## Project Title :- Image Caption Generator

## ABSTRACT

Image caption Generator is a popular research area of Artificial Intelligence that deals with image understanding and a language description for that image.Generating well-formed sentences requires both syntactic and semantic understanding of the language. In this article, we will use different techniques of computer vision and NLP to recognize the context of an image and describe them in a natural language.

We will build a working model of the image caption generator by using CNN (Convolutional Neural Networks) and LSTM (Long short term memory) units.

***Keywords :***

Image, Caption, Xception, Neural Networks, CNN Algorithm, LSTM Algorithm

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Name** | **Page** |
|  | **Certificate** | **2** |
|  | **Acknowledgement** | **3** |
|  | **Abstract** | **4** |
|  | **List of Figures** | **8** |
|  | **List of Tables** | **9** |
|  | **Abbreviation** | **10** |
|  |  |  |
| **1.** | **Introduction** | **11** |
|  | **1.1 Introduction** | **11** |
|  | **1.2 Motivation** | **11** |
|  | **1.3 Purpose** | **11** |
|  | **1.4 Problem Statement** | **11** |
|  | **1.5 Objectives** | **11** |
|  |  |  |
| **2.** | **Literature Survey** | **12-13** |
|  | **2.1 Existing System** | **12** |
|  | **2.1.1 Referred Journal/Conference Papers** | **12** |
|  | **2.1.2 Elaborate on Existing System Applications/Example** | **12** |
|  | **2.1.3 Limitations or challenges in Existing System** | **12** |
|  | **2.2 Proposed System with block diagram** | **12** |
|  | **2.3 Feasibility Study** | **12-13** |
|  |  |  |
| **3.** | **Project Scope and Requirement Analysis** | **14** |
|  | **3.1 Project Scope** | **14** |
|  | **3.2 Requirement Gathering & Analysis** | **14** |

|  |  |  |
| --- | --- | --- |
| **4.** | **Project Design and Modeling Details** | **15-19** |
|  | **4.1 Software Requirement Specification(SRS)** | **15** |
|  | **4.2 System Modules** | **15** |
|  | **4.3 System Modeling and Design** | **15-18** |
|  | **4.4 Database Design** | **18** |
|  | **4.5 System Architecture** | **19** |
|  |  |  |
| **5.** | **Implementation and Coding** | **20-21** |
|  | **5.1 Algorithms** | **20-21** |
|  | **5.2 Flowcharts** |  |
|  | **5.3 Software Requirements with relevant justifications** | **21** |
|  | **5.4 Hardware Requirements with relevant justifications** | **21** |
|  |  |  |
| **6.** | **Testing** | **22-24** |
|  | **6.1 Fundamentals of Testing** | **22** |
|  | **6.2 Test plan of the project** | **23** |
|  | **6.3 Test Cases and Test Results** | **23-24** |
|  |  |  |
| **7.** | **Project Plan and Schedule** | **25-26** |
|  | **7.1 Project Planning and Project Resources** | **25** |
|  | **7.2 Project Scheduling** | **26** |
|  | **7.3 Effort Estimation** | **26** |
|  |  |  |
| **8.** | **Risk Management and Analysis** | **27** |
|  | **8.1 Project Risk Identification** | **27** |
|  | **8.2 Project Risk Analysis** | **27** |

|  |  |  |
| --- | --- | --- |
| **9.** | **Configuration Management** | **28-** |
|  | **9.1 Installation/Uninstallation** |  |
|  | **9.2 User Manual** |  |
|  | **9.2.1 Input Screenshots** |  |
|  | **9.2.2 Output Screenshots** |  |
|  |  |  |
| **10.** | **Conclusion and Future Scope** |  |
|  | **10.1 Conclusion** |  |
|  | **10.2 Future Scope** |  |
|  |  |  |
|  | **References** |  |
|  | **Journal/ Conference Papers** |  |
|  | **Book References** |  |
|  | **Web References** |  |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name** | **Page** |
| Fig.2.2 | Block Diagram | 12 |
| Fig. 4.3.1 | System Architecture | 15 |
| Fig. 4.3.2 | DFD Level 0 | 15 |
| Fig. 4.3.3 | DFD Level 1 | 16 |
| Fig. 4.3.4 | Sequence Diagram | 17 |
| Fig. 4.3.5 | Activity Diagram | 17 |
| Fig. 5.1 | CNN Architecture | 20 |
| Fig. 7.2 | Gantt Chart | 26 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name** | **Page** |
| Table 7.1 | Project Planning | 25 |

## ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Name** |
| AI | Artificial Intelligence |
| CNN | Convolutional Neural Network |
| LSTM | Long Short Term Memory |
| NLP | Natural Language Processing |
| DFD | Data Flow Diagram |

**Chapter 1**

**Introduction**

* 1. **Introduction:**

Making a computer system detect objects and describe them using natural language processing (NLP) in an age-old problem of Artificial Intelligence. This was considered an impossible task by computer vision researchers till now. With the growing advancements in Deep learning techniques, availability of vast datasets, and computational power, models are often built which will generate captions for an image. Image caption generation is a task that involves image processing and natural language processing concepts to recognize the context of an image and describe them in a natural language like English or any other language.

* 1. **Motivation:**
* Generating captions for images is a vital task relevant to the area of both Computer Vision and Natural Language Processing.
* Mimicking the human ability of providing descriptions for images by a machine is itself a remarkable step along the line of Artificial Intelligence.
* The main challenge of this task is to capture how objects relate to each other in the image and to express them in a natural language (like English).
  1. **Purpose:**

The Main purpose of this system is to automatically generate the captions of an image and it includes the labeling of an image with the help of datasets provided during model training.

* 1. **Problem Statement:**

Image caption generator involves computer vision and natural language processing concepts to the develop system that automatically describe the image and display its context in a natural language.

* 1. **Objectives:**
* Image Caption Generator deals with image understanding and a language description for that image.
* To learn and understand image and text generation part, build a working model of Image caption generator by implementing CNN with LSTM.
* Generating well-formed sentences requires both syntactic and semantic understanding of the language.

**Chapter 2**

**Literature Survey**

**2.1 Existing System:**

* There are some existing system in Image caption Generator using RNN. It is having extended collection of data and higher capabilities.
* RNNs can only store some amount of the data for short period time.
* RNN :-RNN are slow and complex. Training an RNN is very difficult task.

**2.2 Proposed System with block diagram:**

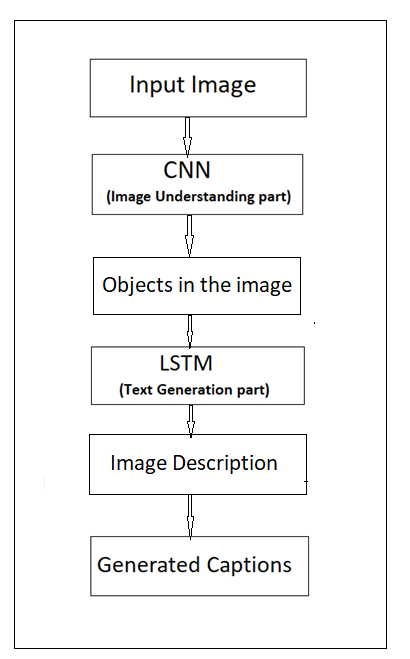


Fig.2.2 Block Diagram

**2.3 Feasibility Study:**

**Technical Feasibility**

The main technologies and tools are associated with project are

1. Language : Python 3

2. Libraries : Keras, Pillow, Numpy, tqdm, TensorFlow

3. IDE : Jupyter Notebook

4. Diagramming Tools : Microsoft Visio 16.0 (Licensed)

Each technology is freely available and required technical skills are manageable. Initially website will be hosted on free web hosting space. For later implementations, it will be hosted on a paid hosting space with sufficient bandwidth. Moderate Internet connection is required for this application, since it does not incorporate any multimedia aspect. From above, it is clear that project “Image Caption Generator” is technically feasible and no risks are associated with technology.

