**CAPSTONE PROJECT:**

**IMAGE CAPTION**

**BY- PRIYANKA BISWAS**

**SHINAKSHI SANKHAYAN**

**GULFAIRUS KAIREDENOVA**

**SANDIPTA SUBIR KHARE**

**AGENDA**

* Problem Statement
* Objective
* Tools & Libraries
* What is Image Caption
* Pros and Cons of Image Captioning
* How to do Image Captioning Implementation
* Techniques for Image Captioning
* Data Preparation
* Natural Language Processing
* NLP Based Techniques
* Working of Tokenizer
* Create Data Generator
* Caption Generation Process
* Deep Learning: Techniques
* Convolution Neural Network(CNN)
* Convolution Network
* LSTM
* Image Caption Generating Model
* Training Model
* Deep Learning: Extracting Feature
* Deep Learning: Xception
* Layers in Model
* Testing Model
* Deployment
* References

**PROBLEM STATEMENT**

Now days, if someone see the image and not able to recognise what image is showing. Then in that Scanerio Image Captioning would help the user to get the final output of the image.

If there is an user with medical conditions like blind and paralysis then they might able to use any technology for recognising the image. Our project Image Captioning with Voice recognition would help the user to get the final output.

**OBJECTIVE**

Image caption, automatically generating natural language descriptions according to the content observed in an image, is an important part of scene understanding, which combines the knowledge of computer vision and natural language processing. The application of image caption is extensive and significant, for example, the realization of human-computer interaction.

**TOOLS & LIBRARIES**

* TOOLS USED:

1. Python

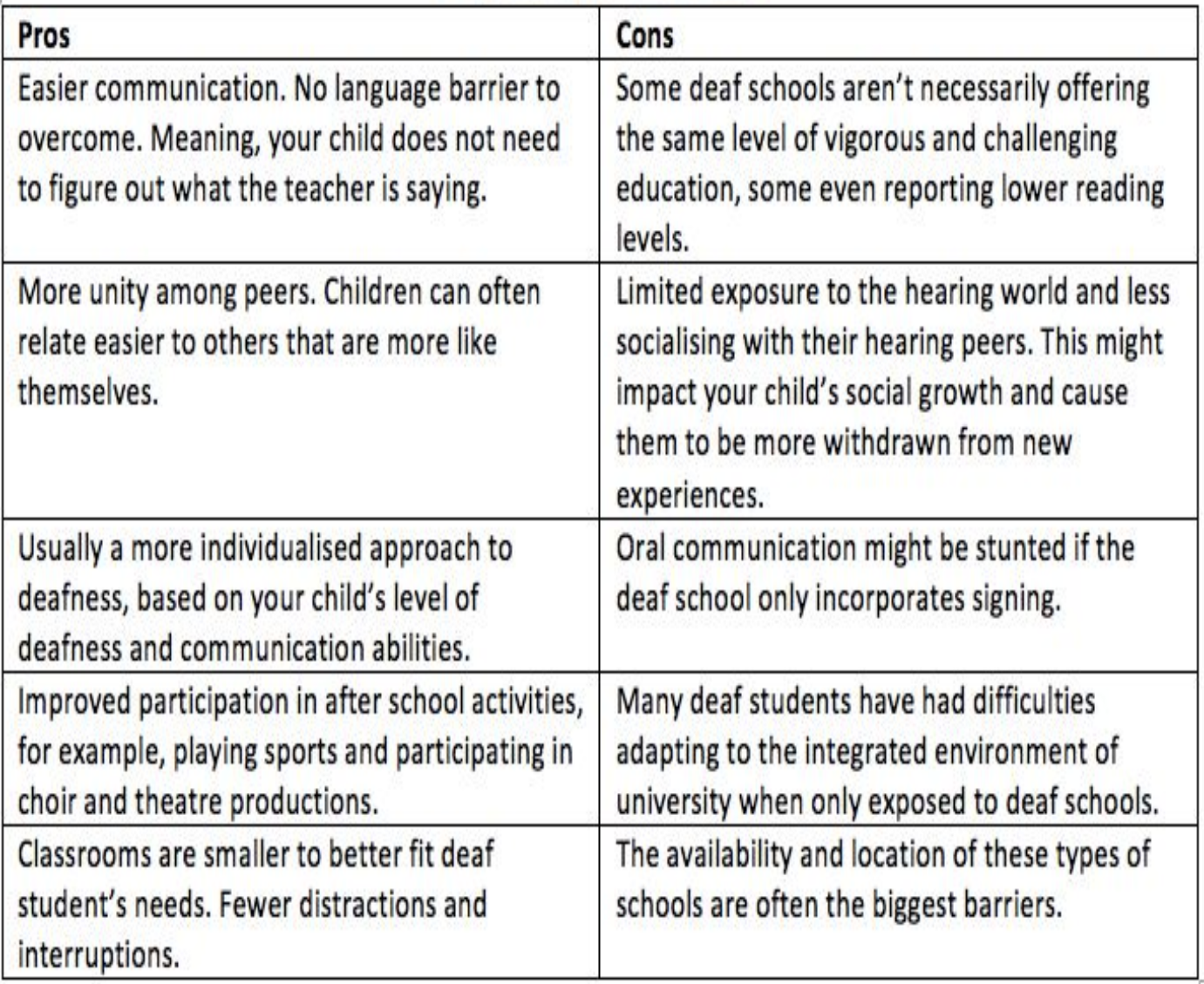
* LIBRARIES:

1. Numpy
2. Pandas
3. Matplotlib
4. Keras

**WHAT IS IMAGE CAPTION?**

**Image Captioning** is the process of generating textual description of an image. It uses **Machine Learning**, **Deep Learning**, **Natural Language Processing** and **Computer Vision** to generate the captions. The dataset will be in the form [**image** → **captions**]. The dataset consists of input images and their corresponding output captions.

**Pros and Cons of Image Captioning**



**How to do Image Captioning Implementation**

The task of image captioning can be divided into two modules logically:

* One is an **image based model** – which extracts the features and nuances out of our image, and
* The other is a **language based model** – which translates the features and objects given by our image based model to a natural sentence.

