Image Captioning by vision encoder decoder models

CSCE 5214 Fall 2021
Project – 2

Aditya Pujari	Hemanth Reddy Yerramreddy	Praveen Kumar Somara	Brinda Potluri	Chandrakanth Mandalapu
11491374	11505484	11525451	11526591	11509665

Abstract

- This project focuses on captioning an image that has been provided as an input.
- A web application using AI would be able to provide in-context captioning to the inputted image
- The project would be deployed on Heroku, and Streamlit for the web-Framework.

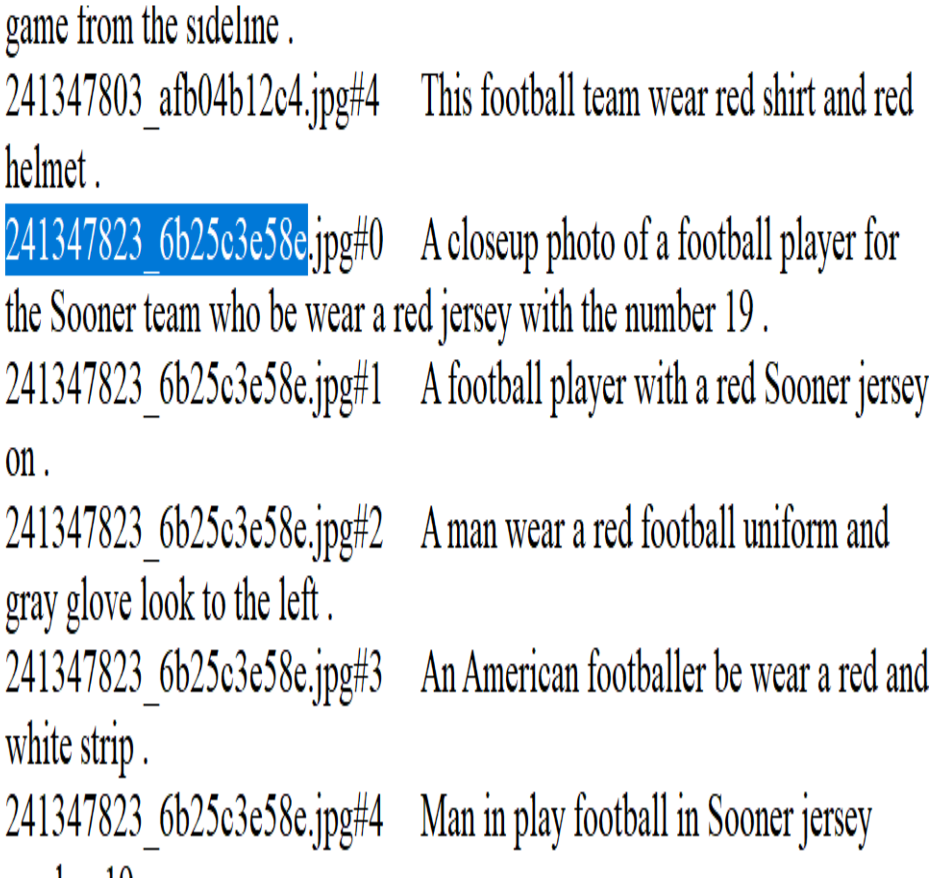
Agenda

- Data Set
- Design and Milestones
- Vision-To-Text Encoder-Decoder framework(Vit), Encoder , Decoder
- Training and Testing
- Modules-to-be-Completed
- References

Dataset

- The data for this project is taken from [Flickr Image dataset]
- The dataset consists of 31.8K images in total.
- The dataset also contains the text files of captions to train the model and a separate test-data to test the model
- The training data consists of 24000 images, Test data consists of 8000. The remaining amount of data would be used validation purposes.

Sample Dataset



game from the sideline .

241347803_afb04b12c4.jpg#4 This football team wear red shirt and red helmet .

