Image Caption Generator User Manual

1 Introduction This document discusses three issues.

- How to download the project and set up the environment to run the project.
- How to predict the caption of any given image.
- How to reproduce the results.

2 Setting Up the Project and Environment

Download the folder named COCO from this linkhttps://drive.google.com/drive/folders/1DSTSjVSSYdCqm21Veo_oIF4L_bUnzTEs?usp=sharing

Now the project folder will contain e folder named COCO which will contain the folders features, models, output and text.

For creating virtual environment we have used Anaconda. First download and install anaconda ***(Python 3.7 version)*** from https://www.anaconda.com/download/.

Then navigate to the folder Image Caption Generator and run the file script.sh located inside the folder using the command ***(only in Linux) ./script.sh***
Finally, activate the new environment created by the script.sh file by using the command

conda activate caption generator

To install everything in Windows, follow these steps:

- Open anaconda prompt and create a conda virtual environment using the following command
  conda create -n caption generator python=3.6
- Activate the new environment
  conda activate caption generator

- Install all the dependencies using these commands
  conda install -c anaconda scikit-learn
  conda install -c conda-forge matplotlib
  conda install -c anaconda pillow
  conda install -c anaconda nltk
  conda install -c anaconda pydot
  conda install -c anaconda tensorflow-gpu
  conda install -c anaconda keras-gpu

To check if everything has been installed correctly, navigate to the downloaded directory named Image Caption Generator and run the command ***python version.py***

3 Generating Caption of an Image

To generate caption of any image:

- Put the image in the ***Image Caption Generator/Generator***directory.
- Run the command ***python Generate Caption.py***
- When input is prompted write the image file's name (say example.jpg) as input.
- The image will be shown with the generated caption on top. You can save the image with caption from here. The caption will also be printed in the command prompt.

4 Reproducing The Results

All required image and text data has already been pre-processed. To reproduce the result you need to train the model using the command

python Train.py

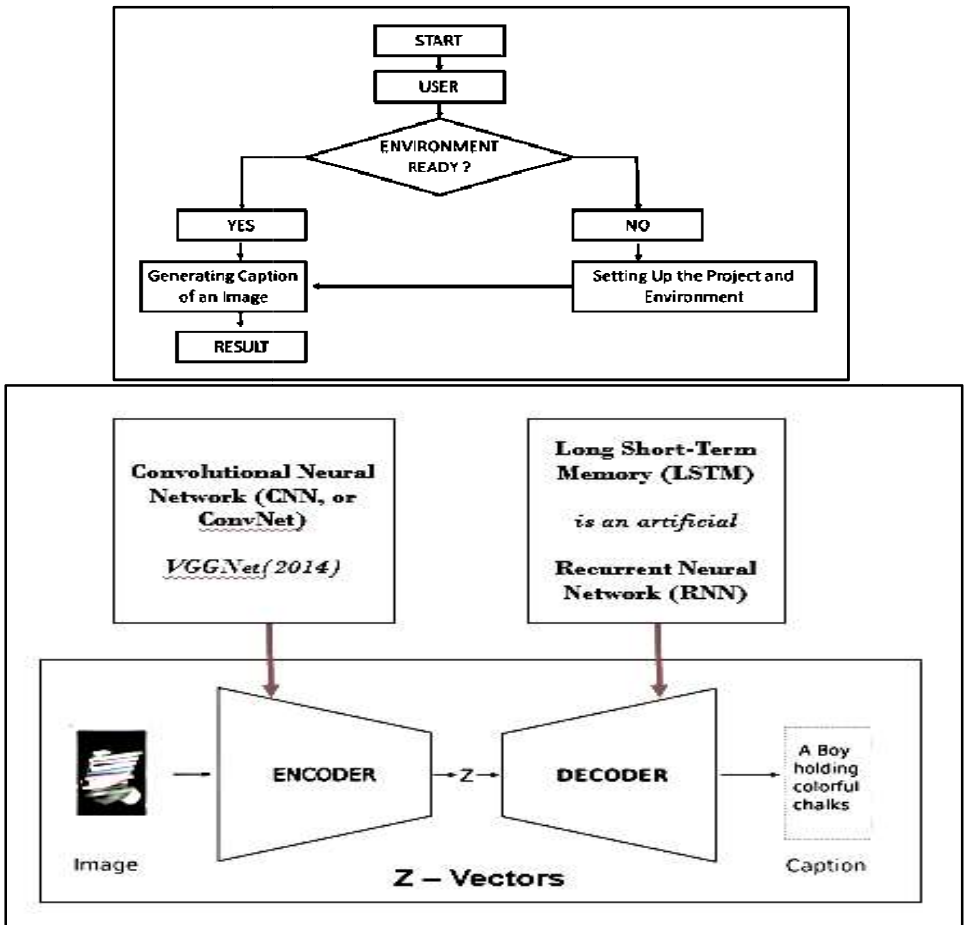
To evaluate the trained model on test data, run the command