**Techniques for Image Captioning**

T he techniques which we may use for the image captioning can be broadly divided into two categories:

* Deep Learning based techniques and
* NLP based techniques.

**Data Preparation**

The **Flickr 8k dataset** have five different descriptions per image, that provide clear descriptions of the noticeable entities and events and are described by actual people. The dataset has 8000 images from Flickr and contains people and animals (mostly dogs) performing some action. We used ¾ of the data for training and ¼ for evaluation.

* lowercase all words
* remove punctuation like .,-<>()
* remove hanging ‘s’ and ‘a’
* remove numbers

**Natural Processing Language**

* Computers don’t understand English words, for computers, we will have to represent them with numbers. So, we have map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a “tokenizer's” pickle file.
* Our vocabulary contains 7577 words.

We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters.

**NLP Based Techniques**

This technique combines Computer Vision and Natural Language Processing techniques to create a model that can describe contents of an image into natural language. This technique is useful for

* Captioning images on the internet, which helps visually impaired people using a screen reader to describe the contents of an image into words.
* Captioning a large database of images to make it easier to search images with description.

**What is Tokenizer**

Tokenization is a common task in Natural Language Processing (NLP). It’s a fundamental step in both traditional NLP methods like Count Vectorizer and Advanced Deep Learning-based architectures like [**Transformers**](https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/?utm_source=blog&utm_medium=what-is-tokenization-nlp)**.**

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or sub words. Hence, tokenization can be broadly classified into 3 types – word, character, and sub word (n-gram characters) tokenization.

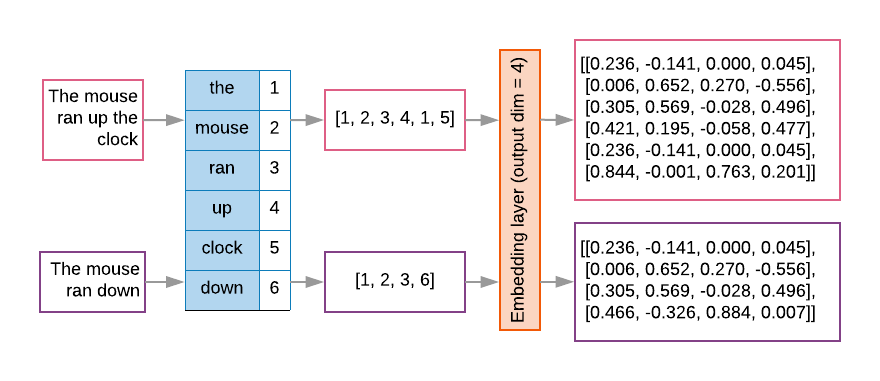
For example, consider the sentence: “Never give up”.

The most common way of forming tokens is based on space. Assuming space as a delimiter, the tokenization of the sentence results in 3 tokens – Never-give-up. As each token is a word, it becomes an example of Word tokenization.

Similarly, tokens can be either characters or sub words. For example, let us consider “smarter”:

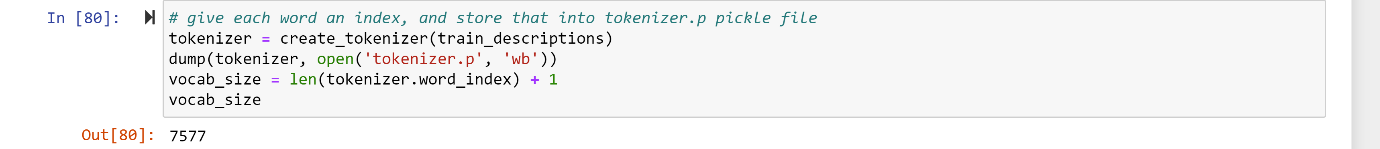
1. Character tokens: s-m-a-r-t-e-r
2. Sub word tokens: smart-er

**Working of Tokenizer**



**Fig 1.1**

vocab\_size:

