CEC21

OPEN SOURCE TECHNOLOGIES

Submitted By : Yatharth Babbar (2017UCO1578)
Abhishek Singh (2017UCO1582)
Navjot Singh (2017UCO1600)

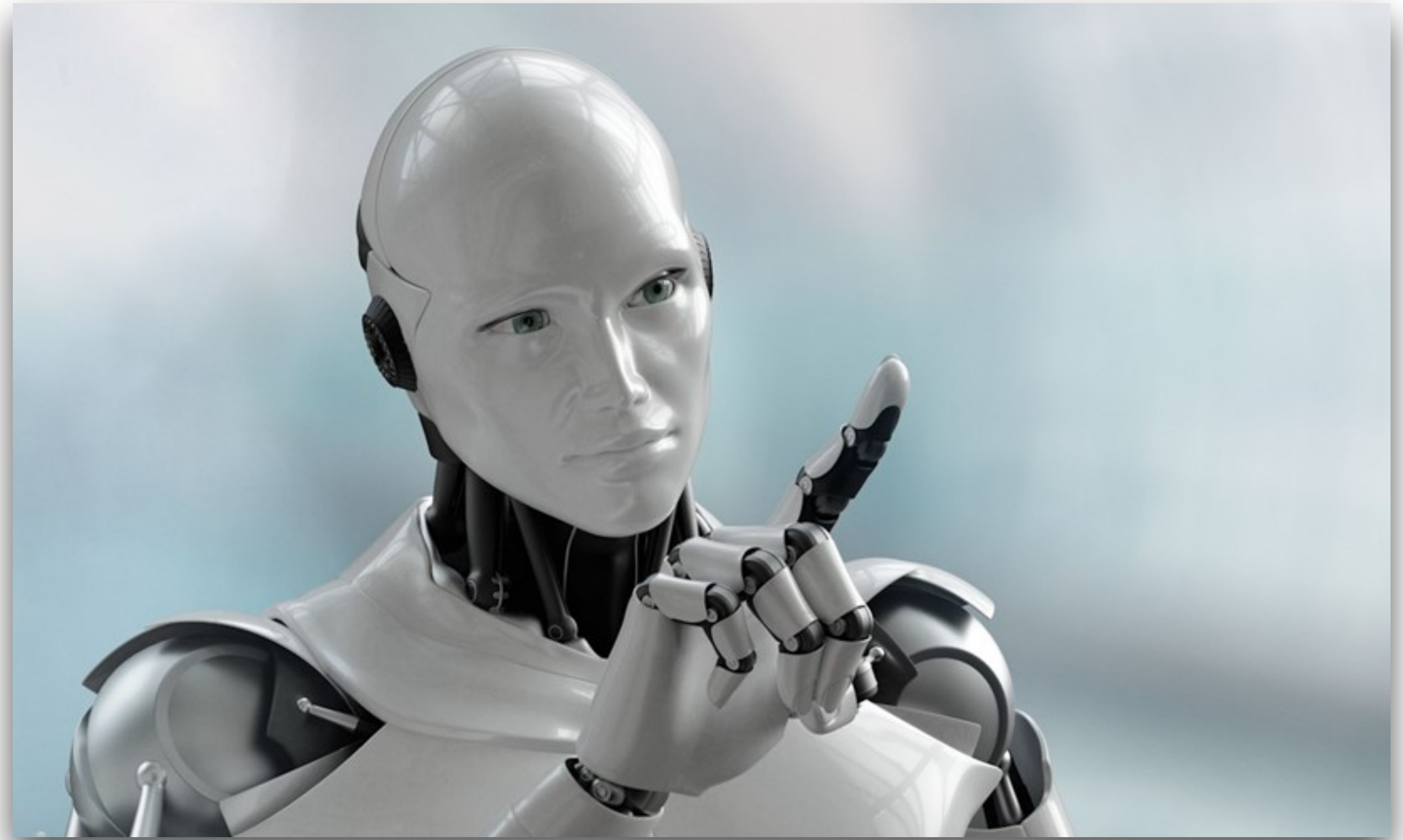
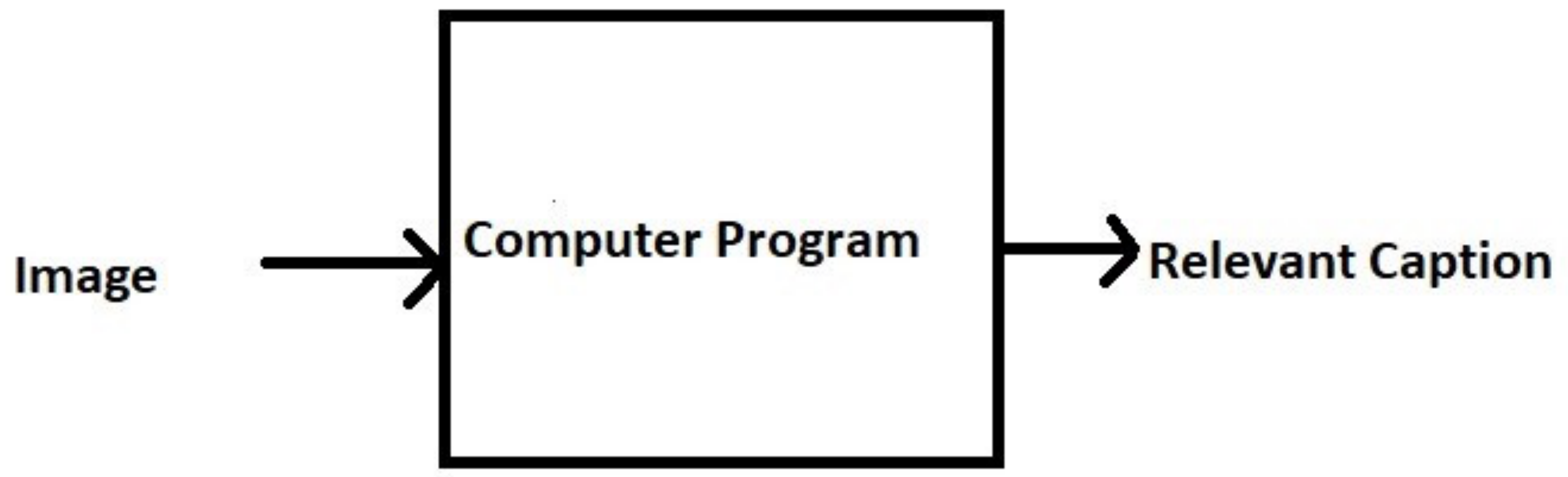


