

Project title:

Image Caption Generation

Guided by :-

Prof. K.S. Kadam



Team Members :

NAME	PRN
Prashant Popat Akkole	17UCS12001XX
Sohel Firoj Faras	17UCS12059XX
Guruputra Channappa Teli	17UCS12061XX
Moinahmad Mujeeb Tahsildar	17UCS12060XX
Krishna Dnyandeo Shinde	17UCS12055XX



Problem Statement

To develop a system which can automatically generate the description of an image with the use of CNN along with LSTM.

Introduction

Automatically describing the content of images using natural languages is a fundamental and challenging task. It has great potential impact.

For example, it could help visually impaired people better understand the content of images on the web. Also, it could provide more accurate and compact information of images/videos in scenarios such as image sharing in social network or video surveillance systems.



Introduction

- What do you see in the picture?



Introduction



Adding captions about image, as human beings.

- ❑ A cow is standing in the field.
- ❑ This is a close up of a brown cow.
- ❑ There is a brown cow with long hair and two horns.
- ❑ There are two trees and a cloud in the background of a field with a large cow in it.

Objectives

- To detect the objects present in the image.
- To generate the sentence based on the captions identified from the image.



Requirement Specifications

<u>SR.NO</u>	<u>Requirement</u>	<u>Essential or Desirable</u>	<u>Description of the requirement</u>
RS1	The dataset should contain considerable amount of data to train the model.	Essential	To train the model collective data required
RS2	The system should have a browsing files function	Essential	There should be an interface for browsing files
RS3	The system should have a uploading functions for sample images to server of system	Essential	There should be an interface for browsing files and uploading images.
RS4	The system should be able to detect objects from the uploaded image.	Desirable	System should be able to identify type of uploading files type[.png, .jpeg, .jpg]



Requirement Specifications

<u>SR.NO</u>	<u>Requirement</u>	<u>Essential or Desirable</u>	<u>Description of the requirement</u>
RS5	The system should return a error status page if something goes wrong.	Essential	Server will respond to error in system operating.
RS6	The system should be able to generate multiple captions for the image.	Essential	This will help in choosing best captions according to their rankings
RS7	The system should be able to generate best caption based on the ranking.	Essential	Server will respond a success page with expected output.





Hardware Requirements

- **Processor:** Intel Corei3 or later
- **RAM:** Minimum 4GB
- **Secondary storage:** Minimum 15GB
- **Graphics card:** NVidia or Google colab

Software Requirements:

- Windows 10 or Ubuntu 18.4
- Web Browser
- Nginx or apache server
- Python3 with Flask Web Library
- Flickr8k Dataset
- DL Libraries {Anaconda Framework}

Data Collection

There are many open-source datasets are available for this project statement.

- **Flickr8k**

- 8000 images, each annotated with 5 sentences via AMT
- 1000 for validation, testing

- **Flickr 30k**

- 30k images
- 1000 validation, 1000 testing

- **MS COCO {Common Objects and contexts}**

- 123,000 images
- 5000 for validation, testing

Dataset of images and sentence descriptions


training image



"A Tabby cat is leaning on a wooden table, with one paw on a laser mouse and the other on a black laptop"