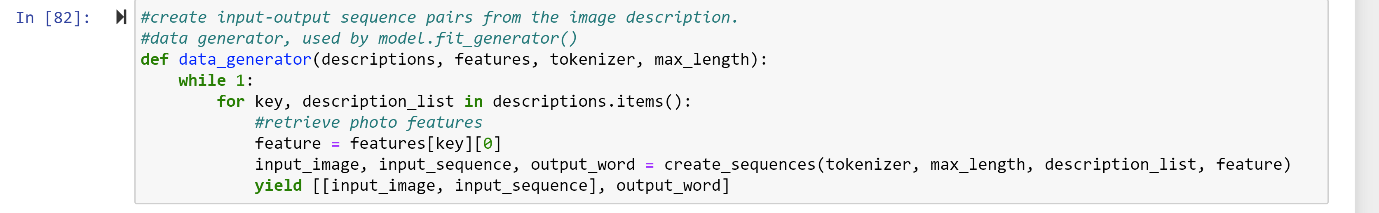


**Fig 1.2**

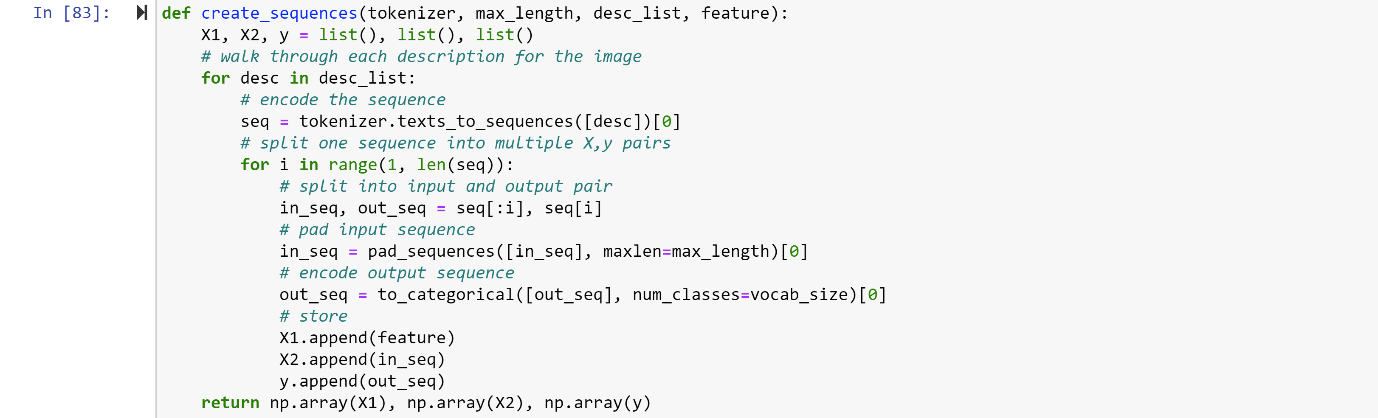
**Data Generator**

* We have train our model on 6000 images and each image will contain 2048 length feature vector and caption is also represented as numbers. This amount of data for 6000 images is not possible to hold into memory so we will be using a generator method that will yield batches.
* The generator will yield the input and output sequence.
* The input to our model is [x1, x2] and the output will be y, where x1 is the 2048 feature vector of that image, x2 is the input text sequence and y is the output text sequence that the model has to predict.