Image Captioning Bot

GOAL

- Just prior to the recent development of Deep Neural Networks captioning an image was inconceivable even by the most advanced researchers in Computer Vision. But with the advent of Deep Learning this problem can be solved very easily if we have the required dataset.

- Deep Learning can be used to solve this problem of generating a caption for a given image, hence the name **Image Captioning.**



APPLICATIONS

- Self driving cars
- Visual Aid for the blind - Guide them while travelling
- Image Search - Search an image based on its caption
- Automatic Surveillance - CCTV Cameras

Prerequisites

- Neural Networks (Multilayer Perceptron, Convolutional Neural Networks, Recurrent Neural Networks)
- Language Model (Natural Language Processing)
- Word Embeddings
- Transfer Learning
- Python syntax and data structures
- Libraries like keras, tensorflow, numpy, nltk, pandas, etc

Steps Involved

- Data Collection
- Understanding the data
- Data Cleaning
- Loading Training set
- Data preprocessing - Captions and Images
- Data Preparation using generator function
- Word Embeddings
- Model Architecture
- Prediction

Data Collection

- There are many open source datasets available for this problem, like Flickr 8k (containing 8k images), Flickr 30k (containing 30k images), MS COCO (containing 180k images), etc.
- We have used Flickr 8k Dataset available at https://www.kaggle.com/shadabhussain/flickr8k
- This dataset contains 8000 images each with 5 captions (total 40000).

