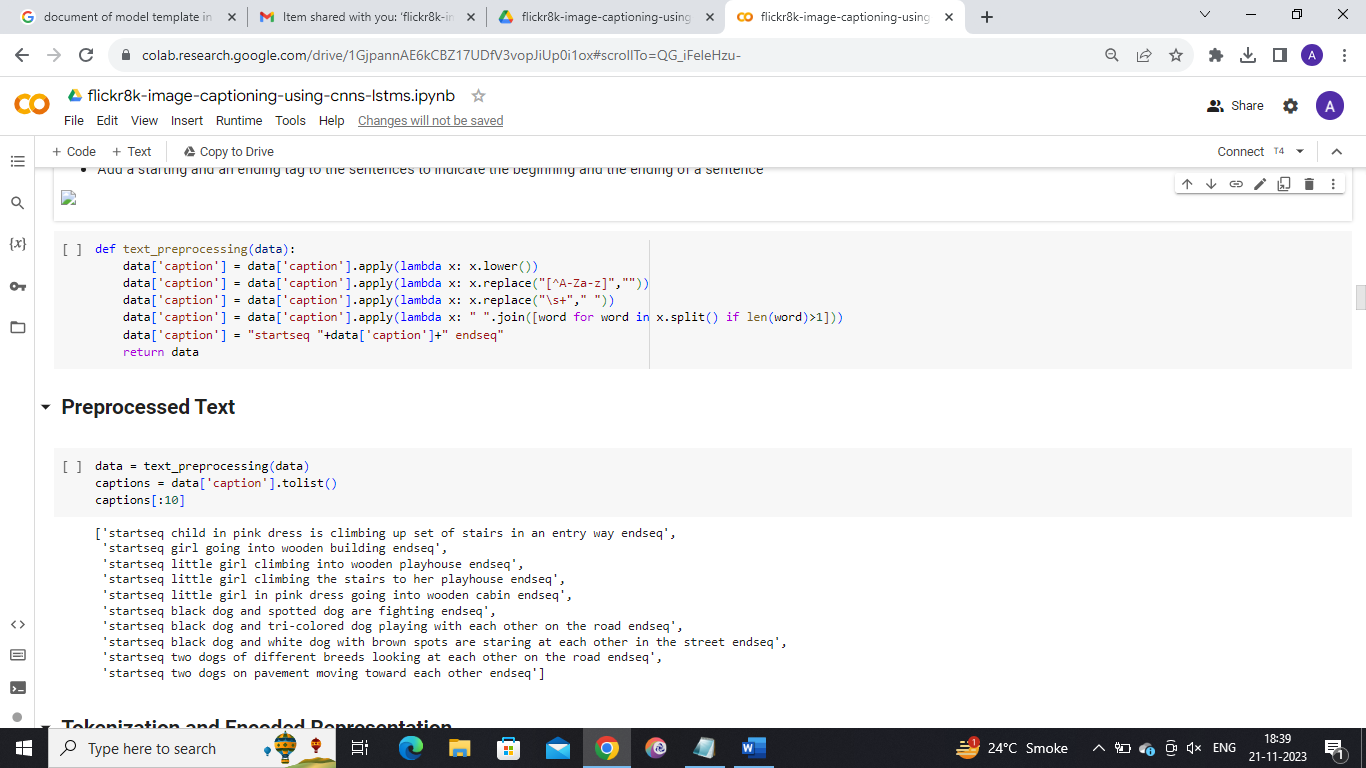
* To perform Image Captioning we will require two deep learning models combined into one for the training purpose
* CNNs extract the features from the image of some vector size aka the vector embeddings. The size of these embeddings depend on the type of pretrained network being used for the feature extraction
* LSTMs are used for the text generation process. The image embeddings are concatenated with the word embeddings and passed to the LSTM to generate the next word
* For a more illustrative explanation of this architecture check the Modelling section for a picture representation

