Software Requirements Specification

for

Image Captioning

Version 1.0 approved

Prepared by:

NAMES	**CMS IDS**
Mir Waqas Ahmed	41157
Sheikh Ismail	41084
Ahsan Iqbal	42233

BUITEMS

24-02-2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Purpose

It is a very challenging task to automatically describe image contents using human-readable sentences, but it could have many benefits. This task is harder than the common image classification or object recognition, and it gained a major focus in computer vision [1]. Also, a textual description must capture not only the objects contained in an image, but it must also describe the relationship between these objects in a natural language which means that with image processing we also need a language model to process the descriptions [4].

There are several ways to provide an automatic description of image contents. One of the popular ways is to use the human-annotated training set which describes the image contents [5]. The motivation of our project came from recent research in Machine Translation where a sentence is translated from one language to another. It was done by translating words individually, aligning the words, and ordering them but recently it has been proven that it can be achieved more easily using Recurrent Neural Networks (RNNs) [3] [2] [6]. There are two phases, Encoder, and Decoder. In the first phase, RNN reads the input and transforms it into a fixed-length vector representation, which is then used in the second hidden phase called decoder RNN that generates the actual output. In our model, we will replace the RNN encoder with a deep convolutional neural network (CNN) as we are working with images. Because CNNs can produce a very good representation of the input image by embedding it to a fixed-length vector, which can be used for a variety of computer vision tasks [6]. We will train it for an image classification task and use the last hidden layer as an input to the RNN decoder that will generate descriptions. We will be training a model called Image Caption Generator. This model combines Computer Vision with Natural Language Processing. Initially, we will be using the Flickr8k dataset to train our model, then if we had available hardware resources, we will train it on Flickr30k or MSCOCO datasets. All these datasets are publicly available. For evaluation, we will be using the BLEU score.

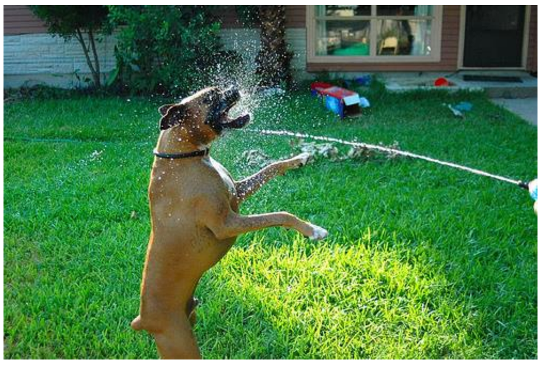
Figure 1: A brown dog is sprayed with water. (Flickr8k)

1.2 Document Conventions

This document follows the “Time New Roman” font. Additionally, the main headings of every chapter having a font size “18” and font-weight are “**BOLD**” whereas the font size of sub-headings is“14” and **“BOLD”** as well. The texts within headings or sub-headings having a font size “12”. The sub-headings within headings will be indented and if the need arises, bullets and numbering will be also used.

1.3 Intended Audience and Reading Suggestions

This SRS will facilitate both developers and end users. For developers, going through this document, it will help them find out the technical stuff and the logical things that make up the software. Whereas, a random person from reading the document will get familiar with this software, its basic use, its purpose and functionality of the software. Each chapter of this document sheds light on these matters.

Chapter 1 will provide a detailed introduction of the project. Chapter 2 will provide the overall description related to the project. Chapter 3 will provide the external interface requirements of the project. Chapter 4 will provide in-depth details about the system feature. Chapter 5 will be about the non-functional requirements of this project and the last chapter 6 is consistent with the diagrams for a better understanding of the logical design of the system. Let suppose if the supervisor or coordinator wants to know how the database of the project will work they can simply jump to the chapter 6 to get information about that particular thing.

1.4 Product Scope

Image Captioning can be beneficial in many ways, for instance on ***Social Media*** sites it can be used to suggest automatic descriptions of the image to the user. It can also be implemented along with some ***Surveillance*** Camera software to capture the frames from security footage, generate descriptions, and pass them through some language processing models to detect suspicious behavior. In the case of ***Visually Impaired*** people, when using a screen reader in a browser when it reaches any image, it reads the alternative text of that image, aka ‘alt’ attribute in HTML img tag <img alt=`company_logo` src=`logo.jpg`>. Instead of a title, we can place the complete description generated from our model to help the visually impaired better understand the content on the web. Image captioning can also be beneficial in ***File Indexing,*** when all the images all tagged with their captions, the user can just search, for instance, *beach*, and all of his pictures taken on the sea view will be filtered out and displayed. Last but not the least, in ***Robotics***, captions can be used as a representation of the environment and surroundings using frames of the images captured through the camera on its head, then those results can be passed to any NLP model to interpret the meaning and help the Robotic algorithm take appropriate actions based on his surroundings.