Understanding the Data

- In the dataset, along with the images, there are some text files related to images.
- One of the files is “Flickr8k.token.txt” which contains the name of each image along with its 5 captions.
- We create a dictionary named “descriptions” which contains the name of image as key and a list of 5 captions for the corresponding image as value.

Data Cleaning

- We performed some basic cleaning on text(captions) like lower-casting all words, removing special tokens and numbers.
- Create a vocabulary of all unique words present across the 40000 captions.
- We consider only those words that occur at least 10 times in the entire corpus of 40000 captions.
- Reducing the vocabulary size results in less overfitting and less computation.
- Total Words - 373837
  Unique Words - 8424
  Filtered Unique Words - 1845

Data Preprocessing (Images)

- Input to our model is an image. We convert every image into a fixed sized vector which can be fed as input to the neural network.
- For this purpose, we opt for transfer learning by using ResNet-50 model (pre-trained model). This model was trained on Imagenet dataset to perform image classification. We just remove the last softmax layer from the model and extract the feature vector for every image.
- We pass every image to this model to get the corresponding feature vector. A dictionary named “encoding_train” maps the image_id(key) to the feature vector(value).
- Similarly, we create a dictionary “encoding_test” for test images.

Data Preprocessing (Captions)

- We represent each unique word in the vocabulary by an integer(index).

- All the 1845 unique words in the corpus will be represented by an integer between 1 and 1845.

- We create two dictionaries namely "idx_to_word" (returns the word at a particular index) and "word_to_idx" (returns the index of a particular word).

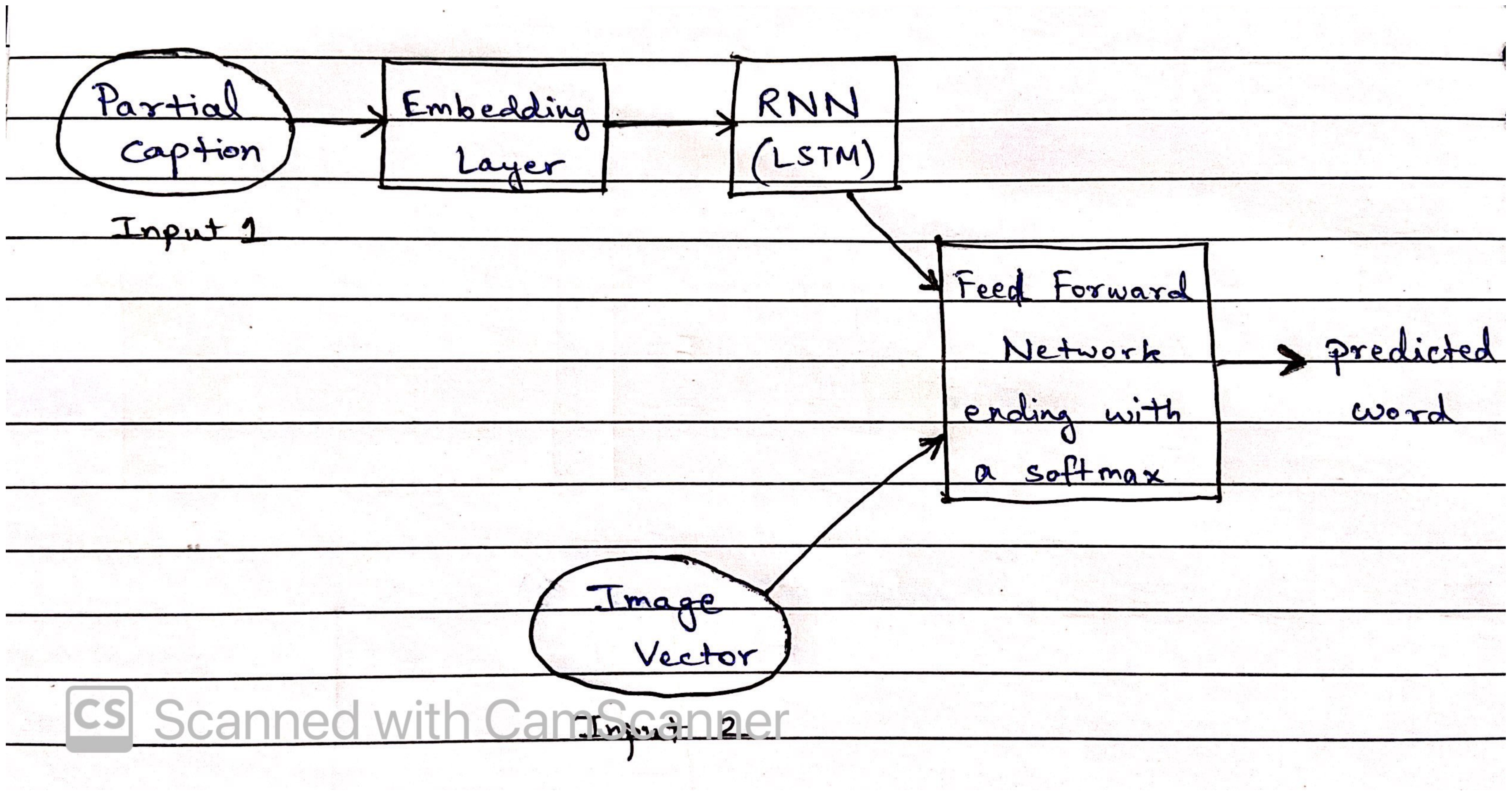
- We also calculate the maximum length of a caption (35 - for our dataset).

