**Image Description Generator**



**Junaid Khan**

**Umm e Maryam**

**Atiqa Javed**

*Supervised By*

**Iqra Chaudhary**

*Submitted for the partial fulfillment of MCS degree to the Faculty of Engineering & CS*

DEPARTMENT OF COMPUTER SCIENCE

**NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD**

**September, 2021**

**ABSTRACT**

Humans have the basic ability to describe an image with a variety of information about it with only a fast glance. Image description generator, produces natural language according to the content observed in an image, and it is an important part of scene understanding, which tells us the about what is happening in the image. This project entails constructing a simple html page with an image uploader, after which the image will be sent to a trained model, which will provide an image description. This model is used to generate the text from the given input images. The learning model generates natural language for the images. Dataset contains 8 thousand images for the description.

Current research in computer vision and machine learning has demonstrated some great abilities at detecting and recognizing objects in natural images. Current state-of-the art results in object detection, classification and localization in ImageNet Challenges have the validation accuracy for top 5 predictions for classification to be at 3.08% while similar classification experiments run by trained humans report an accuracy of 5.1%.

While some people might argue that human accuracy is a function of training time it can be said with great confidence that automated classification models are at least as good as trained humans in classification problems. The ability of these models to analyze and describe complex images, however, is still an active area of research.

**CERTIFICATE**

Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Final Approval**

It is certified that the project report titled **“Image Description Generator”** submitted by **Junaid Khan, Umme Maryam** and **Atiqa Javed** in partial fulfillment of the requirement of **“Master’s Degree in Computer Science”** is approved.

**COMMITTEE**

Dr. Basit Shahzad

Dean Engineering & CS: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Sajjad Haider

HOD Computer Science: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Mehwish Sabih

Head Project Committee: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Asim Rehan

Project Coordinator: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Miss Iqra Chaudhary

Supervisor: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARATION**

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade, and our Master's degree may be revoked.

Group members:

**Name Signature**

Junaid khan (6502) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Umme Maryam (6515) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Atiqa Javed (6497) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PLAGIARISM CERTIFICATE**

This is to clarify that the project entitled “**Image Description Generator**”, which is being submitted here with for the award of the “**Degree of Masters**” in “**Computer Science**”. This is the result of the original work by **Umme Maryam**, **Junaid khan**, and **Atiqa Javed** under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of any degree or compatible certificate or similar tile of this for any other diploma/examining body or university to the best if my knowledge and belief.

**Turnitin Originality Report**

Submission date: 29-Aug-2021 06:10PM (UTC+0500)

Submission ID: 1637594769

File name: Image\_Description\_Generator\_-\_2.docx (792.69K)

Word count: 8132

Character count: 42476

Similarity Index 12%

Similarity By Source

Internet Source: 7%

Student Papers: 7%

Publications: 0%

Date: 29-08-2021 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iqra Chaudhary (Supervisor)

**TURNITIN ORIGINALITY REPORT**

Group based application **Image Description Generation** [MCS] **by Junaid Khan, Umme Maryam** and **Atiqa Javed.**

From: Iqra Chaudhary

Submission date: 29-Aug-2021 06:10PM (UTC +0500)

Submission ID: 1637594769

Word count: 8132

Character count: 42476

Similarity Index 12%

Similarity by Source

Internet Source: 7%

Students Papers: 7%

Publications: 0%

**SOURCES:**

1. 5% match from student papers

Submitted to Higher Education Commission Pakistan

1. 5% match from internet source

[scholarworks.rit.edu](https://scholarworks.rit.edu/)

1. 1% match from student papers

Submitted to National Institute of Technology, Hamirpur

1. <1% match from student papers

Submitted to University of Mauritius

1. <1% match from internet source

[brage.bibsys.no](https://www.unit.no/ugyldig-lenke-til-dokument-i-vitenarkiv)

1. <1% match from internet source

[deivasathish.blogspot.com](https://deivasathish.blogspot.com/)

1. <1% match from student papers

Submitted to Queen Mary and Westfield College

1. 1<% match from student papers

Submitted to CollegeAmerica Services, Inc.

1. <1% match from student papers

Submitted to University of Computer Studies

1. <1% match from student papers

Submitted to Midlands State University

1. <1% match from student papers

Submitted to Western International College (WINC London)

1. <1% match from publication

Xiao-Dong Li, J.K.L. Ho, T.W.S. Chow. "Approximation of dynamical time-variant systems by continuous-time recurrent neural networks", IEEE Transactions on Circuits and Systems II: Express Briefs, 2005

1. <1% match from internet source

[umpir.ump.edu.my](http://umpir.ump.edu.my/)

1. <1% match from student papers

Submitted to De Montfort University

1. <1% match from internet source

[www.flipkart.com](file:///C:\Users\junaid%20khan\Desktop\mam%20ko%20send%20krnay%20wali%20reports\www.flipkart.com)

**ACKNOWLEDGMENT**

All the greatness is for **ALMIGHTY ALLAH** who gave us potential, capability and influence to complete the project. We are highly thankful to **Faculty of Computer Science** for continuous supervision, compensation and auspices. It gives us a huge gratification to receive the supervision, intention, intimation, judgment, and inspiration provided by the project supervisor **Ms. Iqra Chaudhary**. We are much obliged to his for careful consideration, gratitude and supervision throughout the project. His moderate way was also offered to us all through the project. We are really grateful to our parents who always motivated us and remember us in their special prayers for achieving goals in our lives and all other friends who boosted up our motivations. And special Thank to **ALMIGHTY ALLAH** once again.

**TABLE OF CONTENTS**

**Chapter Page**

CHAPTER1: INTRODUCTION………………………………………...……………….. 1

[1.1. Project Domain ……………………………………………………………………..... 3](#_Toc74042322)

[1.2. Problem Identification:](#_Toc74042323) ……......…………………………………………...……….... 4

[1.2.1. Proposed Solution ……….………………………….…………..….................. 4](#_Toc74042324)

[1.2.2. Objectives: …………...….………………………………………………….… 4](#_Toc74042325)

[1.2.3. Scope of the Project ………………………………...…….…………………........... 5](#_Toc74042326)

[1.3. Effectiveness / Usefulness of the System …………………………….…………....… 5](#_Toc74042327)

[1.4. Resource Requirement: …………………………..…………………………..…….... 6](#_Toc74042328)

[1.4.1. Hardware Requirement: ………….………………………………………..….. 6](#_Toc74042329)

[1.4.2. Software Requirement …….…………………………………………………... 6](#_Toc74042330)

1.4.3. Data Requirement ………………………………………………………...…… 8

[1.5. Report Organization …………………………………………………….………...…. 9](#_Toc74042331)

[CHAPTER 2: BACKGROUND AND EXISTING SYSTEMS ………...…………...….. 10](#_Toc74042332)

[2.1. Related Literature Review…………………………………………………………... 11](#_Toc74042334)

[2.2. Related Systems/Applications …………………………………………………..….. 13](#_Toc74042334)

[2.3. Identified Problem from Existing Work …………………………………………..... 1](#_Toc74042339)3

[2.4. Selected Boundary for Proposed Solution ………………………………………...... 14](#_Toc74042341)

[CHAPTER 3: SYSTEM REQUIREMENT AND SPECIFICATIONS ……...………..... 15](#_Toc74042342)

[3.1. System Specification …………………….………………………………………..... 16](#_Toc74042344)

[3.2. System Modules …………………………………….…………………………........ 16](#_Toc74042345)

[3.2.1. Flask Module …..……………………………………………………...……... 16](#_Toc74042346)

[3.2.2. Deep Learning Module …..………………………………………...……….... 17](#_Toc74042347)

[3.2.3. Webpage Module …………………….…………….…………………..…..... 17](#_Toc74042348)

[3.3. Functional Requirements /Software Features: …………………..………...………... 17](#_Toc74042350)

[3.3.1. Image Upload …………………………………………………………...….... 17](#_Toc74042351)

[3.3.2. Flask ………………………………………………………..………………... 17](#_Toc74042352)

[3.3.3. Machine Learning Module ……….……….…………………………….......... 17](#_Toc74042353)

