Automated Image Captioning

Group 3 – Cohort 18

**Team**

-Sujatha Kancharla

-Monica Nukarapu

-Suhail Pasha Kotwal

-Chaitanya Harkara

**Mentor**

-Nayan Jha

Table of Contents

[INTRODUCTION 3](#_Toc113182637)

[Project Description 3](#_Toc113182638)

[Objective 3](#_Toc113182639)

[Timelines 3](#_Toc113182640)

[Dataset 4](#_Toc113182641)

[Deliverables 4](#_Toc113182642)

[Technology 4](#_Toc113182643)

[Understanding of the problem 5](#_Toc113182644)

[SOLUTION APPROACH 7](#_Toc113182645)

[Proposed Solution 8](#_Toc113182646)

[Option1 – Base Model ( LSTM) 8](#_Toc113182647)

[Option2 – Attention 12](#_Toc113182648)

[Option3 – TranSFORMERS 15](#_Toc113182649)

[CHALLENGES 20](#_Toc113182650)

[Sample Auotmated Captions generated 21](#_Toc113182651)

[Data Set 23](#_Toc113182652)

[Conclusion 27](#_Toc113182653)

# InTRODUCTION

Image caption Generator is a popular research area of Artificial Intelligence that deals with image understanding and a language description for that image. Generating well-formed sentences requires both syntactic and semantic understanding of the language. Being able to describe the content of an image using accurately formed sentences is a very challenging task, but it could also have a great impact, by helping visually impaired people better understand the content of images.

This task is significantly harder in comparison to the image classification or object recognition tasks that have been well researched.

The biggest challenge is most definitely being able to create a description that must capture not only the objects contained in an image, but also express how these objects relate to each other.

# Project Description

Captioning the images with proper description is a popular research area of Artificial Intelligence. A good description of an image is often said as “Visualizing a picture in the mind”. The generation of descriptions from the image is a challenging task that can help and have a great impact in various applications such as usage in virtual assistants, image indexing, a recommendation in editing applications, helping visually impaired persons, and several other natural language processing applications. In this project, we need to create a multimodal neural network that involves the concept of Computer Vision and Natural Language Process in recognizing the context of images and describing them in natural languages (English, etc). Deploy the model and evaluate the model on 10 different real-time images.

# Objective

Build an image captioning model to generate captions of an image using CNN

# Timelines

Start - 18-Jun and End (Delivery) –11-Sep

# Dataset

Flickr8k, Flickr30k & COCO

# Deliverables

* Project Technical Report
* Project presentation with desired Documents
* Summary of 3 research Papers

# Technology

* **Tools** : Natural Language Toolkit, TensorFlow, PyTorch, Keras
* **Deployments:** FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

# Understanding of the problem

Automated Image captioning involves in creating an automated caption for an Image by deriving the best context of the contents of the image.

Broadly the solution should

1. Identify multiple objects within the image
2. Derive the relationship between the objects in the image based on their attributes
3. Derive the caption based on the derived context of the image in Natural language (English)

|  |  |
| --- | --- |
| A picture containing green, toy, colorful, close  Description automatically generated  Yoga for… | Trick Photography… |
| A picture containing grass  Description automatically generated  Tennis on sand … | A green frog figurine  Description automatically generated with medium confidence  Make money … |

**Key inputs**

Historically Image captioning solutions were that were developed have been template based, which were heavily hand designed and rigid in terms of Text generation.

**Key References**

Based on the latest solutions of Text generation Using Recurring Neural Networks (RNN), there are multiple recommendations (Research papers) to develop an Image captioning solution using a combination of CNN (encoder) and multiple options for Decoders like RNN(Decoder). LSTM, Attention and Multi head attentions

The research papers

* Show and Tell – “A Neural image caption generator “
* AICRL – Automate Image Captioning Resnet50 LSTM
* Attention Is All You Need

# SOLUTION APproaCH

We plan to solve to this problem using an **Encoder-Decoder** model with three different options

**Option1**

We call this a base model. In this model we used InceptionV3 and Resnet50 for feature extraction combined with Word Embedding( generated using Glove ) passed it to LSTM to generate Predictions for captions of an image

**Option2**

We plan to implement soft attention on the Input and LSTM as decoder. In this model we used InceptionV3 and Resnet50 for feature extraction combined with Word Embedding( generated(Using Keras Tokenizer) and built a Attention layer and passed the output to LSTM to generate Predictions for captions of an image

**Option3**

# Proposed Solution

## Option1 – Base Model ( LSTM)

* Solution Architecture

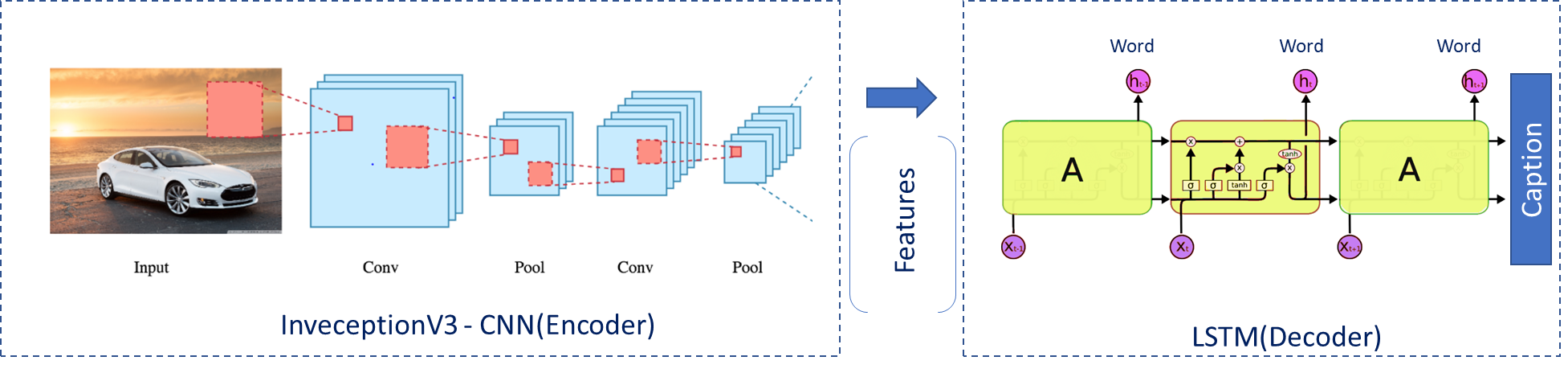


Figure 1:Automated Image captioning using LSTM

Encoder Model will build a combination of both the Encoded form of the image and the encoded form of the text caption and Feed to the Decoder.