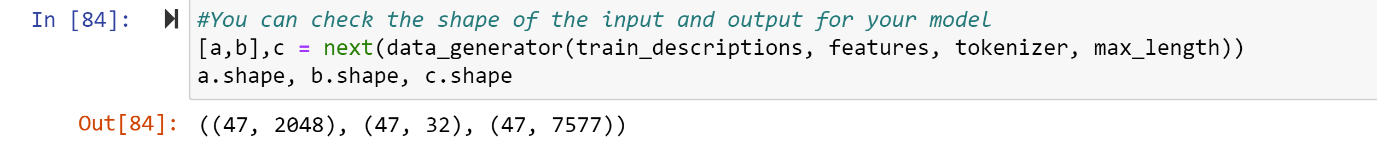
**Create Data Generator:**

****

**Fig 1.3**

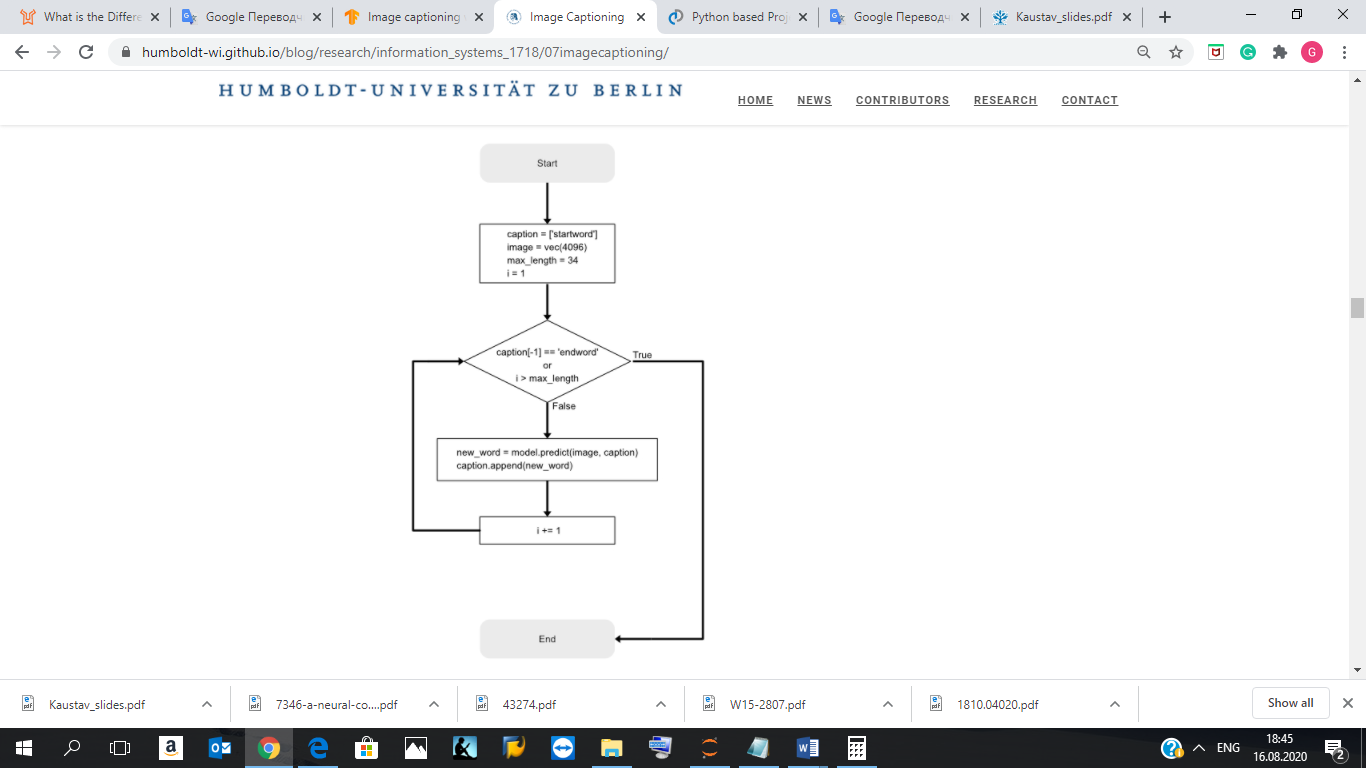
****

**Fig 1.4**

****

**Fig 1.5**

**Caption Generation Process**



**Fig 1.6**

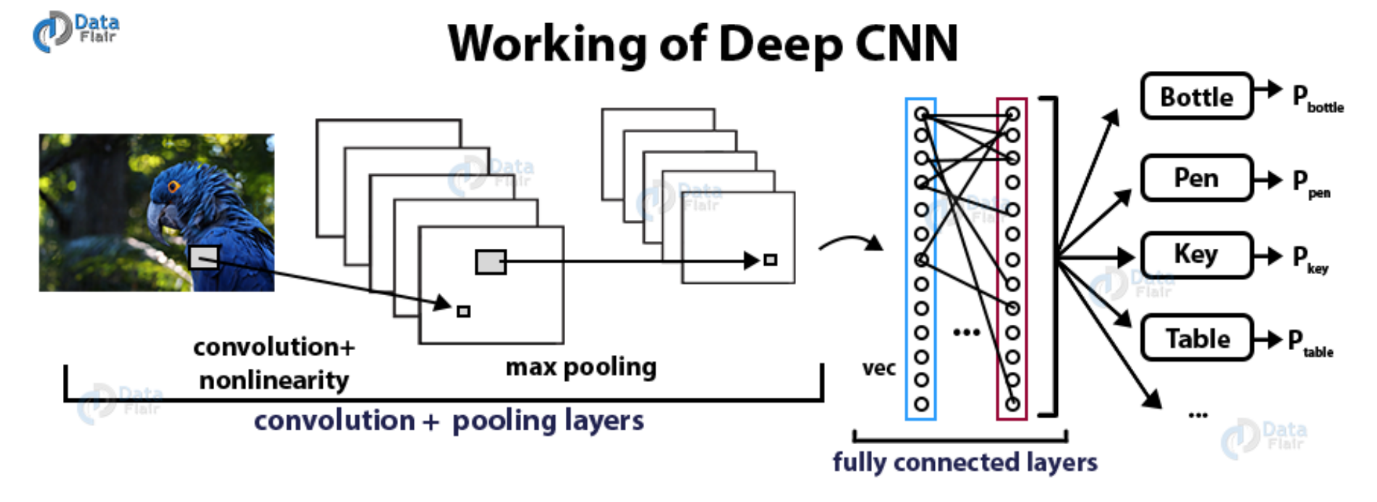
**Deep Learning Techniques**

* In deep learning based techniques, features are learned automatically from training data and they can handle a large and diverse set of images and videos. For example, Convolutional Neural Networks (CNN) are widely used for feature learning, and a classifier such as SoftMax is used for classification. CNN is generally followed by Recurrent Neural Networks (RNN) in order to generate captions.
* An embedding layer will handle the textual input, followed by the LSTM layer.
* Decoder – By merging the output from the above two layers, we will process by the dense layer to make the final prediction. The final layer will contain the number of nodes equal to our vocabulary size.

**What is CNN (Convolution Neural Network)?**

Convolutional Neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images.

CNN is basically used for image classifications and identifying if an image is a bird, a plane or Superman, etc.



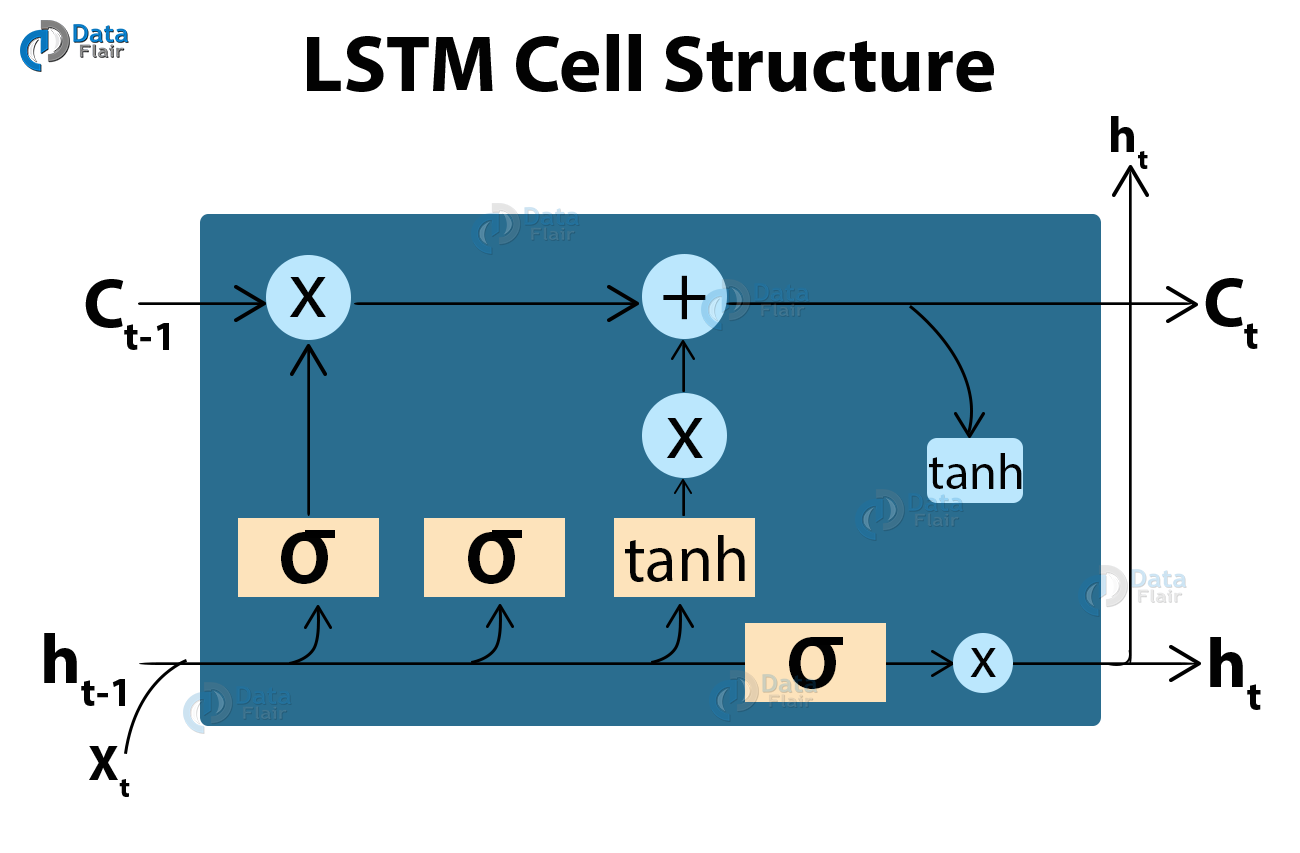
**Fig 1.7**

It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images. It can handle the images that have been translated, rotated, scaled and changes in perspective.