[3.4. Non-Functional Requirements: ……………………………………….…………..... 18](#_Toc74042350)

[3.4.1. Efficiency ………………………………………………………………......... 18](#_Toc74042351)

[3.4.2. Usability ……………………………………………………………………... 18](#_Toc74042351)

[3.4.3. Accuracy ………………………………..………………………………….... 18](#_Toc74042351)

[3.4.4. Testability of System ………………………………………………………..……. 19](#_Toc74042351)

[3.4.5. Required Interface ………………………………………………………………... 19](#_Toc74042351)

[CHAPTER 4: SYSTEM MODELING AND DESIGN …………..……………….....….. 20](#_Toc74042363)

[4.1. System Design and Analysis ……………………………………………………....... 21](#_Toc74042365)

[4.2. Use- Case Diagrams ………………………………………………………..……..... 21](#_Toc74042366)

[4.3. Activity- diagram: ……………………………………………………………..….... 22](#_Toc74042367)

[4.4. Data Flow Diagram …………………………………………………..……….….… 23](#_Toc74042365)

[4.4.1. 0-Level-DFD ………………………………….…………...……………….... 24](#_Toc74042369)

[4.5. Full Dress Use Case/Detailed Use Case …………………………………………...... 24](#_Toc74042367)

[4.5.1. Upload Image ………………………………………………………………... 24](#_Toc74042369)

[4.5.2. Get Description ………………………………..……………………….......... 25](#_Toc74042369)

[4.6. Sequence diagram …………………………………………………………………... 26](#_Toc74042368)

[CHAPTER 5: SYSTEM TESTING AND VALIDATION ………...…..……………….. 28](#_Toc74042371)

[5.1. System testing …………………………………………………………...……….… 29](#_Toc74042373)

[5.2. Testing Techniques ……………………………………………………...……….… 29](#_Toc74042374)

[5.2.1. Unit Testing …………………………………..………………...…................. 30](#_Toc74042376)

[5.2.2. User Testing …………………………………..………………………........... 30](#_Toc74042377)

[5.2.3. The Black Testing …………………………………………….…….………... 30](#_Toc74042376)

[5.2.4. The White Box Testing …………………………..……………………...….... 30](#_Toc74042377)

[5.2.5. Integration testing ………………………………………………….……….... 30](#_Toc74042376)

[5.2.6. Acceptance Testing/User Testing ……………………………...…..……….... 31](#_Toc74042377)

[5.3. Test Cases ……………………….………………………………………...………... 31](#_Toc74042375)

[5.3.1. Test Case 1: User model testing ……………..…………………...…...….…... 31](#_Toc74042376)

[5.3.2. Test Case 2: Flask model testing …………………………..…...…………….. 33](#_Toc74042377)

[5.3.3. Test Case 3: Machine learning model testing ……….…………………..….... 34](#_Toc74042378)

[5.4. Non -Functional Requirements ………………………...…………………….……... 36](#_Toc74042375)

[CHAPTER 6: CONCLUSION ……………….……...…..…………………………........ 37](#_Toc74042382)

[6.1. Conclusion: ……………………………………….……………………………....... 38](#_Toc74042384)

[6.2. Limitations & Future Work ………………………………………………....……… 39](#_Toc74042385)

APPENDIX – I ……………………………………………………………...……..……. 40

[REFERENCES ………….………………………………………………………...……. 42](#_Toc74042387)

**LIST OF FIGURES**

**Figure Caption Page No.**

1.1: **A man is standing on a snow skateboard** …….…………………..……………….…. 3

1.2: The person is riding a surfboard in the ocean ……………..………………..……….. 3

4.1: Use Case Diagram ………………………………………..…………..………..…… 22

4.2: Activity Diagram .…….……………………………….…………..…....................... 23

4.3: DFD Level 0 ....……………………………………….………..……………....…… 24

4.4: Sequence Diagram ……………………………..………...………….....................… 26

4.5: System Sequence Diagram …………………………..…………………...…….…... 27

**LIST OF TABLES**

**Table Caption Page No.**

1.1: Software Requirements ................................................................................................ 6

1.2: Hardwar Requirements ................................................................................................. 6

2.1: Related Work …………...…...................................................................................... 12

2.2: Summary of Existing Systems .................................................................................... 13

4.1: Full Dress Use Case for Image Upload ....................................................................... 24

4.2: Full Dress Use Case for Getting Captions .................................................................. 25

5.1: Test Case1 …………………….…………..……....................................................... 31

5.2: Test Case2 .................................................................................................................. 33

5.3: Test Case3 ………………………..……..………...………………........................... 34

5.4: List of Common Non-Functional Requirements …………………………...……..… 36

**Chapter 1**

**INTRODUCTION**

People who are visibly challenged can use an image description generator to help them visualize the world in the future. It can generate sort of an image using static objects and calculation language models. The optical recognition modeling is used to produce a sentence that represents an image by reducing difficulties. It is a language model taught to measure movement in images. The system creates meanings by finding nouns, verbs, prepositions and scenes from the picture to form a sentence. A visual examination framework that gathers items, designs, and connections in an image and converts them into a request for semantic trees, which are used to study grammar and make descriptive meanings. The neural network is used to locate objects in the image to produce a natural language. The convolutional neural network is used in the project to detect the objects in the images.

The most important problem involving computer visualization and regular language handling is the programmed showing of a picture. There have been many attempts at image classification, object identification, and image annotation, but there has been very little focus on creating posted descriptions. Because of the complexity of the task and to create a machine learning model is a difficult task. As a result, the goal is to develop a system that can create image descriptions.

Humans have a basic ability to interpret the image with so much knowledge about it. Creating a computer program that mimics human skills. In the fields of machine learning and artificial intelligence, the long-term desire of researchers has been to create human like machines. Creating a computer system that can detect imagery and provide natural language description. It is very difficult task to create a model that can mimic the ability of a human but with enough training of the model it can be possible because huge datasets of images are available for the model to train.

For instance, a boy playing with his ball at the beach looking at the number of people in the image, or what he is doing with the ball and where he is playing, retrieving a simple and fast-moving context that describes the maneuver of people in an image.

For Example, a monkey is snathing a banana from a person and he is also very angery in the image, or what he is doing wih the banana and where is the banana, getting information from the image different objects and then describes the image.



Figure 1.1: **A man is standing on a snow skateboard**



Figure 1.2: The person is riding a surfboard in the ocean

# 1.1. Project Domain

In the image description generator, we insert the image and create a natural language description for the visual content of the image. Understanding of image and language production is both parts of this work are a challenge to the open problems in their fields. Understanding image algorithms are used to find visual frameworks and concepts in images.

# 1.2. Problem Identification

Using natural languages ​​to automatically describe image content is an important and difficult task. It has a lot of power. For instance, it can help people who seem to be disabled to understand the content of web images. In cases such as sharing images on social media, it can also send accurate and precise image information. Because cameras are everywhere nowadays, if the system can provide good captions, we can raise alarms immediately when dangerous work is done. This can help reduce some crime and accidents in the city. An image description generator can help Google Image Search look more like Google Search, because every time someone searches for an image, the picture can first be transformed into a caption, and a search can be made based on the caption.

1.2.1. **Proposed Solution**

The proposed program has a simple html page where the user can upload an image. The image will be uploaded to the flask node and the bottle will serve as a bridge between the in-depth reading model and the html front page. A mature solution to create a model that can access a variety of objects and images using a CNN model [1]. After using the CNN model, we will send the image elements to the RNN model to create a global sequence using LSTM [2].

**1.2.2. Objectives**

The main objectives of this project work are that:

1. **Object detection:** The CNN model is used to detect the objects in the input image.
2. **Feature Extraction:** The image's features are extracted using NumPy's principal component analysis. Scene classification is handled by CNN, while object detection and human traits are handled by RNN.
3. **Creating attributes:** The properties with their label strings were defined using the characteristics gathered by the neural networks.
4. **Encoder and Decoder:** The label strings were fed through a decoder RNN, which encoded them into a format that could be converted into an appropriate descriptive phrase.
5. **LSTM:** Long short-term memory is a machine learning model that is used to predict the next sequence of words during generating description of the images. LSTM uses the words vocabulary to predict the sequences of the words.

