STROKE PREDICTION USING MACHINE LEARNING

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

##### 

##### 

##### 

##### 

##### **STROKE PREDICTION USING MACHINE LEARNING**

##### **A Major Project Report**

***in partial fulfilment for the award of the degree***

***of***

##### **BACHELOR’s OF TECHNOLOGY**

***In***

**COMPUTER SCIENCE ENGINEERING**

###### **by**

##### **MAYANK SINGH, PRITIKA RANA and SHIVAM RATHI**

**Guided by**

**Dr SUNITA TIWARI**

**Assistant. Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**G.B. PANT GOVT. ENGINEERING COLLEGE, NEW DELHI.**

(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)

**MAY, 2021**

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the Major Project entitled **“Stroke Prediction Using Machine Learning”** in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering / Information Technology affiliated to **Guru Gobind Singh Indraprastha University, New Delhi** and submitted to the Department of Computer Science and Engineering G. B. Pant Govt. Engineering College is an authentic record of my work carried out during a period from March 2021 to May 2021. The matter represented in this report has not been submitted by me for the award of any other degree of this or any other institute/university.

**Date: - 10-05-2021 Name and Roll No:**

**Mayank Singh (00420907218)**

**Pritika Rana (02420902717)**

**Shivam Rathi (41420902717)**

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

#### *Signature of Supervisor*

#### **Date: - Name & Designation**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**LIST OF FIGURES i**

**ACKNOWLEDGMENT iii**

**ABSTRACT iv**

**CHAPTER 1. INTRODUCTION 1**

1.1 OVERVIEW 1

1.2 MOTIVATION 1

1.3 PROBLEM STATEMENT 1

1.4 SCOPE OF PROJECT 1

1.5 ACCESS TO USERS 2

**CHAPTER 2. RELATED WORKS 3**

2.1 INTRODUCTION 3

2.2 EXISTING SYSTEM 3

2.3 PROPOSED SYSTEM 4

2.3.1 IMAGE FEATURE EXTRACTION 4

2.3.2 THE LANGUAGE MODEL 5

2.3.3 FITTING THE MODEL 6

2.3.4 CAPTION GENERATION 6

2.3.5 SUGGESTION API 6

**CHAPTER 3. PROBLEM DESCRIPTION & SPECIFICATION 7**

3.1 PROBLEM OVERVIEW 7

3.2 SPECIFICATION 7

3.3 REQUIREMENTS 8

3.3.1 SOFTWARE REQUIREMENTS 8

3.3.2 HARDWARE REQUIREMENTS 8

3.3.3 OTHER REQUIREMENTS 9

**CHAPTER 4. SYSTEM DESIGN 10**

4.1 INTRODUCTION 10

4.1.1 DATA SOURCES 10

4.1.2 CONVOLUTION NEURAL NETWORK 10

4.1.3 RECURRENT NEURAL NETWORK 11

4.2 ARCHITECTURE DIAGRAM 11

4.3 UML DIAGRAM 12

4.4 DATA FLOW DIAGRAM 14

4.5 FLOW CHART DIAGRAM14

**CHAPTER 5. IMPLEMENTATION AND RESULT 16**

5.1 INTRODUCTION 16

5.2 MODEL SELECTION 16

5.3 IMAGE PREPROCESSING 17

5.4 TEXT PREPROCESSING 19

5.5 CREATING VOCABULARY 21

5.6 PREPARING THE MODEL 23

5.7 CONFIGURING THE MODEL 25

5.8 TRAINING AND SAVING THE MODEL 28

5.9 PREDICTING THE CAPTIONS 29

5.10 WEB PAGE DEVELOPMENT 31

5.11 API INTEGRATION 32

5.12 PROVIDING USER WITH CAPTIONS32

**CHAPTER 6. CONCLUSION AND FUTURE SCOPE 34**

6.1 CONCLUSION 34

6.2 FUTURE SCOPE34

**CHAPTER 7. REFERENCES 36**

**APPENDICES 39**

**APPENDIX 1: DETAILED TEST STRATEGY AND TEST CASES 39**

**APPENDIX 2: USER GUIDE 40**

**List of Figures**

**Figure No. Figure Title Page No.**

1 Flow Diagram of the project 8

2 Architecture of the image captioning model 11

3 Illustration image of functioning of the captioning model 12

4 Use Case Diagram for the captioning system 12

5 DFD Level 0 13

6 DFD Level 1 13

7 DFD Level 2 14

8 State Chart Diagram of the system 14

9 Flow chart for the caption generation process 15

10 Code Snippet showing the selection and adjustment of the model 16

11 Output representing different layers of the selected model 17

12 Code snippet showing preprocessing of images 17

13 A sample picture fetched from the dataset 18

14 Code snippet - Extracting feature map from passed images 18

15 Output representing the process of feature map creations 19

16 Code snippet showing text preprocessing 20

17 Sample captions fetched from Flickr8k.token.txt 20

18 Code snippet showing the process of caption dictionary creation 21

19 Sample view of caption dictionary 21

20 Code snippet showing Vocabulary Creation Process 22

21 Sample output showing the count of different words 22

22 Sample output showing the word dictionary 23

23 Sample output showing the integer form captions 23

24 Code snippet showing the max length of the caption present 24

25 Output showing the length of the largest caption in our collection 24

26 Code snippet showing the input preparation for the model 24

27 Code snippet showing the conversion to NumPy array form 25

28 Sample output showing the shapes of input 25

29 Configuring process of the image and language model 26

30 Output showing the image sequential model 26

31 Output showing the language sequential model 27

32 Output showing the combined mode 27

33 Detailed layout of the model along with shapes of inputs 28

34 Code snippet showing the training process 29

35 Sample output of the training process 29

36 Sample output of the inverse dictionary 29

37 Code snippet showing the image preprocessing for testing 30

38 Code snippet showing the prediction process and results 30

39 Some sample results generated by the model 31

40 Layout of the Home page of the Web App 33

41 Sample final results on the Web Platform 33

A1 Sample image showing loss and accuracy obtained by the model 38

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to my supervisor’s **Dr Sunita Tiwari** & **Dr Anu Saini** for providing their invaluable guidance, comments and suggestions throughout the project. I would specially thank Dr Sunita Tiwari for constantly motivating me to work harder and suggesting me different ways to tackle the problems encountered.

Also, I would like to thank my team members for their assistance in the project. I would like to express my gratitude for the Department of Computer Science Engineering, Gobind Ballabh Pant Government Engineering college to provide me with the opportunity to work on this project and to all the teachers and mentors guiding along the way.

* Mayank Singh (00420907218)
* Pritika Rana (02420902717)
* Shivam Rathi (41420902717)

**ABSTRACT**

The project Stroke Prediction Using Machine Learning provides user an ability to check the chances of stroke to him/her, based on different parameters asked to him/her. The user interacts with the project with the help of a Web Platform which contains a form having different key parameters which constitutes in stroke. On providing these inputs the model predicts the chances of stroke to him/her.

Stroke is a deadly disease and a leading cause of death. In most of the case the patient suffers grave consequences, along with the patient the family and relatives also suffer a lot. But studies suggest that 80 percent of stroke can be pre-diagnosed and, in that scenario, patient can be saved. There are some key parameters which contributes to chances of stroke. There parameter includes age, gender, hypertension, blood pressure, heart disease, average glucose level, BMI, smoking habits, work type, place of residence, marital status, previous history of stroke or heart attack etc. of the patient. As a patient is mostly well aware of these factors, he/she can get a diagnosis in the case of suspicion about chances of stroke. A lot of machine learning methods are developed and are currently under research to predict the chances of stroke based on different parameters.

In this project we have taken 10 different machine learning classifiers and tested them on our dataset “Stroke dataset” for accuracy. Then selected the best one for prediction. We have also performed k-fold validation and hyper-parameter optimization on the model. Along with that we have tried to identify the best features for the classifiers. The inputs gathered from user via the web platform is provided to the model using the flask backend system and the saved model is used for predicting the output which is displayed to the user on the result page.

The project provides user the facility to check the chances of stroke to them based on inputs features at the choice of location and time through their web enabled devices. The project will be hosted online on web app or mobile app and can be used by user from anywhere. Along with this our objective is also to compare various machine learning algorithm and learn how they perform on the dataset for predicting the stroke risk. What are differences among different model and also understand which classifiers performs best and what may be the reasons behind it.

**CHAPTER-1**

**INTRODUCTION**

**1.1 OVERVIEW**

# **Stroke: -**

A stroke occurs when the blood supply to part of your brain is interrupted or reduced, preventing brain tissue from getting oxygen and nutrients. Brain cells begin to die in minutes.

A stroke is a medical emergency, and prompt treatment is crucial. Early action can reduce brain damage and other complications.

Stroke is the [fifth leading cause of deathTrusted Source](https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm) in the United States and a leading of death even in India. In fact, nearly [800,000 people](https://www.cdc.gov/stroke/facts.htm) have a stroke each year. That equates to around one person every 40 seconds.

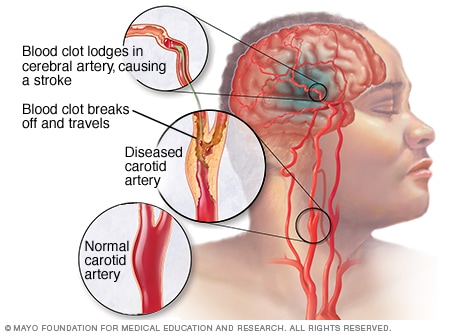
## **Causes**

There are [three main types](https://www.stroke.org/en/about-stroke/types-of-stroke) of stroke:

* **Ischemic stroke:**

### Ischemic stroke

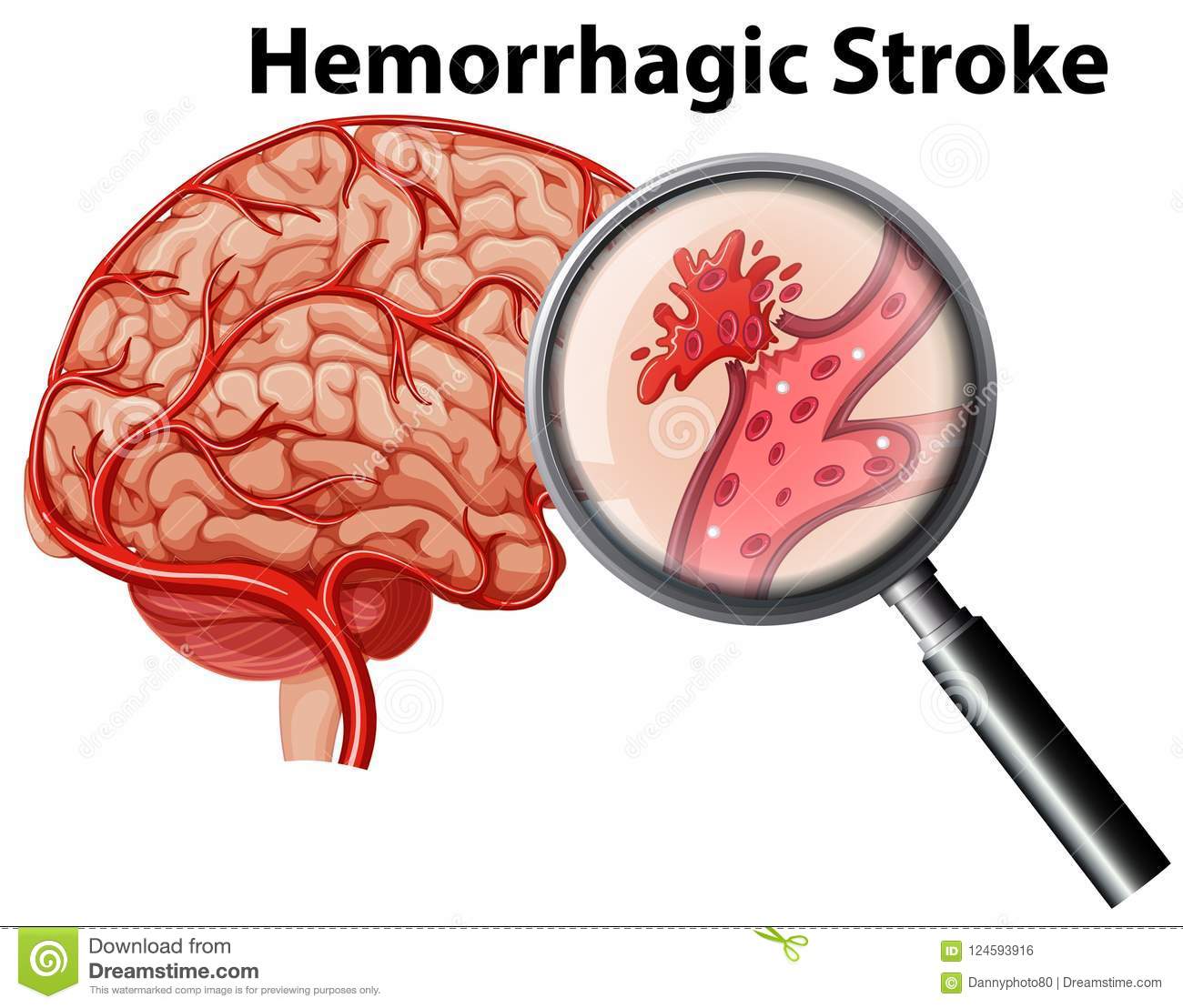
This is the most common type of stroke, making up 87% of all cases. A blood clot prevents blood and oxygen from reaching an area of the brain. It happens when the brain's blood vessels become narrowed or blocked, causing severely reduced blood flow (ischemia). Blocked or narrowed blood vessels are caused by fatty deposits that build up in blood vessels or by blood clots or other debris that travel through your bloodstream and lodge in the blood vessels in your brain



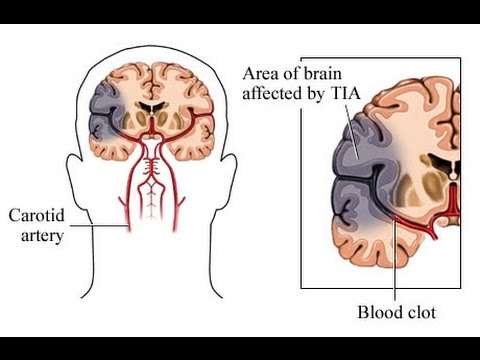
* **Hemorrhagic stroke:**

Hemorrhagic stroke occurs when a blood vessel in your brain leaks or ruptures. Brain hemorrhages can result from many conditions that affect your blood vessels. These are usually the result of aneurysms (an **aneurysm** refers to a weakening of an artery wall that creates a bulge, or distention, of the artery). Factors related to hemorrhagic stroke include:

* Uncontrolled high blood pressure
* Overtreatment with blood thinners (anticoagulants)
* Bulges at weak spots in your blood vessel walls (aneurysms)
* Trauma (such as a car accident)
* Protein deposits in blood vessel walls that lead to weakness in the vessel wall (cerebral amyloid angiopathy)
* Ischemic stroke leading to hemorrhage



* **Transient ischemic attack (TIA):** This occurs when blood flow to a part of the brain is inadequate for a brief period of time. It is a temporary period of symptoms similar to those you'd have in a stroke. Normal blood flow resumes after a short amount of time, and the symptoms resolve without treatment. Some people call this a [ministroke](https://www.medicalnewstoday.com/articles/164038.php). A TIA doesn't cause permanent damage. They're caused by a temporary decrease in blood supply to part of your brain, which may last as little as five minutes.



Symptoms

If you or someone you're with may be having a stroke, pay particular attention to the time the symptoms began. Some treatment options are most effective when given soon after a stroke begins.

Signs and symptoms of stroke include:

* **Trouble speaking and understanding what others are saying.** - confusion, slur your words or have difficulty understanding speech.
* **Paralysis or numbness of the face, arm or leg.** - sudden numbness, weakness or paralysis in your face, arm or leg. This often affects just one side of your body. Eg. one side of your mouth may droop when you try to smile.
* **Problems seeing in one or both eyes.** You may suddenly have blurred or blackened vision in one or both eyes, or you may see double.
* **Headache.** A sudden, severe headache, which may be accompanied by vomiting, dizziness or altered consciousness, may indicate that you're having a stroke.
* **Trouble walking.** You may stumble or lose your balance. You may also have sudden dizziness or a loss of coordination.

### Medical assistance -

**Seek immediate medical attention** if you notice any signs or symptoms of a stroke, even if they seem to come and go or they disappear completely. Think "FAST" and do the following:

* **Face.** Ask the person to smile. Does one side of the face droop?
* **Arms.** Ask the person to raise both arms. Does one arm drift downward? Or is one arm unable to rise?
* **Speech.** Ask the person to repeat a simple phrase. Is his or her speech slurred or strange?