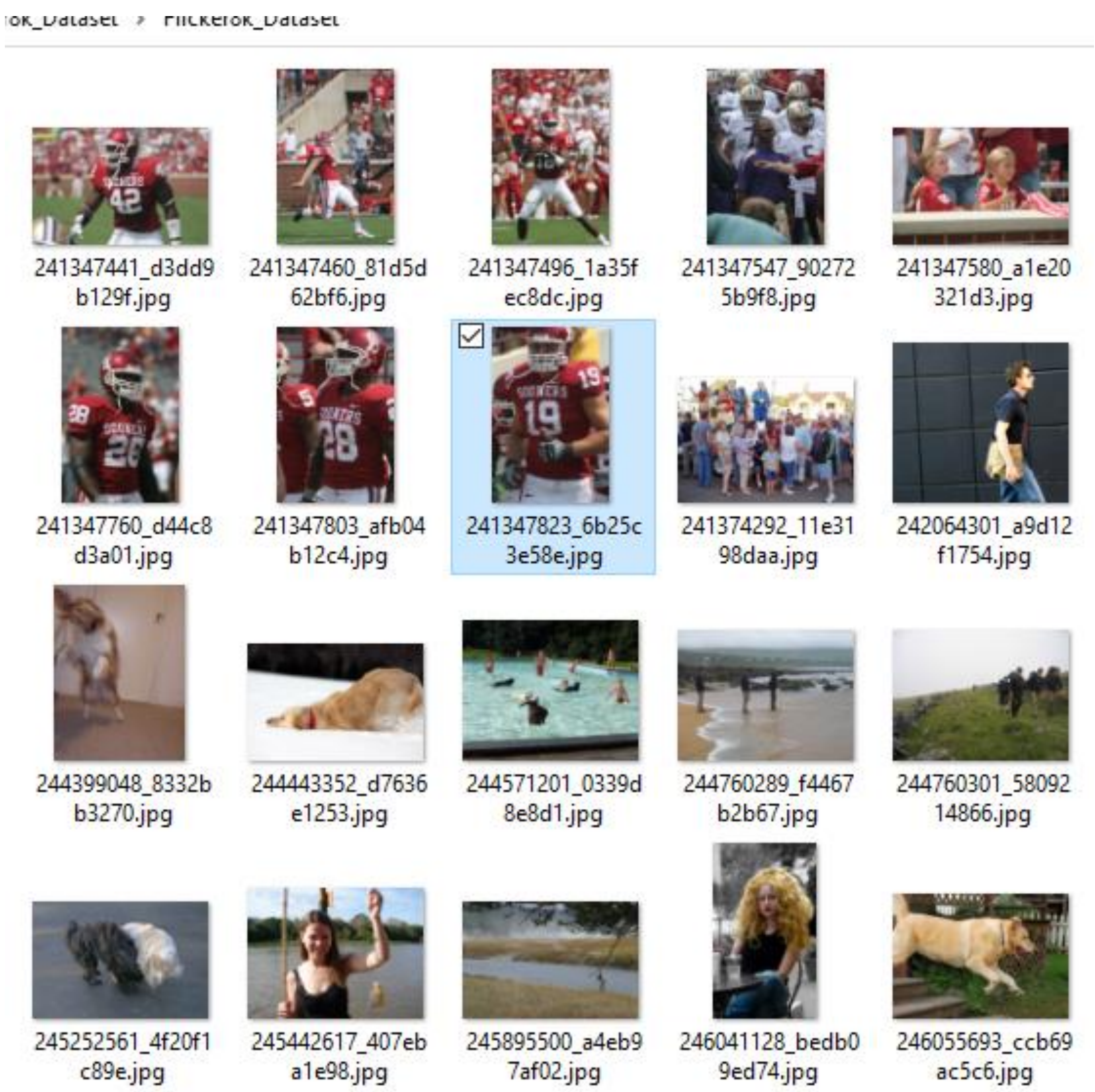
241347823_6b25c3e58e.jpg#0 A closeup photo of a football player for the Sooner team who be wear a red jersey with the number 19 .

241347823_6b25c3e58e.jpg#1 A football player with a red Sooner jersey on .

241347823_6b25c3e58e.jpg#2 A man wear a red football uniform and gray glove look to the left .

241347823_6b25c3e58e.jpg#3 An American footballer be wear a red and white strip .