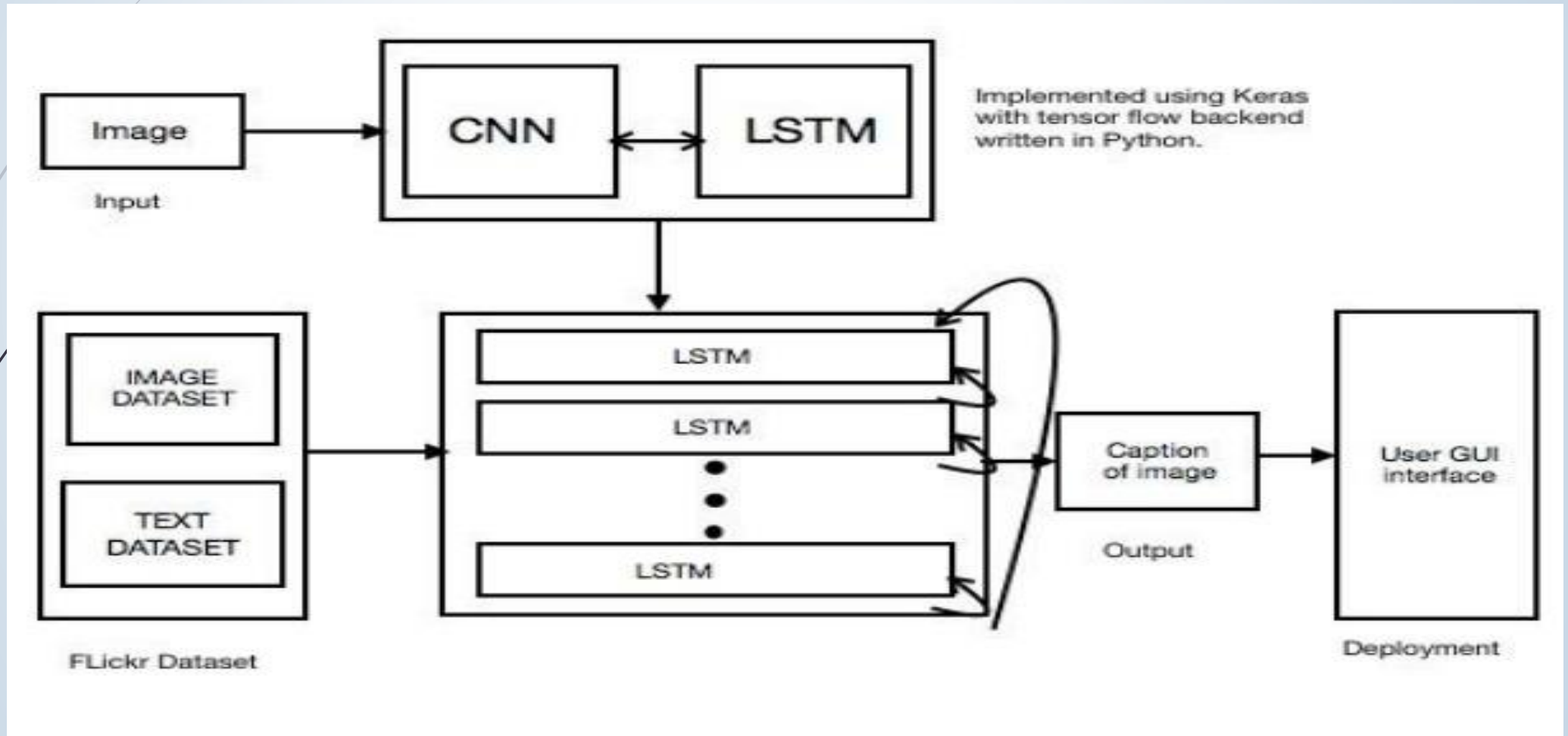


Captioning Model



- ▶ A captioning model relies on two main components, a CNN and RNN. Captioning image is all about merging the two to combine their most powerful attributes i.e.,
 - ▶ CNNs (convolutional Neural Networks) excel at preserving spatial information and recognizes objects in images.
 - ▶ RNNS (Recurrent Neural Networks) work well with any kind of sequential data, such as a generating sequences of words.
- ▶ So by merging the two, you can get a model that can find patterns and images, and then that information to help generate a description of those images.

Model of Image Captioning



Examples of captions



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."



"little girl is eating piece of cake."



"baseball player is throwing ball in game."



"woman is holding bunch of bananas."



"black cat is sitting on top of suitcase."



"a young boy is holding a baseball bat."



"a cat is sitting on a couch with a remote control."



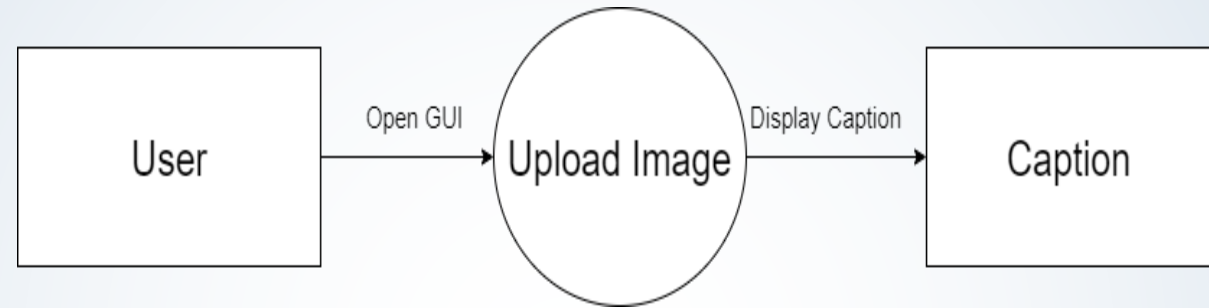
"a woman holding a teddy bear in front of a mirror."