## **Risk factors**

Many factors can increase your stroke risk. Potentially treatable stroke risk factors include:

### Lifestyle risk factors

* Being overweight or obese
* Physical inactivity
* Heavy or binge drinking
* Use of illegal drugs such as cocaine and methamphetamine

### Medical risk factors

* High blood pressure
* Cigarette smoking or secondhand smoke exposure
* High cholesterol
* Diabetes
* Obstructive sleep apnea
* Cardiovascular disease, including heart failure, heart defects, heart infection or abnormal heart rhythm, such as atrial fibrillation
* Personal or family history of stroke, heart attack or transient ischemic attack
* COVID-19 infection

Other factors associated with a higher risk of stroke include:

* **Age** — People age 55 or older have a higher risk of stroke than do younger people.
* **Race** — African Americans have a higher risk of stroke than do people of other races.
* **Gender** — Men have a higher risk of stroke than women. Women are usually older when they have strokes, and they're more likely to die of strokes than are men.
* **Hormones** — Use of birth control pills or hormone therapies that include estrogen increases risk.

## **Prevention**

Knowing your stroke risk factors, following your doctor's recommendations and adopting a healthy lifestyle are the best steps you can take to prevent a stroke. If you've had a stroke or a transient ischemic attack (TIA), these measures might help prevent another stroke.

Many stroke prevention strategies are the same as strategies to prevent heart disease. In general, healthy lifestyle recommendations include:

* **Controlling high blood pressure (hypertension).**
* **Lowering the amount of cholesterol and saturated fat in your diet.** Eating less cholesterol and fat, especially saturated fat and trans fats, may reduce the buildup in your arteries.
* **Quitting tobacco use.**
* **Managing diabetes.** Diet, exercise and losing weight can help you keep your blood sugar in a healthy range.
* **Maintaining a healthy weight.**
* **Eating a diet rich in fruits and vegetables.**
* **Exercising regularly.**
* **Treating obstructive sleep apnea (OSA).** A sleep disorder that causes you to stop breathing for short periods repeatedly during sleep. Treatment for OSA includes a device that delivers positive airway pressure through a mask to keep your airway open while you sleep.
* **Avoiding illegal drugs.**

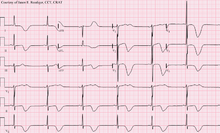
## **Diagnosis**

Stroke is diagnosed through several techniques: a neurological examination (such as the [NIHSS](https://en.wikipedia.org/wiki/NIHSS)), CT scans (most often without contrast enhancements) or [MRI scans](https://en.wikipedia.org/wiki/MRI_scan), [Doppler ultrasound](https://en.wikipedia.org/wiki/Doppler_ultrasound), and [arteriography](https://en.wikipedia.org/wiki/Arteriography). The diagnosis of stroke itself is clinical, with assistance from the imaging techniques. Imaging techniques also assist in determining the subtypes and cause of stroke. There is yet no commonly used [blood test](https://en.wikipedia.org/wiki/Blood_test) for the stroke diagnosis itself, though blood tests may be of help in finding out the likely cause of stroke.

### **Physical examination**

A [physical examination](https://en.wikipedia.org/wiki/Physical_examination), including taking a [medical history](https://en.wikipedia.org/wiki/Medical_history) of the symptoms and a neurological status, helps giving an evaluation of the location and severity of a stroke. It can give a standard score on e.g., the [NIH stroke scale](https://en.wikipedia.org/wiki/NIH_stroke_scale).

### Underlying cause

[](https://en.wikipedia.org/wiki/File:Left_MCA_Stroke.png)

12-lead ECG of a patient with a stroke, showing large deeply inverted [T-waves](https://en.wikipedia.org/wiki/T_wave). Various ECG changes may occur in people with strokes and other brain disorders.

Notes-

**High levels of physical activity reduce the risk of stroke by about 26%**. There is a lack of high-quality studies looking at promotional efforts to improve lifestyle factors. Nonetheless, given the large body of circumstantial evidence, best medical management for stroke includes advice on diet, exercise, smoking and alcohol use. Medication is the most common method of stroke prevention; [carotid endarterectomy](https://en.wikipedia.org/wiki/Carotid_endarterectomy) can be a useful surgical method of preventing stroke.

#### **Blood pressure**

[**High blood pressure**](https://en.wikipedia.org/wiki/Hypertension) **accounts for 35–50% of stroke risk**. Blood pressure reduction of 10 mmHg systolic or 5 mmHg diastolic reduces the risk of stroke by ~40%. Lowering blood pressure has been conclusively shown to prevent both ischemic and hemorrhagic strokes. It is equally important in secondary prevention.

#### **Blood lipids**

**High cholesterol levels** have been inconsistently associated with (ischemic) stroke. [**Statins**](https://en.wikipedia.org/wiki/Statins) **have been shown to reduce the risk of stroke by about 15%.**

#### **Diabetes mellitus**

[**Diabetes mellitus**](https://en.wikipedia.org/wiki/Diabetes_mellitus) **increases the risk of stroke by 2 to 3 times.** While intensive blood sugar control has been shown to reduce small blood vessel complications such as [kidney damage](https://en.wikipedia.org/wiki/Diabetic_nephropathy) and [damage to the retina of the eye](https://en.wikipedia.org/wiki/Diabetic_retinopathy) it has not been shown to reduce large blood vessel complications such as stroke.

#### **Diet**

Nutrition, specifically the [Mediterranean-style diet](https://en.wikipedia.org/wiki/Mediterranean_diet), has the potential for decreasing the risk of having a stroke by more than half.

### **Women**

A number of specific recommendations have been made for women including taking aspirin after the 11th week of pregnancy if there is a history of previous chronic high blood pressure and taking blood pressure medications during pregnancy if the blood pressure is greater than 150 mmHg systolic or greater than 100 mmHg diastolic. In those who have previously had [preeclampsia](https://en.wikipedia.org/wiki/Preeclampsia) other risk factors should be treated more aggressively.

### **Previous stroke or TIA**

Keeping blood pressure below 140/90 mmHg is recommended. Anticoagulation can prevent recurrent ischemic strokes. **Among people with nonvalvular atrial fibrillation, anticoagulation can reduce stroke by 60% while antiplatelet agents can reduce stroke by 20%**. However, a recent meta-analysis suggests harm from anticoagulation started early after an embolic stroke. **Stroke prevention treatment for atrial fibrillation is determined according to the** [**CHA2DS2–VASc score**](https://en.wikipedia.org/wiki/CHA2DS2%E2%80%93VASc_score). The most widely used anticoagulant to prevent thromboembolic stroke in people with nonvalvular atrial fibrillation is the oral agent [warfarin](https://en.wikipedia.org/wiki/Warfarin) while a number of newer agents including [dabigatran](https://en.wikipedia.org/wiki/Dabigatran) are alternatives which do not require [prothrombin time](https://en.wikipedia.org/wiki/Prothrombin_time) monitoring.

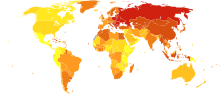
## **Prognosis**

Disability affects 75% of stroke survivors enough to decrease their ability to work. Stroke can affect people physically, mentally, emotionally, or a combination of the three. The results of stroke vary widely depending on size and location of the lesion.

### **Physical effects**

Some of the physical disabilities that can result from stroke include muscle weakness, numbness, [pressure sores](https://en.wikipedia.org/wiki/Pressure_sore), [pneumonia](https://en.wikipedia.org/wiki/Pneumonia), [incontinence](https://en.wikipedia.org/wiki/Urinary_incontinence), [apraxia](https://en.wikipedia.org/wiki/Apraxia) (inability to perform learned movements), difficulties carrying out [daily activities](https://en.wikipedia.org/wiki/Activities_of_daily_living), appetite loss, [speech loss](https://en.wikipedia.org/wiki/Aphasia), [vision loss](https://en.wikipedia.org/wiki/Vision_loss) and [pain](https://en.wikipedia.org/wiki/Pain). If the stroke is severe enough, or in a certain location such as parts of the brainstem, [coma](https://en.wikipedia.org/wiki/Coma) or death can result. Up to 10% of people following a stroke develop [seizures](https://en.wikipedia.org/wiki/Seizure), most commonly in the week subsequent to the event; the severity of the stroke increases the likelihood of a seizure. An estimated 15% of people experience urinary incontinence for more than a year following a stroke. 50% of people have a decline in sexual function ([sexual dysfunction](https://en.wikipedia.org/wiki/Sexual_dysfunction)) following a stroke.

## **Epidemiology**

[](https://en.wikipedia.org/wiki/File:Stroke_world_map-Deaths_per_million_persons-WHO2012.svg)

Stroke deaths per million persons in 2012

  58–316

  317–417

  418–466

  467–518

  519–575

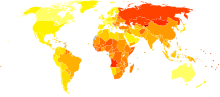
  576–640

  641–771

  772–974

  975-1,683

  1,684–3,477

[](https://en.wikipedia.org/wiki/File:Cerebrovascular_disease_world_map_-_DALY_-_WHO2004.svg)

[Disability-adjusted life year](https://en.wikipedia.org/wiki/Disability-adjusted_life_year) for cerebral vascular disease per 100,000 inhabitants in 2004.[[208]](https://en.wikipedia.org/wiki/Stroke#cite_note-208)

|  |  |
| --- | --- |
| no data    <250    250–425    425–600    600–775    775–950    950–1125 | 1125–1300    1300–1475    1475–1650    1650–1825    1825–2000    >2000 |

Stroke was the second most frequent cause of death worldwide in 2011, accounting for 6.2 million deaths (~11% of the total). Approximately 17 million people had a stroke in 2010 and 33 million people have previously had a stroke and were still alive.

[Map

Description automatically generated](https://www.google.com/search?q=stroke+stats&client=firefox-b-d&tbm=isch&source=iu&ictx=1&fir=D_hrSkM_yRrvlM%252CMQZozmkToGaRvM%252C_&vet=1&usg=AI4_-kTggl4bO53wQe859Qe4bh-6y-u7rQ&sa=X&ved=2ahUKEwj1t5mnqKnxAhXEYysKHSPPCv8Q9QF6BAgIEAE" \l "imgrc=D_hrSkM_yRrvlM)

In 2018, 1 in every 6 deaths from cardiovascular disease was due to **stroke**. Someone in the United States has a **stroke** every 40 seconds. Every 4 minutes, someone dies of **stroke**. Every year, more than 795,000 people in the United States have a **stroke**.

* Each year, 795,000 people suffer a stroke.
* On average, someone has a stroke every 40 seconds in the United States.
* [Ischemic strokes](https://www.verywellhealth.com/what-is-a-ischemic-stroke-3146288) account for 87% of all strokes, while 10% are [hemorrhagic strokes](https://www.verywellhealth.com/hemorrhagic-strokes-3146286) and 3% are subarachnoid hemorrhage strokes.

A picture containing graphical user interface

Description automatically generated

**The risk of stroke** [**increases exponentially**](https://en.wikipedia.org/wiki/Exponential_growth) **from 30 years of age, and the cause varies by age.** Advanced age is one of the most significant stroke risk factors. **95% of strokes occur in people age 45 and older, and two-thirds of strokes occur in those over the age of 65**. A person's risk of dying if he or she does have a stroke also increases with age. However, stroke can occur at any age, including in childhood.

Family members may have a genetic tendency for stroke or share a lifestyle that contributes to stroke. Higher levels of glycoprotein that is present in the plasma are more common amongst people who have had ischemic stroke for the first time. Having had a stroke in the past greatly increases one's risk of future strokes.

**Men are 25% more likely to suffer strokes than women, yet 60% of deaths from stroke occur in women**. Since women live longer, they are older on average when they have their strokes and thus more often killed. Some risk factors for stroke apply only to women. Primary among these are pregnancy, childbirth, [menopause](https://en.wikipedia.org/wiki/Menopause), and the treatment thereof ([HRT](https://en.wikipedia.org/wiki/Hormone_replacement_therapy_(menopause))).

**1.2 MOTIVATION**

The stroke prediction model provides the chances of predicting stroke to the user. This is done by collecting the inputs from user from the web platform and use the trained model to predict the chance of stroke to the user. It is easy to use and always available and can be accessed from anywhere and anytime through the web enabled device. Whenever a person feels symptoms of stroke like loss in vision in one eye, inability to move or feel on one side of the body, problem in understanding or speaking, dizziness these could be symptoms of stroke. There are some key risk factors like blood pressure, smoking habits, high blood cholesterol, obesity, diabetes, kidney problem etc. which are often known to user. In this case he/she can do a self-diagnosis test for chances for stroke risk to him/her and then consult a doctor. Timely detection can help to a large extend and can even save life in some instances. The ease to access app will provide accessibility. Also, the project will help us understand the capability and factors useful for machine learning classifiers for predicting such results.

**1.3 PROBLEM STATEMENT**

To compare various machine learning classifiers based on accuracy and select the best one to build a system for predicting stroke risk.

**1.4 SCOPE OF PROJECT**

This project provides user an easy to access app which they can use to check the chances of stroke to them. The aim of the project is to make such app available to everyone. This project also plans to learn which machine learning classifiers are best suited for this kind of job and understand the reason behind it.

**1.5 ACCESS TO USERS**

The project aims to host the web app for the users making it easily available for everyone at all time. The interface aims to be very basic and user friendly. The major objective is this app can be used by users on phone instantaneously and does not require any expertise.

**CHAPTER - 2**

**RELATED WORK**

**2.1 INTRODUCTION**

In this section, we will look into some of the earlier works presented in the direction of this project. There has been an extensive amount of work that has already happened in the field of image recognition and natural language processing. While preparing this project we have gathered ideas from different research materials and models. Here, we are going to enlist them.

**2.2 EXISTING SYSTEM**

Lately a lot of research has been done on automatic image captioning. The research can be briefly categorized into three different categories including the template-based approaches, retrieval-based approaches, and novel image caption generation approaches.

The template-based approach is aimed at generating captions by using fixed templates with a number of blank slots, in which way different objects, attributes, and actions are detected first and then the blank spaces in the templates are filled. For example, Farhadi use a triplet of scene elements to fill the template slots for generating image captions. Li et al extract the phrases related to detected objects, attributes, and their relationships for this purpose. Kulkarni et al. adopt a conditional random field (CRF) method to infer the objects, attributes, and prepositions before filling in the gaps. Template-based methods can generate grammatically correct captions. However, templates are predefined and length of captions cannot be variable.

The retrieval-based approach tries to generate description for an image by selecting the most semantically similar sentences from sentence pool or directly copying sentences from other visually similar images. For example, Gong et al. utilize stacked auxiliary embedding method to generate image descriptions from millions of weakly annotated images. Ordonez find similar images in the Flickr database and return the descriptions of these retrieved images to query based on millions of images and their corresponding descriptions. Sun use semantic similarity and visual similarity scores to cluster similar terms and images together first and then retrieve the caption of target image from captions of similar images in the same cluster. Hodosh establish a ranking-based framework to treat sentence-based image description as the task of ranking a set of captions for each test image. These methods generate general and syntactically correct captions. However, it is difficult for them to generate image-specific and semantically correct captions.

Different from the mentioned two categories, novel caption generation approaches mainly use deep learning and machine learning to generate the new captions. A general implementation of this method is to analyze the visual content of the image first and then generate image captions from the visual content using a language model. For instance, Vinyals et al. use CNN as an encoder for image classification and LSTM as a decoder to generate sentence for the description. The main drawbacks of the work are the quick model overfitting, so they use the heavy and expensive GoogLeNet with 22 hidden layers and the absence of attention layer that significantly improved the description accuracy. Karpathy investigate the possibility of generating an image description in natural language. Their approach uses image datasets and their description in natural language and seeks an intermodal correspondence between words from the description and visual data. The first model aligns the fragments of sentences to the visual areas, then forms a single description by multimodal embedding. This description is treated as learning data for a second model of a recurrent neural network that learned to a generate caption. Xu et al. use a convolutional neural network to extract feature maps and LSTM to describe the input image, by processing already extracted feature maps. The limitation of this work is the using of obsolete and expensive Oxford VGGnet, where the quality of image classification is low in the modern CNN . Some researchers have put their attention on classification as Yu. who propose a SVM classification-based two-side cross-domain algorithm by inferring intrinsic user and item features (CTSIF-SVMs), a two-side cross-domain algorithm with expanding user and item features via the latent factor space of auxiliary domains (TSEUIF).

**2.3 PROPOSED SYSTEM**

Our proposed model, for automatic image captioning, is based on ResNet50 and LSTM. The ultimate purpose of this model is to generate the proper description for the given images. To do so, the model is designed with an encoder-decoder architecture based on CNN and RNN. In particular, to extract visual features, we use the ResNet50 network as the encoder to generate a one-dimensional vector representation of the input images. After that, to generate the description sentences, we adopt the LSTM as the language model for the decoder to decode the vector into a sentence.

**2.3.1. Image Feature Extraction**

For image feature extraction we used CNN, RestNet50, which is a deep network that has 50 layers. The image feature extractor needs an image of 224x224x3 size. The model uses ResNet50 pretrained on ImageNet dataset where the features of the image are extracted just before the last layer of classification. Another dense layer is added and converted to get a vector of length 2048.