Our model will treat CNN as the ‘image model’ and the RNN/LSTM as the ‘language model’ to encode the text sequences of varying length. The vectors resulting from both the encodings are then merged and processed by a Dense layer to make a final prediction of the caption

We created a merge architecture in order to keep the image out of the RNN/LSTM and thus be able to train the part of the neural network that handles images and the part that handles language separately, using images and sentences from separate training sets.

To encode our image features we made use of transfer learning. We used Pre trained CNN based models InceptionV3 and ResNet50. To encode our text sequence we will map every word to a 200-dimensional vector. We did this using a pre-trained Glove model. This mapping will be done in a separate layer after the input layer called the embedding layer. To generate the caption, we used Greedy Search and Blue score for Quantitative evaluation. These methods will help us in picking the best words to accurately define the image.

Following describes step by step process and various inputs at each stage of execution of the End to End implementation for InceptionV3 and Resnet

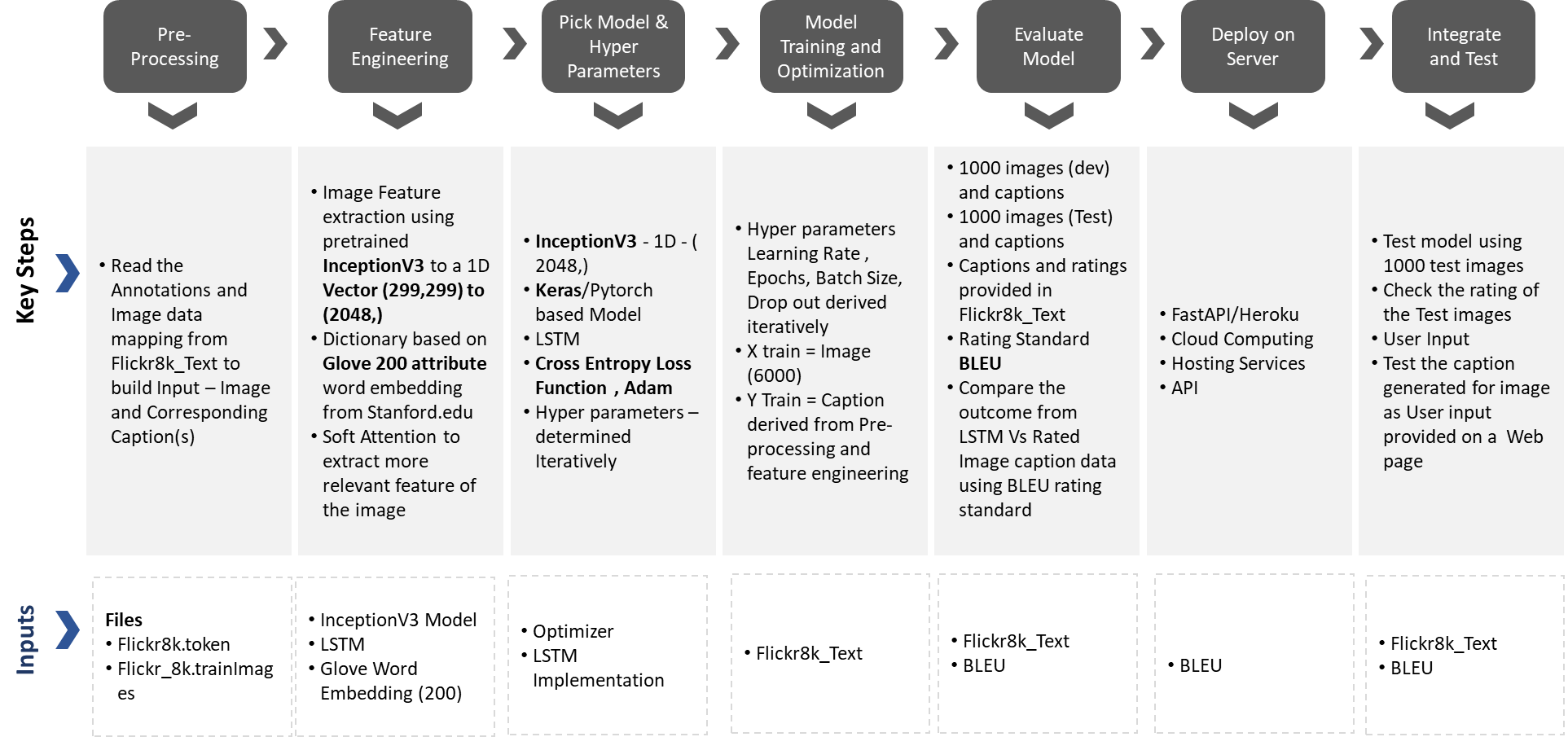


Figure 2: Step by Step process for End to End Implementation using InceptionV3 for Feature extraction