**1.2.3. Scope of the Project**

Precise interpretation of the photos and videos can be of great help to clinical, and for security purposes, and military applications, and can even be utilized to disclose metaphorical items to the visually impaired peoples. Captioning an image can be a daunting task due to uncertainty of various situations, the main discovery of the subject, the discovery of the object, the recognition of the action, and the understanding of the scene. Notwithstanding, the difficulties and challenges, researchers in the field of computer science and natural language research have made great strides in recent times. The project is about improving prevailing systems to produce more comprehend results by training the model with larger image sets.

**1.3. Effectiveness / Usefulness of the System**

The system has two machine learning models CNN [1] and LSTM, which use the power of CNN and LSTM. The LSTM [4] model is used to produce word sequences and the CNN model is used to locate objects in an image. These LSTMs are used to embed captions, while CNN renders image and language features available separately before being listened to in a shared presentation in the merging step. This makes the integration of learning models of different machines, to my knowledge, the first research of its kind. The different categories of neural networks eliminate the need for the allocation of parameters while learning in different ways and allowing each method to be trained using a technique that best reflects its four different characteristics. In addition, by enabling the different functions of RNN and CNN, as well as increasing the size of the words will affect the various functions of the operation and operation of the image description model, it is being investigated in this study. In inclusion, due to the general nature of LSTM-based architecture, the project can also be used in to solve multi-module learning problems.

**1.4. Resource Requirement**

The website should serve as a source for users to communicate with a qualified model. The user will upload the image to the website, which will use the machine learning model. As a result of its findings, the website should display a description of the image to the user. The system must receive an image from the user as input. To see image elements visible, the system must use the CNN model. To understand the text in pictures, the system must use RNN. Obviously, the system should use the RNN model in generating sentences. The image must be defined by the system.

**1.4.1. Hardware Requirement**

The following table lists the minimum hardware requirements for this project:

Table 1.1: Hardware Requirements

|  |  |
| --- | --- |
| **Serial No.** | **Requirement** |
| 1 | Computer/Laptop |
| 2 | Intel core i7 processor 4th gen |
| 3 | 8GB RAM or more. |
| 4 | A good graphical processing unit |
| 5 | 500 Gb hard drive |

**1.4.2. Software Requirement**

The following are the minimum software requirements for this project:

Table 1.2: Software Requirements

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Tool** | **Purpose** |
| 1 | Vs Code | Microsoft's Visual Studio Code (Vs code) is a freeware source-code proofreader for Windows. |
| 2 | Html | it is a language for creating web pages. |
| 3 | Flask | Flask is a well-known Python web framework, or a Python library for building web applications. |
| 4 | Python-3 | Best language for Ai models. |
| 5 | Tensor Flow | It's a machine learning library that may be used for things like computer vision and natural language processing. |
| 6 | NumPy | It’s a python library through which you can modify the pixel values of an image and to create array of the images. |
| 7 | CNN | CNNs, or Convolutional Neural Networks, were created to map picture data to a variable. They've proven to be so effective that they're now the go-to method for every form of picture data prediction challenge |
| 8 | RNN | RNN is used to build LSTM and dense layers with embedding to predict the next work in sequences |
| 9 | Matplotlib | Matplotlib is a python library for data visualization and graphical plotting and it is a numerical extension of NumPy. |
| 10 | Glob | The glob module in Python is utilized to discover documents/pathnames that match an example. |
| 11 | OpenCV python | The OpenCV is a python library that is used to read images from dataset. In CV2, the imread () method loads an image from a specified file. |
| 12 | Tqdm | It is used to show a smart progress meter for a loop in the project. |

**1.4.3. Data Requirement**

Artificial intelligence built on data. People quickly discover that a large amount of information can be used to find many difficult rules. There are many rich and colorful databases available in the image description that do the job, such as Flickr8k dataset, MSCOCO dataset, Flickr30k database, PASCAL 1K, AI Challenger databases, and STAIR captions, which continue to be the subject of debate. The number of images in every database is summarized, and each image in the database has five source explanations. To describe an image each database image has different independent meanings, the database uses replacement captions to describe the same image. Having multiple captions for each image helps the model to focus on the various aspects of the scenes to produce accurate captions.

The Microsoft COCO Captions dataset [7], which focuses on location comprehension, includes images from complex everyday situations and can be used for many applications including image acquisition, classification, and description. A coco database that uses the Amazon Mechanical Turk app to produces five different captions, an aggregate of in excess of 1.5 million sentences. There are 82,783 photos on the training set, 40,504 photos on the confirmation set, and 40,775 photos on the test set. Its 2014 data system contains 20 gigabytes of images and 500-megabyte annotations files that show the relationship between a single image and its meanings. [MSCOCO, which highlights important elements in an image, such as capital letters, shapes, functions, and many more. Chinese phrases generally have more flexibility in syntax and lexicalization than English databases used in the same scientific research activities, and the algorithm's initial barriers are likewise more noteworthy. STYLE. Japanese picture description information database, created using images from the MSCOCO collection. STAIR has 164,063 images and 820,311 Japanese depictions like every one of the five pictures. It is the biggest picture depiction database in Japan. In the project we use a Flickr8k dataset [3] with 8,000 images and each image has 5 captions.

**1.5. Report Organization**

The report is organized as follows. Chapter 1 is about introduction like what technologies are used in the project and what solution the project is providing. Basically, the report is divided into two sections. The first section is for analysis and the second section is about current solutions to the problem.

Chapter 2 briefly describes past work that has been done in this area of field that have motivated this work and how the work incorporates and expands on those methods. This chapter is about identifying the problem in the existing system and solving them and also coming up with new ideas. Chapter 3 presents software requirements and system specification like which type of hardware is required to process the images and to generate descriptions for it plus what type of software we will be using to get accurate descriptions.

Chapter 4 is about System modeling, flow diagrams, use cases, system sequence diagram. This chapter describes the research process and results. In this chapter the diagram is used in the report to help the reader understand the project easily.

Chapter 5 presents the testing phase of the project in which all the modules of the project are testes separately to ensure the proper working of the project. The tests are conducted with a lot of care to sort out any errors in the projects

The chapter 6 is about overall conclusion and ideas for future work. This chapter describes the research process and results. Future works that can be done in this field are discussed in this chapter.

**Chapter 2**

# BACKGROUND AND EXISTING

# SYSTEMS

Using deep learning process state of the captioning for artistic captions was showed. These strategies investigate visual information, characterize and find items and activities, and use captions to define static frames and videos. All of these tasks utilize a regulated learning strategy where subtitles are utilized to prepare the organization. The release of visual element, convolutional neural networks (CNNs) are used, and the reconstruction of network-based network structures, either simple repetitive network or LSTM-based craftsmanship, are utilized to become familiar with the language model and ensuing definitions. This undertaking work is roused by their work, takes on a portion of the standards utilized underway, and reaches out to those methodologies to assist with conquering their constraints to accomplish better outcomes. This section describes the changed concepts and briefly guides students on the methods used in the above lessons.

**2.1. Related Literature Review**

The first work is being narrated by Karpathy et al. [5]. It employs a CNN that has been pre-prepared on ImageNet and calibrated on ImageNet challenge datasets.In addition, a caption generator is created using a Simple Recurrent Network (SRN). During preparing, the SRN got the picture include descriptor from the CNN, just as the catchphrase START at the first-run through example, and each word in the ground truth picture inscription from the preparation information at each time occasion, just as the concealed state from the past time step. Subsequent to preparing with adequate models, the SRN learns the language semantics and foresee the coming word with a superior precision dependent on either the past work or the picture standards through the weight refreshes. During testing, the picture include descriptor extricated from the CNN is utilized as the principal contribution to the SRN alongside the catchphrase START. The image feature descriptor predicts the first word of the image generator. The previous prediction, as well as the previous hidden state, are used to make the next prediction. The measure proceeds until the finish of sentence has been experienced. Before the training and testing, Karpathy preprocessed the words by mapping them into the same vector space as the image feature vector extracted from the CNN such that the dot product of a word vector with its corresponding image vector is maximized.