**Legal Feasibility**

1. The project “Image Caption Generator” is a complete Web Application

Project “ Image Caption Generator is absolutely legal and doable

1. It meets all legal and ethical requirements as per the Information Technology Act,2000 Government of India
2. Project “AI based virtual quiz” uses freely available/open source software development tools
3. No threats to customer’s/institute’s/organization’s confidential data

From above, it is clear that project “Image Caption Generator” is legally feasible with no potential infringement

**Chapter 3**

**Project Scope and Requirement Analysis**

**3.1 Project Scope:**

* Inscope :

1. Using this model of caption generator and image description we use jpg images used by students.
2. The captions generated for the images are quite accurate.

* Outscope:

1. The model will be fail if the other processing is found like video and livedemo.
2. The model depends on the data, so it cannot predict the words that are out of its vocabulary.

**Assumptions/Dependencies:-**

1. This portal require moderate Internet connection.
2. Browser to visit the website.(like chrome, firefox , etc)

**3.2 Requirement Gathering and Analysis:**

Research Paper :-

<https://www.ijitee.org/wp-content/uploads/papers/v10i3/C83830110321>.

**Functional Requirements**

1.The user should able to provide the image through .jpg file.

2.The system should calculate the accurate result.

3.The system should able to capture the correct objects from the image.

**Non-functional Requirements**

1.Performance:- The system must be able to handle as many users as possible at a time.

2.Maintainability:- The system should be easy to maintain.

3.Usability:- The system must be easy to use for any user.

**Chapter 4**

**Project Design and Modeling Details**

**4.1 Software Requirement Specification (SRS):**

**Introduction:**

The SRS is a document, which describes completely the external behaviour of the software. This section of SRS describes the general factors that affect the product and its requirements. The system will be explained in its context to show how system interacts with other systems and introduce the basic functionalities of it.

**Functional requirements:**

1. Operating System – Windows 8, 64 bit

2. Programming language – Python 3

3. Libraries – Keras, Pillow, Numpy, tqdm, TensorFlow

4. IDE – Jupyter Notebook

**4.2 System Modules:**

**1.CNN**:- CNNs are powerful image processing, artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/image-recognition)) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing ([NLP](https://www.techtarget.com/searchbusinessanalytics/definition/natural-language-processing-NLP)). CNN is used for extracting features from the image. We will use the pre-trained model Xception.

**2,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. LSTM will use the information from CNN to help generate a description of the image.