# Visualization(Images and their corresponding captions)



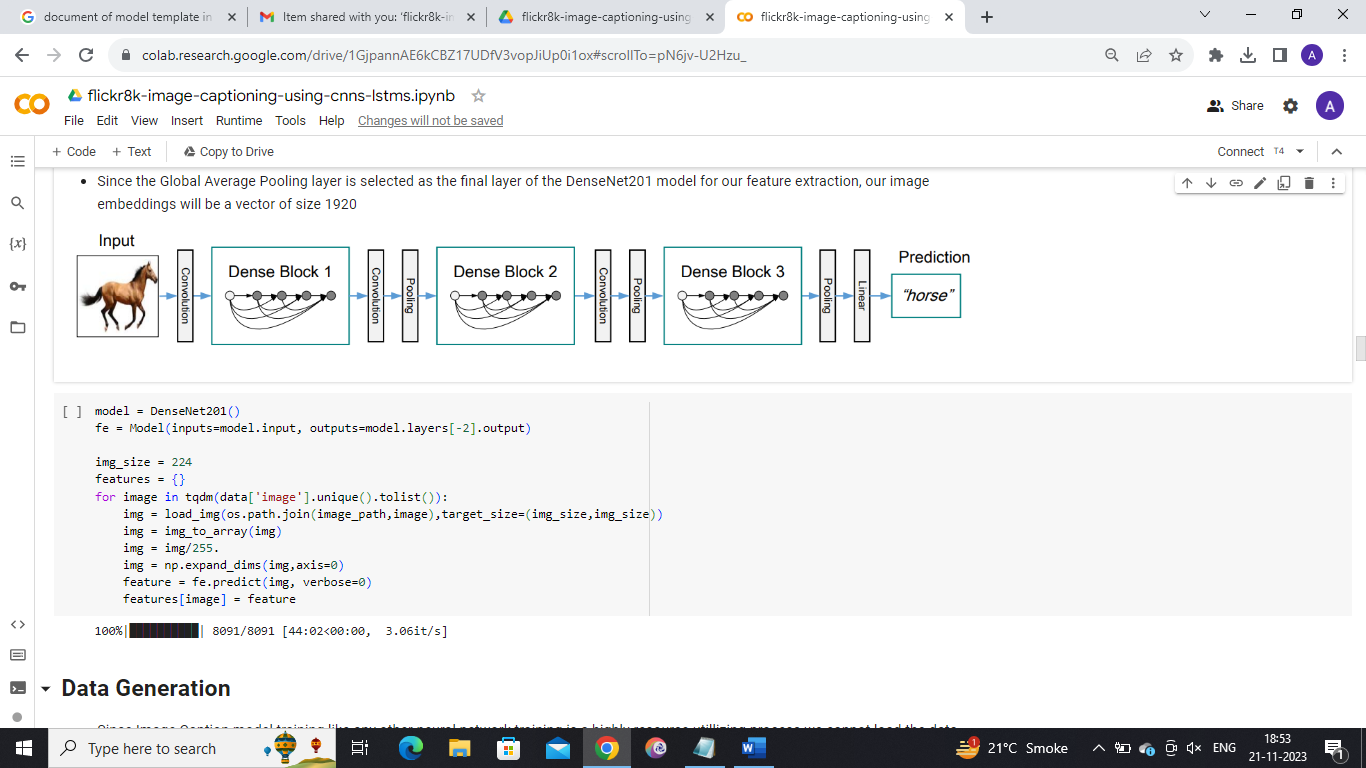
# Caption Text Preprocessing Steps

* Convert sentences into lowercase
* Remove special characters and numbers present in the text
* Remove extra spaces
* Remove single characters
* Add a starting and an ending tag to the sentences to indicate the beginning and the ending of a sentence



# Image Feature Extraction

* DenseNet 201 Architecture is used to extract the features from the images
* Any other pretrained architecture can also be used for extracting features from these images
* Since the Global Average Pooling layer is selected as the final layer of the DenseNet201 model for our feature extraction, our image embeddings will be a vector of size 1920