1.5 References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014. 1, 2

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, 2014. 1, 2, 3

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3156-3164, doi: 10.1109/CVPR.2015.7298935.

[5] Zia, Usman & Riaz, M. & Ghafoor, Abdul & Syed, Sohaib. (2020). Topic sensitive image descriptions. Neural Computing and Applications. 32. 10.1007/s00521-019-04587-x.

[6] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In NIPS, 2014. 1, 2, 3

Chapter 2: Overall Description

2.1 Product Perspective

Security is something that can't be neglected. The main intention behind building up this model is to improve the security of the people and the organization (BUITEMS) as well. The web-based application would be set up and installed in the main entrance of the organization to detect the prohibited things not allowed in the university. One question arises, how does this model work? Let us consider a scenario where a person comes to the entrance gate, with the help of our Machine learning model, it would first detect the objects in the image that the camera captures using a CNN model like (cloths, things that they have been carrying, etc.) and then generates a concise caption with the help of RNN regarding the person's action. For example, if a person comes to the gate then our model firstly, captures its picture and generates a caption that is --A person with a white shirt and jeans enter the university carrying a grey bag.

2.2 Product Functions

The main functions of this systems are:

1. Capture the image.
2. Identify objects in the image.
3. Generate the concise caption of the image

2.3 User Classes and Characteristics

The following departments will use this project:

2.3.1 Surveillance Team

This system will be installed in the surveillance operating room. This model can be integrated with the live streaming of the CCTV. Frames from the video stream would be used as an input for our model. From these frames/images our model will detect objects and the relationships they keep with each other, from them the captions will be generated. In case of any suspicious activity detected or any captions depicting any violation a notification would be generated.

2.3.2 Hostel Warden:

Most of the universities have offered facilities to the students from the outside of the city to stay in the university by providing rooms. As we know, restrictions have to be imposed on the students to maintain discipline in the university. Can be used by the hostel management to identify things that are not allowed in the hostel premises. For example, if someone got captured smoking cigarettes in the hostel premises, this model would alter the hostel management via tha caption generated.

2.4 Operating Environment

- The hardware environment needed for this project are:
  1. Required 16GB(ram), Recommended 32GB(ram).
  2. CPU i7.
  3. 200 GB of free disk space..
  4. GPU:RTX 2080 or better.
- The software environment needed for this project are:
  1. Web Browser (any).
  2. Compatible with MACOS, UBUNTU, Windows operating systems.
  3. Window 7 or above and MACOSX 10.8 or above.
  4. Anaconda
  5. Python 3
  6. Tensor flow and Keras

2.5 Design and Implementation Constraints

Accuracy of the captions generated depends a great deal on the size of dataset used and the number of epochs model has been trained on. CNN to perform any image processing techniques requires high computing power. When using a Flickr8K dataset a single epoch took approximately 2 hours. To improve accuracy and decrease loss atleast 30 epochs are required. It is recommended to use 32-64 GB of memory to train these inception V3 models on larger datasets i.e. Flickr30K or MSCOCO which can’t be done on personal computers. The model would be integrated with a web interface for end users to use it.

2.6 User Documentation

The system would be user-friendly, a module of how to use will be delivered containing step to step instructions for users' use. Frequently Asked Question (FAQ) and a Q&A module will be integrated to help used with their queries.

2.7 Assumptions and Dependencies

Due to the unavailability of hardware resources we started using cloud computing resources for commercial use. Draw backs of using these commercial based cloud computing platforms is that when the session expires all variables, the temporary files stored and any progress at all will be lost. So we have to start over from the beginning again. However, the premium packages do allow us to have a persistent storage in our cloud computing environment. After many evaluations, we came to the conclusion that inception V3 is the best among other models used for image captioning. Inception V3 is the model developed by google but in case of change of policies or un accessibility of inception V3 we can always switch back to VGG-16 or Mobile Net Model.

Chapter 3: External Interface Requirements

3.1 User Interfaces

This model image the caption is a web-based project. The web is composed of 2 pages. The main page has a navigation-bar at the top of the page after that, we will build a section where we have the title of the project, and then we will build another section where we asked users to upload the image that they want to know. Last not least, at the bottom, we have a footer where we will add the project logo at the left bottom side, and on the right bottom side, we will add some social media icons to access different machine learning model pages. Besides this, we have another page where we will add the detailed descriptions and you tube tutorial if necessary, from a random user can get familiar with the project.