python Evaluate Model.py

5 Results

All the results and generated captions are available in the folder named Results.

Block Diagram



Environment

Anaconda (is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system conda.)

- *Feature extractors:VGG-19 Net;*
- *Language models:LSTM; RNN; CNN;*
- *Methods: Encoder-decoder;*
- *Results on datasets:MS COCO; Flickr30k; Flickr8k;*
- *Evaluation metrics:BLEU-1; BLEU-2; BLEU-3; BLEU-4;*

Used libraries and tools:

1. tensorflow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.
2. Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.
3. OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez.

4. NLTK: The Natural Language Toolkit, or more commonly NLTK is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
5. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
6. Pillow is a Python Imaging Library (Fork). Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors.
7. numpy main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of non-negative integers. In NumPy dimensions are called axes.
8. h5py uses straightforward NumPy and Python metaphors, like dictionary and NumPy array syntax. For example, you can iterate over datasets in a file, or check out the .shape or .dtype attributes of datasets.

DATA SETS:

1. Flickr8k
2. Flickr30k
3. MS COCO

- Training Dataset: The sample of data used to fit the model.
- Validation Dataset: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.
- Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

Dataset: **MS COCO**

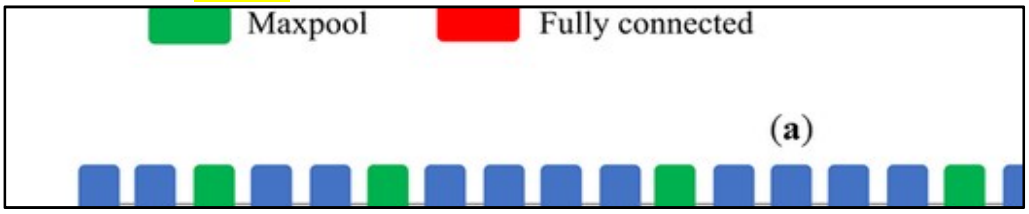
- Captions of test images (around 40k) are not public
- We randomly chose 35k images as training set
- 5k images for test set
- 10k images for validation set

Dataset:**Flickr8k**

- There are 8092 JPEG photos in this collection.
- A training dataset (6,000 photos), a development dataset (1,000 pictures), and a test dataset are all included in the dataset (1,000 images).

Preprocessing: **Image**

- All images were resized to 224x224 size
- Pre-trained VGG-19 model was used to extract features



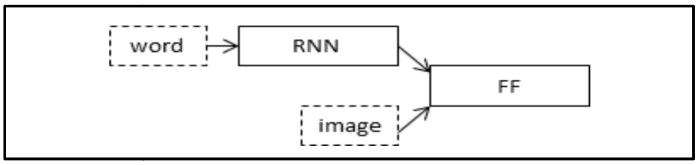
- Output of the layer before last layer was chosen as features
- Each image was converted to 4096 size vector