Data Preparation

- We give an image as input to the model and expect a caption (or sentence) as output.
- But, the model cannot generate entire sentence at once.
- We also provide a partial caption (read using Recurrent Neural Networks) as input to the model along with the image. A single word in the vocabulary is given as output which is appended to the partial caption and fed to the model again. Like this, we generate the entire sentence or caption.

Model Architecture

- High Level Architecture of the model



- Model Summary

```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_3 (InputLayer)             (None, 35)            0
____________________________________________________________________________________________________
input_2 (InputLayer)             (None, 2048)          0
____________________________________________________________________________________________________
embedding_1 (Embedding)          (None, 35, 50)        92400       input_3[0][0]
____________________________________________________________________________________________________
dropout_1 (Dropout)              (None, 2048)          0           input_2[0][0]
____________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 35, 50)        0           embedding_1[0][0]
____________________________________________________________________________________________________
dense_1 (Dense)                  (None, 256)           524544      dropout_1[0][0]
____________________________________________________________________________________________________
lstm_1 (LSTM)                    (None, 256)           314368      dropout_2[0][0]
____________________________________________________________________________________________________
add_17 (Add)                     (None, 256)           0           dense_1[0][0]
                                                                   lstm_1[0][0]
____________________________________________________________________________________________________
dense_2 (Dense)                  (None, 256)           65792       add_17[0][0]
____________________________________________________________________________________________________
dense_3 (Dense)                  (None, 1848)          474936      dense_2[0][0]
====================================================================================================
Total params: 1,472,040
Trainable params: 1,472,040
Non-trainable params: 0
__________________________________________________________________________________________________
```

- LSTM (Long Short Term Memory) is a specialised Recurrent Neural Network to process the partial captions.

- The weights of the model will be updated using backpropagation algorithm and the model will learn to output a word , given an image feature vector and a partial caption.