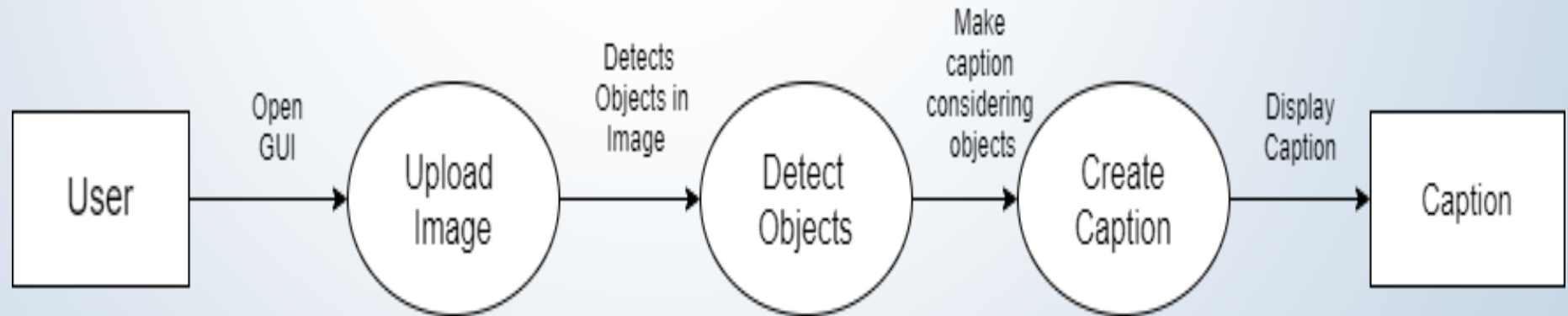
"a horse is standing in the middle of a road."

UML Diagrams:

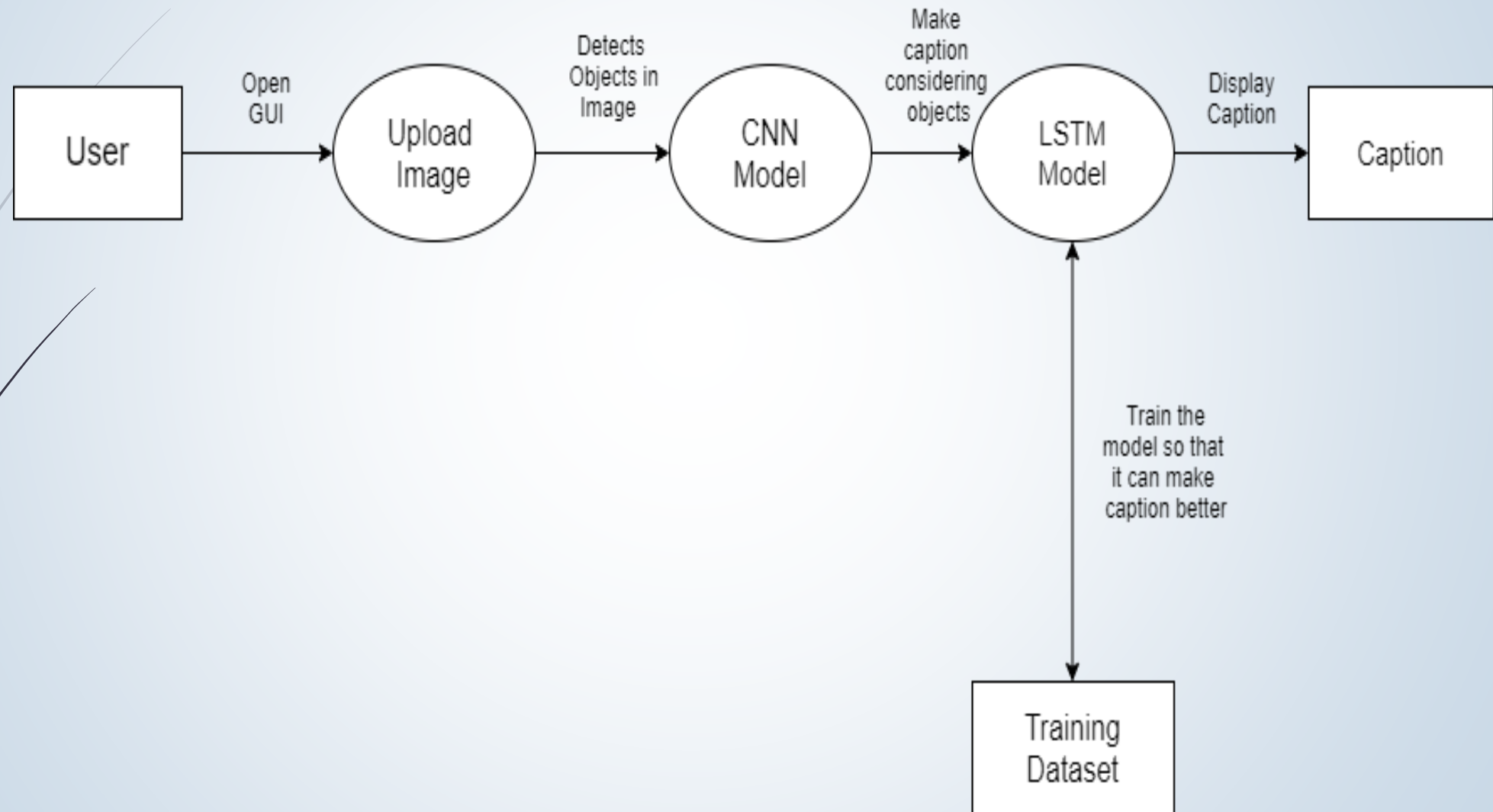
DFD LEVEL 0 :