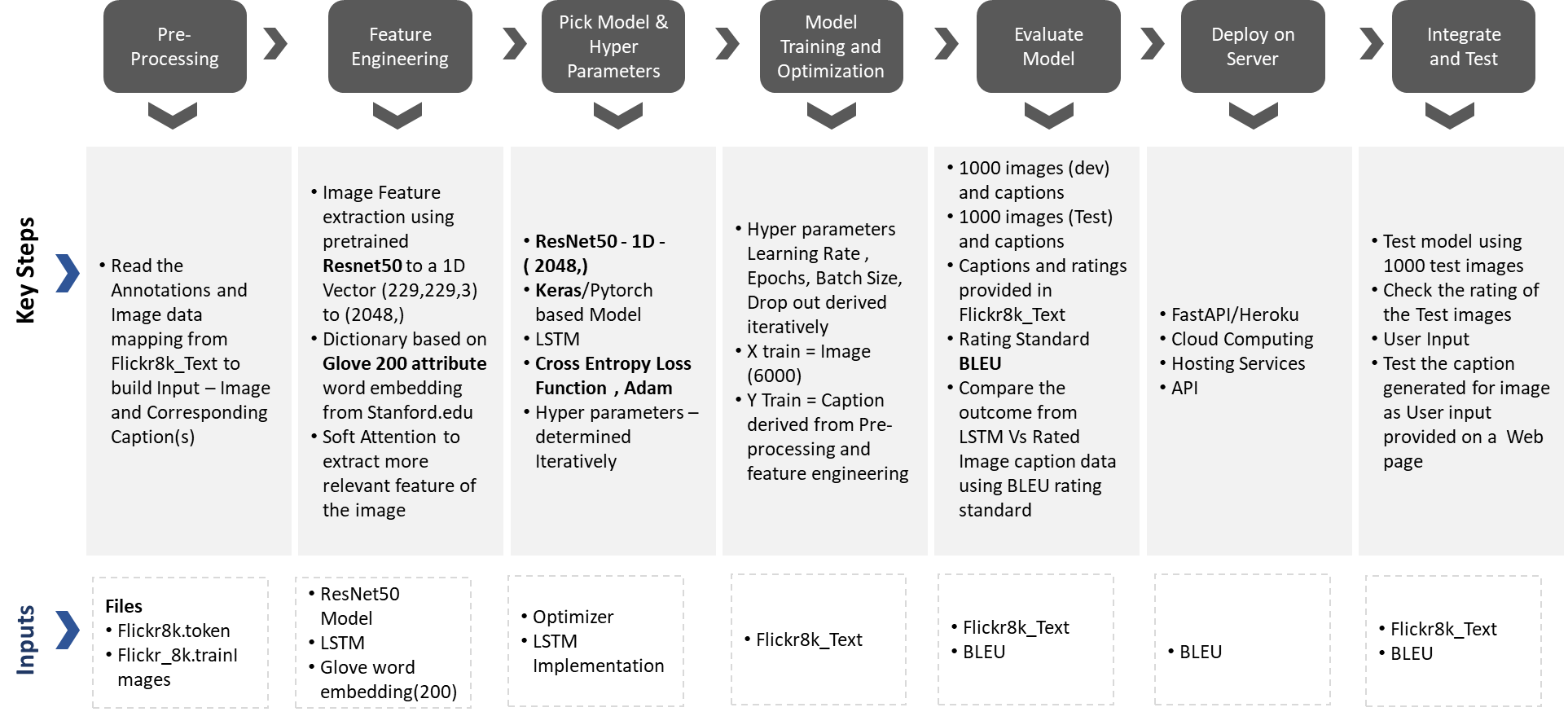


Figure 3 Step by Step process for End to End Implementation using Resent50 for Feature extraction

Following diagram represent the implemented model and Data structure at each stage of execution for InceptionV3 and Resnet50

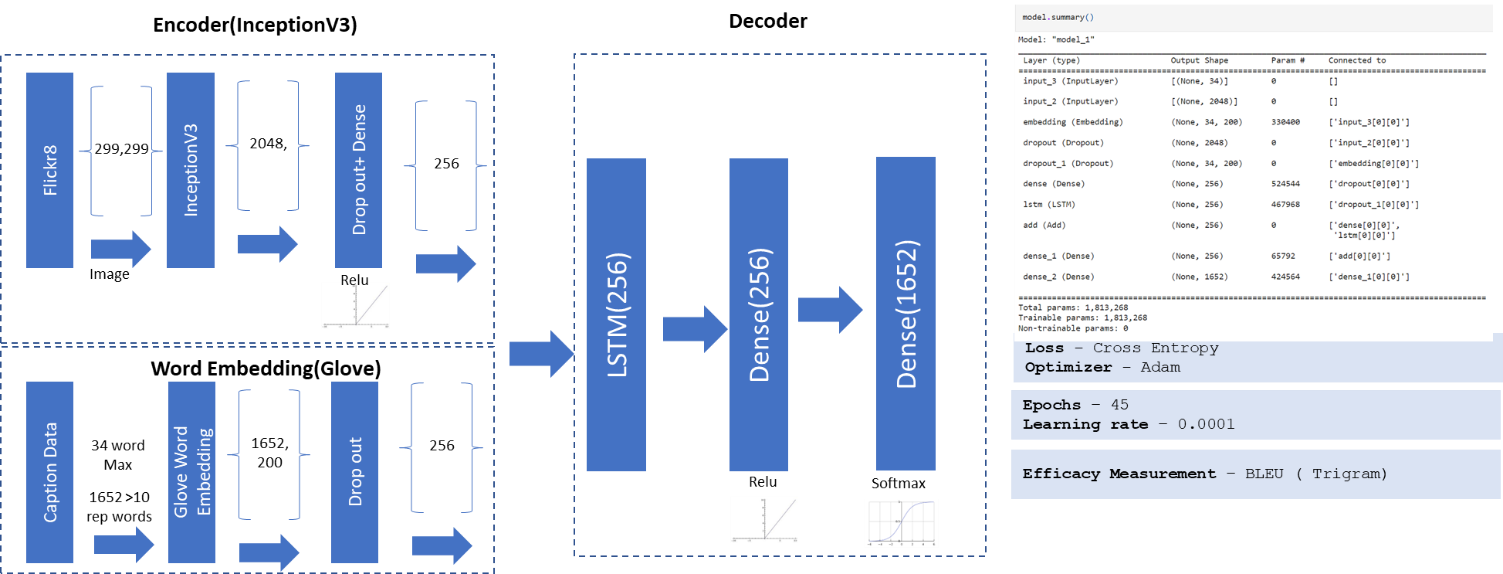


Figure 4:Representation of different layers Implemented LSTM Model and the Data propagation using InceptionV3