**What is LSTM?**

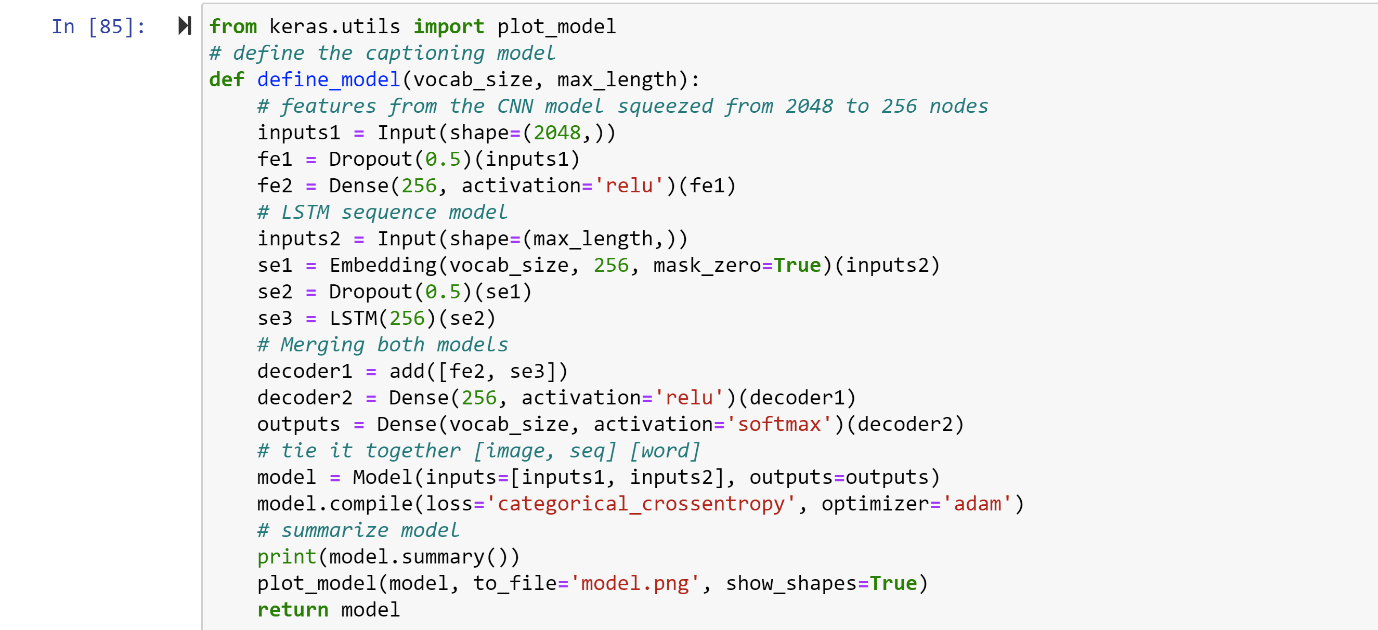
LSTM stands for **Long short term memory**, they are a type of RNN (**recurrent neural network**) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

This is what an LSTM cell looks like –



**Fig 1.8**

Defining CNN-RNN Model with LSTM:



**Fig 1.9**

### Image Caption Generator Model

So, to make our image caption generator model, we will be merging these architectures. It is also called a CNN-RNN model.

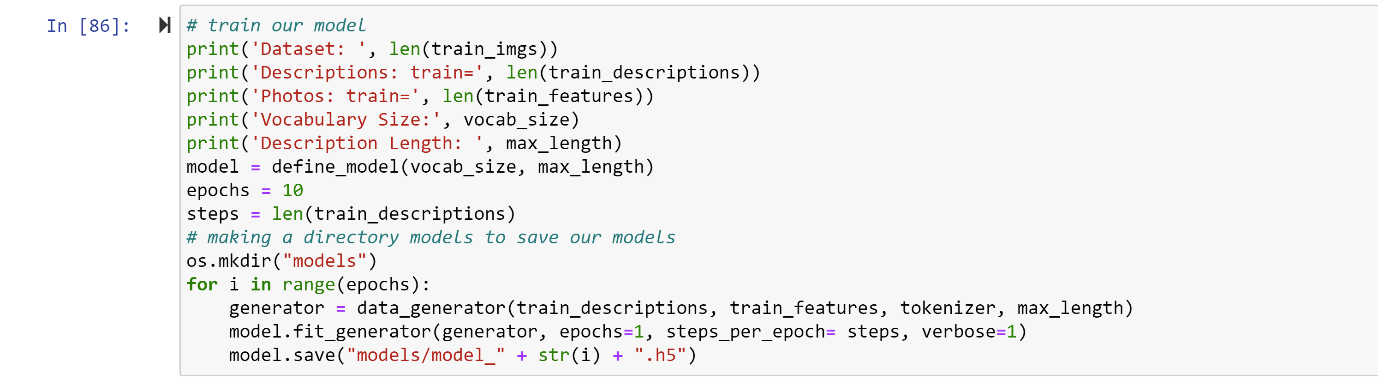
* CNN is used for extracting features from the image. We will use the pre-trained model Xception.
* LSTM will use the information from CNN to help generate a description of the image.