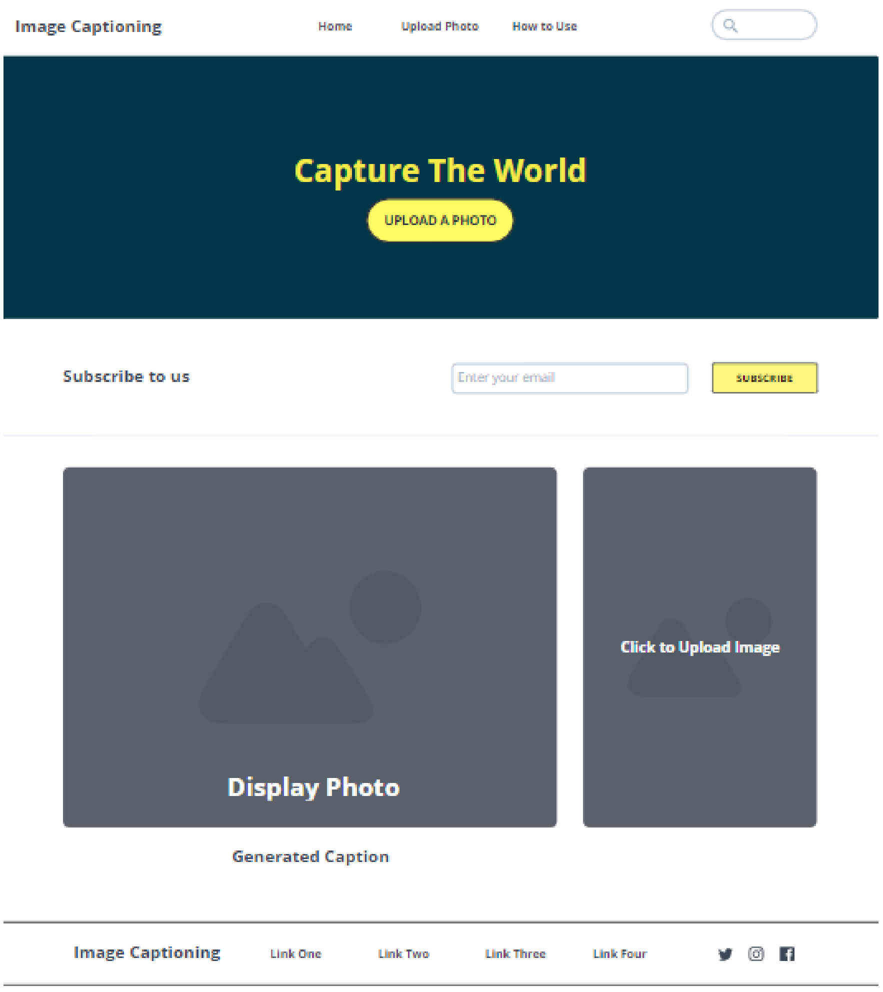

Figure 2: Landing Page UI MockUp

3.2 Hardware Interfaces

The hardware interface needed for this project are:

- Minimum required 16GB ram and recommended 32GB ram.
- CPU core i7.
- 200 GB of free space on drive.
- GPU: RTX 2080 or better.

3.3 Software Interfaces

- For UI designing, HTML has been used for skeleton, refined and designed using CSS and Bootstrap.
- A landing page designed using above mentioned components where an end user can upload images.
- Model will receive uploaded images as a parameter and start processing them.
- Model use has been trained on several datasets found on Kaggle i.e. Flickr 8K, Flickr 16K and MSCOCO.
- Generated caption will be saved in a sql or nonsql database, as well as rendered on the web page for the user to see, both these tasks will be done simultaneously.

3.4 Communications Interfaces

After building up a model, the next thing we will do is to upload the finalized model to the server and the user interacts with the web-based UI with the help of the browser. The only thing a user has to ensure that they have a good internet connection because a user needs to upload images on the web in order, to find the caption of the image. The user at least has maximum internet speed which is 2MBPS. the protocol which we will use is FTTP for security purpose.

Chapter 4: System Features

In this chapter, we will discuss the project system feature in detail. The system features of our project are:

1. Capture the image.
2. Identify objects in the image.
3. Generate the concise caption of the image.

4.1 Image Upload:

4.1.1 Description and Priority

In order, to find the caption of the image the user needs to upload the image, and this feature is intended for the end user. On landing page of the website there would be a pane designed to upload any images (any format). File uploader will upload the given files and send it to the model as an input. A prediction function will use the model and generate caption for the image, the function will then return the caption and it would be rendered for the end user on the screen. This feature has high priority because if they do not upload any image then our model will not have an input to work-on.

4.1.2 Stimulus/Response Sequences

To find the upload the image, the user needs to perform serval steps which are first,

1. Users should have internet service at least they have a minimum of 2MBPS internet speed otherwise, they wouldn't able to use the web.
2. When users connected to the internet service, they have to access the website of the project.
3. Once they reach out to the website, they will see the button where they have to upload the image. If they have a bad internet connection it will take several minutes to upload the image otherwise they will see an error to upload the image again.
4. There is a situation that comes when the user does not upload the correct format of the picture, so they will get an error to upload the correct format of the picture

4.1.3 Functional Requirements

REQ-1: user should have any medium (like mobile, laptop etc.).