241347823_6b25c3e58e.jpg#4 Man in play football in Sooner jersey



ok_Dataset › Flicker8k_Dataset

241347441_d3dd9b129f.jpg
241347460_81d5d62bf6.jpg
241347496_1a35fec8dc.jpg
241347547_902725b9f8.jpg
241347580_a1e20321d3.jpg
241347760_d44c8d3a01.jpg
241347803_afb04b12c4.jpg
☑ 241347823_6b25c3e58e.jpg
241374292_11e3198daa.jpg
242064301_a9d12f1754.jpg
244399048_8332bb3270.jpg
244443352_d7636e1253.jpg
244571201_0339d8e8d1.jpg
244760289_f4467b2b67.jpg
244760301_5809214866.jpg
245252561_4f20f1c89e.jpg
245442617_407eba1e98.jpg
245895500_a4eb97af02.jpg
246041128_bedb09ed74.jpg
246055693_ccb69ac5c6.jpg

Design and Milestones

- The project will be done in Python Programming language.
- Data pre-processing would be done to resize images to 2048 for efficient runtime and smoother web-experience
- The model we are training in is a shared google colab where all team members can share their modules.
- The model would be trained using pre-trained ViT models from hugging face for efficient training time without losing out on accuracy.

Programming Language: Python

Model Training: Jupyter Notebook (Google Colab pro -> P100 GPUs)