**Fig 1.10**

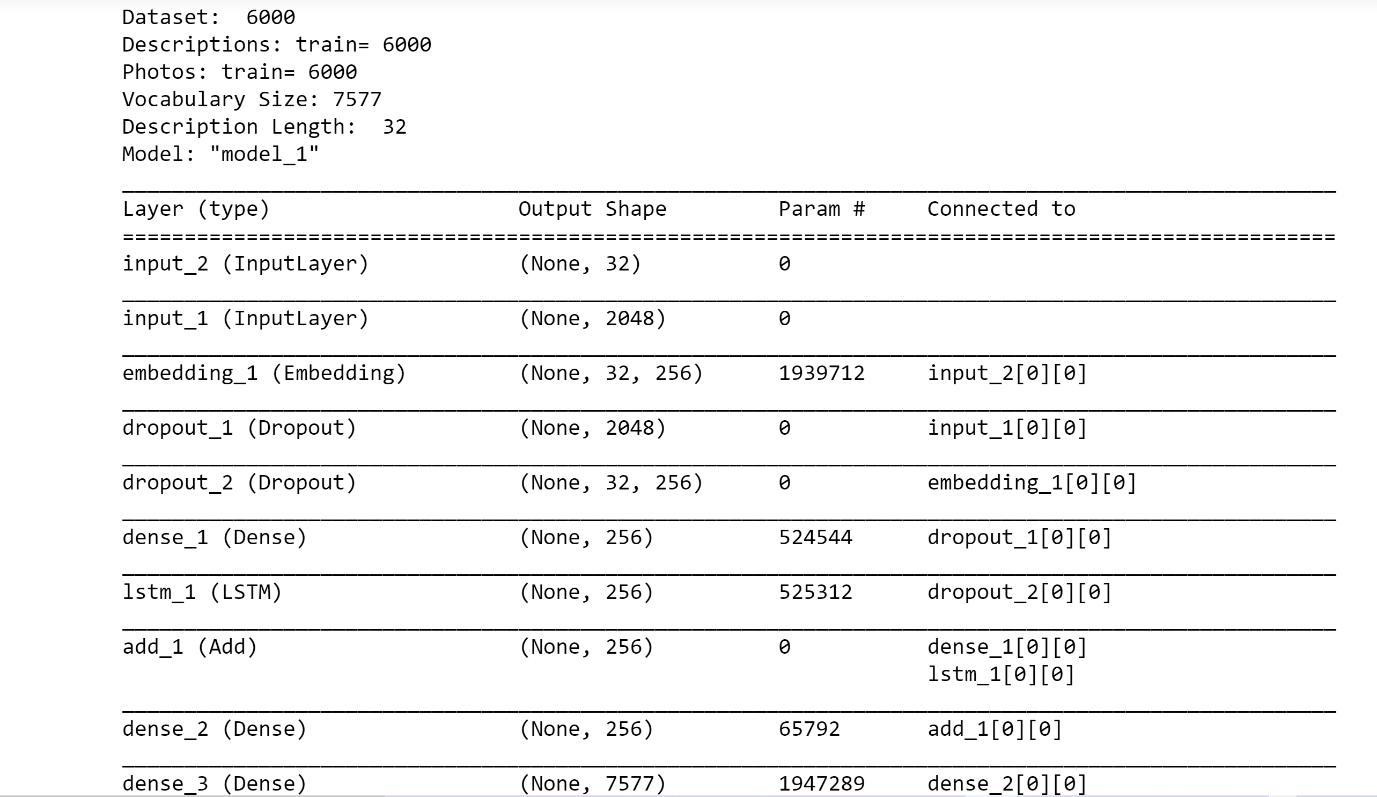
**Training Model**

Training the model:

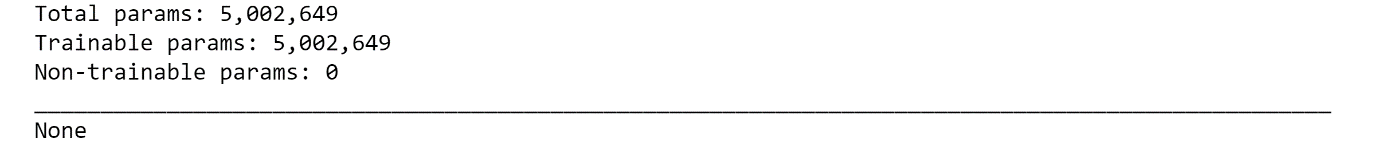
****

**Fig 1.11**

Output:



**Fig 1.12**



**Fig 1.13**

**Extracting Features**

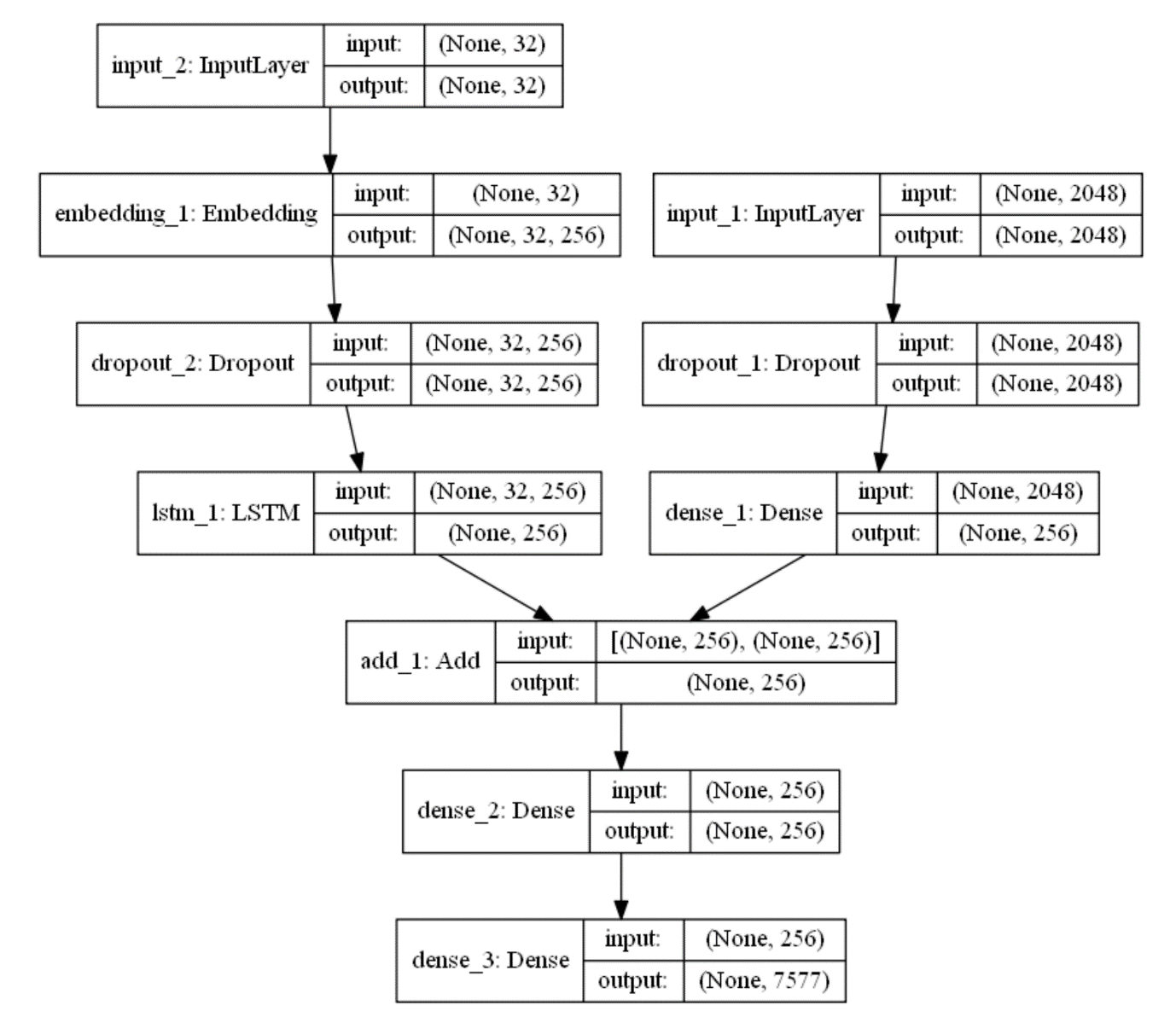
Extracting features from the image CNN is Xception Model.