Graphical user interface, text

Description automatically generated

Figure 5Representation of different layers Implemented LSTM Model and the Data propagation using Resnet50

## Option2 – Attention

* Solution Architecture

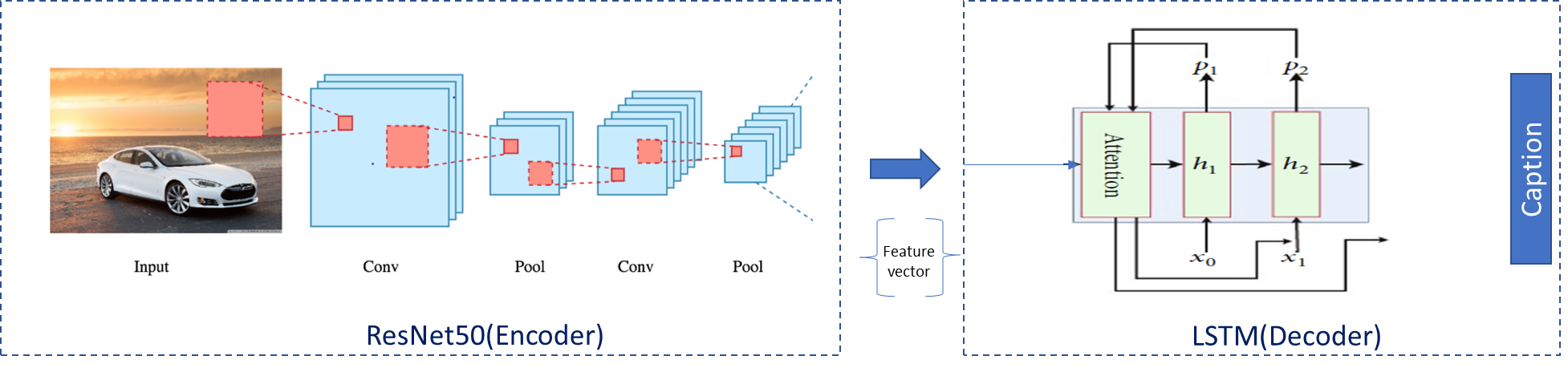


Figure 6:Attention Architecture for Image Captioning

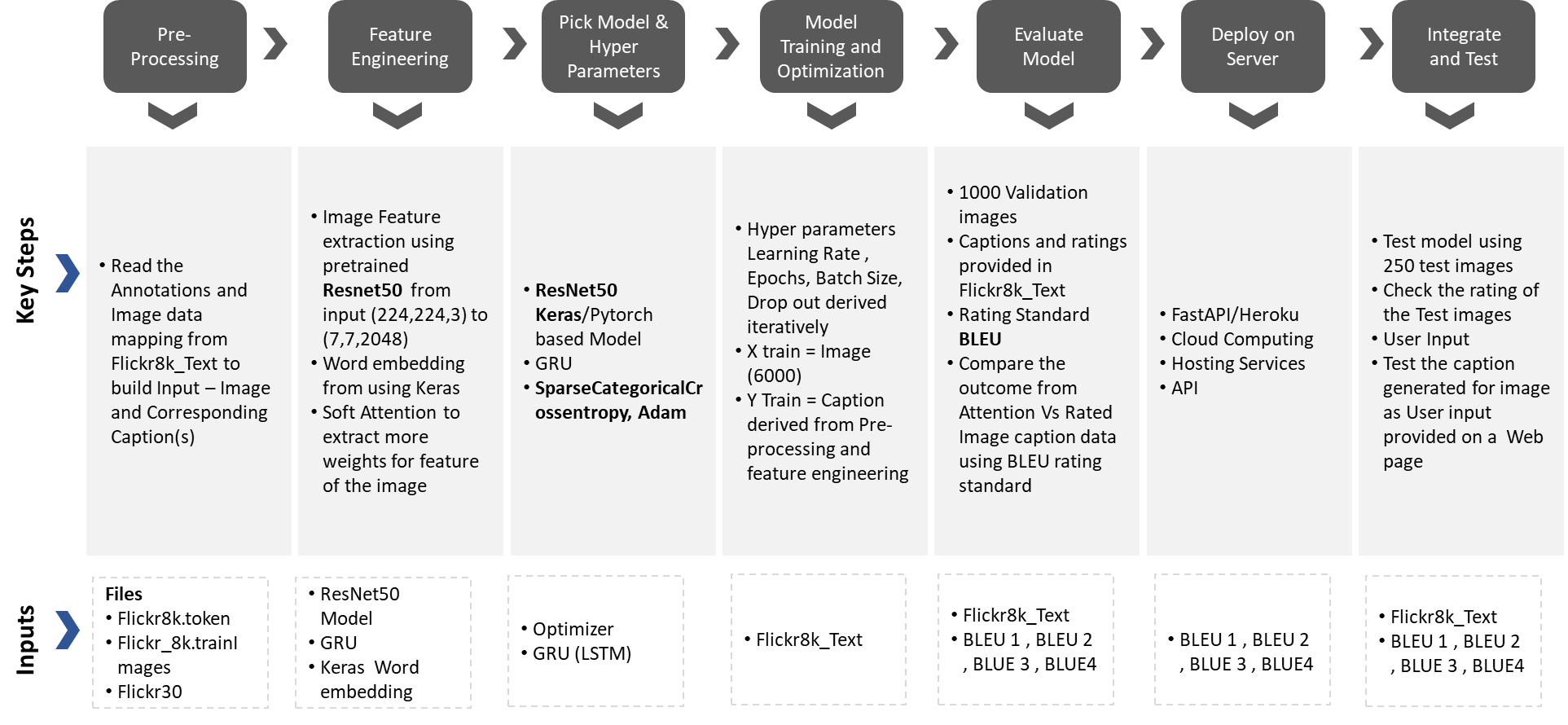
* **Key Highlights**

**Encoder**

* Represent the image, using pretrained convolutional neural network (CNN), ResNet50, which is a very deep network that has 50 layers
* Extract visual features, which use ResNet50 network as the encoder to generate a **Feature vector** representation of the input images

