Project Design
Phase-I Solution
Architecture

Date	23 October 2023
Team ID	Team - 591950
Project Name	Project - IMAGE CAPTION GENERATION
Maximum Marks	5 Marks

Solution Architecture:

CNN – To extract features from the image. A pre-trained model called Exception is used for this.

LSTM – To generate a description from the extracted information of the image.

Steps Involving in architecture:

Dataset for Image Caption Generator:

The Flickr_8K dataset represents the model training of image caption generators. The dataset is downloaded directly from the below links. The downloading process takes some time due to the dataset's large size. In the image below, you can check all the files in the Flickr_8k text folder. The most important file is Flickr 8k.token which stores all the image names with captions.

Building the Image Caption Generator:

We can build image caption generator by jupyter note book by naming the file with train_the_model_caption_generator.

Import all the Required Packages:

Perform Data Cleaning:

As we see all image captions are available in the Flickr 8k.token file of the Flickr_8ktext folder. If you analyze this file carefully, you can drive the format of image storing, each image and caption separated by a new line and carry 5 captions numbered from 0 to 4 along with.

Extract the Feature Vector:

Now we are going to use the pre-trained model called Xception which is already trained with large Datasets to extract the features from these models. Xception was trained on an image net dataset with 1000 different classes to classify the images.

Loading dataset for model training:

A file named Flickr_8ktrainImages.txt is present in our Flickr_8ktest folder. This file carries a list of 6000 Image names that are used for the sake of training.

Tokenizing the Vocabulary:

Machines are not familiar with complex English words so, to process model's data they need a Simple numerical representation. That's why we map every word of the vocabulary with a Separate unique index value. An in-built tokenizer function is present in the Keras library to Create tokens from our vocabulary.

Create a Data generator:

For training the model as a supervised learning task we need to feed it with input and output sequences.

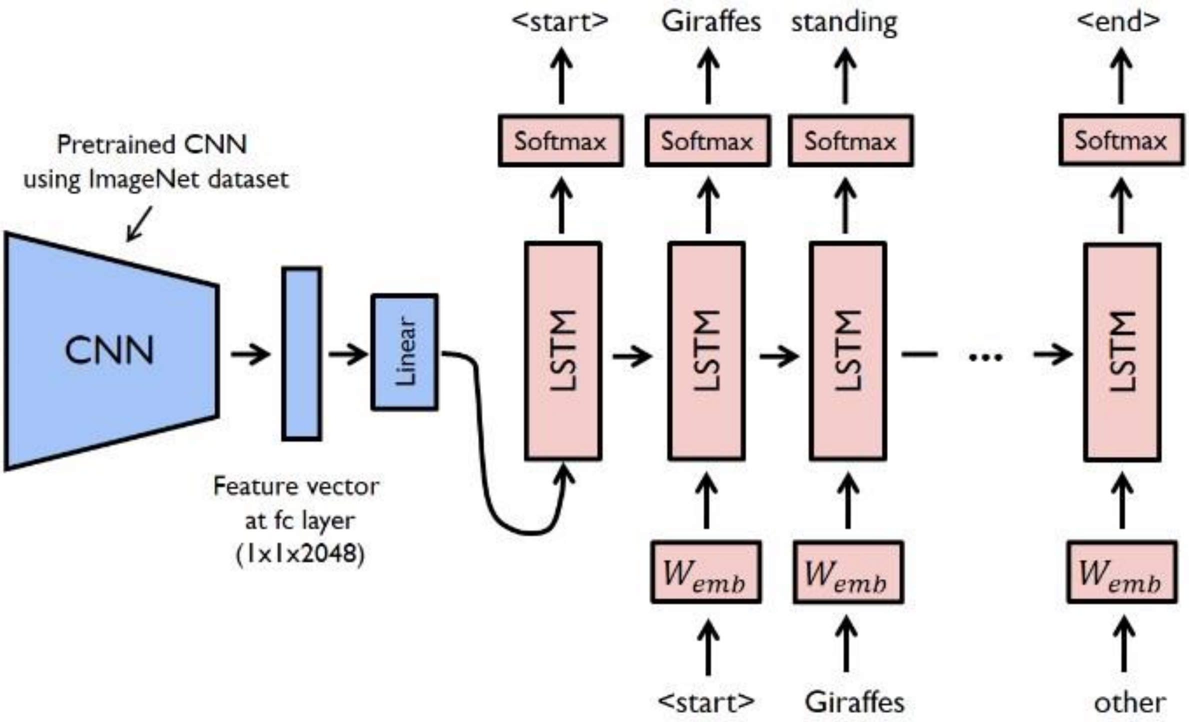
Training the Image Caption Generator model:

We will generate the input and output sequences to train our model with 6000 training images. We Create a function named model.fit_generator() to fit the batches to the model. At last, we save the model To our models folder.

Testing the Image Caption Generator model:

After successful model training, our task is to test the model accuracy by inputting test image data. Let's Create a python file named test_caption.py to load the model and generate predictions.

Example - Solution Architecture Diagram:


Pretrained CNN
using ImageNet dataset
CNN
Feature vector
at fc layer
(1x1x2048)
Linear
<start>
Giraffes
standing
<end>
Softmax
Softmax
Softmax
Softmax
LSTM
LSTM
LSTM
LSTM
...
W_{emb}
W_{emb}
W_{emb}
<start>
Giraffes
other