**Xception model** is thepre-trained model that have been already trained on large datasets and extract the features from these models and use them for our tasks. Xception model has been trained on imagenet dataset that had 1000 different classes to classify. We can directly import this model from the keras.applications . Xception model takes 299\*299\*3 image size as input. We will remove the last classification layer and get the 2048 feature vector.



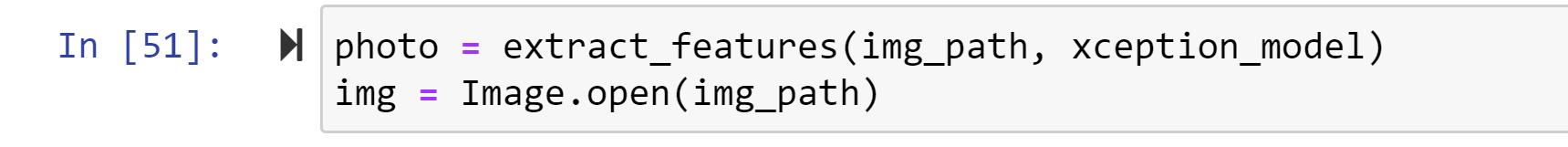
**Fig 1.14**

**Layers in Model**



**Fig 1.15**

**Testing the Model**

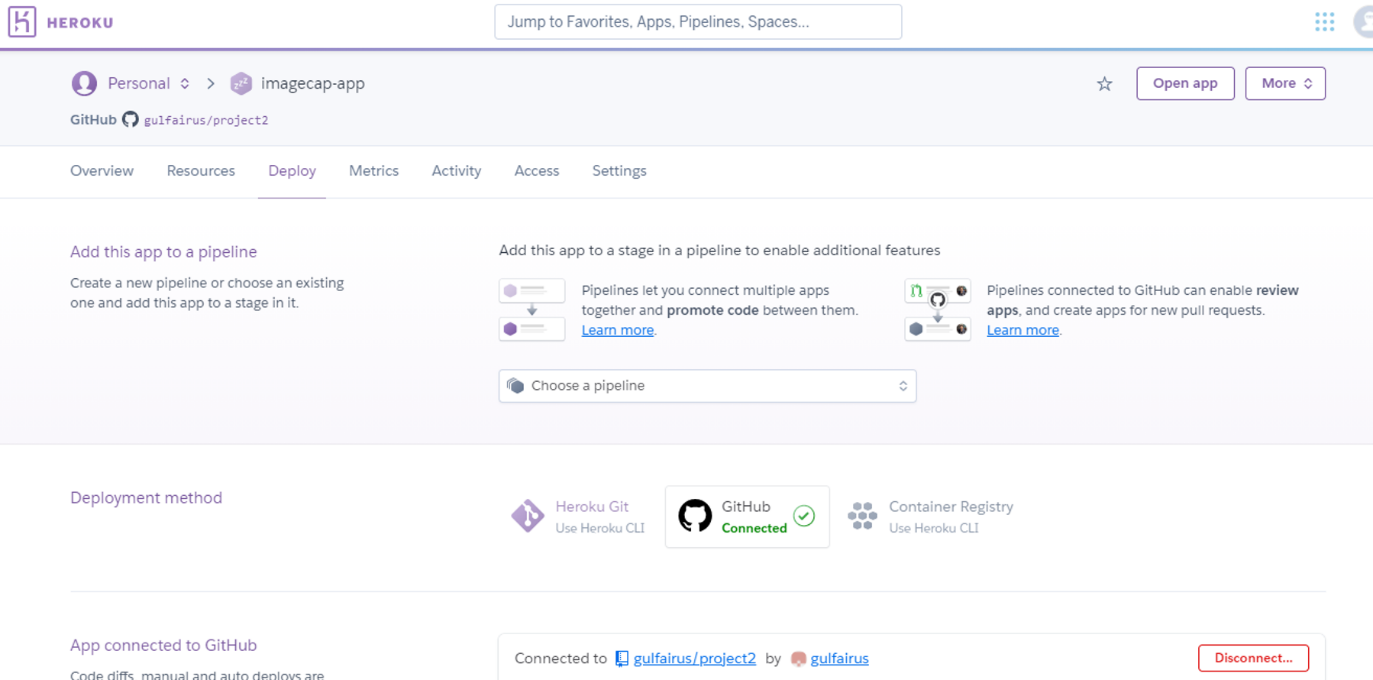


**Fig 1.16**

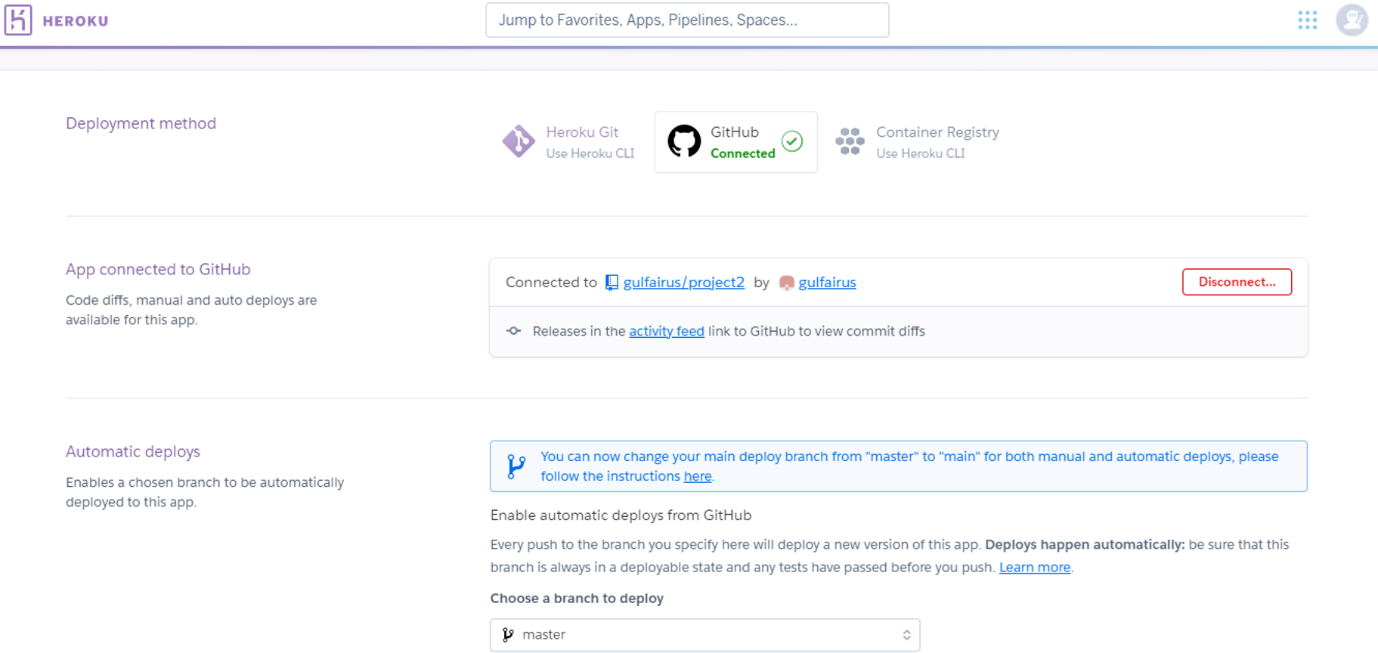


**Fig 1.17**

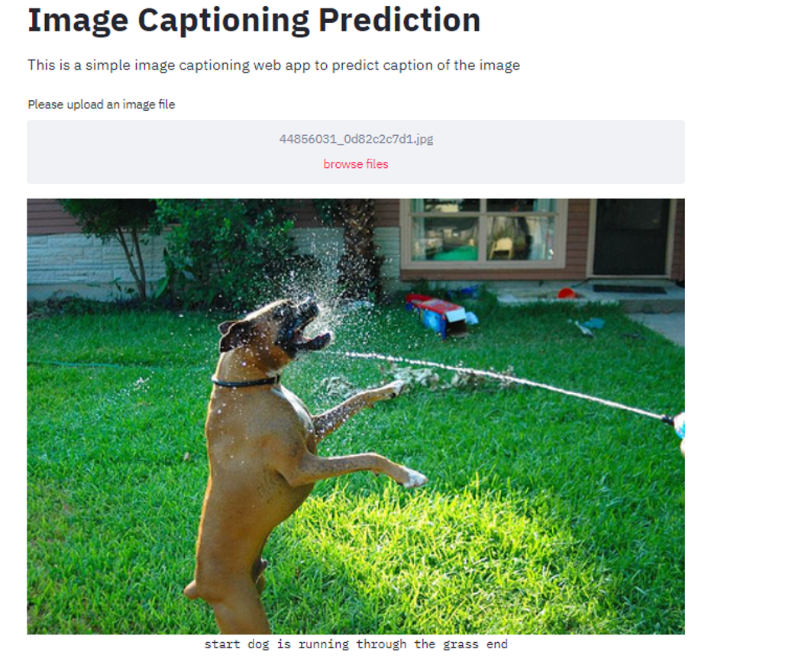