**Decoder**

* Soft attention is implemented by adding an additional input of attention gate into LSTM that helps to concentrate selective attention
* LSTM networks are used to accomplish the tasks of machine translation and sequence generation
* Step by Step Execution Plan



* **Solution Design and Implementation**

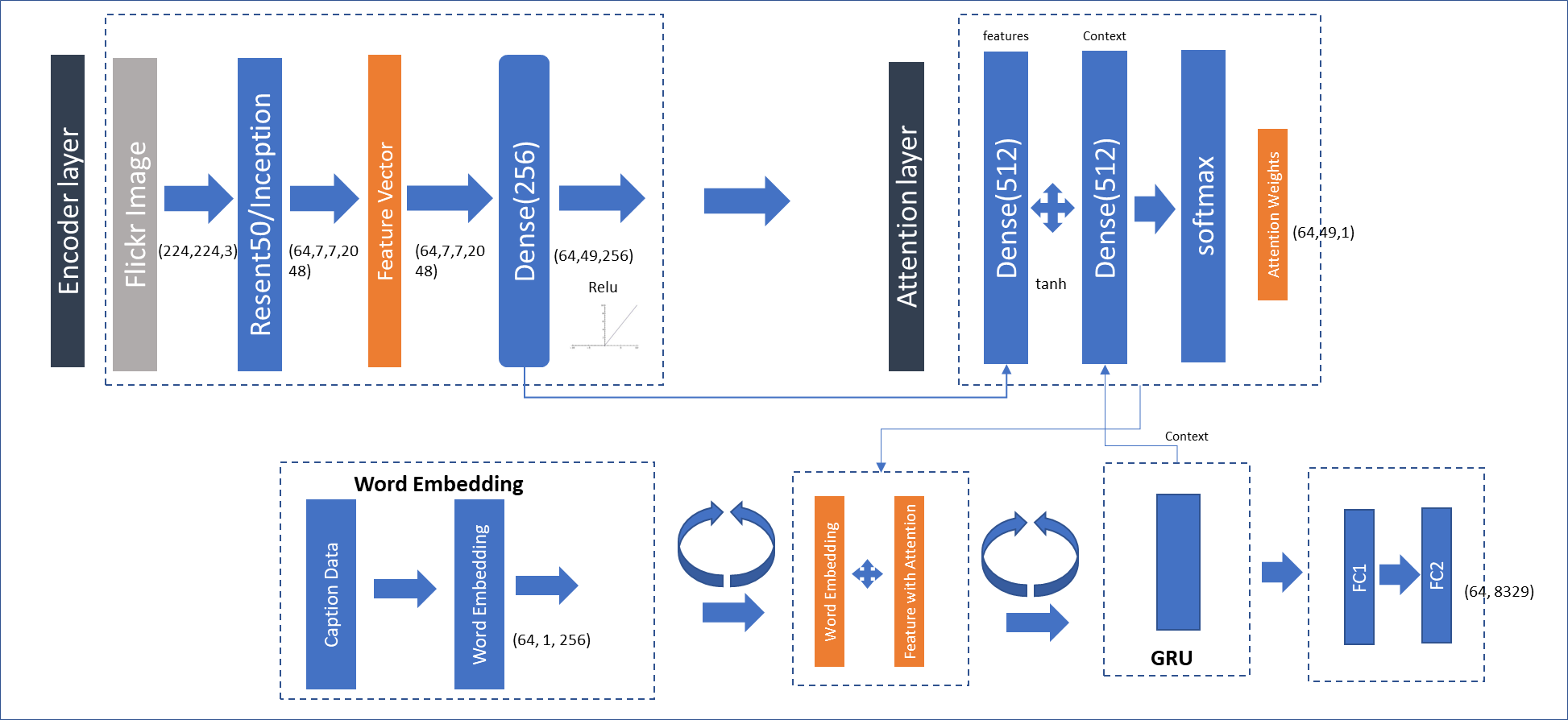
****

Figure 7:Representation of various layer of implementation of Attention Architecture for Image captioning

* **Attention Design**

An attention layer is implemented on top of feature vector extraction from Resnet50. The output from the attention layer in combination with Word embedding is iteratively passed to the LSTM (GRU) to generate the context. The context generated from GRU is passed input to the Attention layer iteratively thus building learning the important features of image for a context .

## Option3 – TranSFORMERS

* Transformer Architecture

Diagram

Description automatically generated

Figure 8:Transformer Architecture

Transformer architecture involves Implementation of Multi Head Attention at two levels . Primary Multi Head attention is implemented on the image feature vector and secondly on the combination of the Word embedding and MHA from Image