This has been achieved by an CNN as proposed by S. Ioffe in [1], which detects the top nineteen regions/objects in an image and generates twenty image feature vectors by passing these nineteen regions jointly with the total image through a CNN. In view of the context oriented data encompassing the word in the two ways and the component vector of the word's relating picture, an SRN architecture called Bidirectional Recursive Neural Network (BRNN) is used to map each word into the same vector space as the image feature vector.

An SRN structure called (BRNN) Bidirectional Recursive Neural Network is utilized to plan each word into a similar vector space as the picture head vector based on the contextual information surrounding the word in both directions and the feature vector of the word's similar image. It also employs a bar search strategy that keeps up with track of conceivable picture captions and afterward picks the expression with the least misfortune, which is the supreme worth of the amount of log probabilities of each word in the subtitle given the past words, instead of an SRN architecture. It also employs a beam search method that maintains track of possible image captions and then chooses the phrase with the least loss, which is the absolute value of the sum of log probabilities of each word in the caption given the previous words, instead of an SRN architecture.

Table 2.1: Related Work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Authors | Contribution | Techniques | Limitations |
| 2015 | S. Ioffe and C. Szegedy | Proposed CNN | CNN | Expressive and diverse |
| 2014 | Karpathy et al | Convolutional image description | FRMM | COCO Caption was used |
| 2016 | S. Hochreiter and J. Schmidhuber | Boosting image description with attributes | LSTM | 82,783 images were tested |

**2.2. Related Systems/Applications**

Systems and applications related to the project are given below:

Table 2.2: Summary of Existing Systems

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | System | Contribution | Tool/Technologies | Limitations | Applications |
| 2018 | Bottom-up and top-down attention for image description. | Bottom-up and top-down attention | RNN | Expressive and diverse | Computer Vision and Pattern Recognition. |
| 2018 | Convolutional image description | Convolutional image description | FRMM | COCO Caption was used | Computer Vision and Pattern Recognition |
| 2016 | Boosting image description with attributes | Boosting image description with attributes | LSTM | 82,783 images were tested | International Computer Vision |

**2.3.** **Identified Problem from Existing Work**

Artificial Neural Networks (ANNs) are programmatic experiences of a worked-on model of billions of neurons of the human mind. Every hub in the ANN is traced back to a neuron in the human cerebrum and is called a neuron. The performance of ANNs with regards to regulated realizing where the information and related results from preparing information are utilized to assist the organization with understanding the connection among info and yield. Every hub in the ANN is associated with different info and yield hubs with a heavy connection to simulate the performance of the tactile framework and catch the connection among information and yield. Going back, where ANN converts load dependent on how well the anticipated yields are like the real factors of the exploratory information, how these weights are studied over time. The following sections will look at this process in more detail. The terminating of neurons in the cerebrum is brought about by solid electrical spikes. Within each node, non-specific startup functions such as sigmoid, tanh, and queued line units are used to replicate these firing in the ANNs. ANN is an assortment of such hubs, every one of which is associated with numerous different hubs. Network input and output layers are made up of nodes associated with touchy and responsive pieces of the framework, while the hidden layer is made up of nodes connected only to other nodes in the network. Indicates the basic structure of ANN. Feedforward Neural Networks (FNN) and Recurrent Neural Networks (RNN) are two types of ANNs that can be classified according to how the hubs in each layer connect (RNN). The term neuron will be used consistently in the following discussion to refer to a hub in ANN as opposed to a neuron in the human cerebrum.

**2.4. Selected Boundary for Proposed Solution**

Donahue et al proposed the Long-term Recurrent Convolutional Networks (LRCNs) [6]. work on the image definition using in-depth reading. It has comparable provisions, for example, start to finish preparing and subtitle generators dependent on CNN/LSTM. It varies from former research in that the word vector is not limited to the same scale as the picture vector, and presents LSTM multi-line structures. It uses end-to-end LRCN build-up training to optimize fully connected CNN layers using CNN-trained formats in the ImageNet database, as suggested by Karpathy et al. [5]. Notwithstanding LRCNs with a solitary LSTM layer, like NIC, it has two unused LSTM structures where the two layers are utilized for image and word input training (excluding LRCN) or then again just a subsequent layer utilized for both picture and vector addition during the principal layer. stored in word vector input (word vector LRCN) (version included). LRCN results show that a factored LRCN produces preferable subtitles over a latent LRCN. This has prompted having separate learning classes for picture vectors and word carriers, so that different method parameters can be shared. The LSTM design is used to examine picture includes in FRMMs in addition to the independent reading categories for each method. The lstm uses captions words that are store in vocabulary to predict the next sequences of the words to generate captions for the images.

**Chapter 3**

**SYSTEM REQUIREMENT AND SPECIFICATIONS**

This chapter contains information about system specifications and system requirements of the project. This chapter also discusses the system's modules, functional requirements, and non-functional needs. For a better understanding, each heading is precisely defined.

The developed system's system specs are given in part 3.1, which is followed by part 3.2, which discusses the developed system's modules. Following that, section 3.3 emphasizes the system's function needs. Finally, we will discuss the non-functional requirements in section 3.4.b.

**3.1. System Specification**

This project is about creating an in-depth learning model that can produce captions for an image. First, there will be a simple html page with the image up loader and the user must upload the image and the image will be uploaded to the flask server. The flask will transfer the image to the machine learning model. The machine model will use forwarding techniques and the CNN model will be used in the image to extract features. After removing the features and inserting the embedding size to match the LSTM size of the RNN model. After this the RNN model will generate sentence for the image using the LSTMs.

**3.2. System Module**

The project is divided into three modules. First, the front module is a simple html page where the user will upload the image and the image will be uploaded to the second flask module. The flask connects the front end to the machine learning module. The flask will transfer the image to an in-depth reading model to generate an explanation and resend the output to the flask module to be displayed on the front-end module. There are three modules deep learning, flask and webpage.

**3.2.1. Flask module**

We used Flask as a connector between the machine learning model and front-end. First, the image will be uploaded to Flask and that will be transferred to a machine learning model to process the image and produce its own definition, and then the flask will send the output to the end. The flask module is a connecting bridge basically to connect the front interface that is html page for uploading the image and sending the image to machine learning model to generate the caption.

**3.2.2. Deep learning module**

  Kera’s Neural Network libraries have been used for this project along with Convolutional Neural Network which holds a Resnet50 pretrained neural network that is 50 layers deep and images can be easily classified into 1000 objects parts: for example, cats, mouse, pencils, cars and jeeps etc. CNN has been used as an encoder to extract the features of the images and then we pass this property vector from a linear layer to resemble the embed size of the RNN LSTM layers. Then RNN LSTM layers with the help of vocabulary will generate order of the words to make a sentence.

* + 1. **Webpage module**

This module is the front-end of the project. It is a simple html page using bootstrap frame work for client and server interaction. It includes a simple image uploader where the image can be uploaded.

3.3. Functional Requirements/Software Features

In collecting the requirement, the user expresses his or her specific performance requirements. Here are a few key performance requirements to consider. The first requirement of the project is to have an html page to upload an image. Secondly the image must be uploaded to the flask and the image will be transferred to an in-depth learning model to produce the image definition. The description should be almost accurate but due to the small database it may be different.

**3.3.1. Image upload**

The user should be able to upload an image in the format of Jpeg only.

**3.3.2. Flask**

The Flask will be able to send the image path to the machine learning model and get the description from the machine learning model.

**3.3.3. Machine learning model**

The machine learning model is the main that is trained on a huge dataset of images to generate the descriptions for the images. The machine learning model contains two models CNN and RNN.

## 3.4. Non-Functional Requirements

Non-functional needs are as important and not as critical as functional requirements. Non-working are the properties that will assist you with characterizing the client's elevated requirements and control item quality. Non-functional requirements are an important and useful element of the system development process. Non-functional needs define system features. Non-functional requirements only affect what the system proposes and the features it has. The system must satisfy the user while in use, which can be achieved with operational and non-operational requirements. The following categories are used to classify these requirements:

The list of non-functional requirements is defined by product type and category but can be categorized into different categories, such as Adaptability, efficiency, and downgrade are examples of these conditions.