REQ-2: having good internet service.

REQ-3: user should have an image**.**

REQ-4: the file format must be correct**.**

4.2 Identify objects in the image:

4.2.1 Description and Priority:

Once a user has uploaded the picture successfully, the next step is to identify the most prominent objects in the picture. This has to be done with the help of the machine learning technique by implementing a conventional neural network(CNN) concept. The priority of this feature is high because if we could not find out or identify the correct objects in the image then our model would not able to create accurate caption of the image.

4.2.2 Stimulus/Response Sequences:

- As we know, we upload our model on the server. If the server is not active, then our model would not get the image.
- when the model is active, then the model will get the image. In response the model will start extracting the most prominent objects from the image and these features would help the model to identify object presents in the image.

4.2.3 Functional Requirements:

REQ-1: user should have any medium (like mobile, laptop etc.).

REQ-2: having good internet service.

REQ-3: user should have an image**.**

REQ-4: the server must be active.

REQ-5: the model must be well-trained.

4.3 Generate the concise caption:

4.3.1 Description and Priority:

The last feature is to create a relationship between objects in the form of a concise caption. When the prominent objects were identified with the help of a recurrent neural network(RNN) which used natural language processing to create captions in the English language. The priority of this feature is also high because our main concern is to build a concise caption of the images.

4.3.2 Stimulus/Response Sequences:

Once image has been uploaded successfully and the prominent objects were found in the image then a concise and accurate caption of the image will be created. There is certain chance the caption which was created not satisfied the context because might be there is lack of accuracy is found in the model.

4.3.3 Functional Requirements:

REQ-1: user should have any medium (like mobile, laptop etc.).

REQ-2: having good internet service.

REQ-3: user should have an image**.**

REQ-4: the server must be active.

REQ-5: the model must be well-trained.

Chapter 5: Other Nonfunctional Requirements

5.1 Performance Requirements

It is evident that a machine learning model takes a fair bit of time when dealing with images and text together. Volume of dataset has a direct relationship with stress on the processing units. So obviously using a bigger dataset would need high performance processing units (GPU and standard processor) and would also need more time to extract features from the images, clean text and train the model on the dataset. Size of dataset can be varied depending on the processing power available. If low end processors were used it would compromise the performance a great deal in training as well as when an end user will upload a picture, because the processing has to be done on the server side these low performing processors will take an awkward amount of time to return the results and the user might feel it has stopped responding.
Other than this, even though there is no need for heavy processing on users' side they still have to upload an image to our server to get the caption back. For that the user would need to have a moderately stable internet connection to upload images easily.

5.2 Safety Requirements

Any harm done to the system will not jeopardize its integrity. User uploaded data will be saved in a secure folder as well as the pixel data of every image will be saved on a remote database. In case of a system failure data can be recovered from one storage or another. However, in case of server failure end users will not be able to access the web page until the system is fixed. Due to this project being a final year, therefore the prototype made will only be uploaded on a single server but it is scalable so more servers can be added anytime.

5.3 Security Requirements

No user authentication will be required as of now. Data uploaded by the users to the server would be strictly in the premises of the developers only, data uploaded will not be used for any purpose other than the sole reason of training the model on these images and making the model more accurate as more and more images get uploaded.

5.4 Software Quality Attributes

Our product offers the following software quality attributes:

Adaptability

Correctness

Maintainability

Portability

Reusability

Usability

5.4.1 Maintainability

Code has comments explaining what each segment of the code does. In case of any errors it will be easy to identify where the problem lies. Each function has error checking associated, good error checking will keep the program from exploding and would make debugging a lot faster. Code written can be easily tested, the sooner an error is found, the cheaper it is to fix it.

5.4.2 Portability

System designed is completely independent of the platform. Any browser can be used to access the web. Similarly, any computer with normal requirements will be able to run the server and render the page on web for it to make accessible for the end users on the internet.

5.4.3 Correctness

The model can be embedded in any complex system and it would work just fine. The overall code does not concern the model. It just has to be properly trained to get accuracy needed to be used in a certain type of system. All the model need is an input to process and give an output accordingly.

5.4.4 Adaptability

Machine learning techniques are highly adaptable. The model is completely independent and does not depend on rest of the code. Hence, the model can be used independently in any other application per requirement, or can adapt to changes made in the current system.

5.4.5 Reusability

Code segments can be executed on their own. Functions can be used elsewhere for similar purposes.

5.4.6 Usability

The application is very easy to use; it is almost intuitive to a user to know what to do. With this there are step to step instructions that a user can follow if they do not get the hold from looking at the layout. An eye catching upload button to upload users file and pass it to our function as an input. The resulted caption from the image then would be shown up in a pop-up including the picture it has been captioned for. Soothing colors are used to make it durable for the eyes, white space and standard layout conventions are followed to make it user friendly. Anyone with a 2-day experience on internet would not have to struggle to use.