The model takes an image and produces a caption, encoded as a sequence of 1-K coded words.



where K is the size of the dictionary and *c* is the caption length. We use CNN in particular, ResNet50, to obtain set annotation vectors like the feature vectors. The extractor produces L-vectors, all of which is a D-dimensional representation of the corresponding part of an image.

##### **2.3.2. The Language Model**

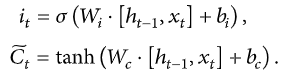
In our design, we adopt LSTM as our language model to generate proper caption based on the input vector from the ResNet50 output.



where the output vector of the previous cell ht-1 with the new element of the sequence xt is concatenated and passed as one vector through the layer with the sigmoid activation function.



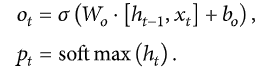
Two created vectors are used to update the state from Ct-1 to Ct. To do this, we multiply the past state by ƒt to “forget” the data recognized as unnecessary in the previous step, then add .



The input gate must determine what values will be updated, and the tanh layer creates a vector of new candidates for  , and values can be added to the cell state.



The obtained values of Ct and ht are transmitted to the neural network input at a time *t*+1.



The multiplicative filters allow to effectively train LSTM, as they are good to prevent the exploding and vanishing gradients. Nonlinearity is provided by the sigmoid 𝝈(.) and the hyperbolic tangent h(). In the last equation, is fed to the softmax function to calculate the probability distribution pt over all the words. This function is calculated and optimized on the entire training dataset. The word with maximum probability is selected at each time step and fed into the next time step input to generate a full sentence.

**2.3.3. Fitting the Model**

After building the model, the model is fit using the training dataset. The model is made to run for 50 epochs and the best model is chosen among the 50 epochs by computing loss function on Flickr8k development dataset. The model with the lowest loss function is chosen for generating captions.

**2.3.4. Caption Generation**

To test our trained model, we input an image to the model. Next the image is fed into the feature extractor to recognize what all objects and scenes are depicted in the image, after resizing it. The process of caption generation is done using the RNN trained model. Then for that image, sequentially, word-by-word the caption is generated by selecting the word with maximum weight for the image at that particular iteration. The indexed word is converted to word and then appended into the final caption. When a tag is detected or the size of the caption reaches 40, the final caption is generated and printed along with the input image.

##### **2.3.5. Suggestion API**

Finally, in the web app, the description string is then passed to API which will suggest a few caption options to the user for selecting their preferred one.

**CHAPTER - 3**

**PROBLEM DESCRIPTION AND SPECIFICATION**

**3.1 PROBLEM DESCRIPTION**

The problem introduces a captioning task, which requires a computer vision system to both localize and describe salient regions in images in natural language. The image captioning task generalizes object detection when the descriptions consist of a single word.

**3.2 SPECIFICATION**

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English.

The aim of image captioning is to automatically describe an image with one or more natural language sentences. This is a problem that integrates computer vision and natural language processing, so its main challenges arise from the need of translating between two distinct, but usually paired modalities. First, it is necessary to detect objects on the scene and determine the relationships between them and then, express the image content correctly with properly formed sentences. The generated description is still much different from the way people describe images because people rely on common sense and experience, point out important details and ignore objects and relationships that they imply. Moreover, they often use imagination to make descriptions vivid and interesting. The process flow diagram below in Figure 1 outlines the working of the project.

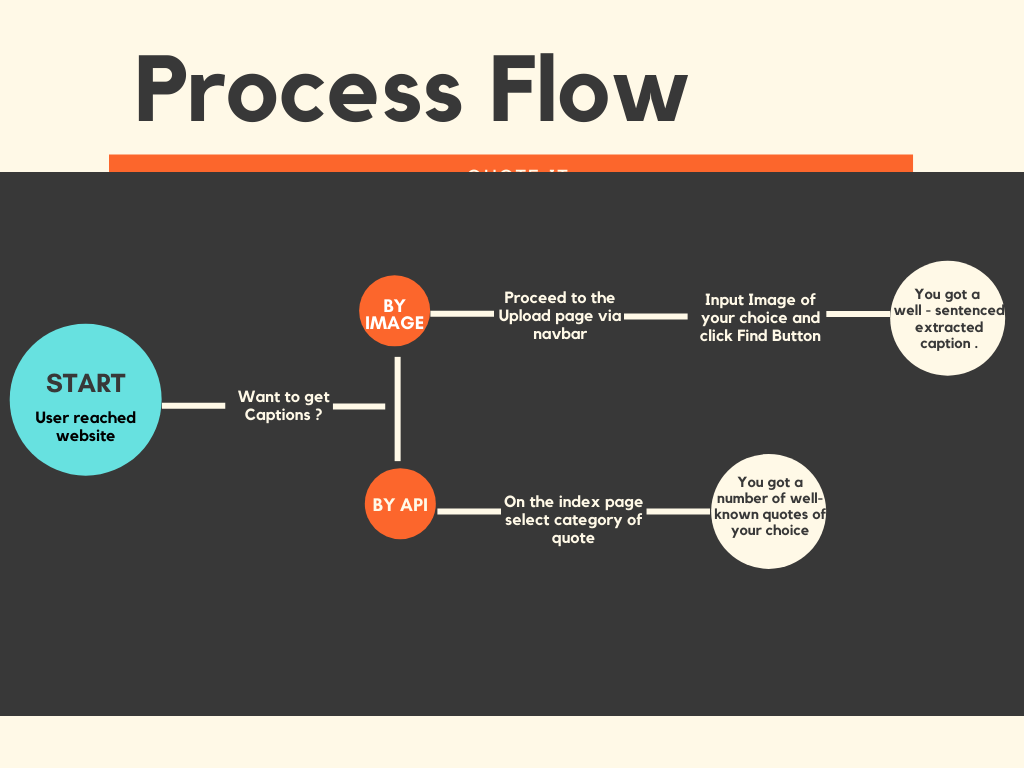


Figure 1: Process Flow Diagram

**3.3 REQUIREMENTS**

**3.3.1 SOFTWARE REQUIREMENTS**

* Python Version 3.6+
* IDE: JupyterLab or PyCharm
* Frameworks:-
  + Flask
  + Bootstrap
* Python Libraries:-
  + Scikit-learn
  + Seaborn
  + Numpy: 1.18.5
  + Pandas: 1.1.14

**3.3.2 HARDWARE REQUIREMENTS**

* Processor:- Intel i3+ or AMD A6+
* RAM:- 2GB/+
* Hard Disk Space:- 2GB

**3.3.3 OTHER REQUIREMENTS**

* Stroke Prediction Dataset
* For training the model - Kaggle Notebook/Google Colab/JupyterLab should be preferred (Recommended)

**CHAPTER - 4**

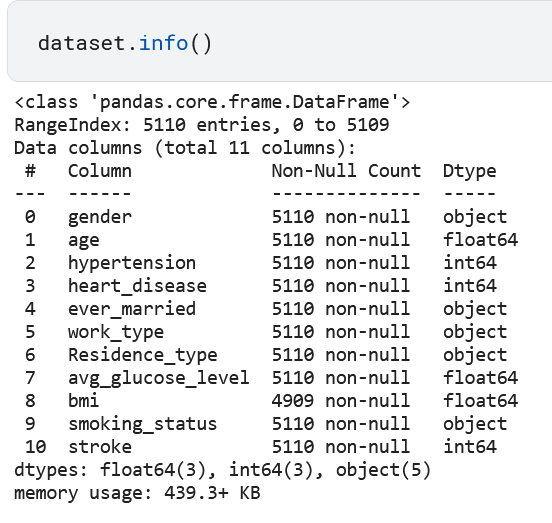
**SYSTEM DESIGN**

**4.1 INTRODUCTION**

We have written data preprocessing scripts to process raw input data (both images and captions) into proper format; A pre-trained Convolutional Neural Network architecture as an encoder to extract and encode image features into a higher dimensional vector space; An LSTM-based Recurrent Neural Network as a decoder to convert encoded features to natural language descriptions; Attention mechanism which allows the decoder to see features from a specifically highlighted region of the input image to improve the overall performance

**4.1.1. Data Sources**

We had selected “Stroke-prediction dataset” obtained from Kaggle as our dataset. This dataset contains of 5110 records having 12 attributes (id, age, gender, hypertension, heart disease, marital status, residence type, work type, average glucose level, BMI, smoking status and stroke occurrence). The below image provides the info about the dataset.



**4.1.2. Convolutional Neural Network (Encoder)**

The encoder needs to extract image features of various sizes and encode them into vector space which can be fed to RNN in a later stage. VGG-16 and ResNet are commonly recommended as image encoders. We chose to modify the pre-trained ResNet50 model provided by Tensorflow library. In this task, CNN is used to encode features instead of classifying images. As a result, we removed the fully connected layers and the max pool layers at the end of the network. Under this new construction, the input image matrix has dimension N ×3×256×256, and the output has dimension N × 14 × 14 × 512. Furthermore, in order to support input images with various sizes, we added an adaptive 2d layer to our CNN architecture.

**4.1.3. Recurrent Neural Network (Decoder)**

The decoder needs to generate image captions word by word using a Recurrent Neural Network - LSTMs which is able to sequentially generate words. The input for the decoder is the encoded image feature vectors from CNN and the encoded image captions produced in data preprocessing stage.

**4.2 ARCHITECTURE DIAGRAM**

The Backend System architecture Diagram is as follows:

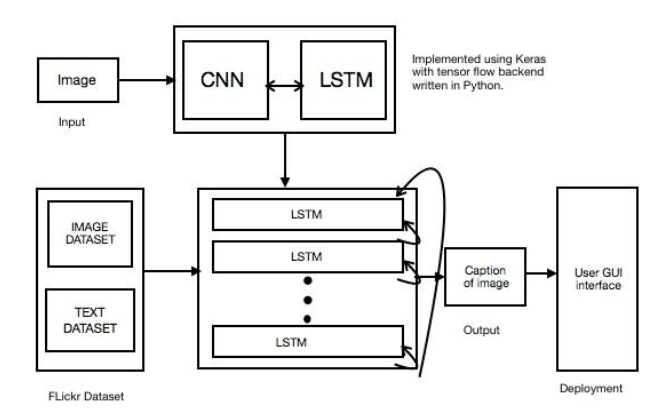
****

Figure 2: Architecture of the image captioning model

The model uses ResNet50 as the CNN model and generated Region features of the passed images which are feed to RNN(LSTM) network which is fed to the model for predicting the next word in the captions on the basis of highest probability obtained. All these next words are added to generate the captions till the final caption is generated. Figure 2 shows a brief architecture for the project. The image is passed and fed to CNN and LSTM model, which internally is composed of various layers and fed with regional image and text dataset. At the user interacts with the project through the GUI deployed for them. Figure 3 explains the process used by CNN i.e, object detection and feature map preparation. These features along with text dataset are given to the vector of LSTM model with predicts the objects in the image and return the word in the sequence which has the highest probability which is used for generating the caption.

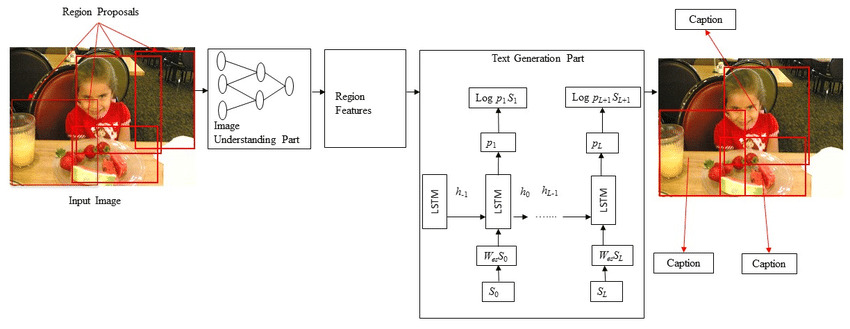


Figure 3: Illustration image of functioning of the captioning model

**4.3 UNIFIED MODELING LANGUAGE(UML)**

**4.3.1 USE CASE DIAGRAM**

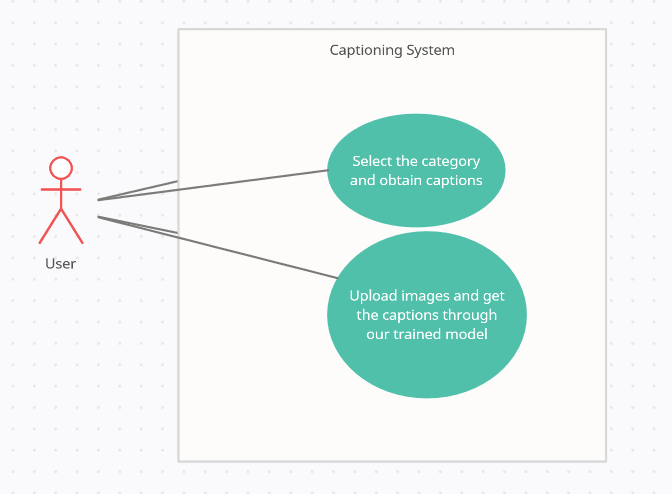
****

Figure 4: Use Case Diagram for the captioning system

Figure 4 shows that there are basically two use-cases available for the user. When the user visits the web site they can choose:

1. Any category available on the home page to obtain the collection of images of that category and then select and copy the caption of their choice.
2. User can navigate to the Upload section and then upload the image from their device and then by the help of the image caption generator model running in the background he/she can obtain the caption for the provided image.

**4.3.2 DATA FLOW DIAGRAM**

**4.3.2.1 Data Flow Diagram Level 0**

****

Figure 5: DFD level 0

In Figure 5 the level 0 DFD shows the top-level view of the data flow. It shows that initially when the user interacts with the GUI, they have two option. The first one to choose a caption of their favourite category and the second option is to provide an image to the model for generation caption.

**4.3.2.2 Data Flow Diagram Level 1**

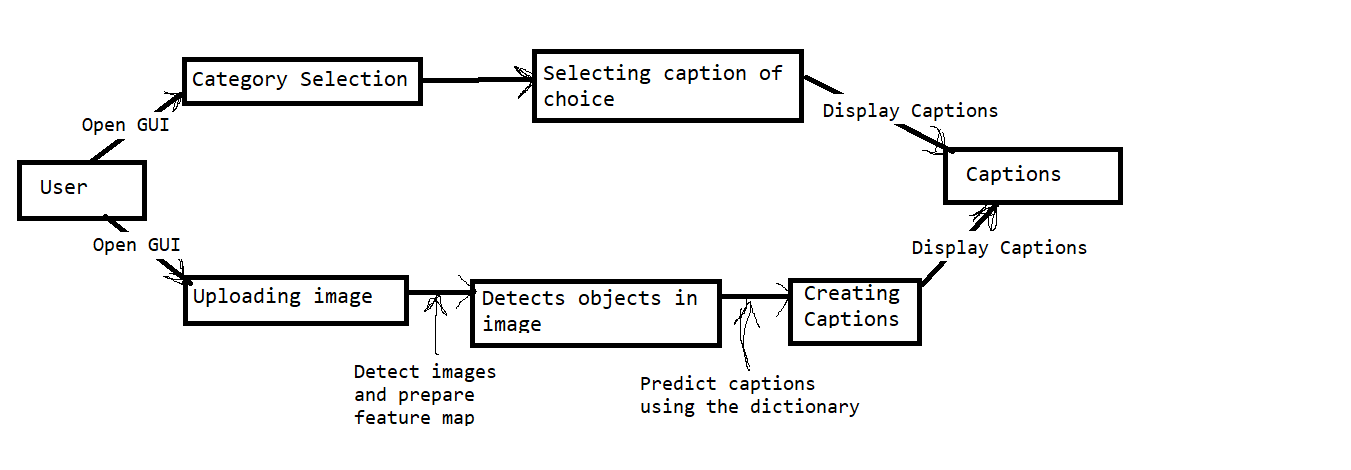
****

Figure 6: DFD level 1

In figure 6 we see the medium level view of data flow. It shows how the input passed goes through a channel of different segments. In our case when a user uploads the image it passes through the CNN model and the features of the image generated are then passed to Language model which gives the captions to the GUI for display.

**4.3.2.3 Data Flow Diagram Level 2**

****

Figure 7: DFD level 2

In figure 7 explains the functioning of the model in great depth. The first option remains the same, but the second option shows the usage of a regional dataset to train the model. This improves the functioning of the model by training the model on our custom dataset.

**4.4 STATE CHART DIAGRAM**

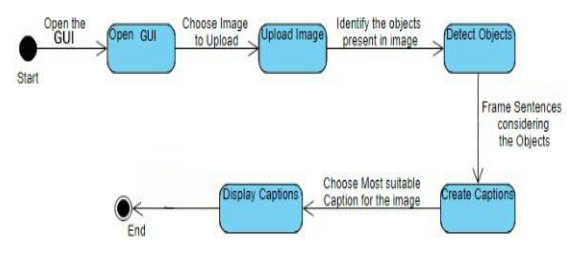
****

Figure 8: State Chart Diagram of the system

Figure 8 shows the state of the program at every deciding state and its output which are fed to the next state as input. The program basically contains five states as stated in the figure. The GUI and uploading of the image are handled by the flask library. The object detection phase uses the CNN model to detect the objects in the figure and then the create caption state uses the LSTM model to generate the captions.