Prediction

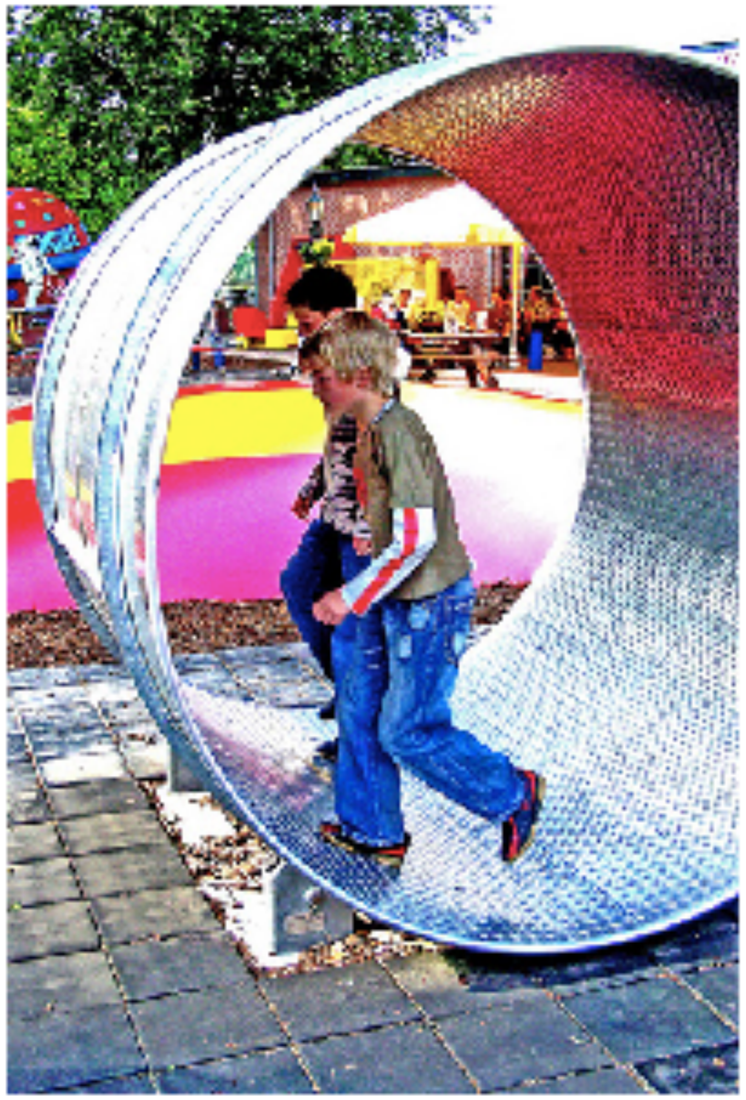
- No model in the world is perfect and so, our model also makes mistakes. Eg. Colors getting mixed with background, incorrect grammar, etc.

- It is important to understand that images used for testing must be semantically related to those used for training the model. For example, if we train our model on images of cats, dogs, etc, we must not test it on images of air planes, waterfalls, etc.

- When the distribution of test and train sets is different, no Machine Learning model will give good performance.

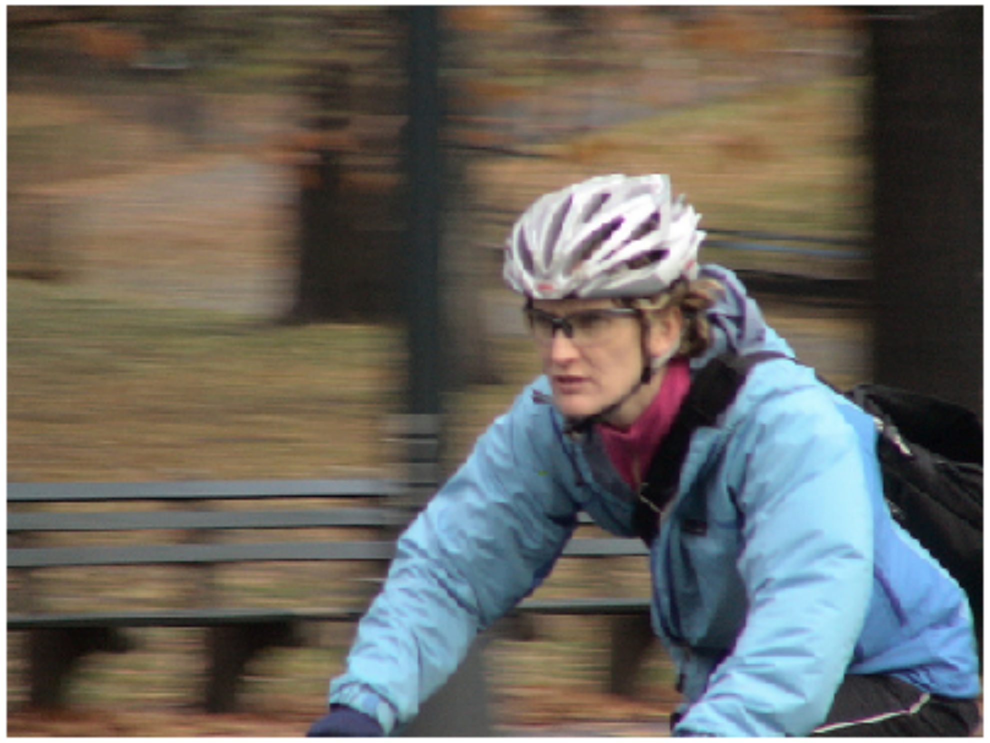
- Some of the captions generated by the model are shown below :