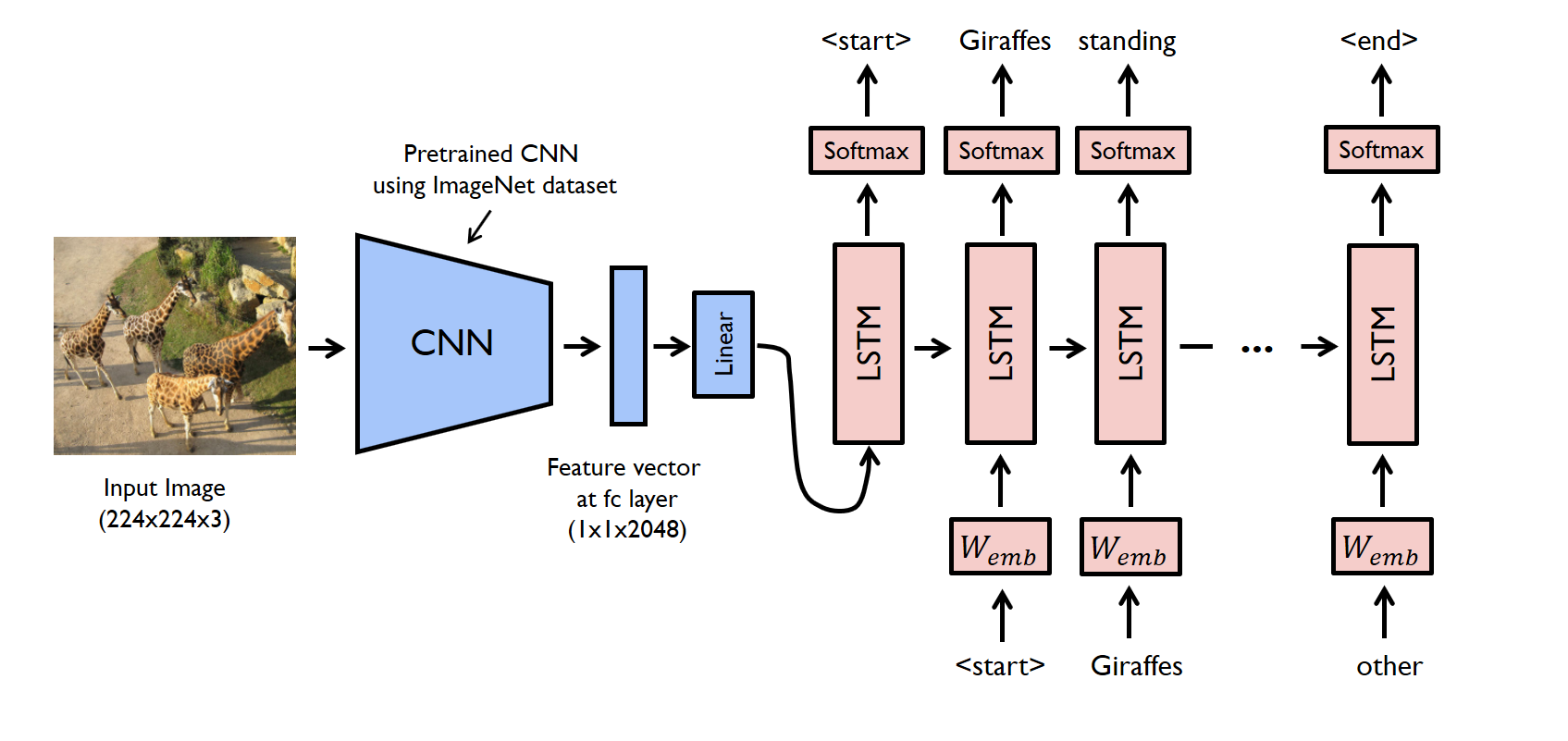


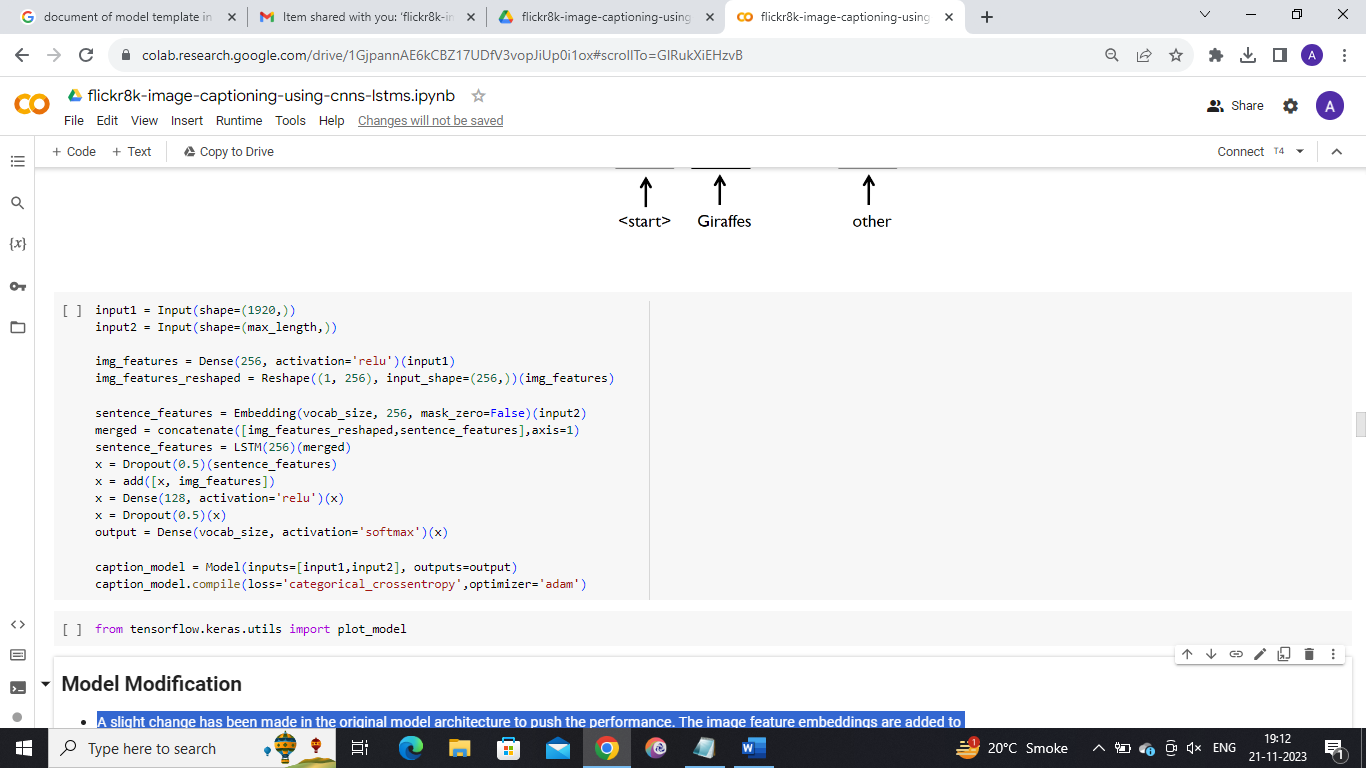
# Data Generation

* Since Image Caption model training like any other neural network training is a highly resource utillizing process we cannot load the data into the main memory all at once, and hence we need to generate the data in the required format batch wise
* The inputs will be the image embeddings and their corresonding caption text embeddings for the training process
* The text embeddings are passed word by word for the caption generation during inference time