**4.5 FLOW CHART FOR CAPTION GENERATION PROCESS**

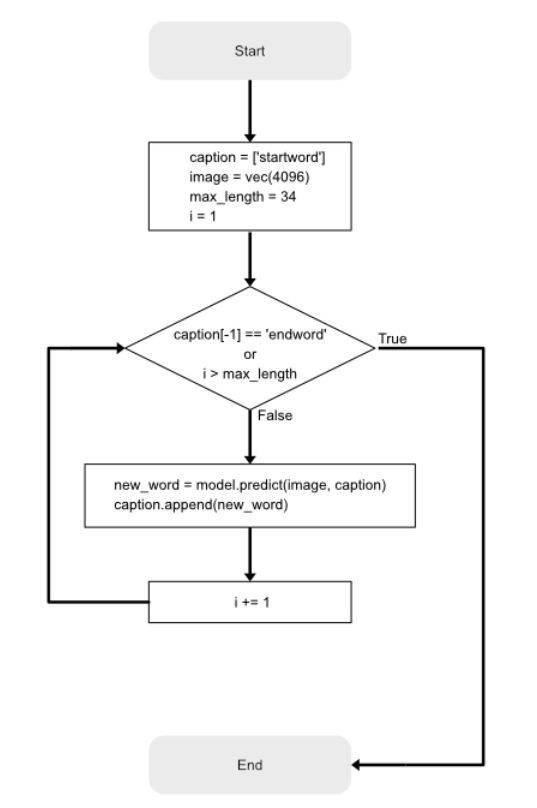
****

Figure 9: Flow chart for the caption generation process

Figure 9 tells us about the functioning of the caption prediction task. Once the image feature of the passed test image is extracted. The trained model tries to predict the next word in the caption by using the Dense layer and agrmax function. The inverse dictionary is used for obtaining the word corresponding to the integer provided. When the “endword” is passed to the model, it stops and the final caption is given in output.

**CHAPTER - 5**

**IMPLEMENTATION AND RESULT**

**5.1 OVERVIEW**

In this section, we are going to cover all the steps we have taken while implementing the program. The whole program consists of nine different sections which we are going to cover in later sections.

**5.2 MODEL SELECTION**

We have used the ResNet50 model of the TensorFlow in our project for image processing as shown in Figure 10. We first import ResNet50 from the Tensorflow module, which can be found in tensorflow.keras.applications module. Once we import the model, it will download the model from the TensorFlow website and save it into the ‘incept\_model’ variable. Then we remove the last two layers of the ResNet50 model as the last to later are not of our use. We also save the model in our local directory using the ‘save’ method; so that it can be used later in offline mode.

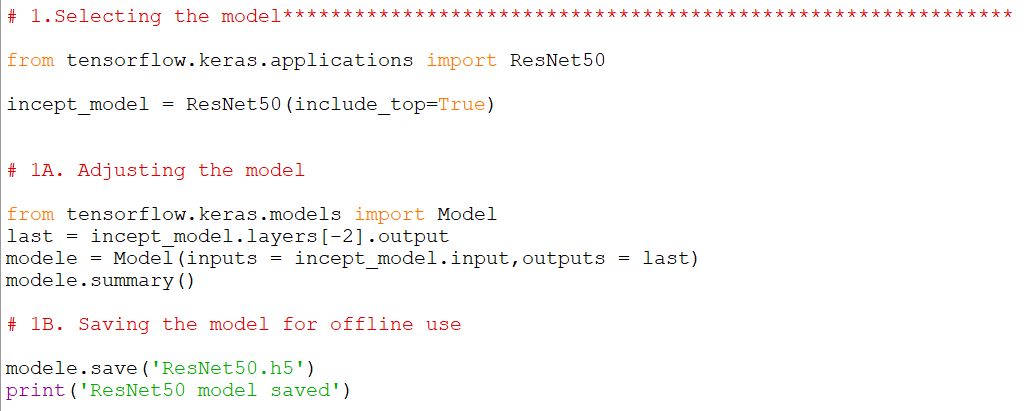
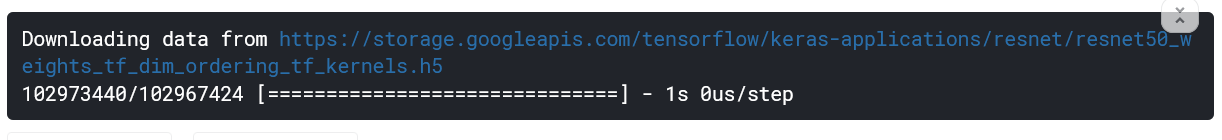


Figure 10: Code Snippet showing the selection and adjustment of the model



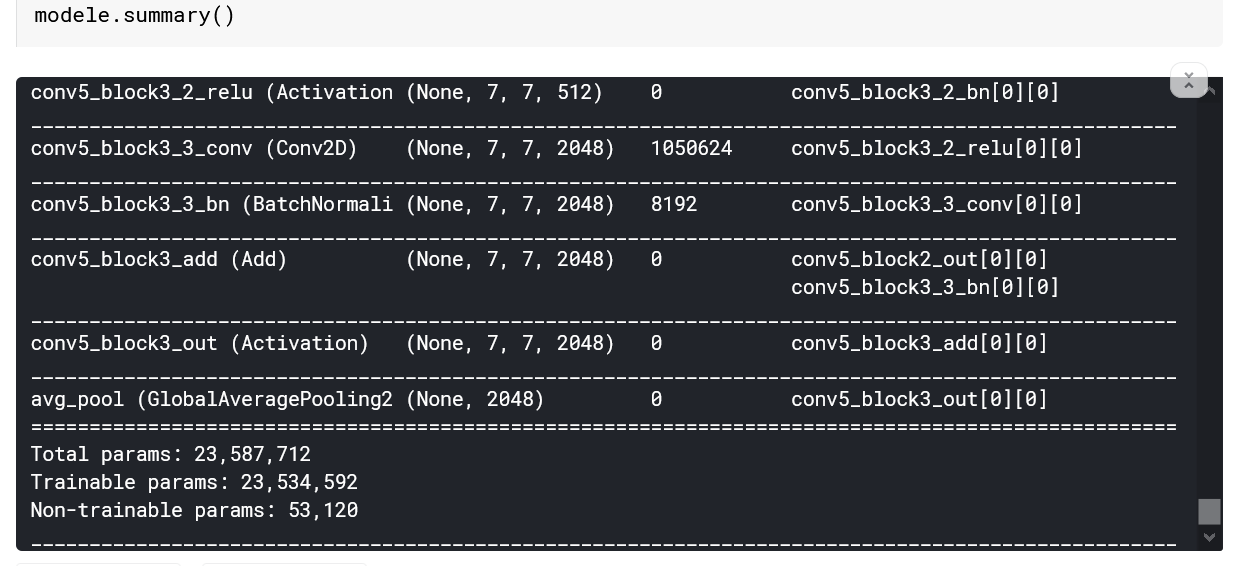


Figure 11: Output representing different layers of the selected model

**5.3 IMAGE PREPROCESSING**

We use ‘Flickr8k Dataset’ which we have downloaded and saved in the local directory as shown in Figure 12. By using the ResNet50 model we are extracting image features and saving it for later use in a pickle file named ‘features.p’.



Figure 12: Code snippet showing preprocessing of images

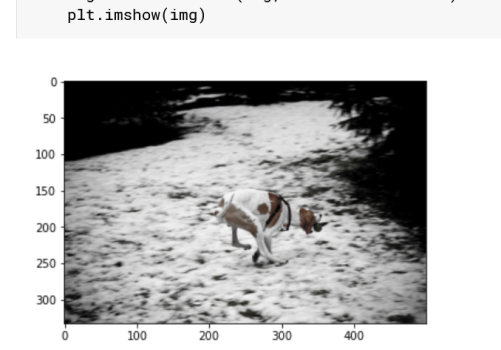


Figure 13: A sample picture fetched from the dataset

Figure 14 shows the process of creating a feature map of a specified limit of images from the regional image dataset. Here we have selected 1500 images for the training purpose because of hardware limitations. The image in the images array is fetched and converted to a coloured and resized image of 224 x 224 x 3 dimension. Then our defined ResNet 50 model as “modele” extracts and store the image feature in the image feature variable in the NumPy array form. The process of extraction is shown in figure 15 and figure 16 shows a snap of image features of some sample images.

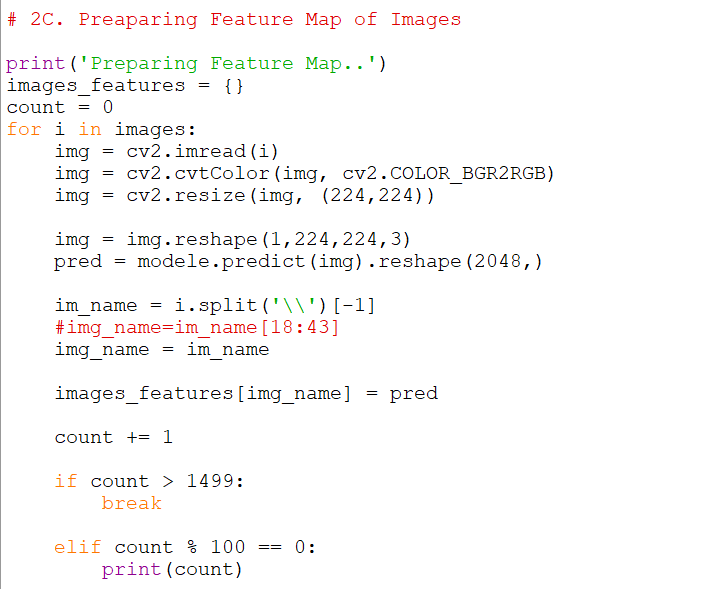
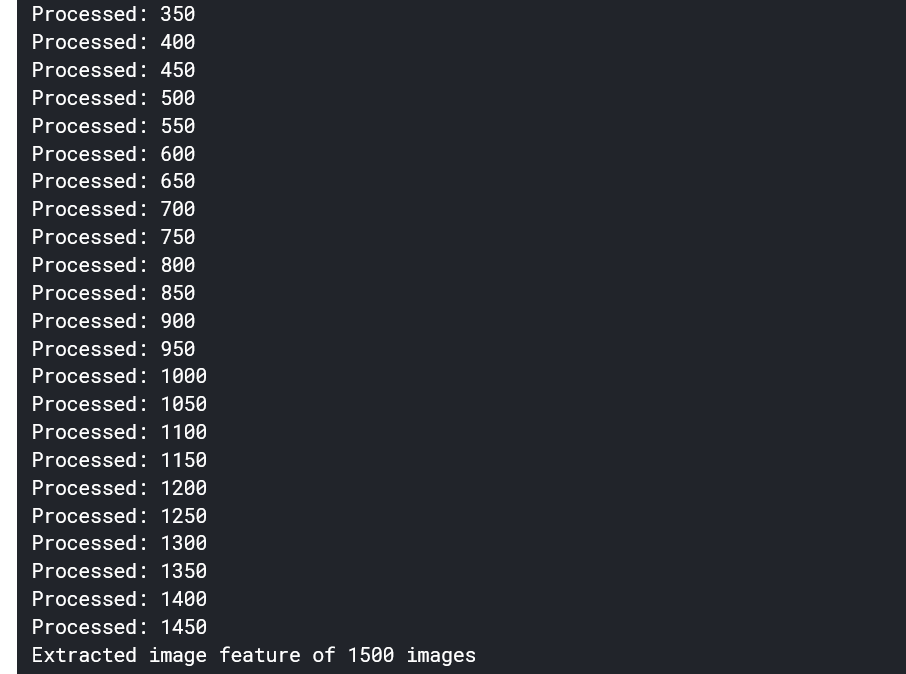


Figure 14: Extracting feature map from the passed images



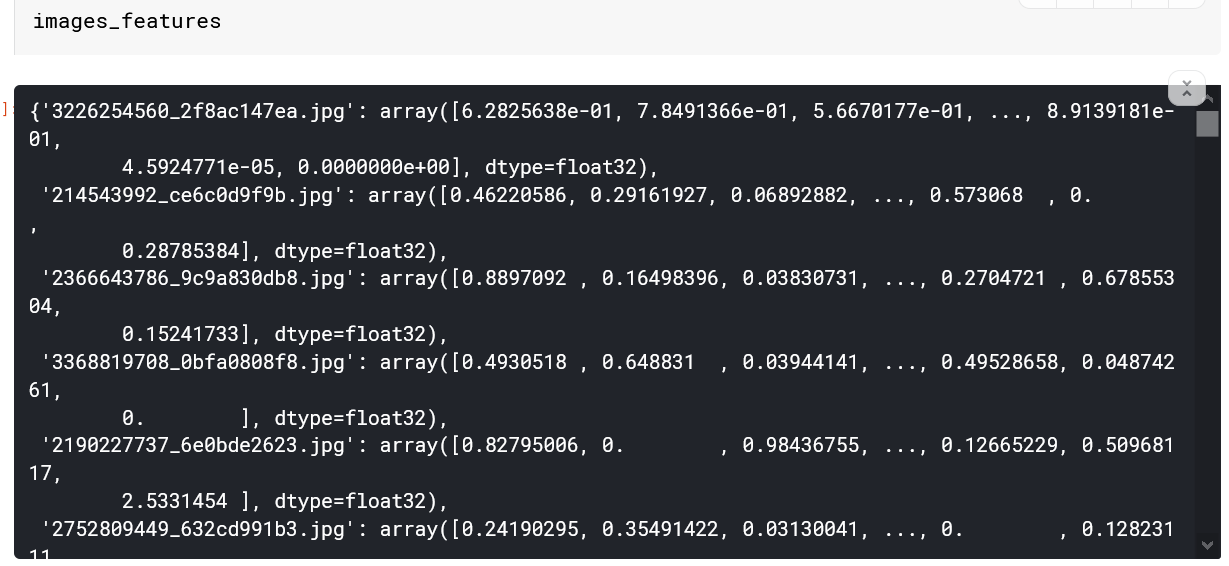


Figure 15: Output representing the process of feature map creations and sample view of the feature map of some images

**5.4 TEXT PREPROCESSING**

We are using ‘Flickr8k.token.txt’ which contains five captions for all the images in the Flickr8k Image dataset. We split them into different lines and convert them in string form, ready for our operation as shown in figure 16 and figure 17. Then we prepare a dictionary consisting of images as the keys and a list of 5 captions as the values for all the images in the feature map that we saved earlier. Then we modify the captions by adding ‘startofseq’ word at the front of every caption denoting the start of the caption and ‘endofseq’ at the end of every caption denoting the end of that particular caption. Then we again attach these modified captions with the respective images as shown in figure 18 and the corresponding results in figure 19. We did this for preparing the caption vector which will later be used by our model for prediction.