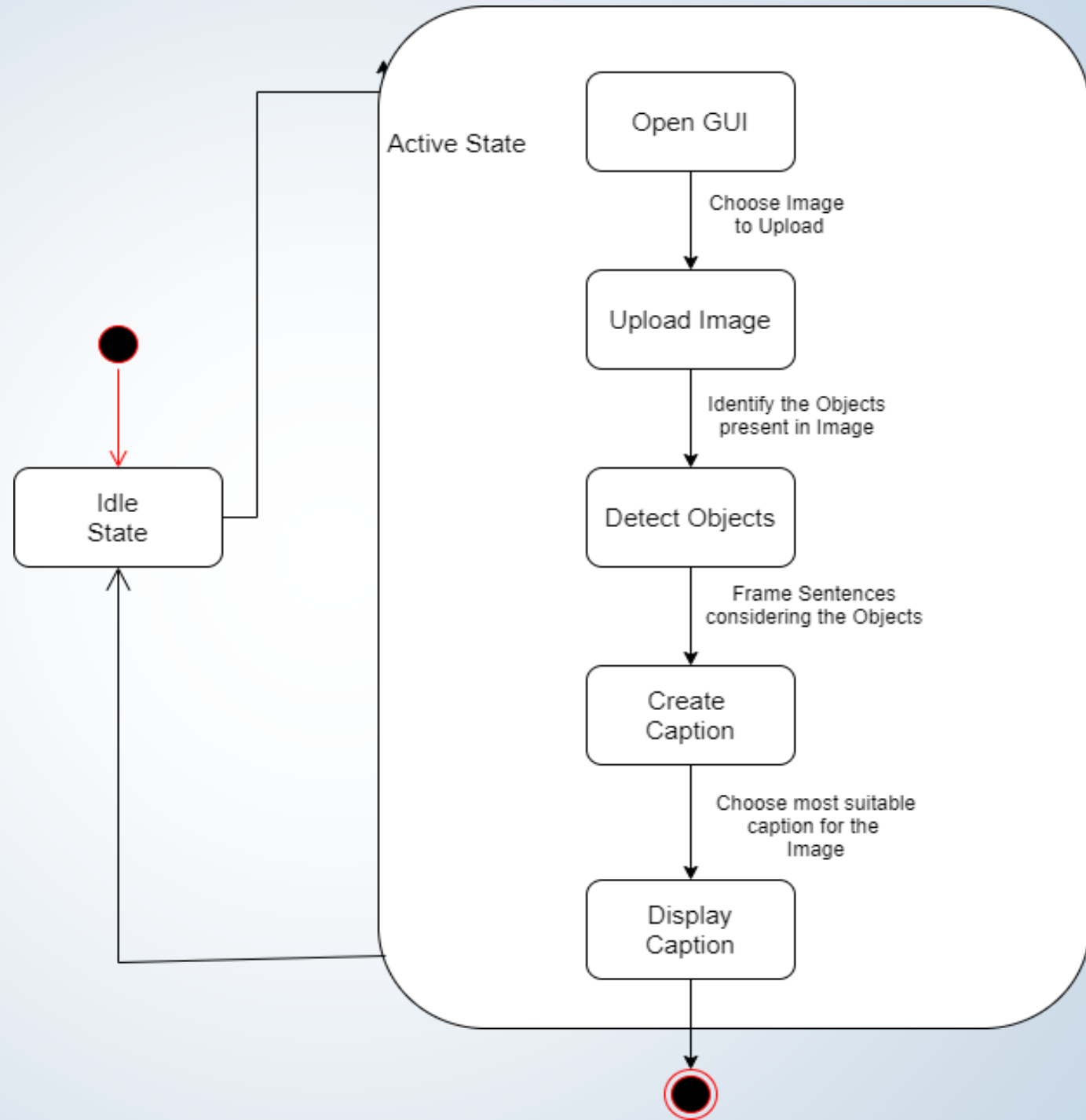
DFD LEVEL 1 :