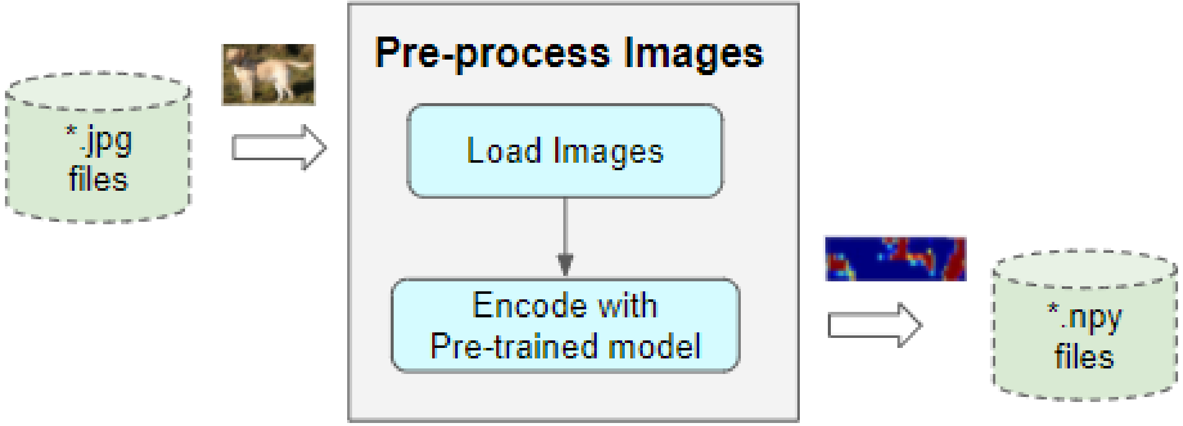
Dataset: Kaggle

Server (Cloud Platform): Heroku

Web App Framework: Streamlit

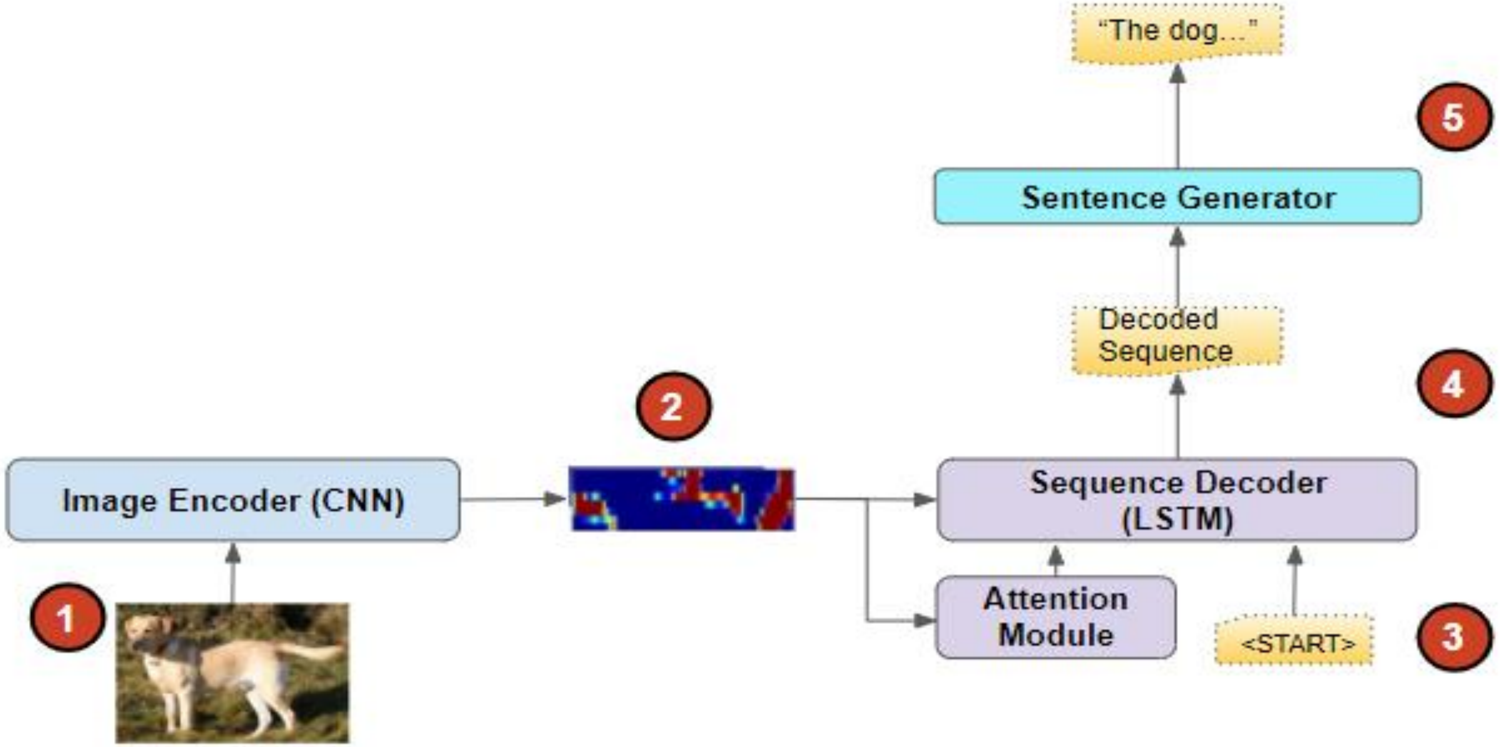
Image Pre-Processing

- The primary segment consists of a collection of CNN layers that constantly remove the relevant highlights from the image in order to provide a reduced element map representation.
- The Classifier, which is made up of a series of Linear layers, is the next part. It takes an image with a map and forecasts a class (such as canine, automobile, or house) in which the element belongs.



Vision-To-Text Encoder-Decoder framework

- ViT models exceed the present state-of-the-art (CNN) in terms of computing efficiency and accuracy by almost 4 times