5.5 Business Rules

- User must have access to internet to surf our site.
- User should have a picture to upload in order to use the facility.
- File format have to match in order for our model to process the image.
- If the uploaded file were invalid server would return a pop up error asking user to upload with file with the specified file format.

Chapter 6: Logical Designs of System

6.1 Use Case Diagrams

6.1.1 Use Case Diagram – Image Upload for Captioning

User can interact with system using the web interface, user can upload the image via web UI and server will process image and generate a caption and return to user via web UI.

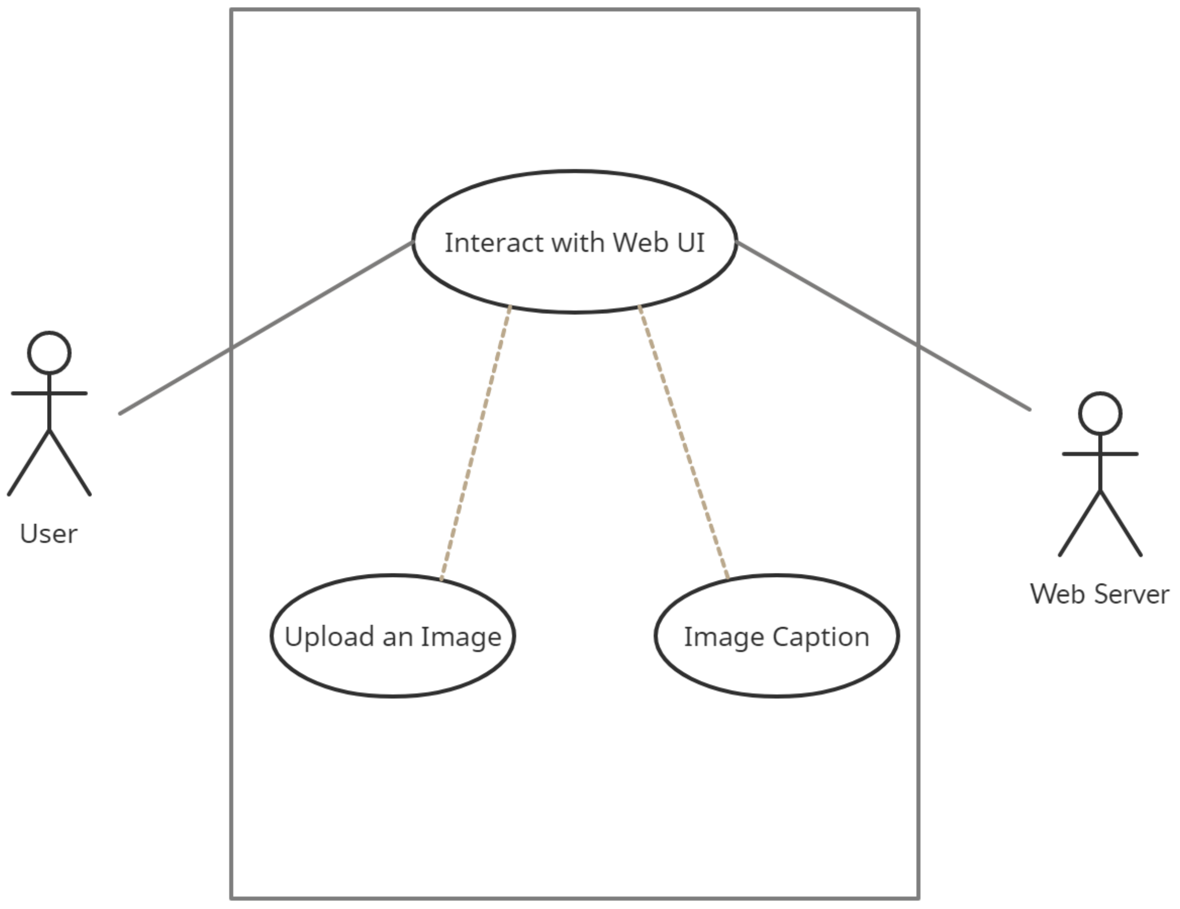

Figure 3: Use case of uploading an image

6.2 Descriptive Use Cases

Table 1: Descriptive Use Case Table

ID:	**Use Case 1**
Title:	Image Upload for Captioning
Description:	*User can interact with system using the web interface, user can upload the image via web UI and server will process image and generate a caption and return to user via web UI.*
Primary Actor:	*End User*
Preconditions:	*User must have an active internet connection*
Post conditions:	*User selected a valid image file which uploaded successfully*
Main Success Scenario:	1. *Open the Website/Web UI* 2. *Click "choose a photo" button* 3. *Select a valid image file and upload it* 4. *Get captions for that image*
Alternative Scenario:	• *User selected an invalid file which failed to upload* • *System is unable to describe image correctly*
Frequency of Use:	*It will be used whenever a user opens the web app and uploads file*