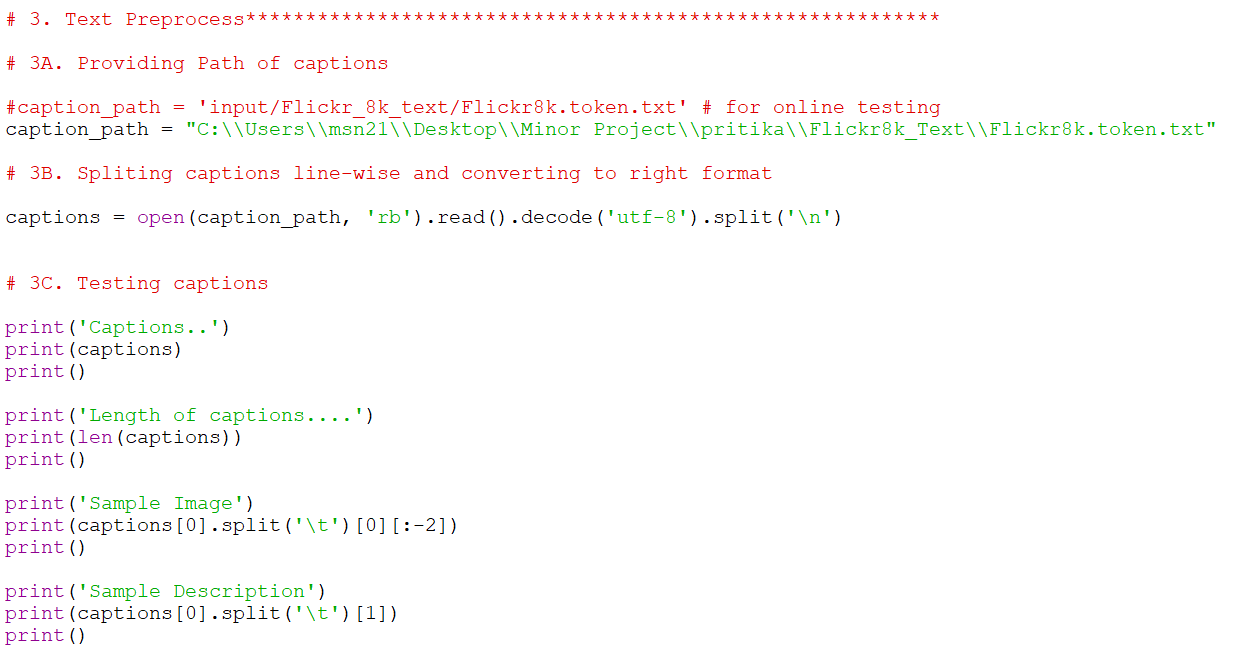


Figure 16: Code snippet showing text preprocessing

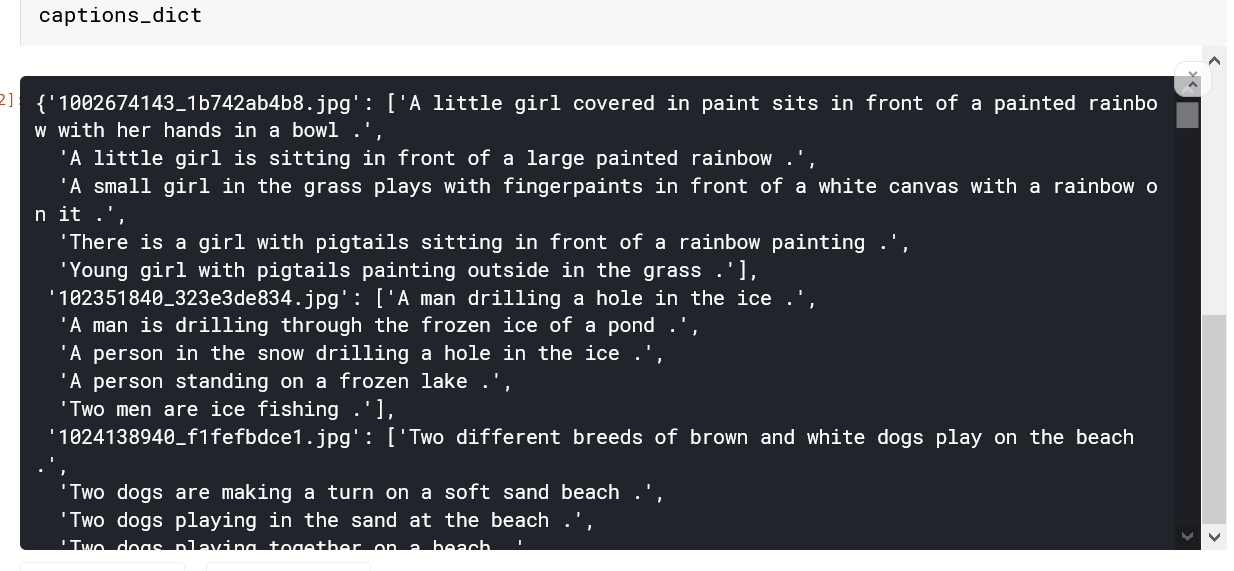


Figure 17: Sample captions fetched from Flickr8k.token.txt

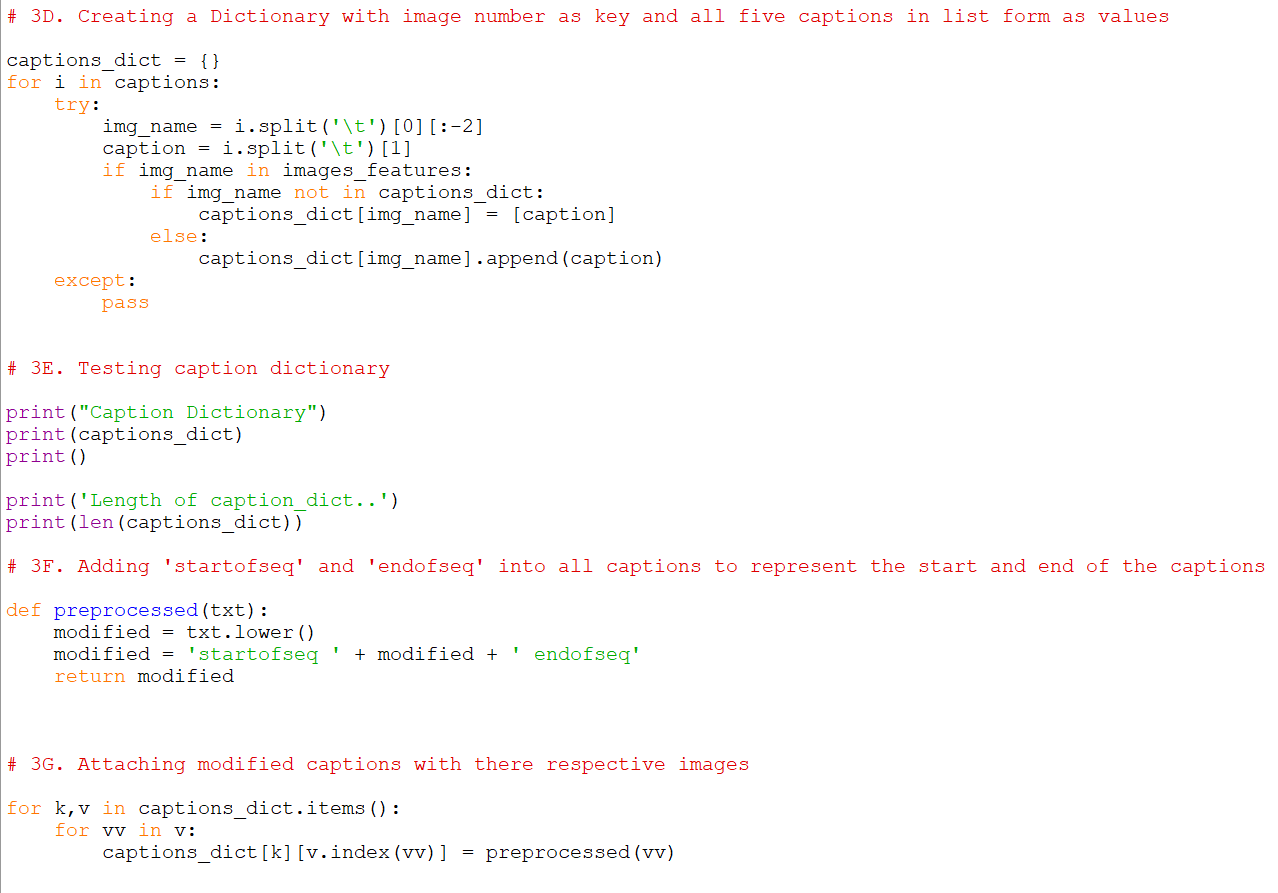


Figure 18: Code snippet representing the modification process of the captions and caption dictionary creation

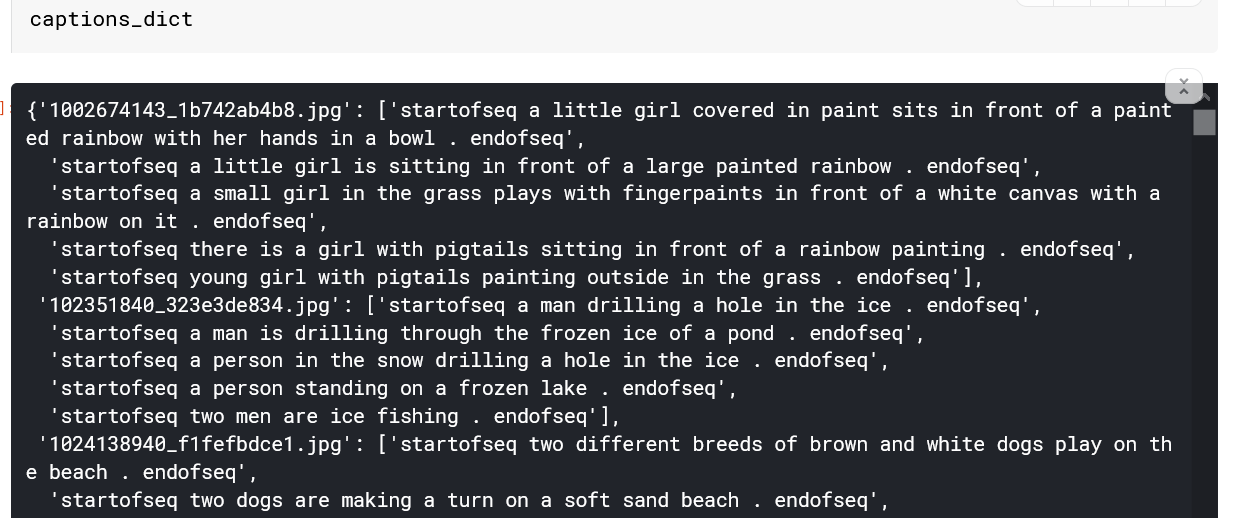


Figure 19: Sample view of caption dictionary

**5.5 CREATING VOCABULARY**

Next, ahead we count the number of times any particular word appears in any of the captions among all the captions present as shown in figure 20 code snippet. Then we provide an integer number to every word forming a dictionary of different words present in the captions as shown in results in figure 22. Then we replace every word in every caption with the integer key corresponding to it as the value in the caption dictionary which earlier had images and the list of its corresponding five captions. The new caption dictionary contains images and corresponding captions but in integer form as shown in figure 23.

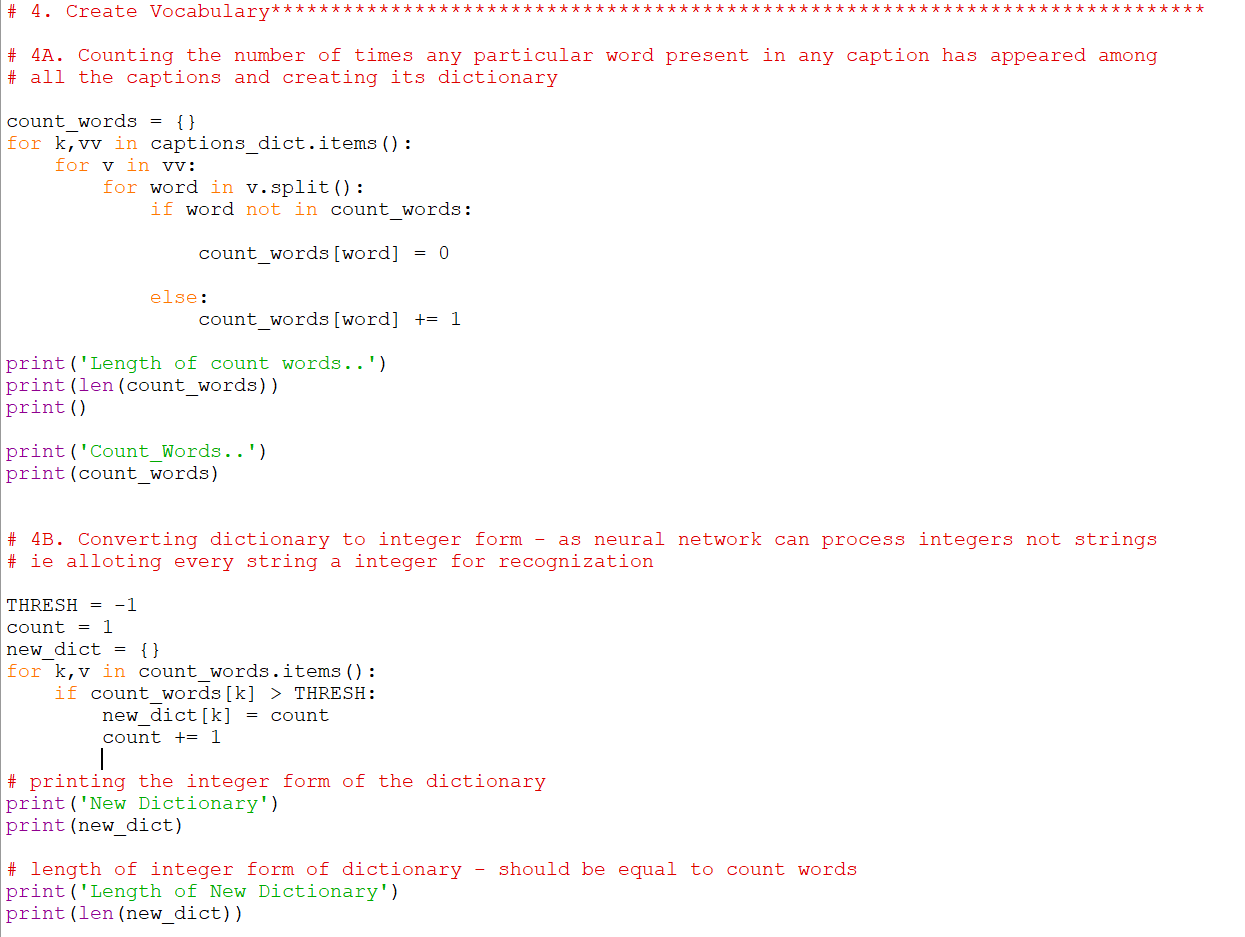


Figure 20: Code snippet showing Vocabulary Creation Process

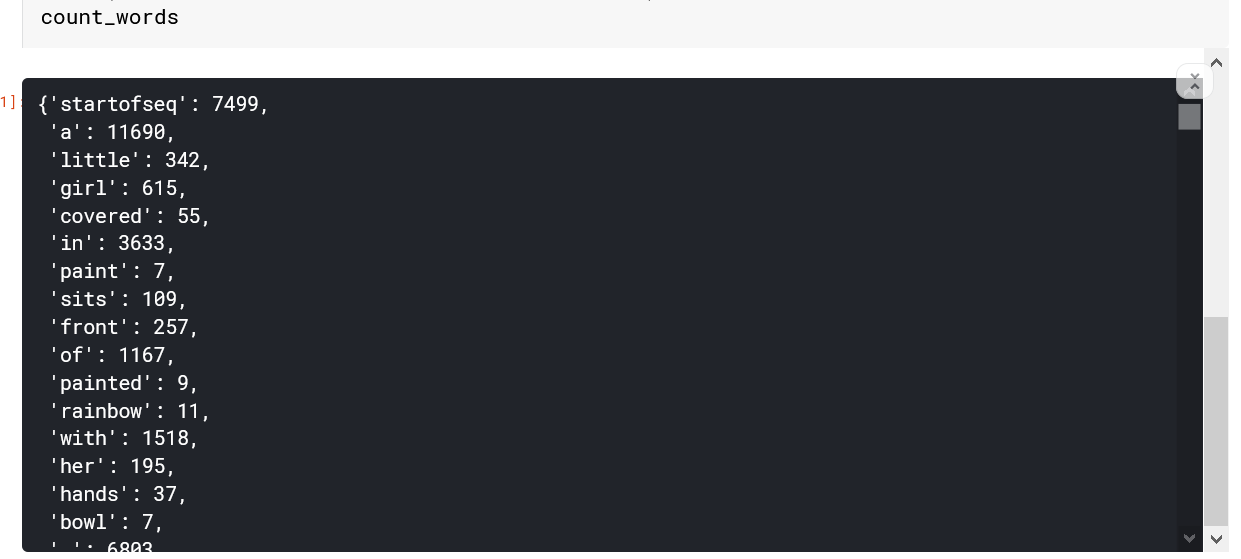


Figure 21: Sample output showing the count of different words

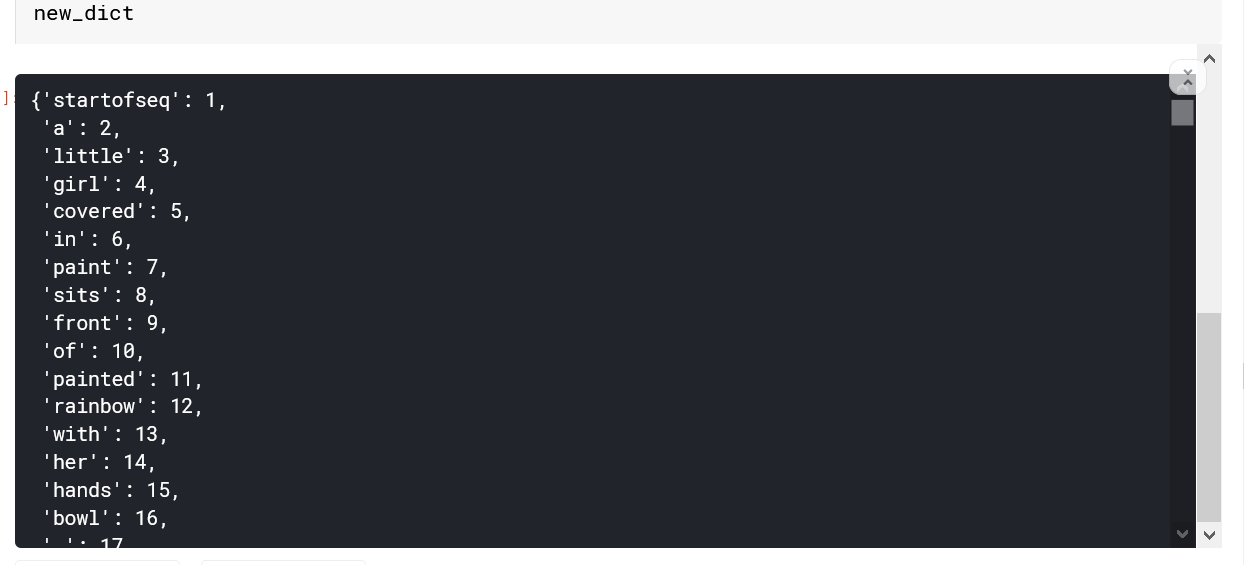


Figure 22: Sample output showing the word dictionary

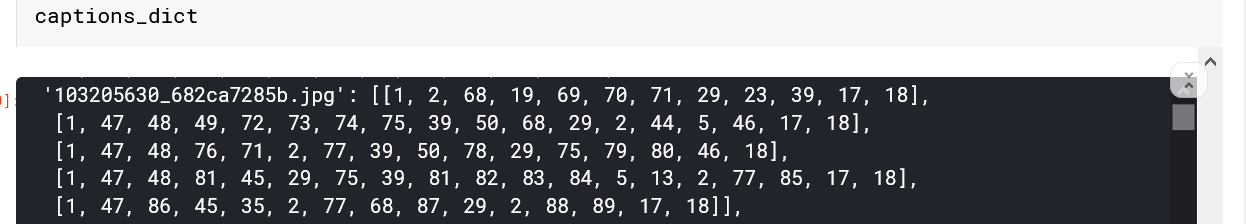


Figure 23: Sample output showing the integer form captions attached with there respective images

**5.6 PREPARING MODEL**

**5.6.1 Preparing Input for the model**

Now we need to obtain the maximum length of the caption that is present in our captions, this will be one of the inputs of the model which will help it decide the length of the caption to be predicted. Vocab size is the number of words we have in the dictionary, this would also be one of the inputs to the model. The process of obtaining the max length caption in our collection is shown in figure 24 with results in figure 25. We use a generator function to prepare the input for training the model. The outputs of the generator function are three NumPy arrays which contain the image and the first words and the next predicted word (the word having the highest probability). The generator method is displayed in figure 26 and a few the variables are created in figure 27 code snippet along with some sample values. Figure 28 shows the length and shape of input which are fed to the model for training.