**Deployment**



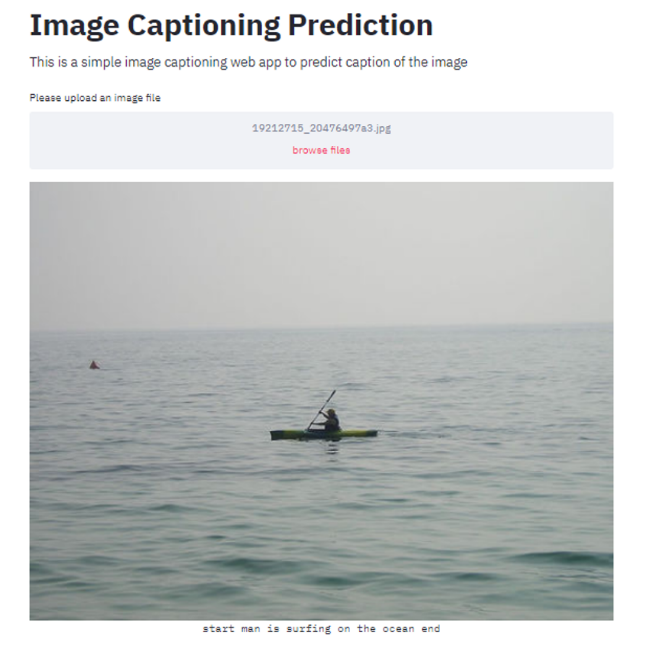
**Fig 1.18**



**Fig 1.19**



**Fig 1.20**



**Fig 1.21**

**Reference**

* <https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/#:~:text=A%20Quick%20Rundown%20of%20Tokenization,-Tokenization%20is%20a&text=Tokenization%20is%20a%20way%20of,words%2C%20characters%2C%20or%20subwords.&text=As%20each%20token%20is%20a,be%20either%20characters%20or%20subwords>.
* <https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/>
* <https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/>
* <https://pypi.org/project/tokenizer/#:~:text=Tokenizer%20is%20a%20compact%20pure,%2C%20URL%2FURI%2C%20etc>.
* <https://www.tutorialspoint.com/python_text_processing/python_tokenization.htm>