**3.4.1. Efficiency**

Any web page responsiveness is an important non-functional requirement. The passage it takes for an application to answer and complete a task is a measure of its quality. The website should resend results in a specific time.

**3.4.2. Usability**

The interface of the web page is very user friendly; the user only has to upload an image and click on a button to generate the description for the image.

**3.4.3. Accuracy**

The correctness of the model is the most vital non-functional requirement, as this regulates how best the model is in creation description for the images. The model is trained on a dataset of 80 thousand images so we are expecting an outcome nearly accurate but as in machine learning achieving 100% result is a difficult task so there will be a few exceptions in the result.

**3.4.4**. **Testability of System**

The program or application is tested on many levels. First, a unit test is performed on each project module, and then a black box test is performed throughout the program to see if there are any errors. In simple terms it can be defined as the magnitude of how a system can be tested. To measure software testing, you need to measure software control, visibility, availability and simplicity.

**3.4.5. Required Interface**

In interface requirement we have simple html page interface in which user have a button “upload image button” and a button to select an image.by clicking on the select button user will be able to select the desire image.

After selecting image user will click on upload image button to upload the image on the flask server. Then the interface will have a button “get description” by clicking on the image the interface will display the selected image and the description that is predicted by the machine learning model.

**Chapter 4**

SYSTEM MODELING AND DESIGN

The process of establishing a system’s structure, there are components, modules, API, and data tables to meet the specified operational and non-functional requirements. System design is the use of a theory system in product development. Cases of program implementation, as well as class diagram, prescription, sequence diagram, and architectural designs, are discussed in the "System Modeling and Designs" chapter. Engineers and codes can understand anything because all methods or functions are performed independently. To explain in detail how this project works we will draw flow diagrams, flow charts and flow tables.

**4.1. System Design and Analysis**

System information is determined by system design and analysis, which is an important facet of the development process. System analysis and configuration can help the user get a quick overview of the system's capabilities without having to test or use it in person. It assists developers in identifying or understanding the benefits and challenges of a project before its implementation. The program plan provides a template for the program on which it is built. To make the system more efficient and robust to meet market needs, an engineer analyzes system errors before building them so that they can overcome them rather than waste time building the wrong system. The system design helps the user to fully understand the project performance with the help of various flow diagrams. The layout design helps you understand the project's vision through visual presentations.

**4.2. Use Case Diagrams**

Diagrams of use cases are for visual representations of the program and how a different user will interact with its features. Primarily, it is defined as the visible representation of interactions between different elements of a system. Drawings of the various character cases and application cases are used to represent the different functions of the program. The set of functions, services, and functions to be performed by the system are referred to as operating cases. In this context, the program is a whole project in which the user will interact with and perform a different function. "Characters" are people or objects that play out specific capacities inside the framework. Use case outlines are useful for picturing the utilitarian necessities of a framework, which will result design decisions and development priorities.

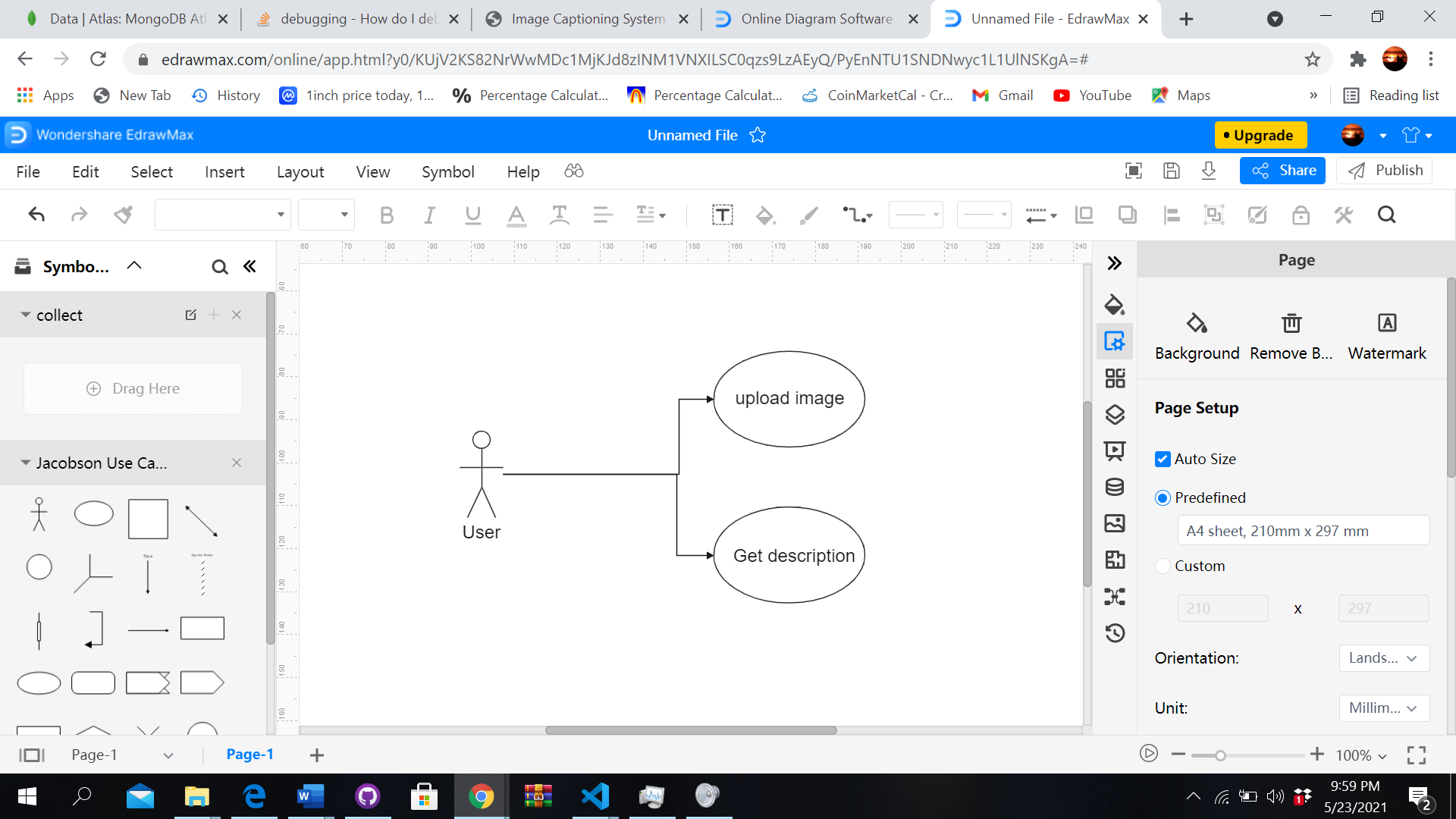


Figure 4.1: Use Case Diagram 1

**4.3. Activity diagram**

Activity diagram is a more intelligent kind of a flow chart that shows the flow of data from one function to the other. It's fragmented down into one or more steps, each with a distinguish shape for a different motive.

The functionality is represented by rectangles, the flow is represented by arrows, the conditions are represented by diamonds, and the start and end are represented by circles.

The activity helps the user to understand the activities that can be performed on the software to interact with the software. Activity diagram is one of the important design modeling diagrams in the report.

Activity diagram can also be used to show both sequential and concurrent processing of activities. They are commonly used in business and process modelling to represent the dynamic features of a system. The main propose of the activity diagram is to tell the reader about the flow of diagram.  For example: To understand a concept, some people prefer written tutorial that contain images.