**4.3 System Modeling and Design:**

**System Architecture:-**

****

Fig. 4.3.1 UML Diagram

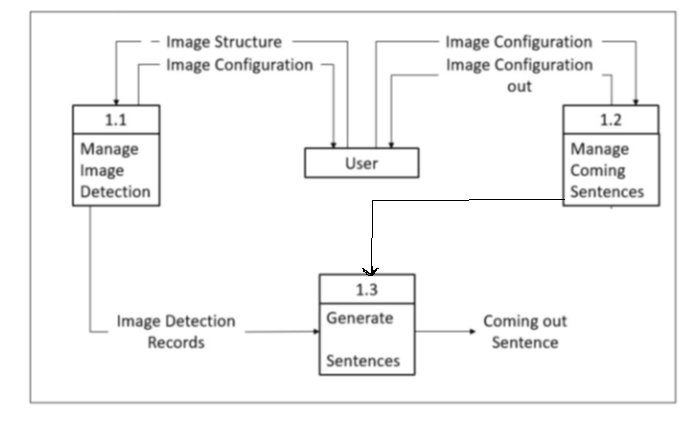
**DFD Diagram:-**

1. Level 0



Fig. 4.3.2 DFD Level 0 Diagram

1. Level 1

****

**Sequence Diagram:-**

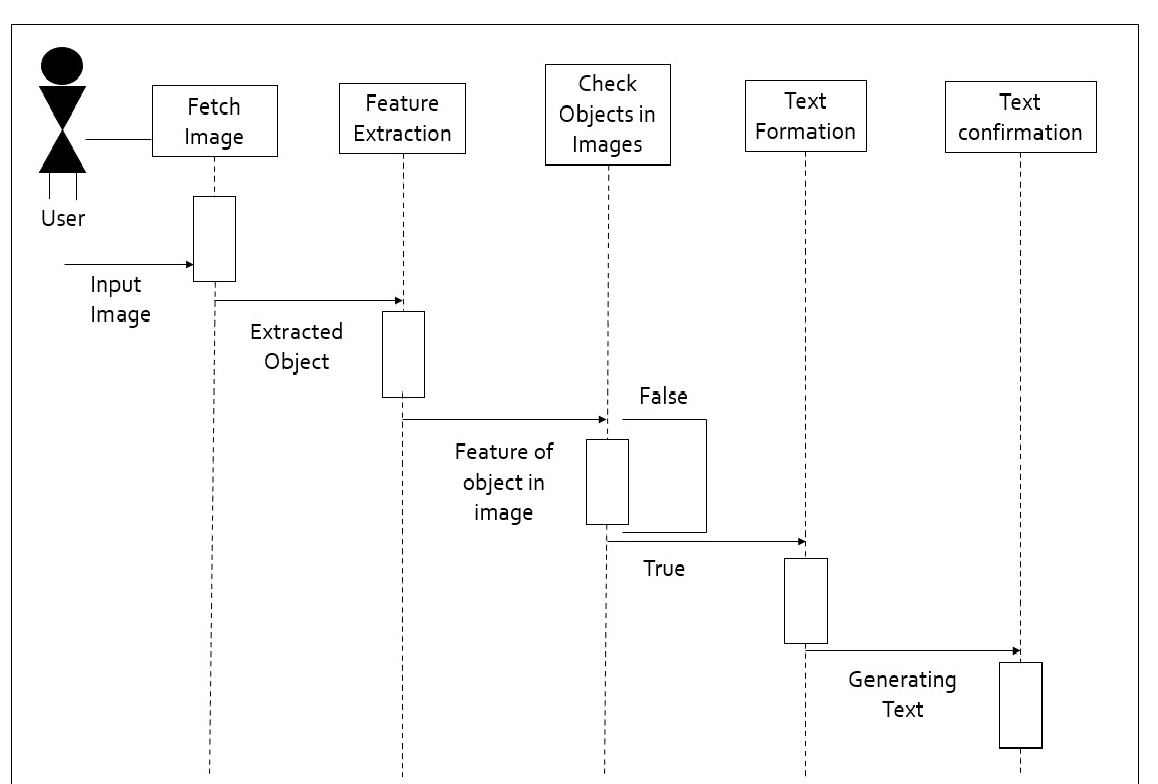


Fig. 4.3.5 Sequence Diagram

**Activity Diagram:-**

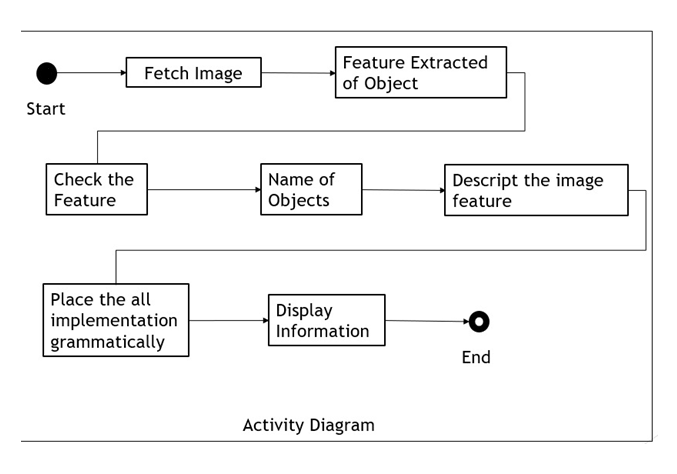


Fig. 4.3.6 Activity Diagram

**Chapter 5**

**Implementation and Coding**

**5.1 Algorithms:**

**CNN Algorithm**

It is assumed that the reader knows the concept of Neural networks.  
When it comes to Machine Learning, [Artificial Neural Networks](https://www.geeksforgeeks.org/implementing-ann-training-process-in-python/) perform really well. Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network there are three types of layers:

1. **Input Layers:** It’s the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).
2. **Hidden Layer:** The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.
3. **Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

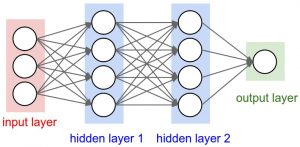
****

Fig. 5.1 CNN Architecture

**Convolution Neural Network**

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (as images generally have red, green, and blue channels).

**Layers used to build Convolutional Networks**

A convolutional networks is a sequence of layers, and every layer transforms one volume to another through a differentiable function.   
**Types of layers:**   
Let’s take an example by running a convolutional networks on of image of dimension 32 x 32 x 3. 

1. **Input Layer:** This layer holds the raw input of the image with width 32, height 32, and depth 3.
2. **Convolution Layer:** This layer computes the output volume by computing the dot product between all filters and image patches. Suppose we use a total of 12 filters for this layer we’ll get output volume of dimension 32 x 32 x 12
3. **Activation Function Layer**: This layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12.
4. **Pool Layer**: This layer is periodically inserted in the convolutional networks and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.
5. **Fully-Connected Layer:** This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

**5.3 Software Requirements with relevant justification:**

1. Operating System – Windows 7 and above, 64bit
2. Programming language – Python 3.9.6
3. Libraries – Keras, Pillow, Numpy, tqdm, TensorFlow
4. IDE – Jupyter Notebook, VS code

**5.4 Hardware Requirements with relevant justification:**

1. Processor – Intel I5

2. RAM – 4 GB Minimum

3. HDD – 100 GB above

**Chapter 6**

**Testing**

**6.1 Fundamentals of Testing:**

**Software Testing:-**

Testing is the process of evaluating a system or its components with intent to find whether it satisfies specified requirements or not. In simple words, the software testing is executing a system in order to identify any gaps, or missing requirements in contrary to the actual requirements. In most the cases, software tester, software developer, project manager, and end users are involved in testing a system within their respective capacities.

**Types of Software Testing:-**

1. Unit Testing

Unit testing focuses on smallest unt of software design. It is often done by programmers by using simple input and observing its corresponding outputs.

1. Integration Testing

Integration testing is defined as the testing of combined parts of an application to determine if the work correctly. It is done by two ways, first is bottom-up testing which starts testing by single unit to higher level components and top-down testing is test first higher-level components to lower level.

1. Black-Box Testing

The technique of testing without having any knowledge of interior working of the application. While performing a black-box testing, a tester will interact with the systems user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

1. White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

1. Grey-Box Testing

This technique to test the application with having knowledge of the internal workings of an application. The grey box testers don‟t rely on the source code; instead they rely on interface definition and functional specification.

1. System Testing

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified quality standards.

1. Manual Testing

Manual testing includes testing software manually i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug.

**6.2 Test Plan of the Project:**

Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

Individual PASS/FAIL criteria are written for each test case. All the tests need to get a PASS result for proper working of an application.

**6.3 Test Cases and Test Results:**

#### **Test Case 1: import necessary packages and performing data cleaning.**

**Objective:** We have to install all necessary libraries and packages, Creating functions for data cleaning.

**Steps:** The following steps have to be followed to carry out the test.

1. load\_doc
2. all\_img\_captions
3. cleaning\_text
4. text\_vocabulary
5. save\_descriptions