# Modelling

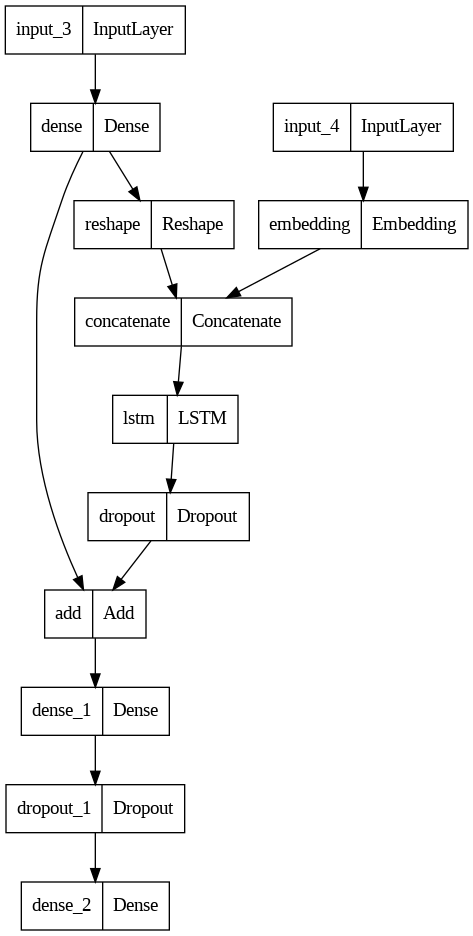
* The image embedding representations are concatenated with the first word of sentence ie. starseq and passed to the LSTM network
* The LSTM network starts generating words after each input thus forming a sentence at the end