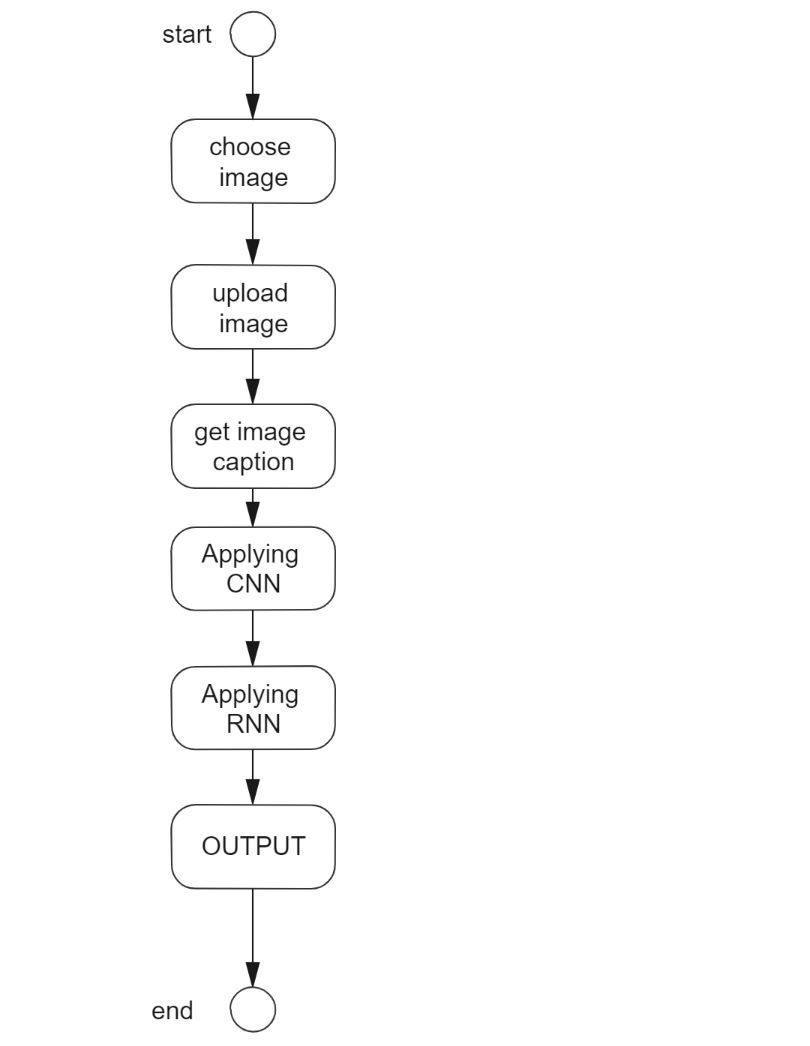


Figure 4.2: Activity Diagram

**4.4. Data Flow Diagram**

A data flow diagram shows the flow of data in a system. It utilizes recognize images like square shapes, circles, and bolts, and short content names, to uncover stream of information 9from one component to another. DFD helps the reader to know the flow of data. DFDs are used to define many entities and their connections using standardized symbols and syntax.

Data flow diagram is a very important diagram for the readers as helps them to easily understand the flow of data.

**4.4.1 0-Level-DFD**

It is otherwise called a setting outline. It's intended to be a deliberation see, showing the framework as a solitary interaction with its relationship to outside elements. It addresses the whole framework as a solitary air pocket with information and yield information showed by approaching/active bolts.

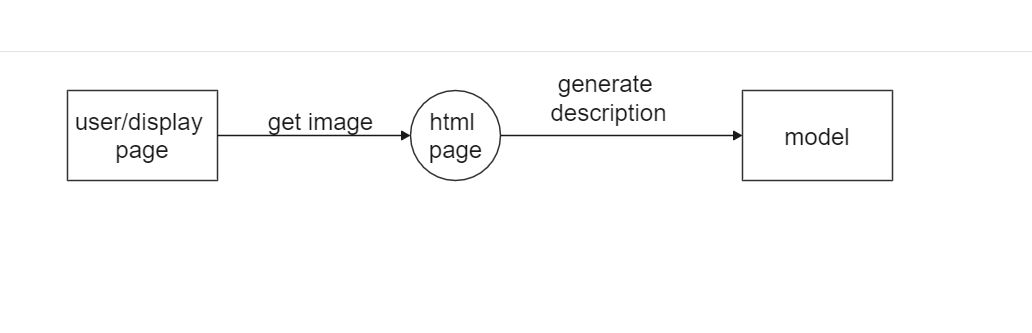
****

Figure 4.3: 0-Level-DFD

**4.5.** **Full Dress Use Case/Detailed Use Case**

Use cases are detailed descriptions of all of the roles that a user performs when using the system. A use case describes every activity he takes or does in detail. Details includes information about who is performing the use case tasks such as who is performing the utilization case assignments, who are the partners, and what are the situation where they are performing the tasks are all presented in Full Dress Use Case. The following is a list of full-dress use cases.

**4.5.1. Upload image**

Table 4.1: Full Dress Use Case for Image Upload

|  |  |
| --- | --- |
| **Use Case Selection** | **Comment** |
| **Use Case name** | Upload image |
| **Scope** | Image description Generator |
| **Level** | User goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | User wants to upload image |
| **Pre-conditions** | User has to open the upload image page |
| **Success Guarantee** | User will successfully able to upload image |
| **Main Success Scenario** | User will successfully able to upload image |
| **Extensions** | If this fails user cannot upload the image. |
| **Special Requirements** | Performance, Reliability, Accuracy, consistency |

**4.5.2. Get Description**

Table 4.2: Full Dress Use Case for Getting Description

|  |  |
| --- | --- |
| **Use Case Selection** | **Comment** |
| **Use Case name** | Get description |
| **Scope** | Image description Generator |
| **Level** | Sub-Function |
| **Primary Actor** | User |
| **Stakeholders/Interests** | User wants to get picture description |
| **Pre-conditions** | Picture must have already uploaded |
| **Success Guarantee** | User will get image description |
| **Main Success Scenario** | User will get image description |
| **Extensions** | If it fails user will not be able to get image description |
| **Special Requirements** | Performance, reliability |

**4.6. Sequence Diagram**

Sequence diagram tells us about the flow of messages by program and is also known as event drawing. Sequential diagram helps to illustrate various dynamic situations. It creates a connection between the sequences of events as they were ordered in sequence or presented simultaneously. In UML, a vertical dotted line will be created that will stretch to the bottom of the page to represent the flow of the message, while the vertical line represents the life line. It involves duplication and branch construction. It scans the real-time system and shows the flow of the message between various items, as well as the ability to quickly update to show system changes. Users and process integration are defined using sequential diagrams. A business relationship, time, or sequence of an item is also shown in the sequence diagram. It can also tell you which classes or procedures are involved in this situation and explain how system discussions are timed and monitored. Sequential drawing is considered important, as it is a tool for design, analysis and writing.

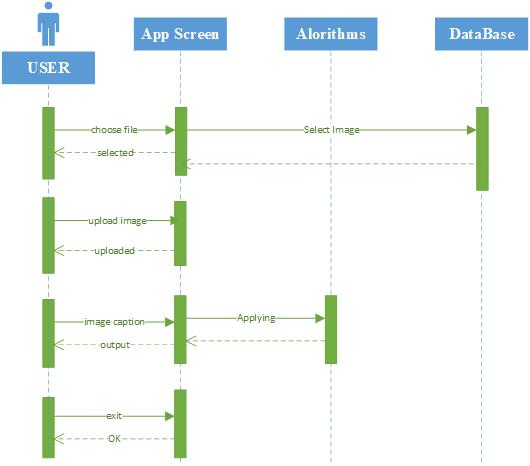


Figure 4.4: Sequence Diagram

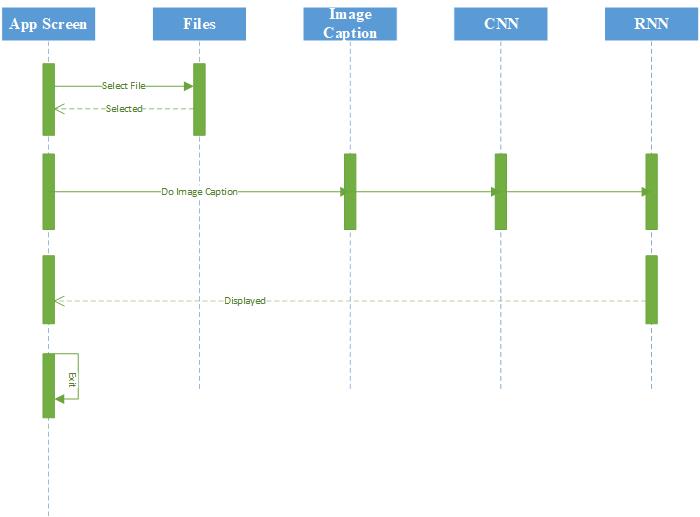


Figure 4.5: System Sequence Diagram

**Chapter 5**

**SYSTEM TESTING AND VALIDATION**

Chapter 5 illustrates the authentication and testing procedures. To confirm that the system is free of errors or flaws, different kind of testing approaches are used. System testing can be performed on the entire finished software or on a specific functionality or element of the system. It is a process for determining whether or not an application that has been construct to deliver is trustworthy and accurate in meeting developer expectations during its use. Validation must be done before software can be released to clients. Some tests are done after the software has been finished. The significance of software testing cannot be overstated.