6.3 Activity Diagram of System

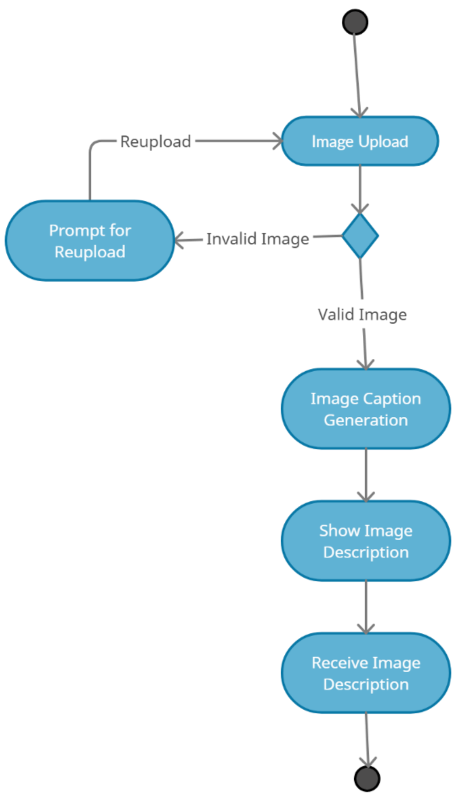

Figure 4: Activity Diagram

6.4 Data Flow Diagrams

6.4.1 Context Level Diagram

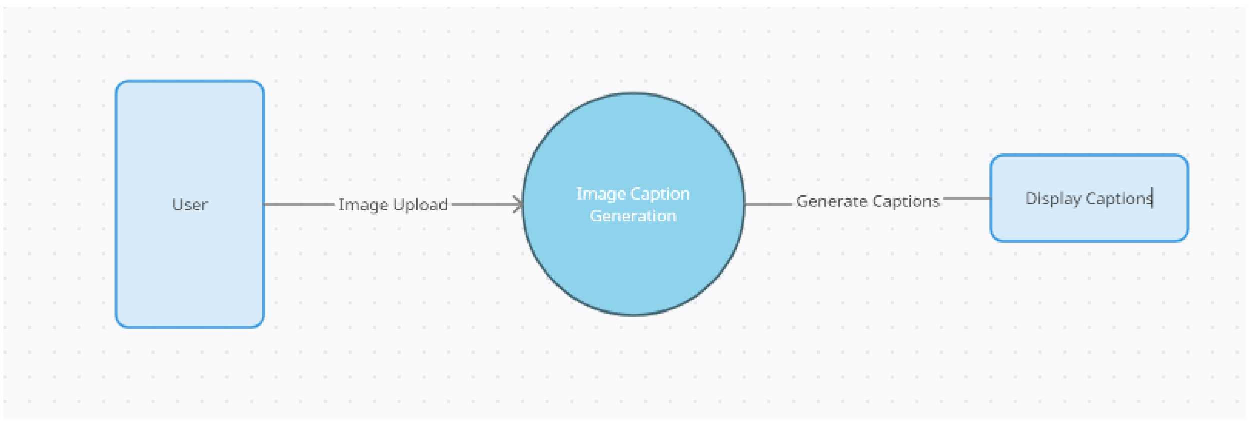

Figure 5: Level 0 DFD

6.4.2 Level 1 DFDs

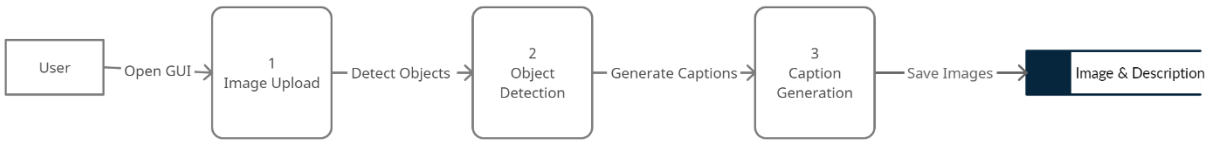