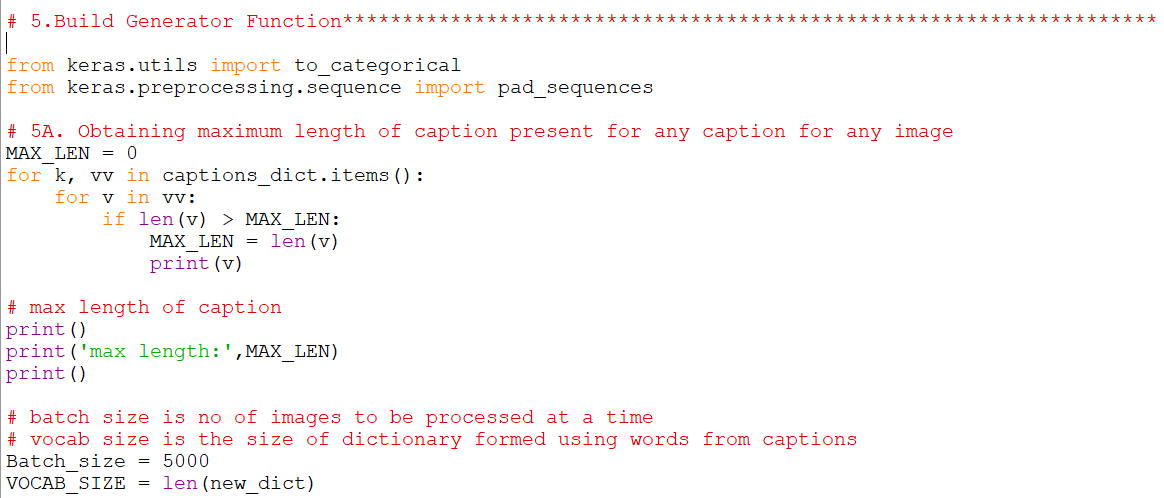


Figure 24: Code snippet showing the max length of the caption present

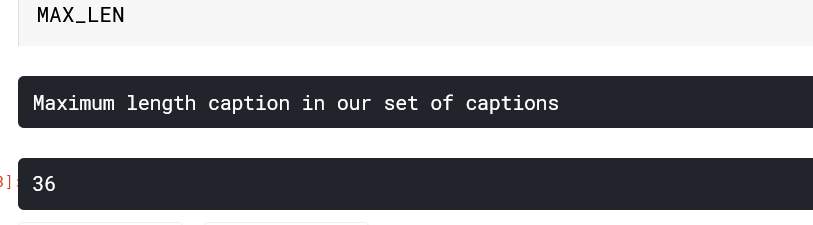


Figure 25: Output showing the length of the largest caption in our collection

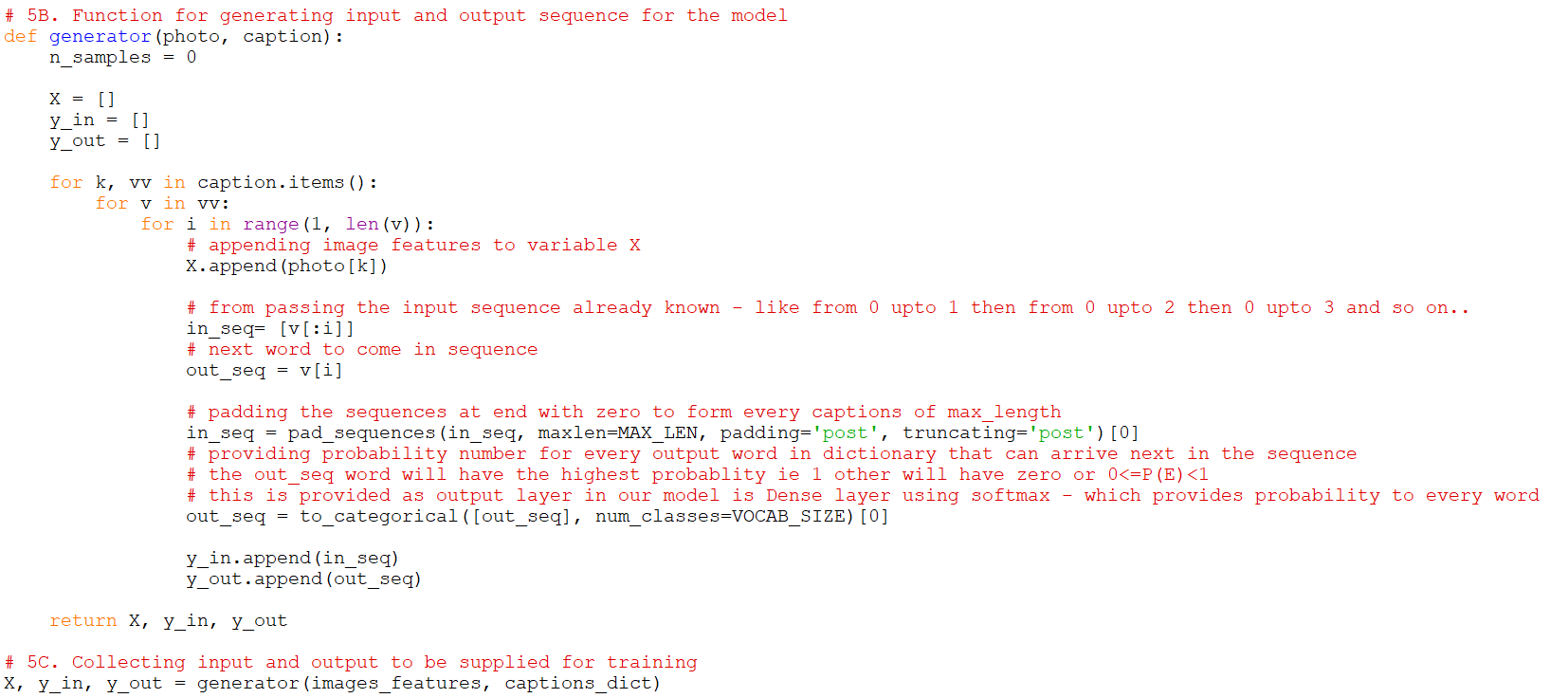


Figure 26: Code snippet showing the input form preparation which will be fed to the model

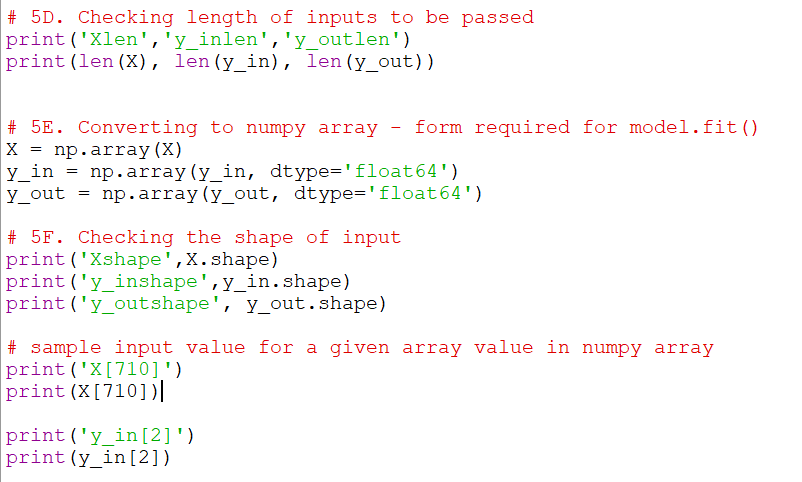


Figure 27: Code snippet showing the testing of shapes of inputs and conversion to NumPy array form

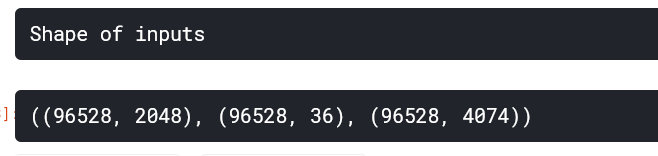
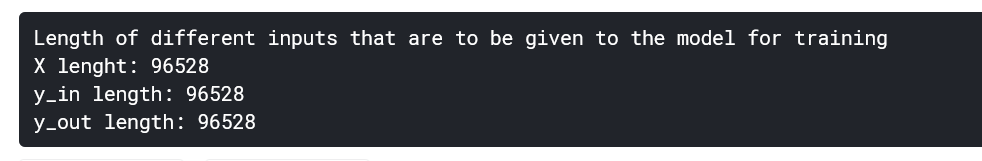


Figure 28: Sample output showing the shapes of input

**5.7 CONFIGURING THE MODEL**

**5.6.2.1 Configuring Image Model**

We use the ‘Sequential’ model present in tensorflow.keras.model library to sequentially add different layers which we want to use for processing the image. We have a ‘Dense layer’ and ‘RepeatVector layer’ and ‘relu’ activation function. The code snippet in Figure 29 highlights the process of configuration of image and language model. Figure 30 and Figure 31 shows the different layers present in the image and language model selected and different parameters they work on.



Figure 29: Code snippet showing the configuring process of the image and language model

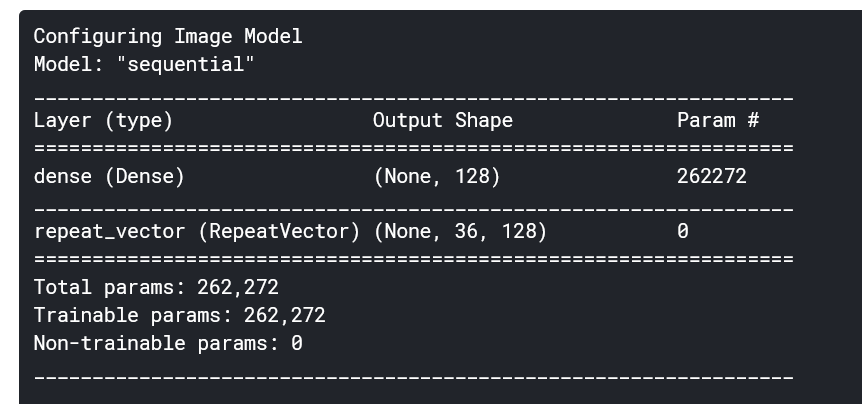


Figure 30: Output showing the image sequential model

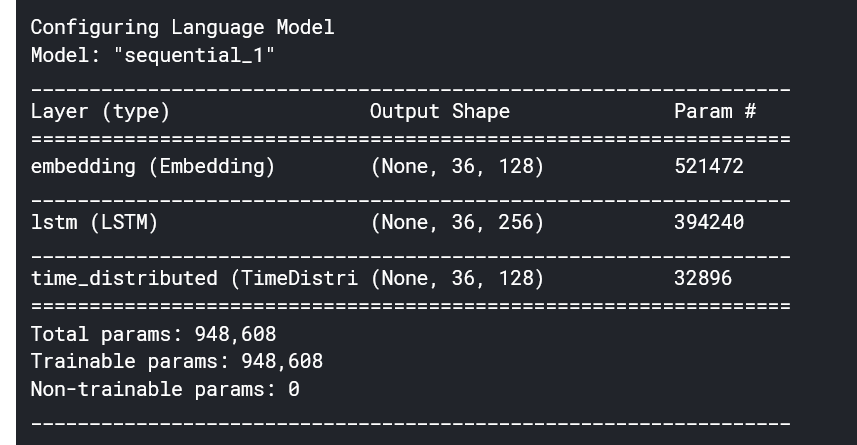


Figure 31: Output showing the language sequential model

Both of the models i.e. image and language model are combined using the concatenate layer and the resulting model layers are shown in figure 32. Figure 33 is the descriptive model generated through Keras plot model method. This shows different layers along with the shapes and when they are added to the model and the output generated by them.