**5.1. System Testing**

The entire system is put through its paces. The system has been tested using a variety of methods to see if it produces the intended output. If not, these errors must be corrected. The system “image description generator “that it can generator captions for different images according to our expectation. To verify that the system is functioning effectively, we use a variety of methods to ensure that all of the modules are performing as expected. Adjustments must be done if the system does not function as planned; otherwise, the system is ready.

**5.2. Testing Techniques**

Testing Techniques are very important they are used to test a system and its components to determine if they meet the desire requirements. Testing a framework can help reveal any holes, blemishes, or missing necessities that contrast from the first prerequisites. By the help of these techniques the developers try to improve the developed system by removing buds and adding new functions to increase functionality, there are different testing techniques which can be used by the developers to test the functionality of the system. Testing techniques such as Unit testing, integration testing, white box testing, black box testing, acceptance testing, performance testing, security testing, and so on are different examples of these types of tests. The following are the techniques widely used for testing developed systems.

**5.2.1. Unit Testing**

In unit testing we test each module individually in case we encounter an error in the module we try to fix it immediately because once all the modules start working together it can be difficult to remove them. Unit testing is simple and helpful because it allows you to test one module at a time and not the whole system. In the project we have tried to test each module separately to look at the unnecessary dependence of the modules. As a result, we removed any errors using the unit test to make the system work better over time.

**5.2.2. User Testing**

In User Test we take the last user's opinion. During user testing, the system is tested to see if there are any bugs or error so the product is not faulty and ready to be delivered and if it meets the user's requirements. This test is for insecurity that does not require alteration or change. In this test, continuous experiments are led to test the exactness of the framework and how it reacts best to client needs.

**5.2.3. The Black box Testing**

Black box testing is a kind of test where the presentation of a framework will be tested without seeing the internal structure and function. It contains tests where we have no idea how it works internally. The test case is performed by feeding the data to the model and evaluating the results.

## 5.2.4. The White Box Testing

White box testing is a kind of software that is used to examines an application's internal workings. The term "white" refers to a see-through box concept in which the contents of the box might be seen. The test is conducted by supplying inputs and observing the outcomes of those inputs.

**5.2.5.** **Integration Testing**

Different parts are initially connected and then tested in integration testing. The major goal of this form of testing is to see if the system functions properly when communicating with other modules. It is mostly used to identify whether or not there are any defects or issues between interconnected systems. System Integration testing can be done either pre or post System Testing.

**5.2.6.** **Acceptance Testing/User Testing**

User acceptance testing determines whether or not the system meets the customer's expectations. This phase is focused on how the user perceives things. All we do is make sure the product is ready to ship or if it needs some tweaking or finishing. Real-time test cases are used in this phase to examine the system's capacity and how it responds to the user.

**5.3. Test Cases**

Test cases are the document that comprises a bunch of various test information, like preconditions, anticipated outcomes, and post conditions for a solitary test situation to test the system functionality compliance with a certain requirement. The main focus of test cases are Functional and non-functional requirements. In order to test different models, we have conducted following test cases on the developed system.

**5.3.1.** **User Model Test Case 1**

If the user may choose an image file from the browser directory to be shown on the webpage, the user model will describe.

Table 5.1: Test Case1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **GENERAL INFORMATION** | | | | | | | | | | | | |
| Test Stage |  |  | Testing the Units  Testing the System Performance | | |  | Testing the System Integration | |  | | Testing the System Integration |  |  |
| **Test Date:** | 05/12/2021 | | | | | | **System Date, if applicable:** | | | 05/12/2021 | | | |
| **Tester:** | Junaid khan | | | | | | **Test Case Number:** | | | TC:1 | | | |
| **Test Case**  **Description:** | User selecting image, clicking on the “Upload Image” button | | | | | | | | | | | | |
| **Results:** |  | Pass | |  | **Incident Number, if applicable:** | | | System Testing | | | | | |
|  | **INTRODUCTION** | | | | | | | | | | | | |
| **Requirement(s) to be tested:** | User must have image in jpeg format to upload. | | | | | | | | | | | | |
| **Roles and**  **Responsibilities:** | The user should click on the choose button and select the desire image. | | | | | | | | | | | | |
| **Stop**  **Procedures:** | User after selecting the image of choice tap on the “Open button” so the image can be selected. | | | | | | | | | | | | |
| **Stop**  **Procedures:** | This Procedure will be stop after user selects the image. | | | | | | | | | | | | |
|  | **ENVIRONMENTAL NEEDS** | | | | | | | | | | | | |
| **Hardware:** | Computer/Laptop, Intel core i7 processor or latest 8-GB RAM or more,500GB hard disk, good internet connection | | | | | | | | | | | | |
| **Software:** | Simple HTML, bootstrap, flask | | | | | | | | | | | | |
|  | **TEST** | | | | | | | | | | | | |
| **Procedural**  **Steps:** | First user has to open webpage on the local host and click on the button “choose file” and then select a jpeg image from the browser window and then click on the Open button. | | | | | | | | | | | | |
| **Expected**  **Results of Case:** | After selecting the image user can see their selected image on the web page. | | | | | | | | | | | | |
|  | **ACTUAL RESULTS** | | | | | | | | | | | | |
| **Output**  **Specifications:** | Selected image will be displayed on webpage. | | | | | | | | | | | | |

**5.3.2.** **Test Case 2: Flask model test**

In the test we will ensure that the image is successfully upload to the static folder using flask, and will test whether the Flask is accurately acting as bridge between front-end and the machine learning model.

The test checks the flask module is successfully transfer the image to machine learning model and after model generate the description will it be able to send back the generated description to the front end.

Table 5.2: Test Case2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **GENERAL INFORMATION** | | | | | | | | | | | | | |
| Test Stage |  |  | Testing the Units  Testing the System Performance | | |  | Testing the System Integration | |  | Testing the System Integration | |  |  | |
| **Test Date:** | 05/12/2021 | | | | | **System Date, if applicable:** | | | | | 05/12/2021 | | | |
| **Tester:** | Atiqa Javed | | | | | **Test Case Number:** | | | | | TC:2 | | | |
| **Test Case**  **Description:** | Ensuring the image successfully uploaded on to the desire folder using flask, and the flask creates an image path and transfer that image path to the machine learning model. | | | | | | | | | | | | | |
| **Results:** |  | Pass | |  | **Incident Number, if applicable:** | | | System Testing | | | | | |
|  | **INTRODUCTION** | | | | | | | | | | | | | |
| **Requirement(s) to be tested:** | Image upload on folder, passing image path to the machine learning model | | | | | | | | | | | | | |
| **Roles and**  **Responsibilities:** | We ensure the flask is getting the image from front end and passing the image path to machine learning model, and passing the output to front end | | | | | | | | | | | | | |
| **Stop**  **Procedures:** | This Procedure will be stop after the flask pass the output to the front-end. | | | | | | | | | | | | | |
|  | **ENVIRONMENTAL NEEDS** | | | | | | | | | | | | | |
| **Hardware:** | Computer/Laptop, Intel core i7 processor or latest 8-GB RAM or more,500GB hard disk, good internet connection | | | | | | | | | | | | | |
| **Software:** | Node.js | | | | | | | | | | | | | |
|  | **TEST** | | | | | | | | | | | | | |
| **Test Items and Features:** | First click on the “Predict” button and ensure the image is uploaded on to the static folder, secondly the flask should pass the image to the machine learning model and return the response to the front end. | | | | | | | | | | | | | |
| **Input**  **Specifications:** | Clicking on to the “Predict button”. | | | | | | | | | | | | | |
| **Expected**  **Results of Case:** | The model will be able to upload the image and get the response from machine learning model. | | | | | | | | | | | | | |
|  | **ACTUAL RESULTS** | | | | | | | | | | | | | |
| **Output**  **Specifications:** | The flask is acting as bridge between html page and machine learning model. | | | | | | | | | | | | | |

**5.3.3.** **Test Case 3: Machine learning model test**

This test is conducted to ensure the accuracy of the model that we trained on coco dataset.