little boy in green shirt is jumping on swing



dog running on the beach

man in blue shirt and helmet is riding bicycle



Modifications

This is just a first-cut solution and a lot of modifications can be made to improve the solution like :

- Using a larger dataset
- Changing the model architecture
- Doing more hyper parameter tuning
- Use cross validation set to understand overfitting

Integrating model with WEB using FLASK

- Deployment of machine learning models means making the model available to the end users or systems.

- We have used Flask API for this purpose.

- Some of the advantages of using FLASK are :
  1. Easy to use
  2. Built in development server and debugger
  3. Integrated unit testing support
  4. RESTful request dispatching
  5. Extensively Documented

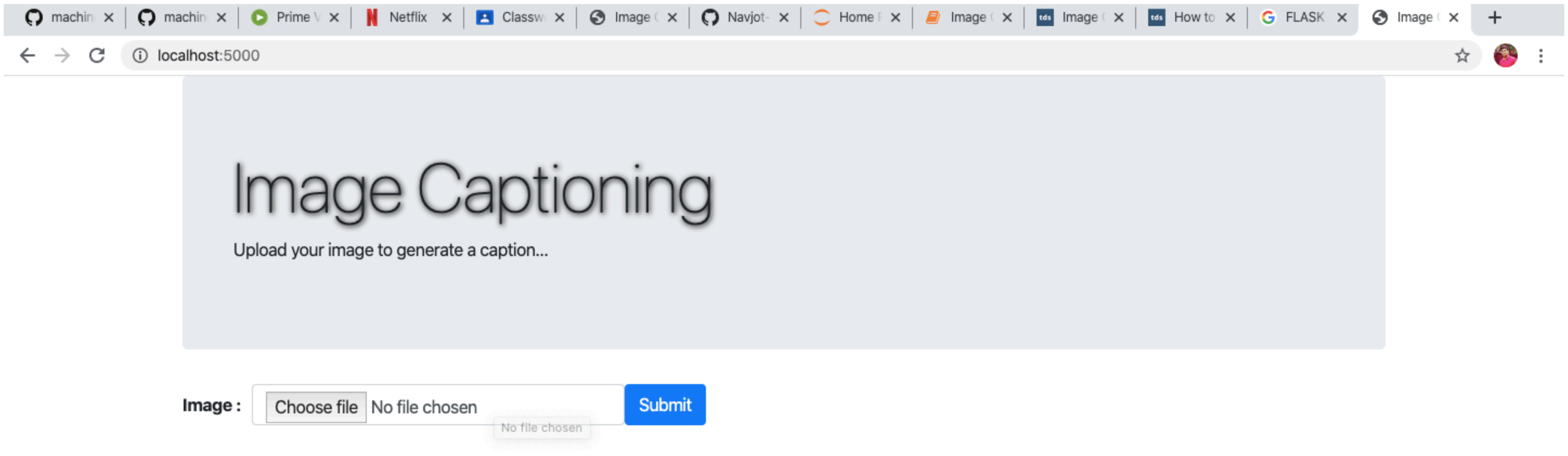


Image Captioning

Upload your image to generate a caption...