**Expected Results:** Creating files to store information

**Result:** Successful.

**Test Case 2: Extracting Features and loading dataset for training**

**Objective:** Extracting features from Xception and loading the training dataset using functions.

**Steps:** The following steps have to be followed to carry out the test.

1**.**load\_photos

2**.**load\_clean\_descriptions

3**.**load\_features

**Expected Results:** Image features will be extracted from Xception which is a trained CNN model.

**Result:** Successful.

#### **Test Case 3: Tokenizing the vocabulary.**

**Objective:** Keras library provides tokenizer function to create tokens.

**Steps:** The following steps have to be followed to carry out the test.

1. Represent English words in numbers.
2. Mapping each word of the vocabulary with unique index value.

**Expected Results:** Create tokens from vocabulary and save as “tokenizer.p” pickle file.

**Result:** Successful

#### **Test Case 4: Training and testing the model**

**Objective:** Train the model using 6000 training images for generating the Captions

**Steps:** The following steps have to be followed to carry out the test.

1.Train the model automatically create the “Model” folder .

2. Prediction contain the Max length of words .

**Expected Results:** To display the Captions for Image.

**Result:** Successful.

**Chapter 7**

**Project Plan & Schedule**

* 1. **Project Planning and Project Resources:**

A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish of the terminal elements and summary elements of the project. Terminal elements and summary elements comprise the work breakdown structure of the project.

|  |  |
| --- | --- |
| WorkTask | Date |
| Problem Definition | September 1st Week |
| Existing System and Requirement Analysis | September 3rd Week |
| Information Gathering | October 2nd Week |
| Design | November 2nd to December 4th Week |
| Initial Report | January 2nd to February 1st Week |
| Implementation | March 2nd to May 1st week |
| System Testing | May 3rd week |
| Deployment | June 2nd to June last week |
| Final Report | July 2nd week |

Table 7.1 Project Planning

**7.2 Project Scheduling:**

**Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ACTIVITY/  MONTH | SEP | OCT | NOV | DEC | JAN | FEB | MAR | APR | MAY | JUN | JUL |
| Problem definition |  |  |  |  |  |  |  |  |  |  |  |
| Existing system and requirement analysis |  |  |  |  |  |  |  |  |  |  |  |
| Information gathering |  |  |  |  |  |  |  |  |  |  |  |
| Design |  |  |  |  |  |  |  |  |  |  |  |
| Initial Report |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |
| System Testing |  |  |  |  |  |  |  |  |  |  |  |
| Deployment |  |  |  |  |  |  |  |  |  |  |  |
| Final Report |  |  |  |  |  |  |  |  |  |  |  |

Fig. 7.2 Gantt Chart

**7.3 Effort Estimation:**

In software development, Effort Estimation is the process of predicting the most realistic amount of effort required to develop or maintain software based on incomplete, uncertain and noisy input. This effort is traditionally measured in hours worked by a person or the money needed to pay for this work.

Effort Estimation is used to help draft project plans and budget in the early stages of the software development lifecycle. This practice enables a project manager or a product owner to accurately predict costs and allocate resources accordingly.

Project manager will estimate the effort required to complete each item. Rather than using time or cost estimates they will look at user stories and story points. A product owner will compare the features of their new project with a previous one to determine the complexity of their user story and assign suitable story points.

**Chapter 8**

**Risk Management and Analysis**

**8.1 Project Risk Identification:**

Account Registration: If user haven’t created account so user cannot access features.

User Ignorance: User having insufficient knowledge about user interface.

**8.2 Project Risk Analysis:**

Create Account: User must create account with proper details.

User friendly interface: Creating simple and compatible user interface for ease of access the features.

**1)Power cut-off**

Analysis – Power cut-off may cause on Result from the Data cleaning and Data training.

Management- Charge the system.

### 2)System Failure

System failure can cause loss of hardware, software, data, or information.

Analysis - System failures may Result from a motherboard can cause a system failure because the computer or operate in general and also like processor, RAM. It is defective and must replace. In other side generally software issues may occur like computer freezes, slow processing etc. for troubleshooting restart web application.

**3)Dataset failure**

Analysis – Dataset Failures may result of training dataset in improper manner for creating the Captions.