Preprocessing: **Text Data**

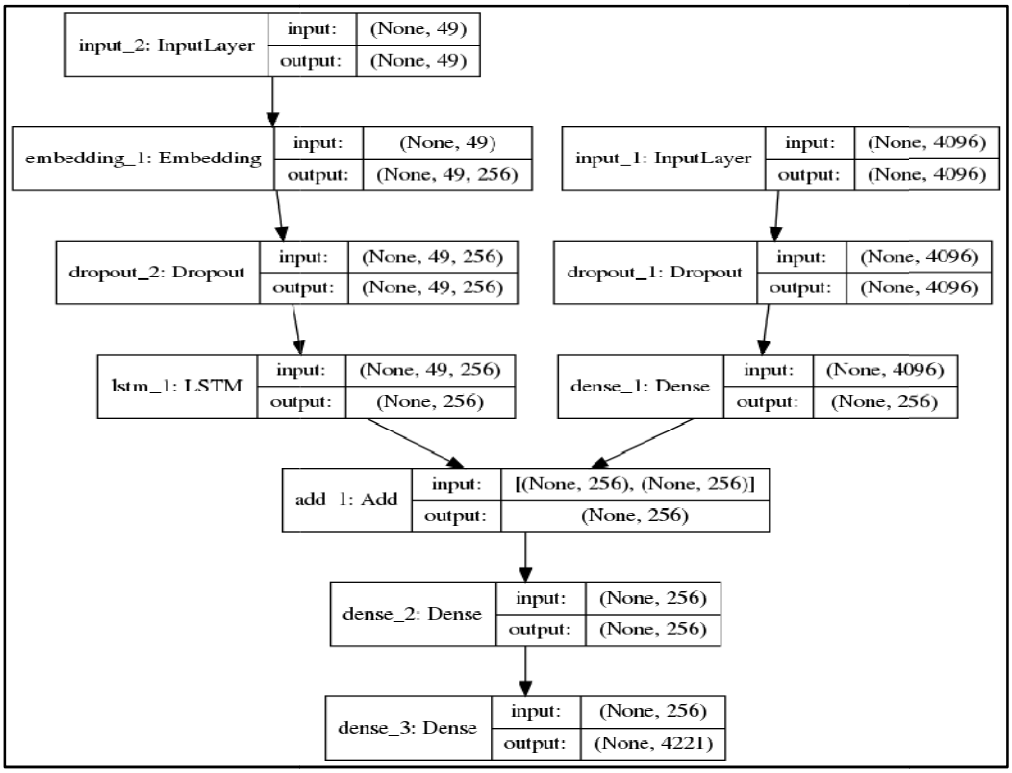
- Firstly, text data was cleaned
- New lines within captions were removed
- All punctuation marks were removed
- At this stage vocabulary size was around 29k (29028)
- Only those words were kept which appeared more than 10 times in the corpus
- Final vocabulary size was 7016

Defining Model

We chose to implement the merge model described in the referenced paper (***Marc Tanti et. al***).



- **Photo Feature Extractor:** VGG-19 (CNN) model was used to extract features.
- **Sequence Processor:** Word embedding layer for handling the text input.
- **Decoder:** Both the feature extractor and sequence processor output a fixed-length vector. These are merged together and processed by a Dense layer to make a final prediction.



Training: **Sample Generation**

X1(Image)	X2 (Text Sequences)	Y(Word)

Image	startseq,	little
Image	startseq, little	girl
Image	startseq, little, girl	running
Image	startseq, little, girl, running	in
Image	startseq, little, girl, running, in	field
Image	startseq, little, girl, running, in, field	endseq

Training

- Workstation:
- Total Epoch: 10
- Training Time:
- Batch Size: All samples generated from 8 images

Evaluation: **About BLEU**

- We used **BLEU** (**bilingual evaluation understudy**) as performance metrics
- BLEU-N: Match of n-tuple in reference and test corpus
- BLEU-4 is standard in image caption related literature

Evaluation: **Our Result (Flickr8k)**

- BLEU-1: 0.581075
- BLEU-2: 0.359521
- BLEU-3: 0.266660
- BLEU-4: 0.143680

Evaluation: **Our Result (Flickr30k)**

- BLEU-1: 0.548999
- BLEU-2: 0.311599
- BLEU-3: 0.224285
- BLEU-4: 0.111142

Evaluation: **Our Result (MS COCO)**

- BLEU-1: 0.581075
- BLEU-2: 0.359521
- BLEU-3: 0.266660
- BLEU-4: 0.143680

Issues Faced:

- High Computation Cost
- Better approach required for reducing vocabulary size
- Due to time constraint could not train with the whole dataset

To generate caption of any image:

1. Put the image in the Image Caption Generator/Generator directory.
2. Run the command python Generate Caption.py
3. When input is prompted write the image let's name (say example.jpg) as input.
4. The image will be shown with the generated caption on top. You can save the image with caption from here. The caption will also be printed in the command prompt.

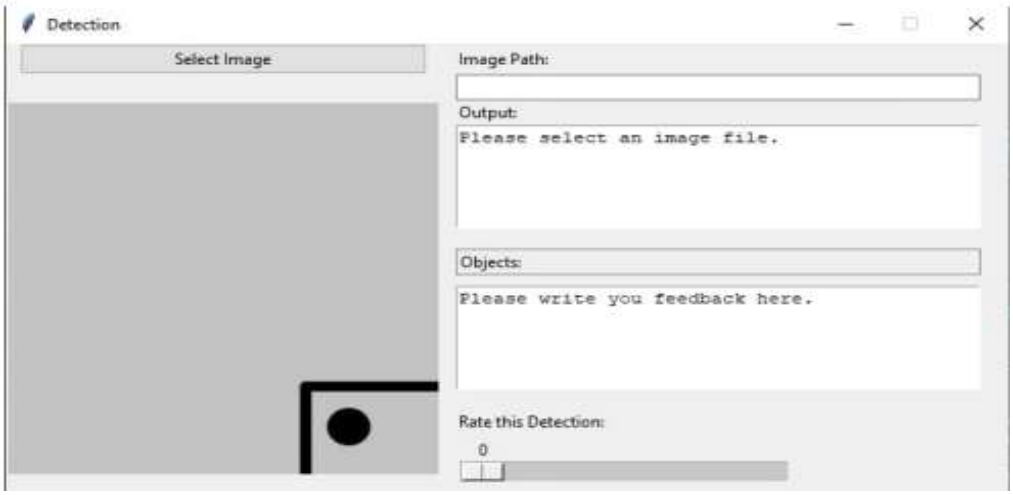
6 GUI

The result shown inside the GUI as output.

How to run GUI:

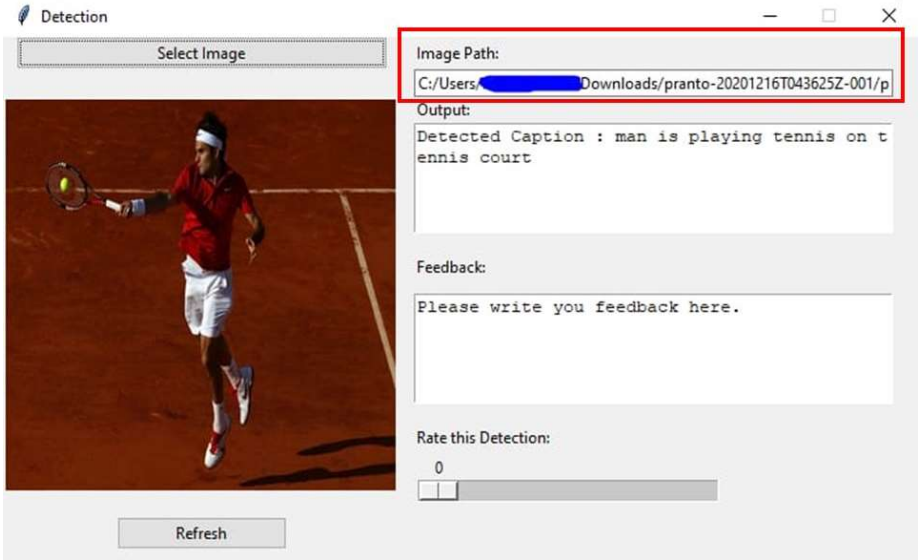
open spyder3 > open and open file generate_captio.py > run