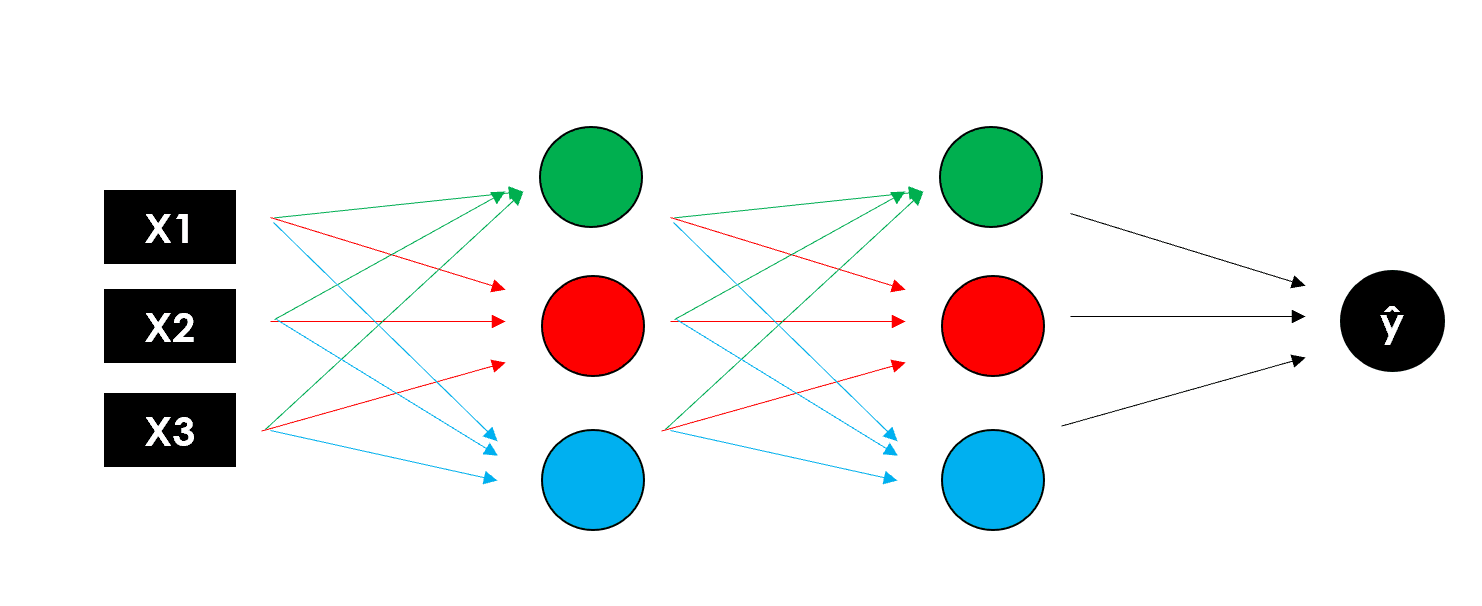


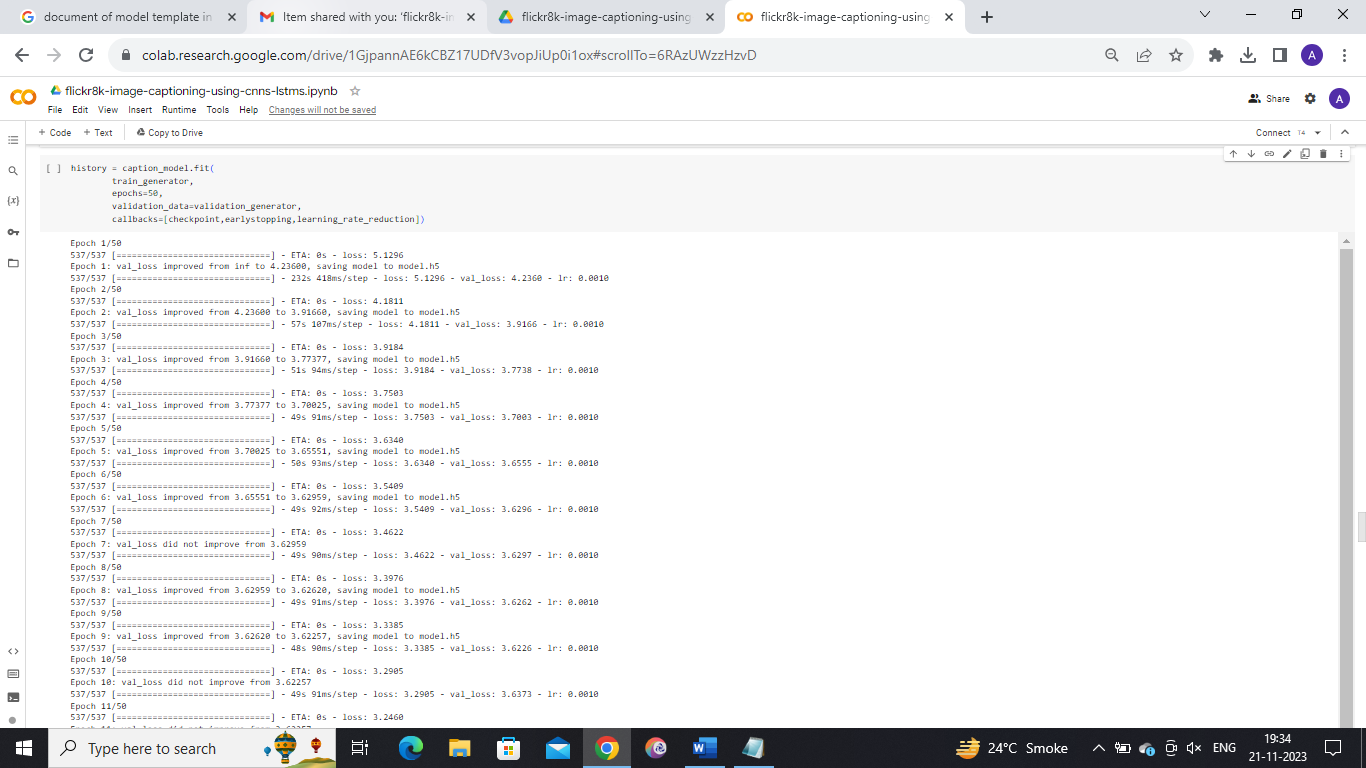
# Model Modification

* A slight change has been made in the original model architecture to push the performance. The image feature embeddings are added to the output of the LSTMs and then passed on to the fully connected layers
* This slightly improves the performance of the model orignally proposed back in 2014: Show and Tell: A Neural Image Caption Generator