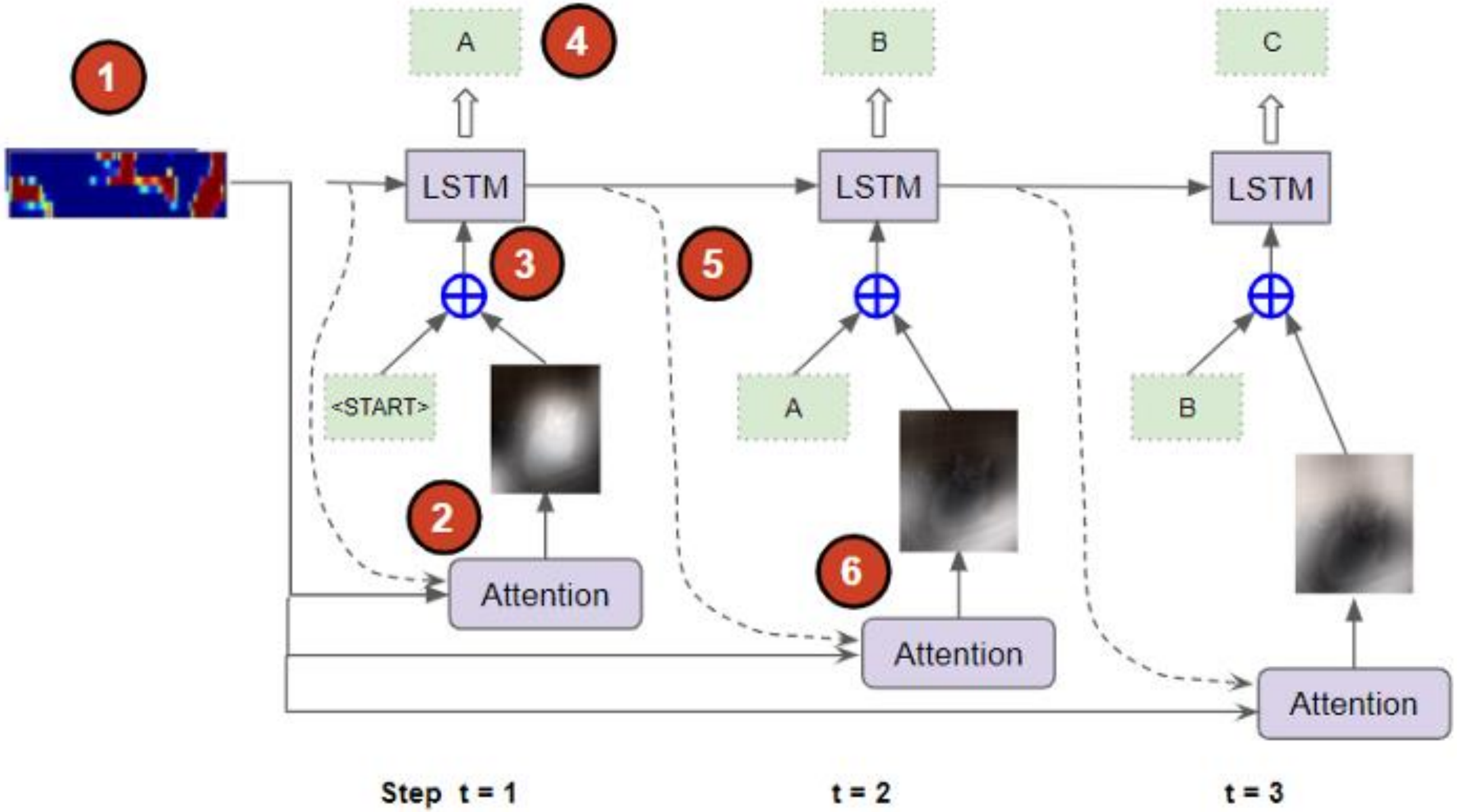


Encoder and Decoder

- The Encoder here is fairly basic, as it is done by the pre-prepared Inception model. It consists of a Linear layer that provides the Decoder with the pre-encoded visual highlights.
- The project would use a Sequence Decoder (GRU) along with attention model
- After passing via an Embedding layer, the captions are passed in as the input in the Sequence Decoder .
- The Attention Module registers the weighted Attention Score based on the encoded image from the Encoder and the hidden state from the Sequence Decoder.

Training and Testing

- We are using pre-trained models to train our model.
- For the first stage, we use move to figure out how to pre-process the raw images with a CNN-based network that has already been trained. This takes the image as input and outputs encoded image vectors that capture the image's main features. We don't need to do anything else to prepare this network.
- Using the pre-prepared model, we extract picture features from the test images.
- Greedy Search is used to predict the output by selecting the term with the highest probability at each timestep.


1
2
3
4
5
6
A
B
C
LSTM
LSTM
LSTM
<START>
A
B
Attention
Attention
Attention
Step t = 1
t = 2
t = 3

Deployment

- ➢ We are using Heroku as the deployment server to test the model trained on real time images.
- ➢ We already created the project and designed the interface.
- ➢ We still need to map the prediction phase to the deployment server.