- **UI**

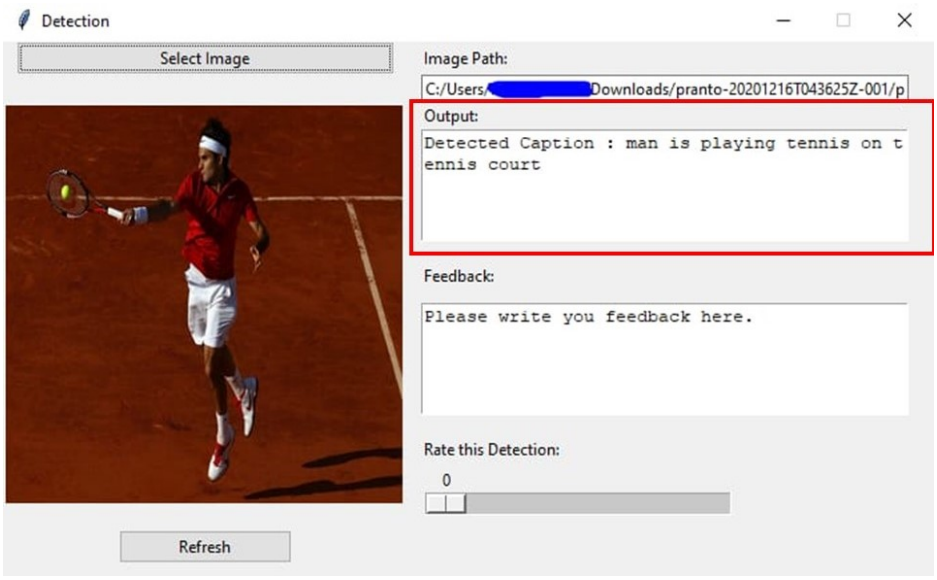


- **Uploading Image**

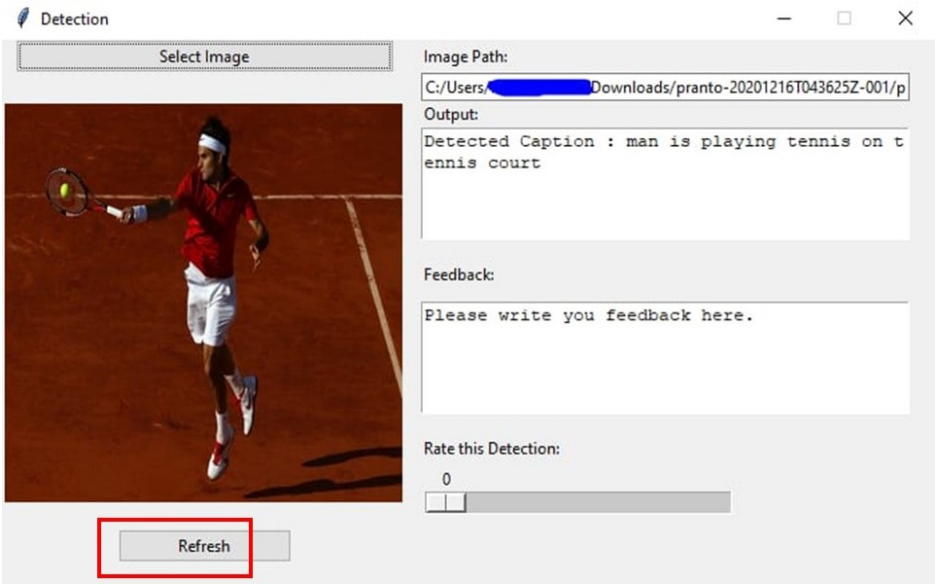
- **Image Path**



- **Result as output**



- **Refresh button**



Citation:

Md. Mijanur Rahman, Ashik Uzzaman, Sadia Islam Sami, "Image captioning using deep neural network based model", Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University. Github Repository, 2021. https://github.com/mijancse/image-captioning-using-deep-neural-network-based-model