Figure 6: Level 1 DFD

6.5 Sequence Diagram

6.5.1 Sequence Diagram – Image Upload

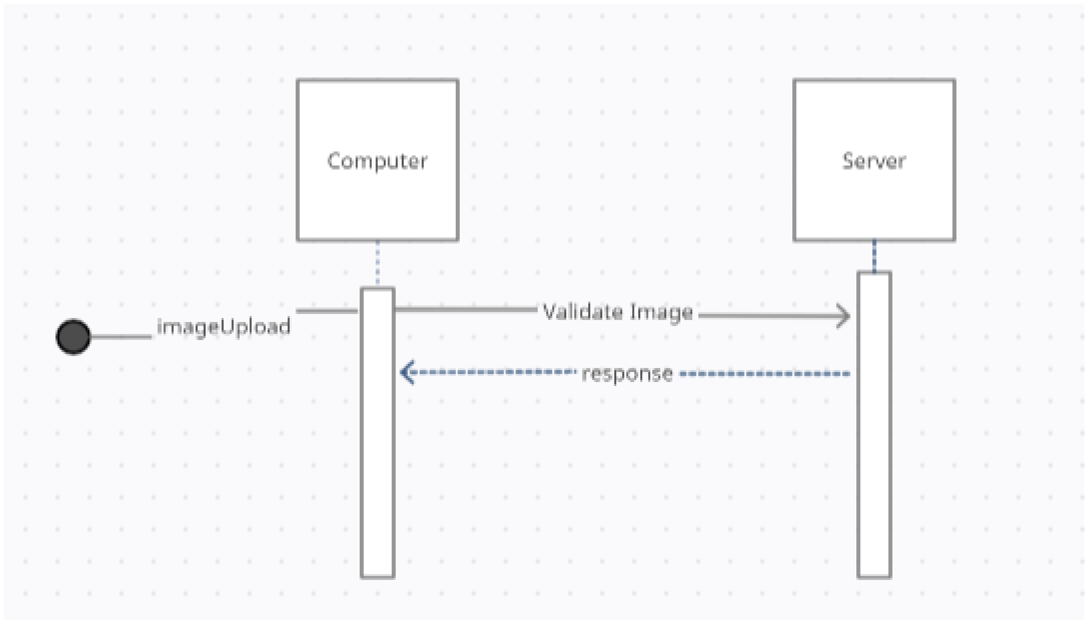

Figure 7: Image Upload Sequence Diagram

6.5.2 Sequence Diagram – Object Detection

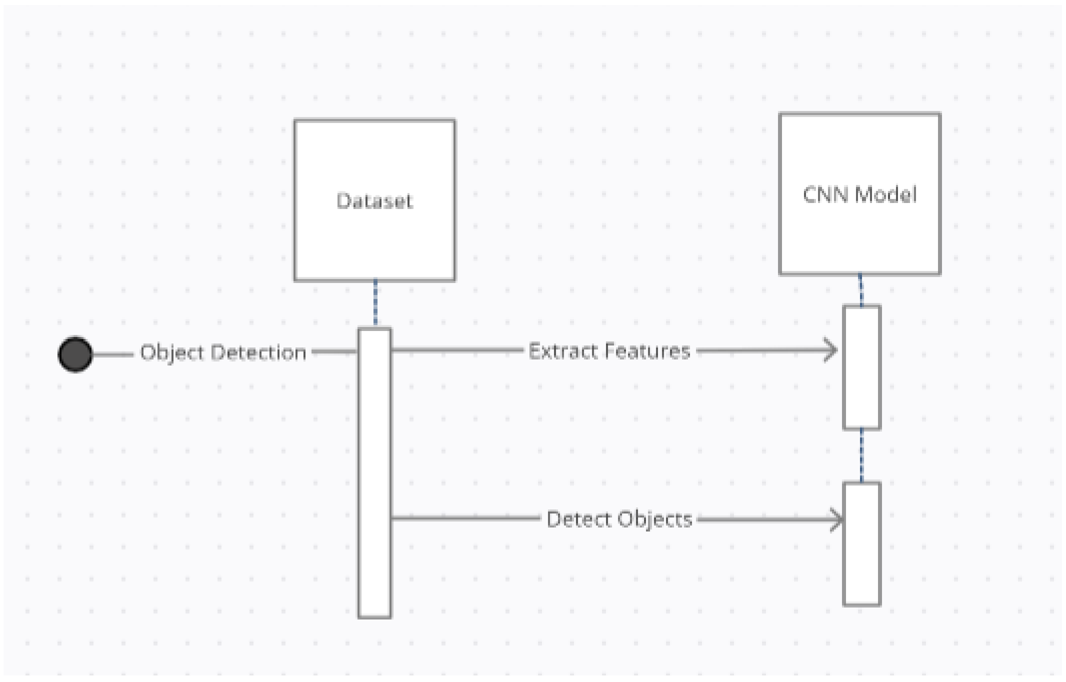