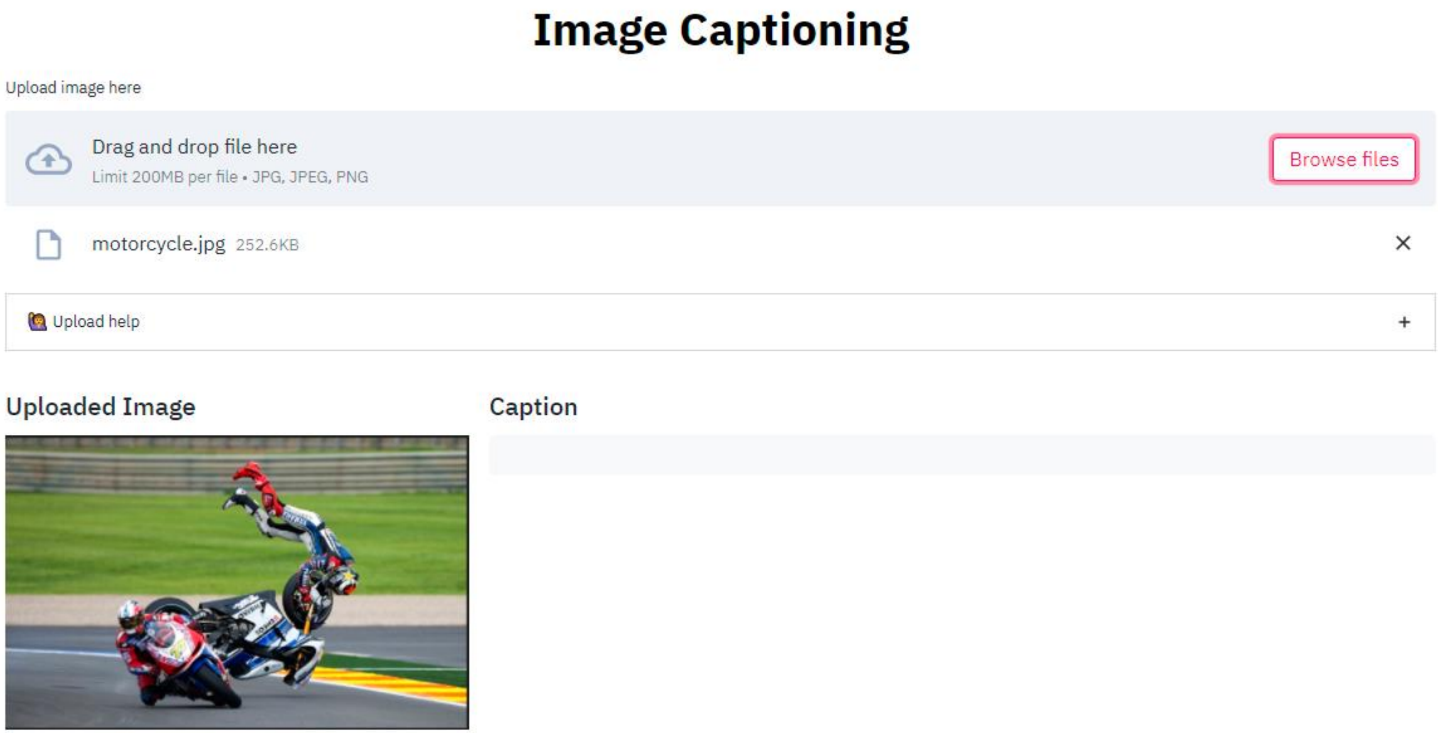


Image Captioning

Upload image here

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

motorcycle.jpg 252.6KB

Uploaded Image

Caption

Modules-to-be-Completed

- Complete the full training of the model in the pre-trained models.
- Integrating the model with the frontend webserver.
- Evaluating test cases and the model's performances on various pre-trained models. to check its reliability

References

1. https://towardsdatascience.com/image-captions-with-attention-in-tensorflow-step-by-step-927dad3569fa
2. https://towardsdatascience.com/image-captions-with-deep-learning-state-of-the-art-architectures-3290573712db
3. http://www.jair.org/papers/paper3994.html

Thank You