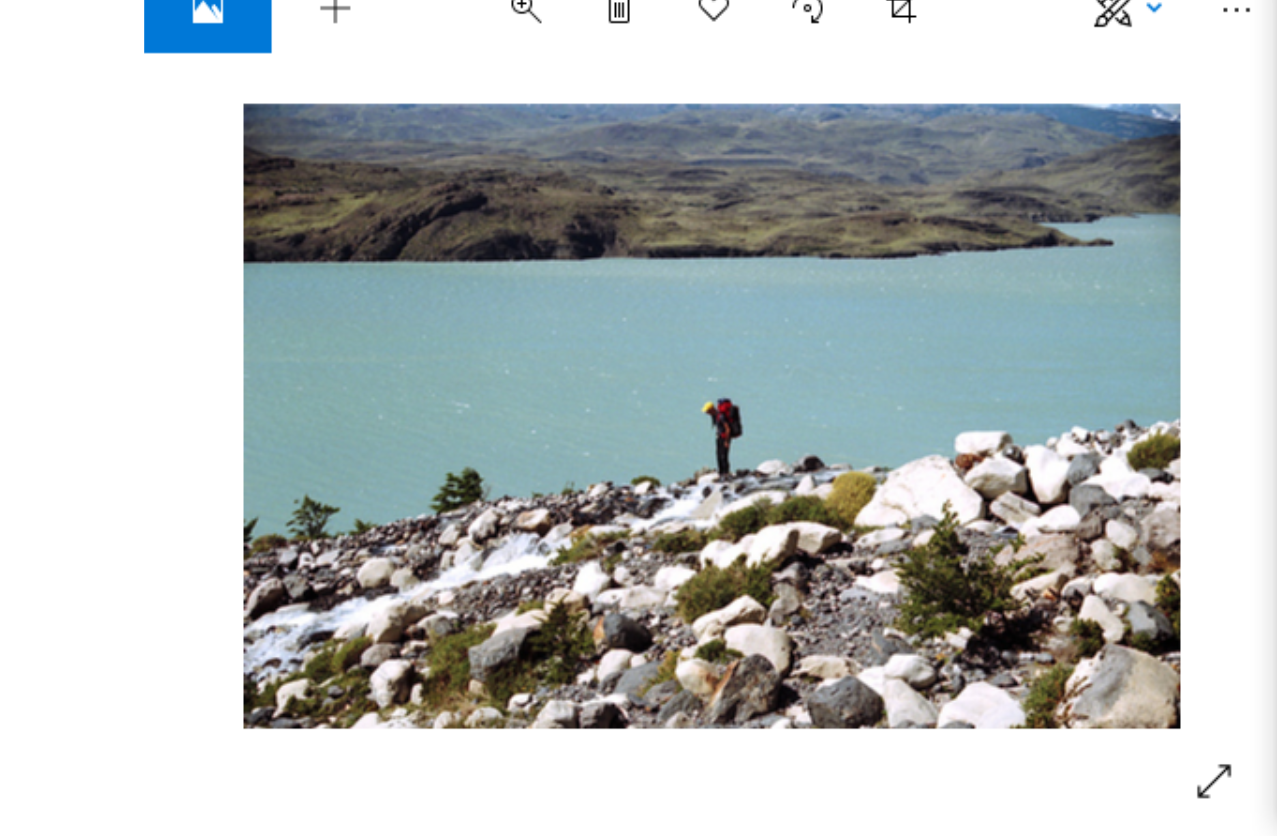
Management – We can use modified method to train the Model in correct manner.