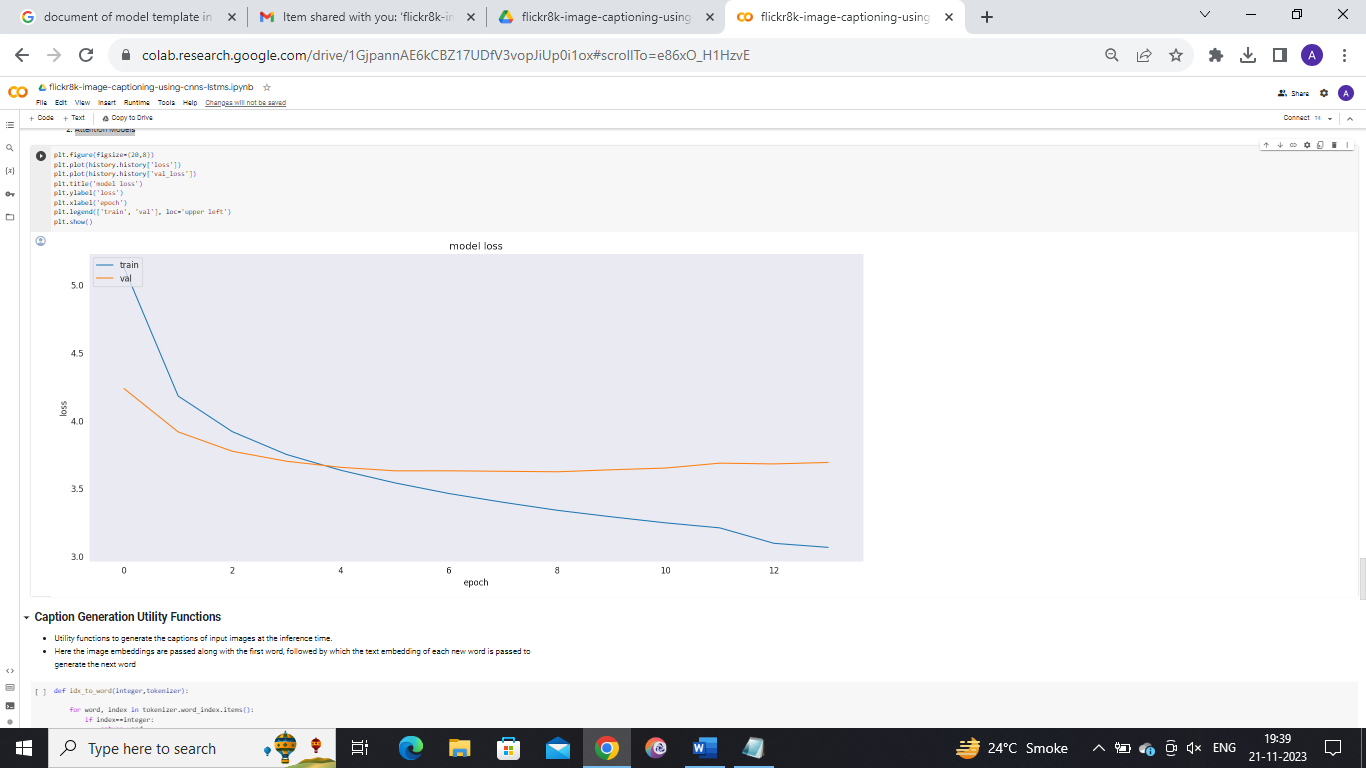
# Train the Model





# Result analysis of model

* Inference
  + Learning Curve (Loss Curve)
  + Assessment of Generated Captions (by checking the relevance of the caption with respect to the image, BLEU Score will not be used in this kernel)
* Learning Curve
  + The model has clearly overfit, possibly due to less amount of data
  + We can tackle this problem in two ways
  + Train the model on a larger dataset Flickr40k
  + Attention Models



# Result

# Caption Generation Utility Functions

* Utility functions to generate the captions of input images at the inference time.
* Here the image embeddings are passed along with the first word, followed by which the text embedding of each new word is passed to generate the next word
* **Taking 15 Random Samples for Caption Prediction**