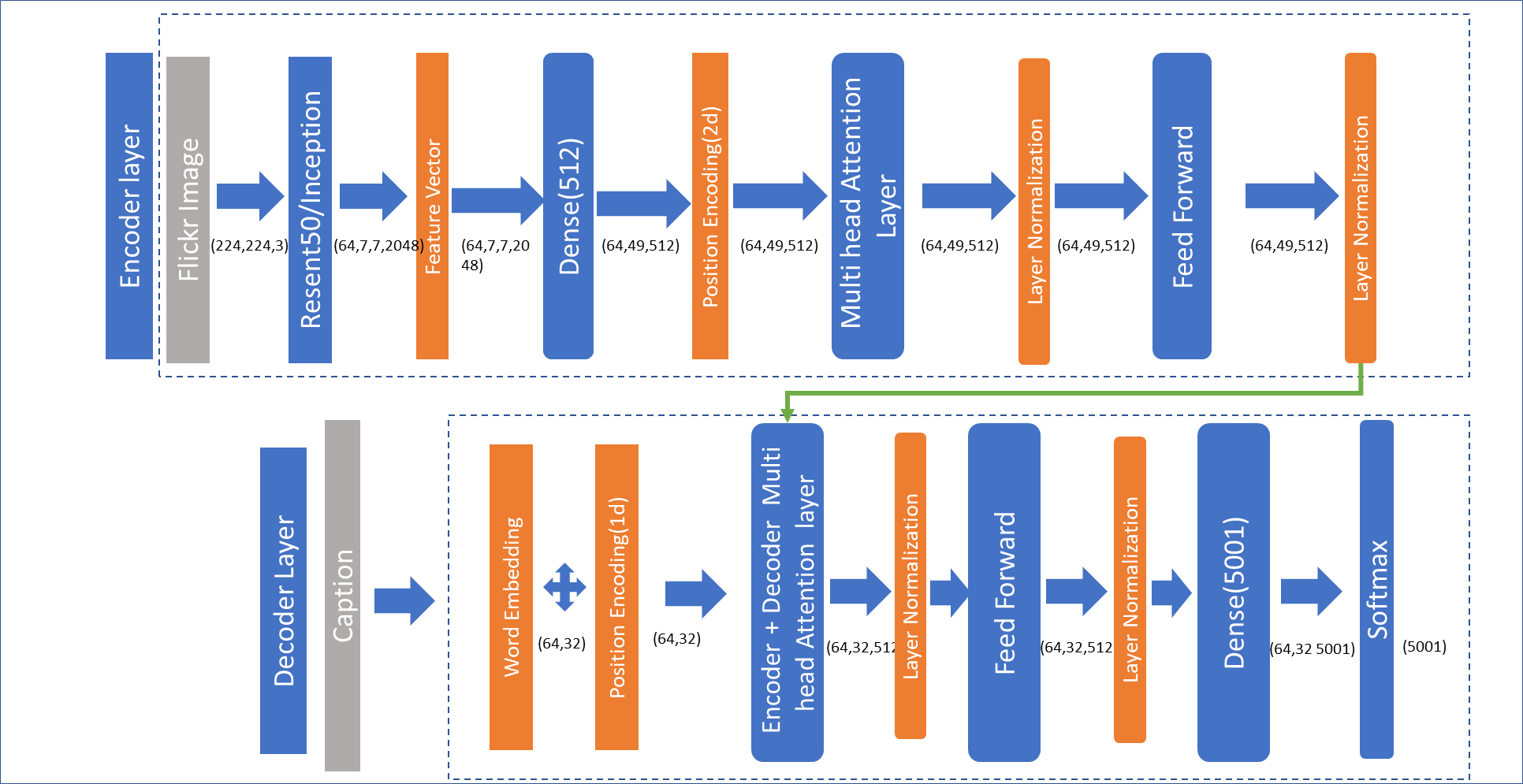


Figure 9:Representation of various layers in Transform Implementation and Data flow ( Resnet50)