**Chapter 9**

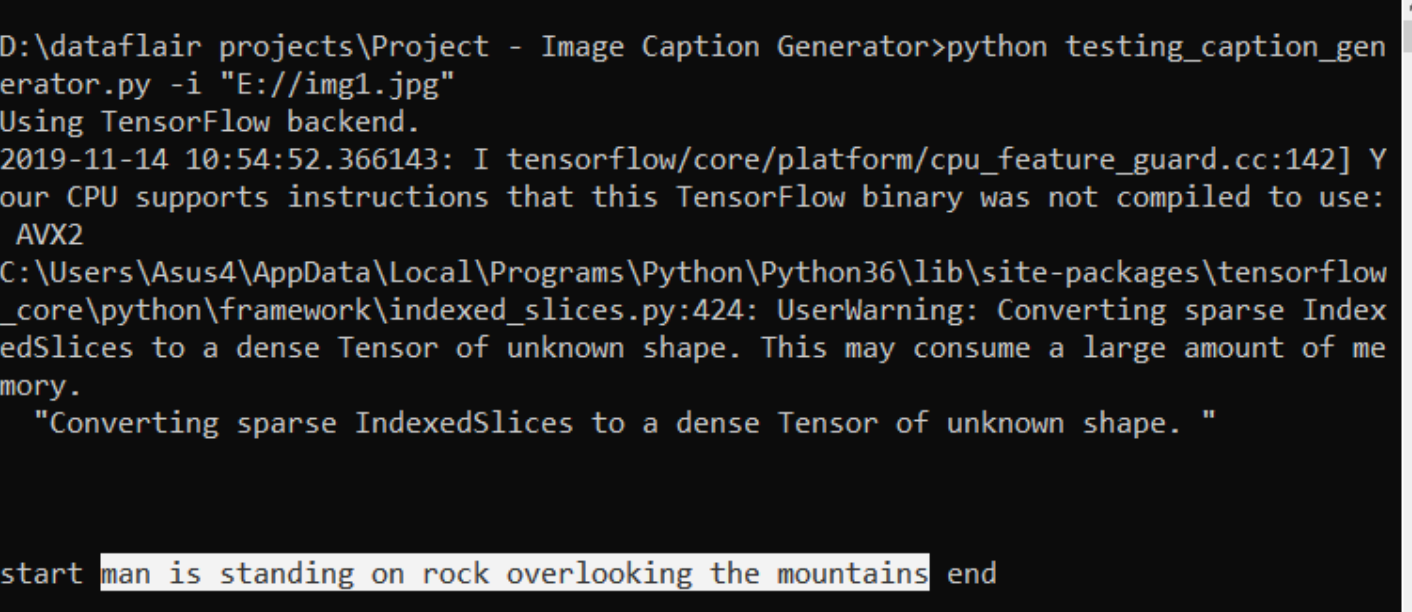
**Configuration Management**

**9.2 User Manual:**

**9.2.1 Input Screenshots**



**9.2.2 Output Screenshots**



**Chapter 10**

**Conclusion and Future Scope**

**10.1 Conclusion:**

We have presented a deep learning model that tends to automatically generate image captions in a grammatically format and display the context in a natural languge.

The CNN and the LSTM works together in proper synchronization, they were able to find the relation between objects in images.

**10.2 Future Scope:**

The future scope of our project is that to help visually impaired people to better understand their surrounding.

**References**

[1] International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075,” Image Caption Generato”, Megha J Panicker, Vikas Upadhayay, Gunjan Sethi, Vrinda Mathur , Volume-10 Issue-3, January 2021.

[2] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7239-72, Published by : Research India Publications. http://www.ripublication.com 7239 “Image Captioning - A Deep Learning Approach “ Lakshminarasimhan Srinivasan1 , Dinesh Sreekanthan2 , Amutha A.L.

[3] Computer vision foundation ”Show and Tell: A Neural Image Caption Generator” published by :IEEE xplore , Oriol Vinyals vinyals@google.com Alexander Toshev toshev@google.com Samy Bengio bengio@google.com Dumitru Erhan [dumitru@google.com](mailto:dumitru@google.com)

[4] “Image Captioning using Visual Attention” Anadi Chaman(12105) and K.V.Sameer Raja(12332) October 4, 2015

**Websites:**

* https://en.wikipedia.org/wiki/Natural\_language\_processing.

* https://en.wikipedia.org/wiki/Convolutional\_neural\_network.