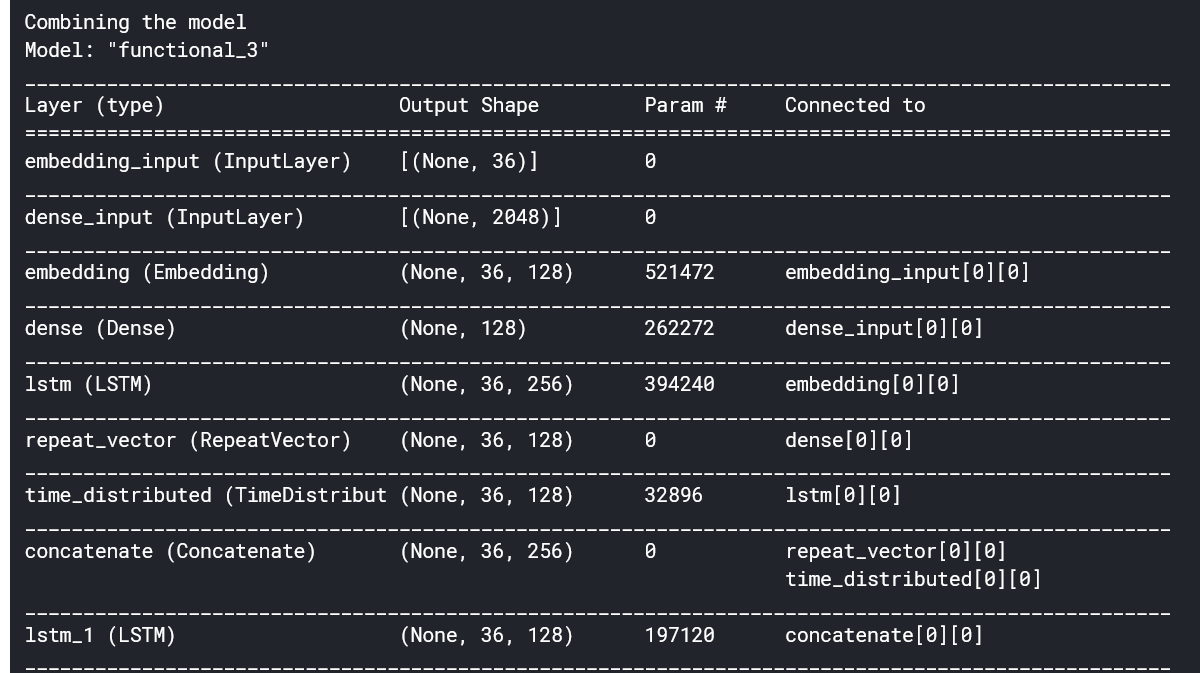


Figure 32: Output showing the combined model

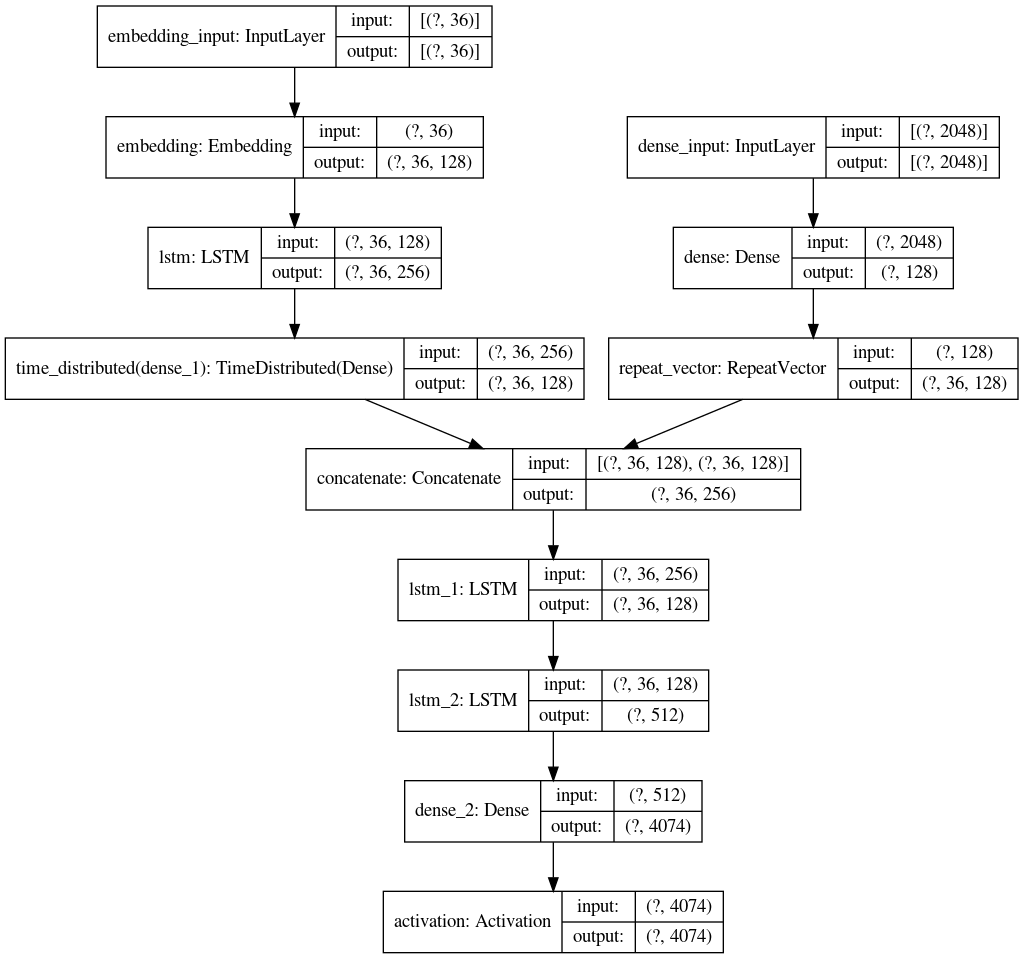


Figure 33: Detailed layout of the model along with shapes of inputs

**5.8 TRAINING AND SAVING THE MODEL**

We put the model on training with the inputs we have generated through the generator function in the above section using TensorFlow model.fit() function for 50 epochs as shown in figure 34. Once the training is complete we save the trained model for later use and operating in offline mode. An output snippet showing the training process is attached in figure 35. Then we prepare an inverse dictionary where every number of the word dictionary is now the key and every word of the dictionary is value. The inverse dictionary sample output is displayed in figure 36. This will be used in prediction when our model will give the number out of the dictionary having the highest probability we will check the corresponding value and return that word.

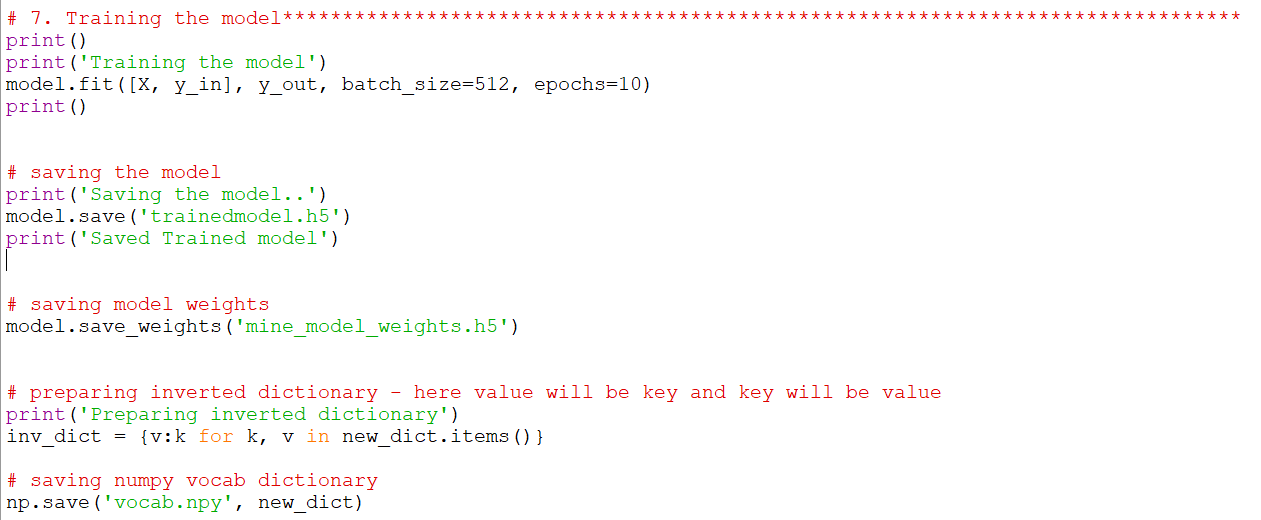


Figure 34: Code snippet showing the training process and creation of an inverse dictionary

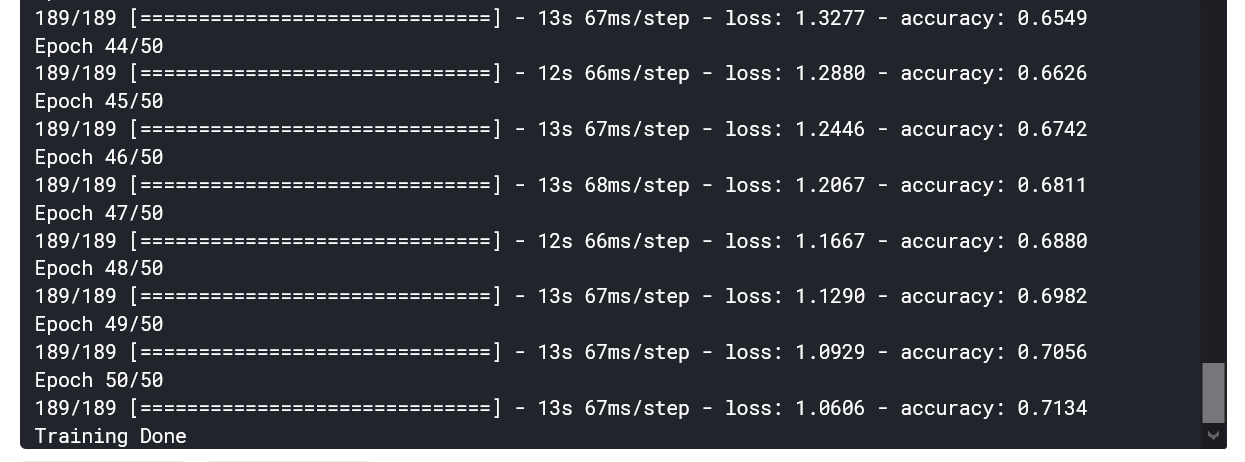


Figure 35: Sample output of the training process

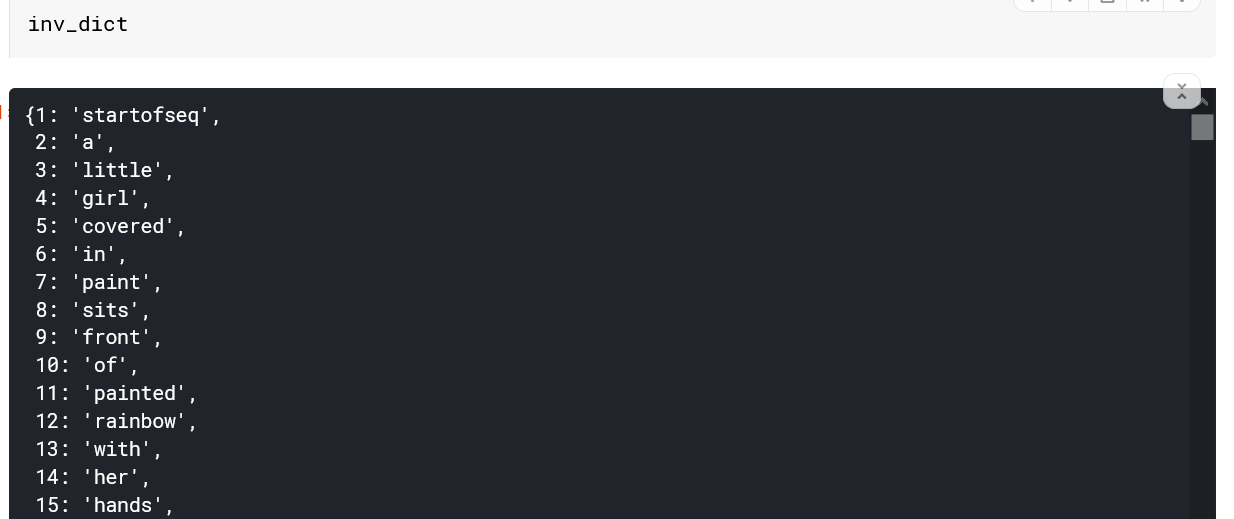


Figure 36: Sample output of the inverse dictionary

**5.9 PREDICTING THE CAPTIONS**

Now we have stored the ResNet50 model, trained model, dictionary of words and inverted dictionary. We can use these stored files to run or model on images to predict the captions. We have prepared a function which can fetch the images from our test folder and provide it one by one for predicting the captions. Figure 37 shows the process of preparation of test image provided by the user.

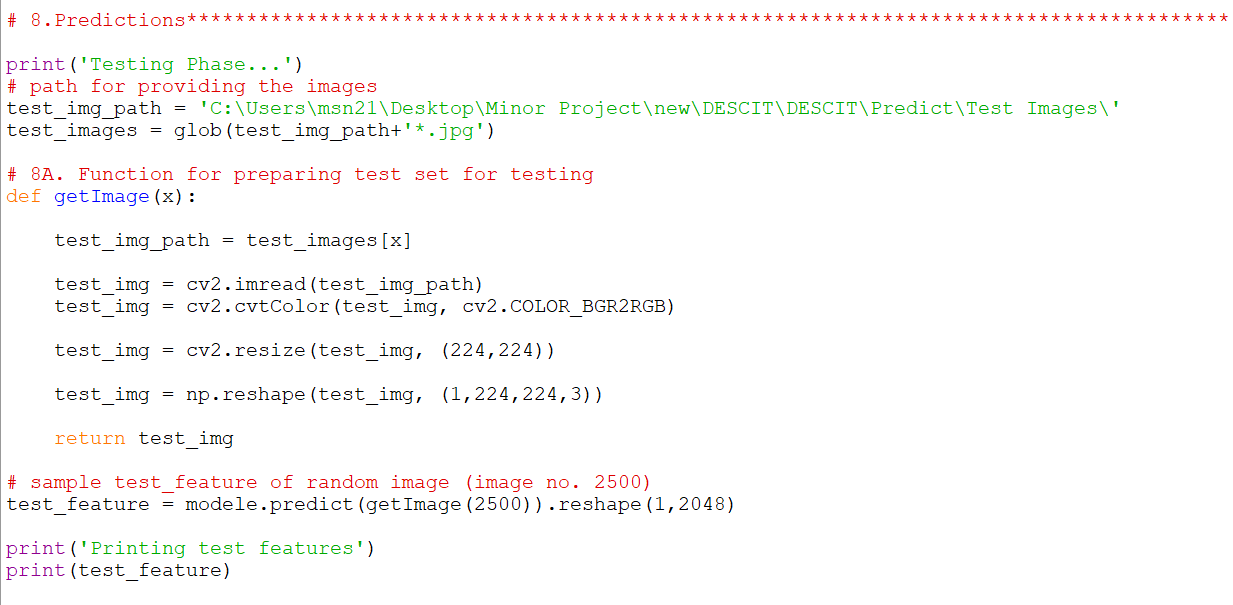


Figure 37: Code snippet showing the image preprocessing for the testing purpose

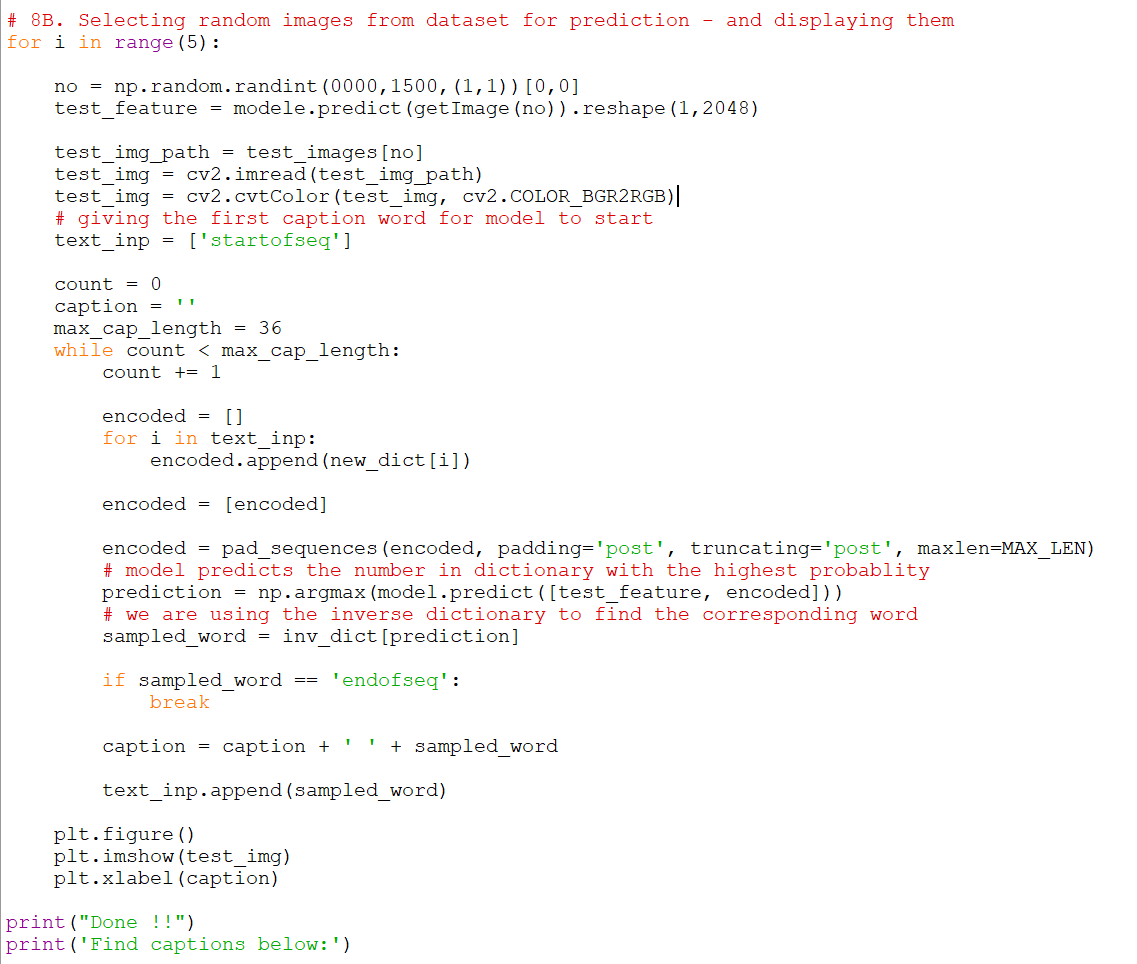
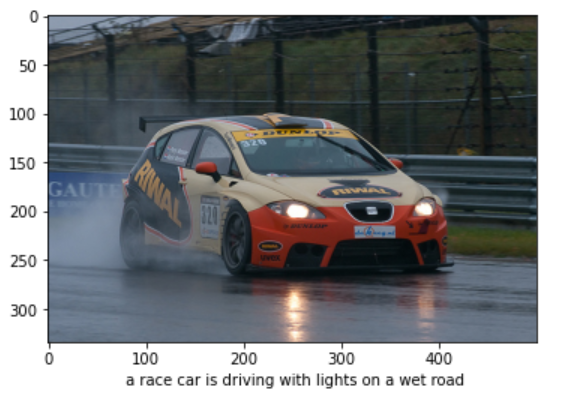


Figure 38: Code snippet showing the prediction process and results



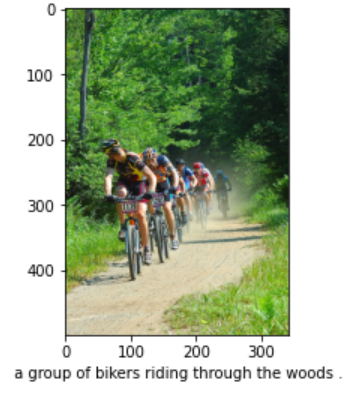


Figure 39: Some sample results generated by the model

Figure 38 code snippet shows the batch mode operation of the model which selects some random images from the test images folder and provides captions for them. Few of the results are attached in Figure 39.