Table 5.3: Test Case3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **GENERAL INFORMATION** | | | | | | | | | | | | | |
| Test Stage |  |  | Testing the Units  Testing the Systems | | |  | Testing the System Integration | | |  | Testing the system | |  |  |
| **Test Date:** | 05/12/2021 | | | | | | | **System Date, if applicable:** | | | | 05/12/2021 | | |
| **Tester:** | Maryam | | | | | | | **Test Case Number:** | | | | TC:3 | | |
| **Test Case**  **Description:** | User can click on the “Get Image Caption” button and the model should generate a description for the image, and the description should be nearly accurate. | | | | | | | | | | | | | |
| **Results:** |  | Pass | |  | **Incident Number, if applicable:** | | | | System Testing | | | | | |
|  | **INTRODUCTION** | | | | | | | | | | | | | |
| **Requirement(s) to be tested:** | A description of the image that is nearly accurate | | | | | | | | | | | | | |
| **Roles and**  **Responsibility** | We ensure that the model be generating accurate results. | | | | | | | | | | | | | |
| **Set Up**  **Procedures:** | The tester should click on the “Predict Caption” button and wait for the model to generate the description of the image | | | | | | | | | | | | | |
| **Stop**  **Procedures:** | This Procedure will be ended when the RNN model generates the description | | | | | | | | | | | | | |
|  | **ENVIRONMENTAL NEEDS** | | | | | | | | | | | | | |
| **Hardware:** | Computer or Laptop, Intel core i7 processor or later 8-GB RAM or more,500GB hard disk | | | | | | | | | | | | | |
| **Software:** | NumPy, Pytorch, Python-3, Pillow, CNN and RNN model | | | | | | | | | | | | | |
|  | **TEST** | | | | | | | | | | | | | |
| **Test Items and Features:** | First click on the “Predict caption” button and wait for the model to generate the image description. | | | | | | | | | | | | | |
| **Input**  **Specifications:** | Click on the “Predict caption button” | | | | | | | | | | | | | |
| **Procedural Steps:** | By clicking on the “Predict caption” button user can get the image description | | | | | | | | | | | | | |
| **Expected**  **Results of Case:** | The model should be able to generate nearly accurate description of the images | | | | | | | | | | | | | |
|  | **ACTUAL RESULTS** | | | | | | | | | | | | | |
| **Output**  **Specifications:** | The model should be able to create a description of the photos that is almost correct. | | | | | | | | | | | | | |

**5.4. Non-Functional Requirements**

Non-functional are those needs that are as vital as functional needs because they determine how well the system is working in the hands of end users. All non-functional criteria are added by the developers without the request of the user.

Table 5.4: List of Common Non-Functional Requirements

|  |  |
| --- | --- |
| **Property** | **Measure** |
| Speed | Responsive time is quick in 2 to 5 second the model will generate the description of the image. |
| Size | 370 mb |
| Ease of use | Around 19-22 hours of data training time |
| Reliability | The reliability of the system is quite good. The webpage is on a local host so the system is quite responsive and reliable. |
| Probability | It is well responsive web page with a flask that can be used in any laptop/computer meeting the hardware requirements. |
| System Availably | All the features of the project work 24 hours without any issue |
| System Testability | Different testing techniques has been to make the system reliable and accurate |

Chapter 6

CONCLUSIONS

The epilogue, limitations, and future work that it necessitates are discussed in this chapter. The initial part discusses the conclusion of the project. In this section, we examine how this machine learning works and what essential features were achieved throughout the development.

**6.1. Conclusion**

An Image description generator is the focal point if the project. In recent years there has been a lot of work to be done to solve the task of interpretation, focusing on the algorithmic idea of various consideration calculations, and attempting to sum up how these algorithms work in this report, in the report we have included all aspects proposed lately to address the assignment of interpretation, and to focus on the algorithmic context of various attention-algorithms. A comprehensive summary of the data and test parameters used in practice is provided.

In addition to the image description can be used to retrieve image, video captions, and video motion detection, and in addition to the wide range of existing image definition programs, test results show that the system still needs a lot of improvements and enhancements to function properly. The challenge is mainly with the three problems listed below: First, how Ai will learn to build sentences in a natural language that is as complete and understandable as one can understand; second, check the grammatical value of the model; third, make the definition of semantics as clear and steady with the components of the picture as possible. We suggest four possible changes for future work.

An image can have many details in it. Instead of simply being able to describe the image the model should be able to produce multiple descriptions of many key elements.

Multilingualism can be defined using corpora descriptive languages, so a picture caption framework should be developed that can handle multiple languages.

Analyzing the effectiveness of indigenous language production systems can be a daunting task. There are many ways but the most ideal approach to discover the precision of the writings created is for language practitioners to scrutinize themselves, but this is difficult to do. Measurement metrics should be improved to make them more compliant with human expert assessments to maximize program performance.

In order to improve the efficiency of the system the speed of training must be measured and the output of the model sentences must be precise.

**6.2. Limitations and Future Work**

In the future, we'd like to look at methods for generating multiple sentences of different material. Combining interesting area detection and image description is one choice. Future study will include using RCNNs models to further improve the performance in generating descriptions of images and determining the efficiency of FRMM models that can be used for other multi-reason applications for instance video depiction, discourse to message transformation, and language interpretation.

**APPENDIX – I**

**Acronyms and Abbreviation:**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **NumPy** | NumPy is a python library through which you can modify the pixel values of an image. |
| **TensorFlow** | It's a machine learning library that may be used for things like computer vision and natural language processing. |
| **RNN** | Recurrent Neural Networks (RNN) are a type of Machine learning algorithm that, in deep learning, can process a sequence of inputs and save their state while processing the following sequence of inputs. |
| **CNN** | A convolutional neural network (CNN) is a pretrained model of Kera’s that is used to detect objects in an image.it has 50 deep layers and can classify images and detect objects such as car, pencil, dog, plane etc. |
| **Flask** | Flask is a well-known Python web framework, or a Python library for building web applications. |
| **Python-3** | A programming language that is thought to be the best for creating AI models. |
| **Keras** | Keras is a Python interface to an open-source software library for artificial neural networks. TensorFlow has a user interface called Keras. |
| **OpenCV-Python** | The OpenCV-Python library is used for reading images from a dataset.In CV2, the imread() method loads an image from a file. |
| **Tqdm** | It is used to show a smart progress meter for a loop. |
| **Resnet50** | ResNet-50 is a convolutional neural network that is used for classifying and detecting the objects in an image. |

# REFERENCES

# [1] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In arXiv:1502.03167, 2015.

# [2] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), 1997.

# [3] “Dataset containing 8000 images with descriptions” Published: 18-02-2019, Contributor: Saurabh Shinde.

[Online] Available: <https://www.kaggle.com/srbhshinde/flickr8k-sau> [Accessed: 03 August ,2020]

# [4] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), 1997.ICML,2014.

# [5] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. NIPS, 2014.

# [6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In ICML, 2014

# [7] “Dataset containing 80 thousand images with descriptions” Published: 12-05-2014, Contributor: Genevieve Patterson.

[Online] Available: <https://cocodataset.org/#download> [Accessed: 06 November ,2020]