Figure 8: Object Detection Sequence Diagram

6.5.3 Sequence Diagram – Generate Captions

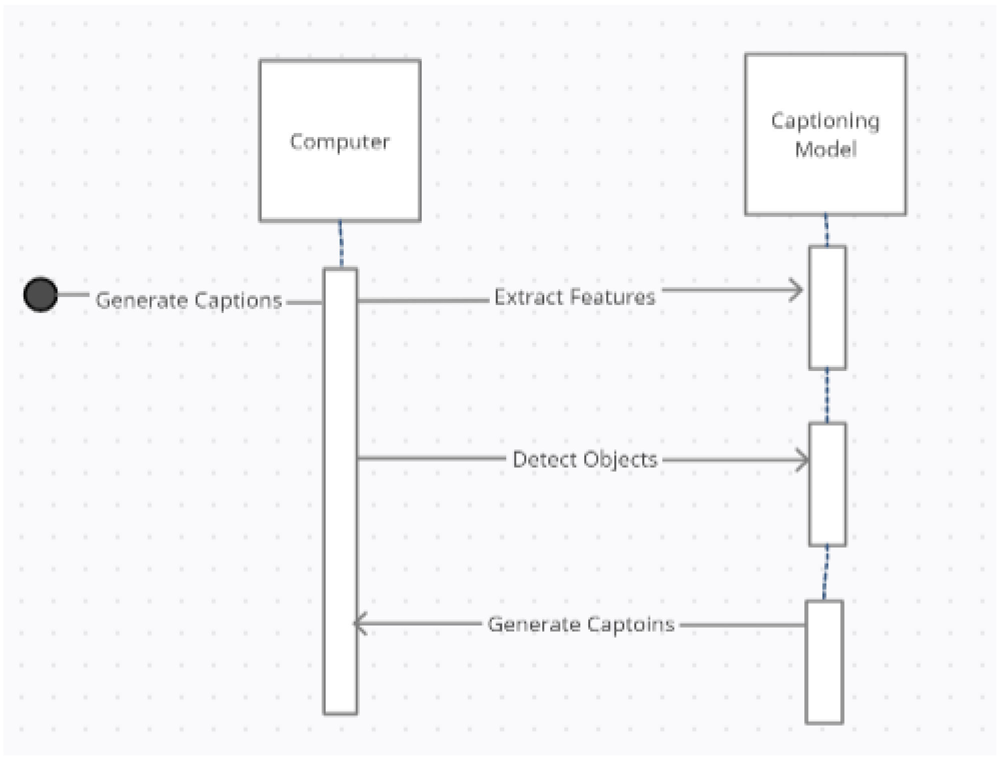

Figure 9: Caption Generation Sequence Diagram

6.6 Entity Relationship Diagram

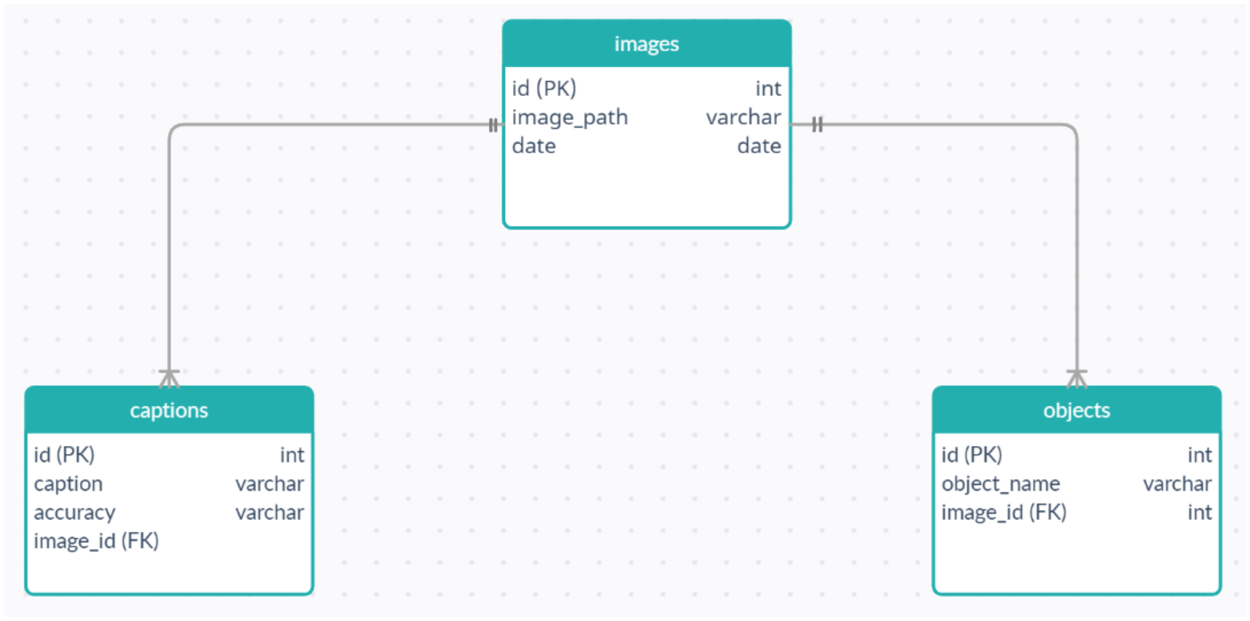

Figure 10: Entity-Relationship Diagram