**5.10 WEB PAGE DEVELOPMENT**

For the purpose of providing interfaces to users, we needed to develop a platform. Thus we chose web development for this. In this project, we created a web app using Python and it’s web framework flask, which reduces development time and allows us to build faster and smarter.

For the Frontend, we have created 5 web pages which include:

1. **Index.html** - Main welcome page of our web app, allows navigation to other pages and displays the categories of available quotes.
2. **Services.html -** It can be considered as the second option in Navigation. It is an important page because a user can actually upload the image here and gets the extracted caption.
3. **About.html** - About Page is for users knowledge about website, developers etc
4. **Contact.html -** In case of any query, enquiry, report, appreciation users can communicate to the development team
5. **Single.html -** Last but not least, this page is not directly accessible. On the index page when a user wants the quotes of a particular category and thus clicks its card, he/she is directed to this page, where they get a number of related quotes of that single category.

**5.11 API INTEGRATION**

In order to provide a vast number of quotes and categories, we connected our app to API. In this project, we used GoodQuotes API. API Integration is done using Javascript.

**5.12 CONNECTING MODEL WITH FRONTEND AND PYTHON CODE**

The backend and frontend both work together to serve a single goal. It’s pretty helpful to keep it in mind at all times. They are made, so a user can access them.

WORKING OF WEBSITE

The user points their browser to one of your website’s URLs and waits for the browser to render the page. The user sees a useful and usable page. The user interacts with the page.

Thus till now, our website was working as static, it was just connected to API. In order to connect it to our working python code, we used flask i.e, micro-framework of python. In this, we created several routes using route decorator and thus it helped us in hosting it to a local server - http://localhost:7000/

**5.13 PROVIDING USER WITH CAPTIONS ON THE PLATFORM**

On reaching Upload Page when a user inputs an image, The services() function is decorated with @app.route so that it is invoked when the browser sends a POST request. Using the request module of Flask it creates an object of the file and saves it in local storage. At the same time, it sends that image to python code by calling its function which in response returns the output string, which is then displayed as a caption.

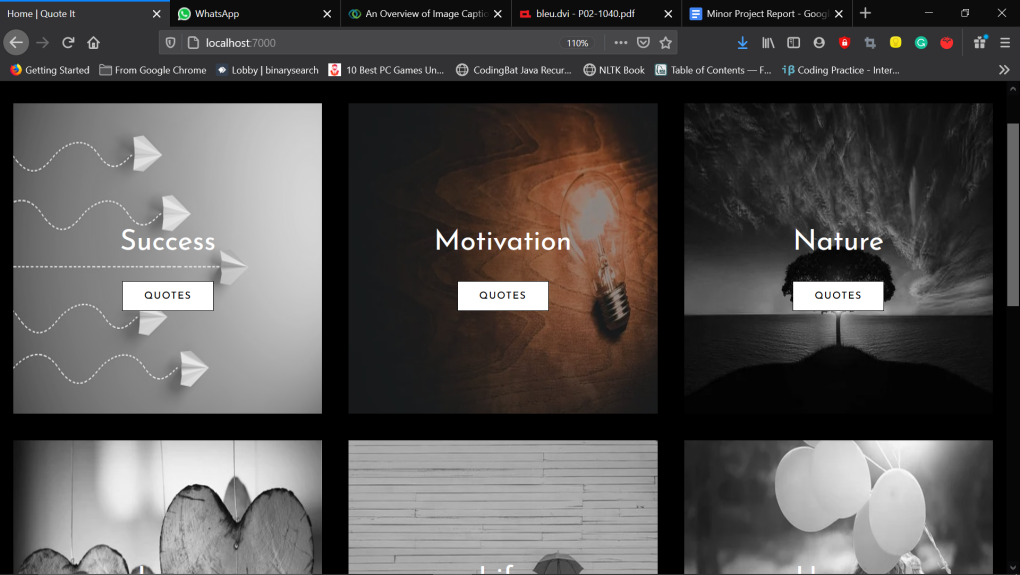


Figure 40: Layout of the Home page of the Web App

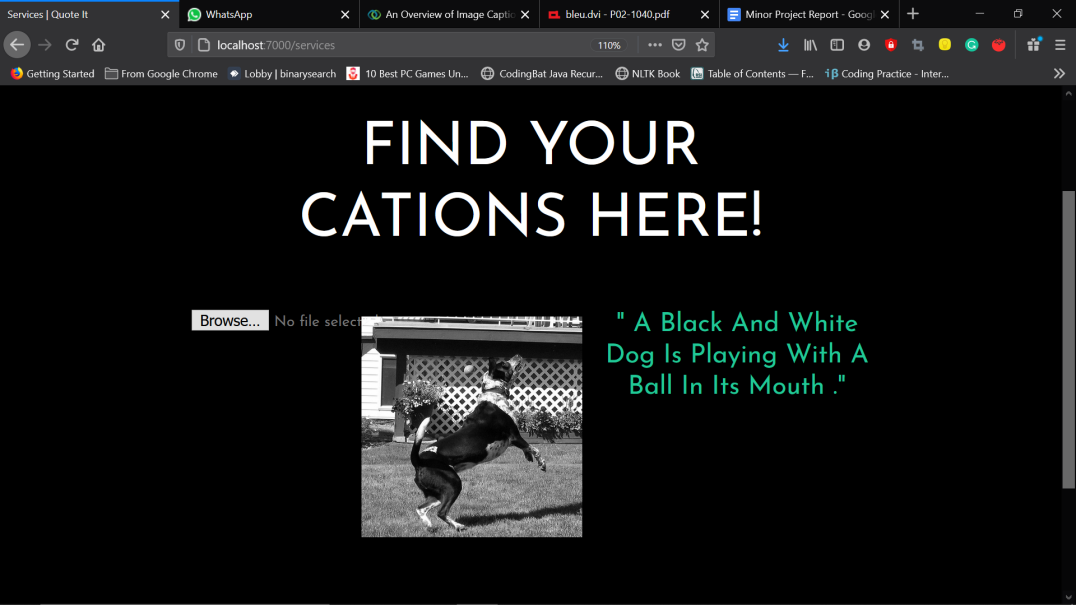


Figure 41: Sample results displayed at the Web Platform when using the model for prediction

Figure 40 shows the front page of the Web App as visible to the user. Here he/she can select the category of their choice to get the captions or they can select the Upload tab in the left top parallel bars. When the user will upload the image and press on the button. The model functions in the background and provide the caption which is shown on the Web App as shown in figure 41. Flask library is used to fetch the caption predicted by the model and display it on the Web Page.

**CHAPTER - 6**

**CONCLUSIONS AND FUTURE SCOPE**

**6.1 CONCLUSIONS**

Through this project, we were able to build a system that can be used by user from the timing, location of his/her choice and check for chances of stroke to him. We compared 10 different machine learning classifiers based on accuracy on our dataset, used the best one and performed k-fold validation and hyper-parameter optimization on it too. We were able to understand the different and working of different classifiers and reason behind there particular behavior on our dataset. We are ultimately able to achieve our objective of learning about various machine learning classifier, comparing them and using the best one to build a system which can be used by anyone from anywhere to test the chances of stroke to him/her.

**6.2 FUTURE SCOPE**

There is a huge feature approach for this model some of which are enlisted below:

* The model is currently trained and tested for accuracy on a small dataset of approx. 5k records. We believe training it on any other dataset, which is bigger in size would enhance or prediction capability even more.
* Right now, the user is only asked for details in the form. We plan to enhance the web platform to display more information and statistics about the stroke.
* An android app for same purpose can be really useful and handy, we will this could be a potential future scope.
* Right now, the model uses 10 parameters for predicting the chances of stroke, adding more crucial parameters which play role in stroke and its prediction could be a really nice enhancement to the model.
* Adding more information about what stroke actually is, it causes, symptoms and other related and relevant information on the web page is also a nice and useful scope.
* Adding functionality to fetch all health record data and then use patient history for predicting chances of stroke could be a super huge enhancement.

**CHAPTER - 7**

**REFERENCES**

**Datasets:**

[1] “Dataset: Cardiovascular health study (CHS).” https://biolincc.nhlbi.nih.gov/studies/chs/. Accessed: 2016-05-8.

[2] "Stroke Prediction Dataset - (fedesoriano)." https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

**Model Classifiers: -**

[3] Scikit-learn Supervised Models - https://scikit-learn.org/stable/supervised\_learning.html

[4] GaussianNB - UPDATING FORMULAE AND A PAIR-WISE ALGORITHM FOR COMPUTING SAMPLE VARIANCES - Tony F.Chan, Gene H. Golub, Randall J. LeVeque:-Dept. Of CS, Standord NOV,1979 {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html> }

[5] MultinomialNB - {<https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html> } {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html> }

[6] BernoulliNB - {<https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html> } {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html> }

[7] CategoricalNB - {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html> }

[8] LinearRegression - {<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html> }

[9] KNeighborsClassifier - {<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm> } {<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> }

[10] Logistic Regression - Minimizing Finite Sums with the Stochastic AverageGradient - Mark Schmidt, Nicolas Le Roux, Francis Bach - 10 May 2016 - {<https://hal.inria.fr/hal-00860051v2> }

[11] Logistic Regression - {<https://en.wikipedia.org/wiki/Logistic_regression> } {<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> }

[12] Logistic Regression - SAGA:A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives - Aaron Defazio, Francis Bach, Simon Lacoste-Julien - 16 Dec 2014 - {arXiv:1407.0202v3 [cs.LG], <https://arxiv.org/pdf/1407.0202.pdf> }

[13] Logistic Regression - Dual Coordinate Descent Methods for Logistic Regression and Maximum Entropy Models - Hsiang-Fu Yu, Fang-Lan Huang, Chih-JenLin - {<https://www.csie.ntu.edu.tw/~cjlin/papers/maxent_dual.pdf> }

[14] Decision Tree Classifier - Classification and Regression Trees Software and New Developments - Francesco Mola - 1998 -{<https://sci-hub.se/10.1007/978-3-642-72253-0_42> }

[15] Decision Tree Classifier - {<https://en.wikipedia.org/wiki/Decision_tree_learning> } {<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> }

[16] Decision Tree Classifier - The Elements of statistical Learning Ebook- {<https://web.stanford.edu/~hastie/Papers/ESLII.pdf> }

[17] Decision Tree Classifier - Understanding Decision Tree Classifier Blog - Tarun Gupta - Oct 2020 - {<https://towardsdatascience.com/understanding-decision-tree-classifier-7366224e033b> }

[18] Support Vector Classification(SVC) - {<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> }

[19] SVC - LIBSVM: A Library for Support Vector Machines - Chih-Chung Chang and Chih-Jen Lin - Jan 2021 - {<https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf> }

[20] SVM - Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods - by John C. Platt - 1999 - {<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.1639> }

[21] SVM - Support Vector Machines with Scikit-learn Blog- Avinash Navlani - Dec 2019 - {<https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>

[22] KMeans-{<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html> }

[23] KMeans Clustering - {<https://en.wikipedia.org/wiki/K-means_clustering> }

[24] KMeans - K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks - Imad Dabbura - Sept 2018 - {<https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a> }

[25] Random Forest Classifier - {<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> }

[26] Random Forest Classifier - Random Forest - LEO BREIMAN - 2001 - {<https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf> }

[27] Random Forest Classifier - {<https://en.wikipedia.org/wiki/Random_forest> }

[28] Random Forest Classifier - Understanding Random Forest - How the Algorithm Works and Why It Is So Effective Blog - Tony Yiu- Jun 2019 -{<https://towardsdatascience.com/understanding-random-forest-58381e0602d2> }

[29] Random Forest Classifier - Understanding Random Forests Classifiers in Python Blog - Avinash Navlani - May 2018 - {<https://www.datacamp.com/community/tutorials/random-forests-classifier-python> }

[30] Gradient Boosting Classifier - {<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html> }

[31] StandardScaler - {<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> }

[32] XGBClassifier - How to Develop Your First XGBoost Model in Python Blog - Jason Brownlee - Aug 2016{<https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/> }

[33] k-Fold Cross Validation - A Gentle Introduction to k-fold Cross-Validation Blog - Jason Brownlee - May 2018 - {<https://machinelearningmastery.com/k-fold-cross-validation/> }

[34] k-Fold Cross Validation - Cross-validation: evaluating estimator performance - {<https://scikit-learn.org/stable/modules/cross_validation.html> }

[35] k-Fold Cross Validation - Permutation Tests for Studying Classifier Performance - Markus Ojala, Gemma C. Garriga - Jun 2010 -{<https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf> }

[36] k-Fold Cross Validation - {<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html> }

[37] k-Fold Cross Validation - Why and how to Cross Validate a Model? Importance and types of Cross-validation techniques Blog- Sanjay M- Nov 2018 - {<https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f> }

[38] Hyper-Parameter Optimization - {<https://en.wikipedia.org/wiki/Hyperparameter_optimization> }

[39] Hyper-Parameter Optimization - An introduction on how to fine-tune Machine and Deep Learning models using techniques such as: Random Search, Automated Hyperparameter Tuning and Artificial Neural Networks Tuning - Pier Paolo Ippolito - Sept,2019 - {<https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d> }

[40] Hyper-Parameter Optimization - Random Search for Hyper-Parameter Optimization - James Bergstra, Yoshua Bengio - 2012 - {<https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a> }

[41] Hyper-Parameter Optimization - Algorithms for Hyper-Parameter Optimization - James Bergstra,Yoshua Bengio,Remi Bardenet,Balazs Kegl - Nov,2011 - {<https://hal.inria.fr/hal-00642998/file/draft1.pdf> }

[42] Hyper-Parameter Optimization - A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring - Yufei Xiaa,Chuanzhe Liua,Yu Ying Lib,Nana Liua - Feb,2017 - {<https://sci-hub.se/10.1016/j.eswa.2017.02.017> }

[43] GridSearchCV - {<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html> }

[44] RandomizedSearchCV -{<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html> }

[45] GridSearchCV - Machine Learning: GridSearchCV & RandomizedSearchCV - Hyperparameter tuning with GridSearchCV and RandomizedSearchCV Blog - Papa Moryba Kouate - Sept,2020 - {<https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10>}

[46] Interpretable Classifiers Using Rules And Bayesian Analysis: Building A Better Stroke Prediction Model - Benjamin Letham, Cynthia Rudin, Tyler H. Mccormick And David Madigan - 2015

[47] Comparison Of Different Machine Learning Approaches To Model Stroke Subtype Classification And Risk Prediction - Luis Garcia-Terriza, Jose L. Risco-Martin, Jose L. Ayala, Gemma Reig Rosello, Juan M. Camarasaltas - April 2019

[48] Prediction Models Aided Postoperative Decision Making Based on Neural Network and Support Vector Machines - Ibrahim Abuaqel, Abdullah Alsaif, Mohammed Almulhim, Mohammed Aljallal, Musab Alghuraibi, Mujahid Alzahrani, Abdulrahman AlHarby, Sunday O. Olatunji. - 2017

[49] Prediction of Stroke Using Deep Learning Model - Pattanapong Chantamit-o-pas and Madhu Goyal - 2017

[50] Classification of stroke disease using machine learning algorithms - Priya Govindarajan, Ravichandran Kattur Soundarapandian, Amir H. Gandomi, Rizwan Patan, Premaladha Jayaraman, Ramachandran Manikandan - Nov 2018

[51] An Integrated Machine Learning Approach to Stroke Prediction - Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu, Honglak Lee - July 2017

**APPENDICES**

**APPENDIX 1**

**1.1 DETAILED TEST STRATEGY AND TEST CASES**

Our project is based on medical data collected from different hospitals. The testing and validation of model is performed using the scikit-learn functions.

Figure: Image of histogram showing the comparison of train, test and validation score.

About the testing in real life scenario, it is to be done on field.

**APPENDIX 2**

**2.1 USER GUIDE**

The user needs to follow the below-mentioned procedure for using the model to predict the chances of stroke to him/her.

Step 1: Visit to the webpage using “https://strokemsp.herokuapp.com”

Step 2: Fill in all the details asked.

Step 3: Click on submit button and wait for a moment for the model to predict.

Step 4: Within a few moments the model will predict the results which is being displayed to you on a new web page.

Step 5: If the model predicts you to have stroke risk, we recommend you to pay a visit to a doctor.

Step 6: Done. If the model predicts you having no risk of stroke; and still you feel any of the symptoms we recommend you to talk to a doctor.

\*\*EOF\*\*