Home Page

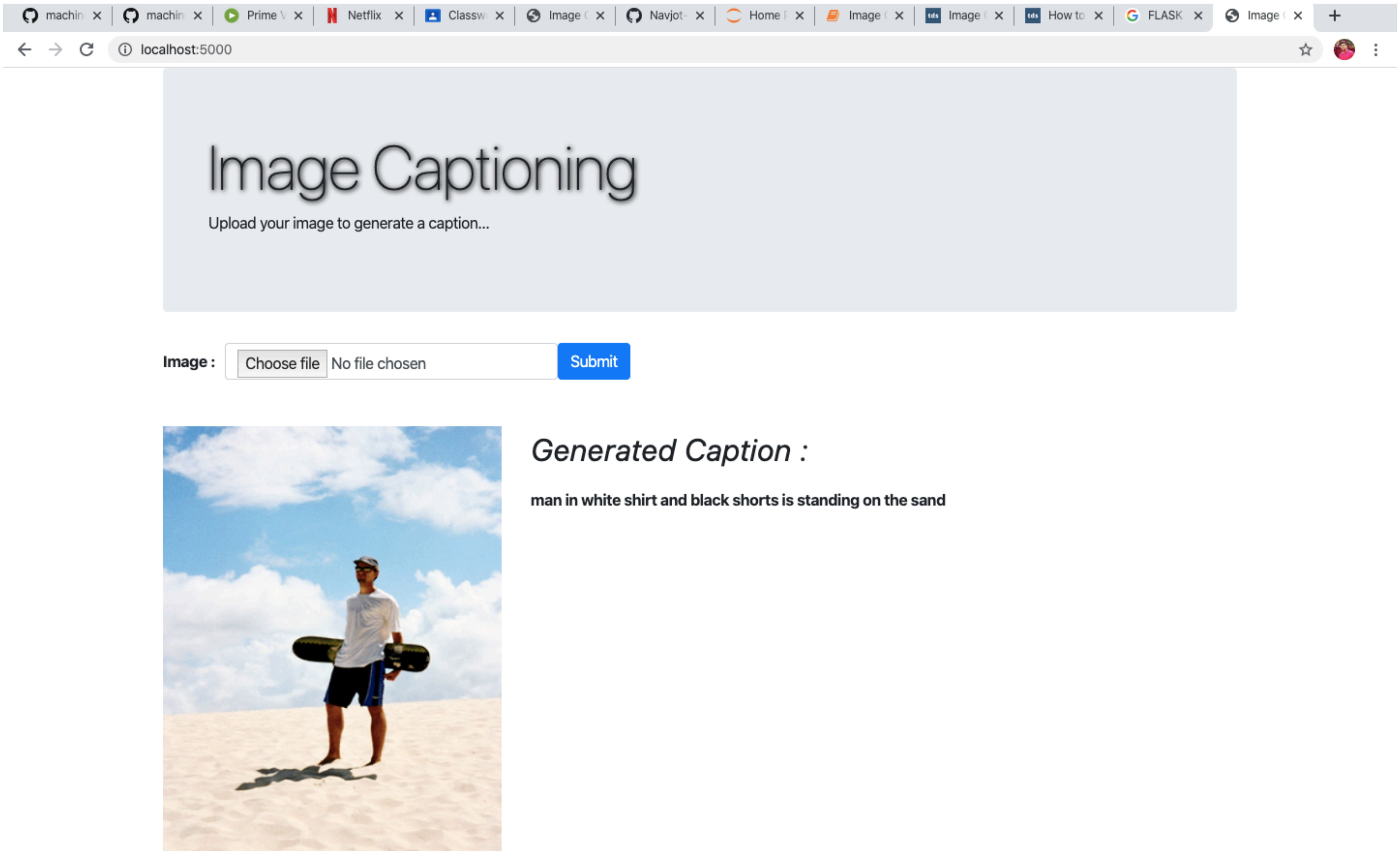


Image Captioning

Upload your image to generate a caption...