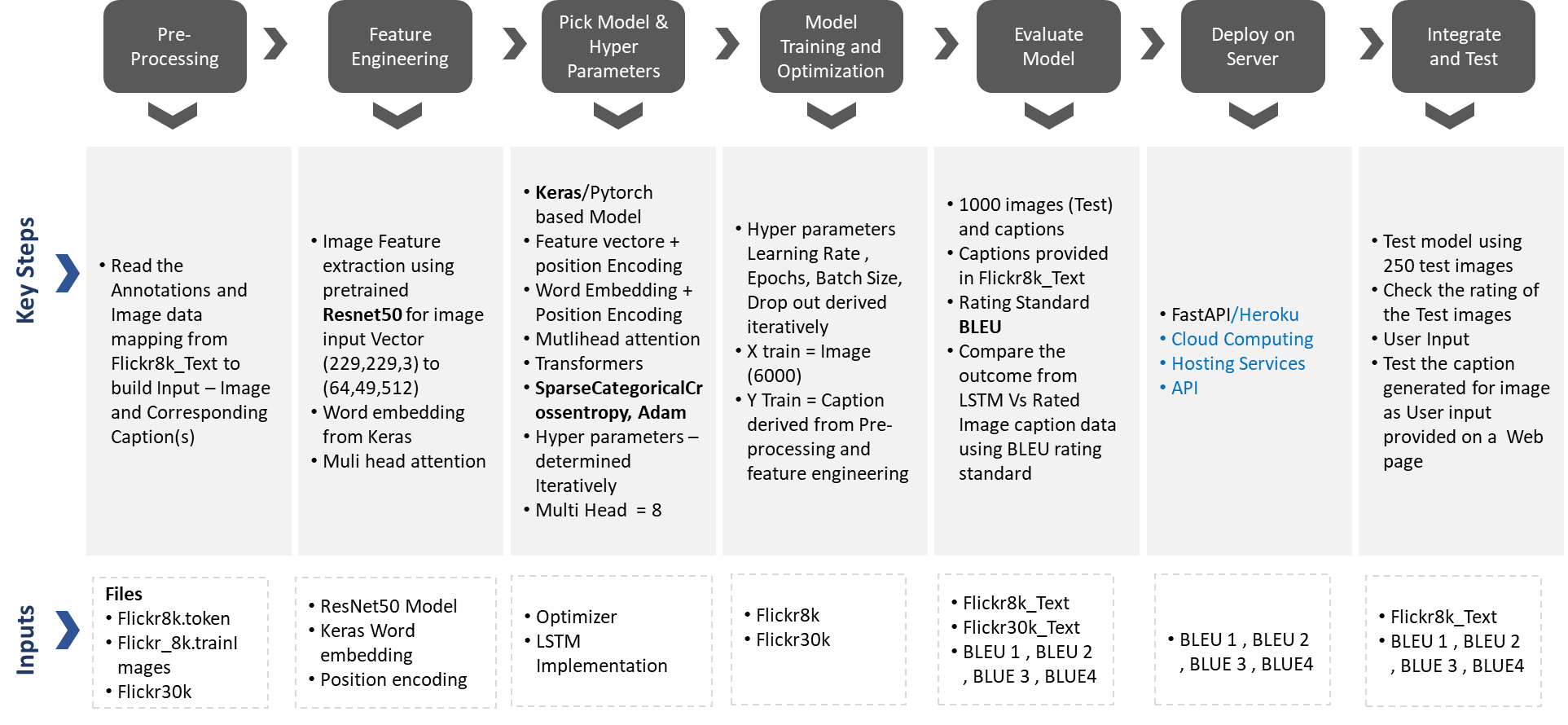


Figure 10:Step by step implementation of Transformer

* **Key Results**

Following key Hyper parameters and the results across multiple implementations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Image Feature extraction** | **Hyper Parameters and training details** | **Data Set** | **Results ( Data captured after running on 250 random Test Images)** | |
| **Base Model - LSTM** | Inception V3 | Loss - Cross Entropy  Optimizer – Adam  Epochs – 45  Learning rate – 0.0001  Training – 8000 images | Flickr8 | BLEU-3 count: 46 | BLEU-3 count average 0.21 |
| Resnet50 | Flickr8 | BLEU-3 count: 61 | BLEU-3 count average 0.2875 |
| **Attention** | VGG16 | Loss - SparseCategoricalCrossentropy  Optimizer – Adam  Epochs – 45  Learning rate – 0.0001  Training – 8000 images | Flickr8 | BLEU-1 count: 176  BLEU-2 count: 71  BLEU-3 count: 26  BLEU-4 count: 4 | BLEU-1 Average: 0.2559  BLEU-2 Average: 0.2275  BLEU-3 Average: 0.2541  BLEU-4 Average: 0.2975 |
| Flickr8 | BLEU-1 count: 178  BLEU-2 count: 89  BLEU-3 count: 35  BLEU-4 count: 14 | BLEU-1 Average: 0.2773  BLEU-2 Average: 0.2501  BLEU-3 Average: 0.2901  BLEU-4 Average: 0.2608 |
| Resnet50 |
| **Transformers** | Inception V3 | Loss - SparseCategoricalCrossentropy  Optimizer – Adam  Epochs – 70  Training – 8000 images  Multi Head - 8 | Flickr8 | BLEU-1 count: 195  BLEU-2 count: 103  BLEU-3 count: 44  BLEU-4 count: 10 | BLEU-1 Average: 0.3115  BLEU-2 Average: 0.2738  BLEU-3 Average: 0.3011  BLEU-4 Average: 0.2824 |
| Resnet50 | Flickr8 | BLEU-1 count: 200  BLEU-2 count: 100  BLEU-3 count: 40  BLEU-4 count: 12 | BLEU-1 Average: 0.2798  BLEU-2 Average: 0.2540  BLEU-3 Average: 0.2973  BLEU-4 Average: 0.2985 |
| Loss - SparseCategoricalCrossentropy  Optimizer – Adam  Epochs – 35  Training – 25000 images  Multi Head – 8 | Flickr30 | BLEU-1 count: 177  BLEU-2 count: 82  BLEU-3 count: 22  BLEU-4 count: 7 | BLEU-1 Average: 0.2516  BLEU-2 Average: 0.2232  BLEU-3 Average: 0.2901  BLEU-4 Average: 0.2837 |

## CHALLENGES

Following are the key challenges corresponding mitigation plan we followed

|  |  |  |
| --- | --- | --- |
| Challenge | Description | Mitigation |
| Flickr30k data set processing | * Flickr30K is of size 4G * The number of files and size created an issue with processing while training the models | * Downloaded the image files on Google drive * Build the Numpy files from Pretrained Resnet50 on Google drive * Incrementally move the Numpy files to Collab and while training and execution |
| Processing Multiple epochs for Flickr30K | * Flickr30K is of size 4G * The number of files and size created an issue with processing while training the models * Performance of training- Each epoch runs ~10 min | * Train the models incrementally for 10-15 epoch at time and build save the weights * Start the next training with from previous training end point by loading saved weights |
| Evaluation of captions using Bleu score | * Improving evaluation criteria of Bleu score for the captions | * Blues score were calculated for Single , Bi gram, Tri gram and N Grams for values >0.1 and averages across all the caption generation for Test images |
| Transformers Implementation | * Additional MHA for Word embedding(captions) was part of the model which had to be removed from traditional approach of transformer implementation | * Captions were created processed only for Word embedding + Positional encoding and directly passed to Encoder decoder MHA block |
| Integration with Fast API /Cloud | * Limitation in Heroku for 512 * Tensorflow models need high configuration (> 1GB RAM) | * To deploy on GCP Cloud * Give higher configuration to deploy on cloud |
| Github upload for huge files( >25 MB) | More than 25 MB files were not opening on github | Deployed github lfs and uploaded files |

.

## Sample Auotmated Captions generated

|  |  |  |
| --- | --- | --- |
| LSTM(Flickr8) | Attention(Flickr8) | Transformer(Flickr8) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* Flikr30k results

|  |  |
| --- | --- |
| Attention(Flickr30k) | Transformers(Flickr30k) |
|  |  |
|  |  |
|  |  |
|  |  |

## Data Set

A number of datasets are used for training, testing, and evaluation of the image captioning methods. The datasets differ in various perspectives such as the number of images, the number of captions per image, format of the captions, and image size. Three datasets: Flickr8k, Flickr30k, and MS COCO Dataset are popularly used.