DFD LEVEL 2



State Chart diagram



Methodology

1. Firstly, we imported all the necessary packages

2. Getting and performing data cleaning

- ▶ We will clean the text in the following ways in order to reduce the size of the vocabulary of words we will need to work with:
 - ▶ Convert all words to lowercase.
 - ▶ Remove all punctuation.
 - ▶ Remove all words that are one character or less in length (e.g. 'a').
 - ▶ Remove all words with numbers in them.
- ▶ Once cleaned, we can summarize the size of the vocabulary. And can save the dictionary of image identifiers and descriptions to a new file named *descriptions.txt*, with one image identifier and description per line.

3. Load Data

- ▶ The train and development dataset have been predefined in the *Flickr_8k.trainImages.txt* and *Flickr_8k.devImages.txt* files respectively, that both contain lists of photo file names. From these file names, we can extract the photo identifiers and use these identifiers to filter photos and descriptions for each set.

Methodology

4. Defining the image caption model[3 parts]

- **Photo Feature Extractor.** This is a 16-layer VGG model pre-trained on the ImageNet dataset. We have pre-processed the photos with the VGG model (without the output layer) and will use the extracted features predicted by this model as input.
- **Sequence Processor.** This is a word embedding layer for handling the text input, followed by a Long Short-Term Memory (LSTM) recurrent neural network layer.
- **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.

The model learns fast and quickly over fits the training dataset. For this reason, we will monitor the skill of the trained model on the holdout development dataset. When the skill of the model on the development dataset improves at the end of an epoch, we will save the whole model to file. At the end of the run, we can then use the saved model with the best skill on the training dataset as our final model.

5. Evaluate the Model

- We will evaluate a model by generating descriptions for all photos in the test dataset and evaluating those predictions with a standard cost function. We will generate predictions for all photos in the test dataset and in the train dataset.
- The actual and predicted descriptions are collected and evaluated collectively using the corpus BLEU score that summarizes how close the generated text is to the expected text.

Methodology

Sentence Generation Results evaluated in BLEU score (B-n)

- ✓ fraction of n-grams in generated string that are contained in reference (human generated) sentences.

6. Generate New Caption

- ➡ We will generate a description for new image using our model.
- ➡ Upload the new image with “xyz.jpg” format.
- ➡ First, we must load the Tokenizer from *tokenizer.pkl* and define the maximum length of the sequence to generate, needed for padding inputs. Then use the modified `featureExtractor()` which only work on a single photo.
- ➡ We can then generate a description using the *generate_desc()* function defined when evaluating the model.

Applications

- ❖ **Google Photos:** Classify your photo into Mountains, sea etc.(image indexing)
- ❖ It can be used to describe video in real time.
- ❖ Social Media Platforms like Facebook can infer directly from the image, where you are (beach, cafe etc), what you wear (color) and more importantly what you're doing also (in a way).
- ❖ Attention Network for object detection.
- ❖ VQA(visual Question Answering) using image and question embedding.
- ❖ It helps in recommendations in editing applications.
- ❖ Automatic captioning can help, make Google Image Search as good as Google Search, as then every image could be first converted into a caption and then search can be performed based on caption. [**Google lens**]

Future Scope & Research

- ❖ **Aid to the Blind person** {converting the scene into text and then the text to voice.} **Nvidia** research is trying to create such a product.
- ❖ In **Web development**, It's good practice to provide a description for any image that appears on the page so that an image can be read or heard as opposed to just seen. This makes web content accessible.
- ❖ **CCTV Cameras** are everywhere today, but along with viewing the world, if we can also generate relevant captions, then we can raise alarms as soon as there is malicious activity going on somewhere. This could probably help reduce some crime and/or accidents.
- ❖ **Self Driving Cars:** automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it can give a boost the self driving system.

References

- <https://stackoverflow.com/questions/tagged/face-captioning>
- Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570.
- Show & Tell: A Neural Image Caption Generator, 2015.
[<https://arxiv.org/abs/1411.4555>]



Thank You !