Generated Caption :

man in white shirt and black shorts is standing on the sand

Predicted Caption for Image 1

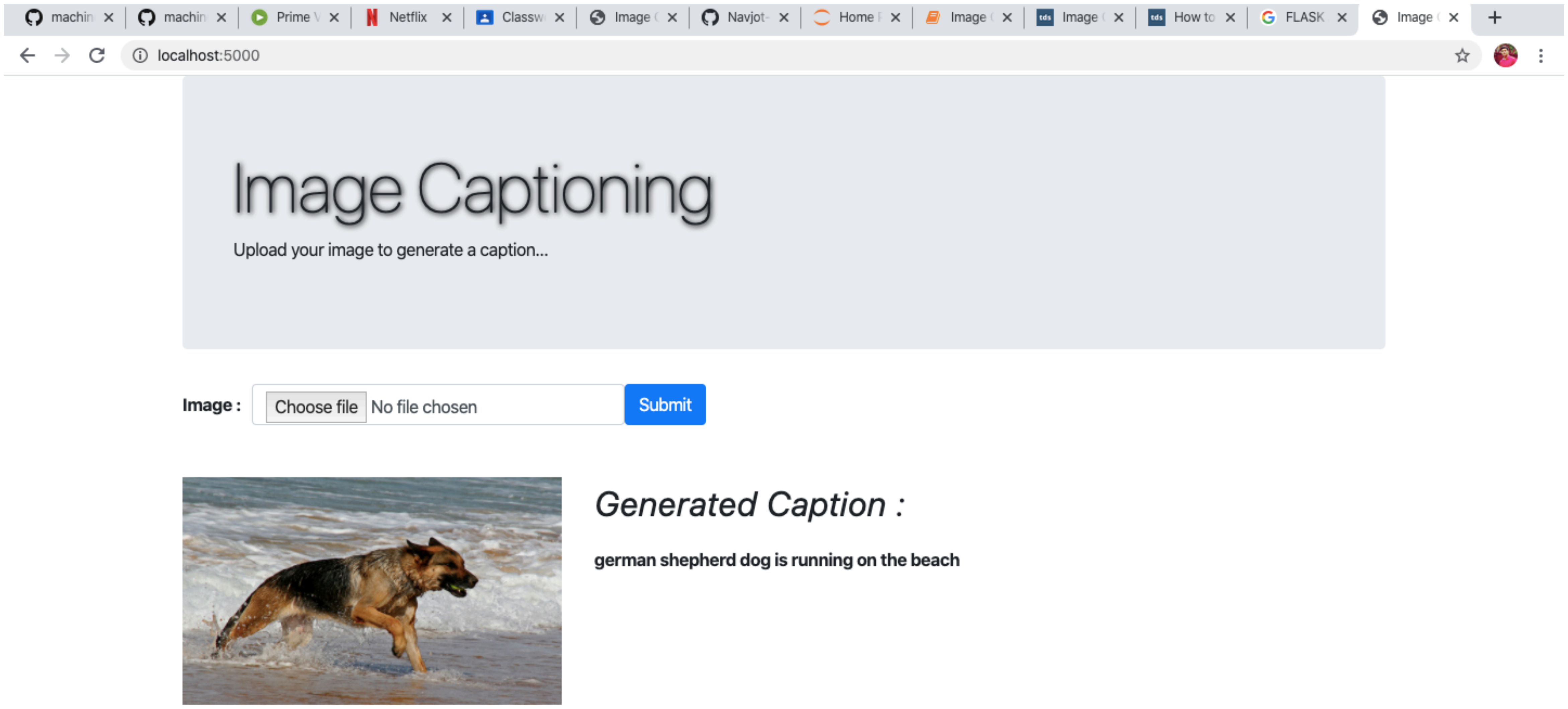


Image Captioning

Upload your image to generate a caption...

Generated Caption :

german shepherd dog is running on the beach

Predicted Caption for Image 2

References

1. https://cs.stanford.edu/people/karpathy/cvpr2015.pdf

2. https://arxiv.org/abs/1411.4555

3. https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/

4. https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8

5. https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8fe34d4

THANK YOU