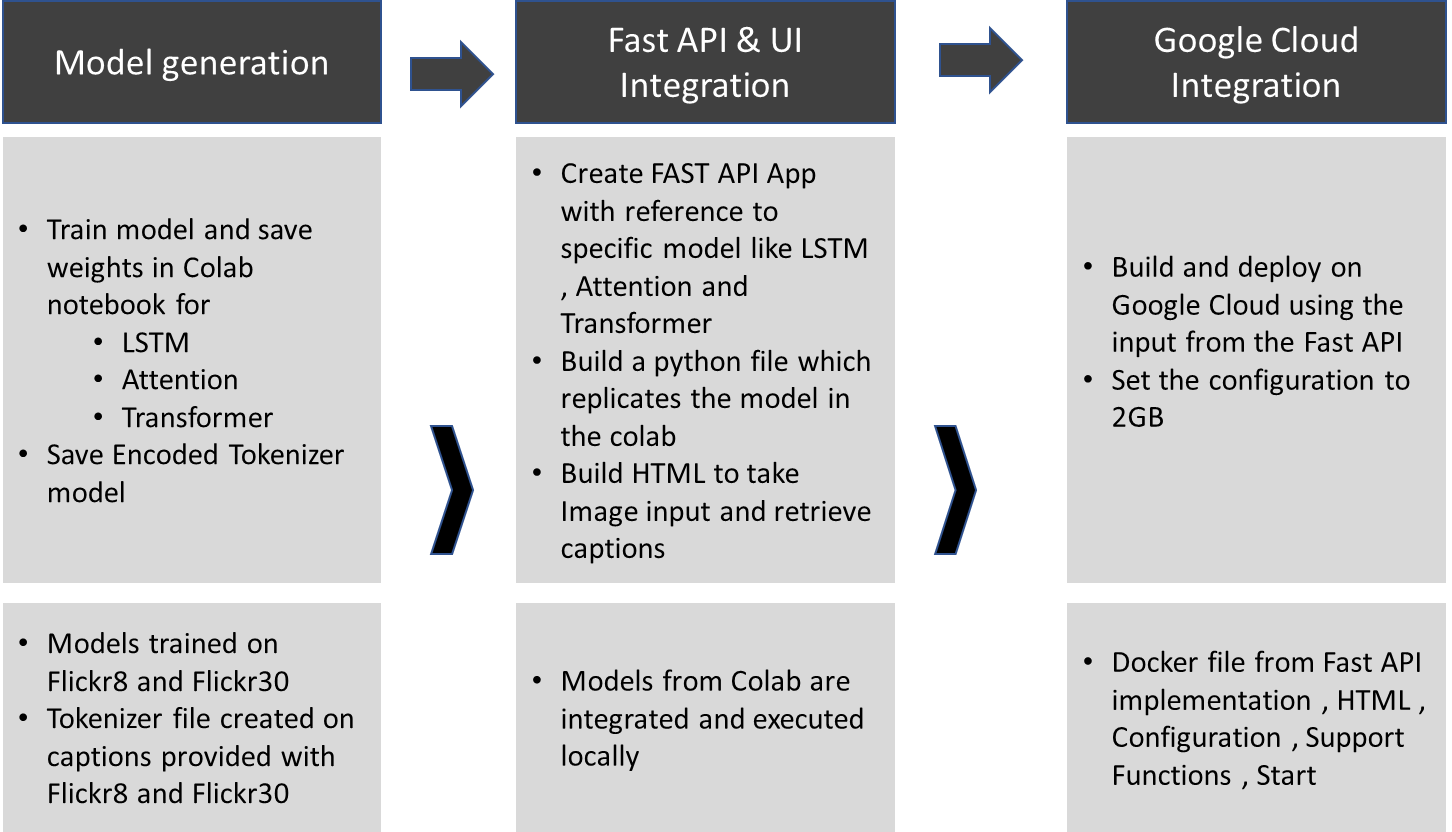
In the Flickr8k dataset, each image is associated with five different captions that describe the entities and events depicted in the image that were collected. By associating each image with multiple, independently produced sentences, the dataset captures some of the linguistic variety that can be used to describe the same image.

Flickr8k is a good starting dataset as it is small in size and can be trained easily on low-end laptops/desktops using a CPU.

We have used Flickr8 and Flick30

* Flickr8k
  + Flick8k\_Dataset/ :- contains the 8000 images
  + Flick8k\_Text/
    - Flickr8k.token.txt:- contains the image id along with the 5 captions
    - Flickr8k.trainImages.txt:- contains the training image id’s
    - Flickr8k.testImages.txt:- contains the test image id’s
* Flickr30
  + Flickr30\_Dataset/: contains the 30000 images
  + Captions.txt/- Expert captions for all 30000 images

# Deployment



|  |
| --- |
| 1)Create application in Google cloud and install Cloud SDK  2)Open command prompt on local machine and enter below commands:  cd C:\Model Deployment\deploy\_aic\_lstm-main  gcloud init --it sets your gmail  gcloud config get-value project  gcloud builds submit --tag gcr.io/deploy-aic-lstm/aiclstm --deploy-aic-lstm is project id created in cloud and aiclstm is service name we intend to give  3)Got to Gcloud UI  ->cloud run  ->Create service aiclstm and enter details |

# ConclusioN

To be determined based on the final solution and results

# References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio and Dumitru Erhan, “Show and tell: A Neural Image Caption Generator,” in Proceedings of the IEEE

conference on computer vision and pattern recognition, pp. 3156–3164, Boston, MA, USA, 2015.

[2] Yan Chu , Xiao Yue , Lei Yu, Mikhailov Sergei and Zhengkui Wang, “Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention,” This

is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any

medium, provided the original work is properly cited..

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin, “Attention Is All You

Need,” in Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[4] https://www.analyticsvidhya.com/blog/2020/11/attention-mechanism-for-caption-generation/.

[5] https://www.analyticsvidhya.com/blog/2021/01/implementation-of-attention-mechanism-for-caption-generation-on-transformers-using-tensorflow/