**University of Mumbai**

**Text based Image Retrieval**

Submitted at the end of semester VI in partial fulfillment of requirements

**Of Bachelors in Technology in Computer Engineering**

by

**Aakash Saroop**

**1911001**

**Bhairav Narkhede**

**1911003**

**Nidhi Bhanushali**

**1911004**

Guide

**Ms Sheetal Pereira**

**Mrs Mansi Kambli**



**Department of Computer Engineering**

**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**

**Batch 2019 -2022**

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**Certificate**

This is to certify that the MINIPROJECT report entitled **Text based Image Retrieval**  submitted by Aakash Saroop, Bhairav Narkhede, Nidhi Bhanushali, at the end of semester VI of TY B. Tech are bona fide record for partial fulfillment of requirements for the degree of Bachelors in Computer Engineering of University of Mumbai

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

Guide Head of the Department

Date:

Place: Mumbai-77

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**Certificate of Approval of Examiners**

We certify that this MiniProject report entitled **Text based Image Retrieval** is bonafide record of Mini project work done by Aakash Saroop, Bhairav Narkhede, Nidhi Bhanushali during semester VI.

This Mini project work is submitted at the end of semester IV in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

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

Internal Examiner 1

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

Internal Examiner 2

Date:

Place: Mumbai-77

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**DECLARATION**

We declare that this written report submission represents the work done based on our and / or others’ ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

|  |  |
| --- | --- |
| IMG_6672.jpg  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  1911001  **Roll No.** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  1911003  **Roll No.** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  1911004  **Roll No.** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Roll No.** |

**Date: 05-05-2022**

**Place: Mumbai-77**

**Abstract:**

With the exponential increase in the amount of data present in an individual’s device, it has become increasingly difficult for a user to manually retrieve data from their device. Retrieving image data has been particularly difficult because unlike textual data, the facility to search the target entity based on keywords is not available in image data. The Image Captioning Deep Learning models have been improving rapidly recently. Using the technology of these recently improved image captioning models, we have built an image retrieval system in which the user can describe the image in the form of text and find the image with its caption closely matching the query.

**Table of Content:**

**List of Figures:**

**List of Tables:**

**Nomenclatures:**

**Chapter 1 : Introduction**

**1.1 Background/Motivation**

**1.2 Problem definition**

**1.3 Scope**

**1.4 Hardware and software requirements**

**Chapter 2 : Literature Survey**

**Chapter 3 : Project Design**

**3.1 Proposed System model**

**3.2 Software Project Management Plan**

**3.3 Software Requirement Specification Document**

**3.4 Software Design Document ( All applicable UML diagrams)**

**Chapter 4 : Implementation and experimentation of one of the issues related to project work/ topic**

**4.1 Proposed system model implementation**

**4.2 Inclusion of Any additional details as suggested by Project Guide/during progress seminar**

**4.3 Software Testing  (Software testing reports at various levels)**

**4.4 Experimental results and its analysis**

**Chapter 5 : Conclusion and future work**

**5.1 Conclusion**

**5.2** **Scope for future work**

**Bibliography**

**Acknowledgements**

**List of figures:**

Model architecture

Use case diagram

UI screenshots

Sample Caption model outputs

**List of tables:**

Literature survey

Software project management plan

Test cases

**Nomenclatures:**

Image-retrieval

Captioning

Caption generation model

NLP string similarity model

Indexing

**Chapter 1:**

**Introduction**

**Introduction:**

As studied in Software Engineering, developing a successful product (software: including the code and documents) needs a systematic approach. In this experiment you will prepare the basic documents required to develop a product, a software system, a website or a mobile app to provide certain services or facilities.

**Motivation of the project:**

Most users have thousands of images stored in their smartphone's photo gallery. This makes it difficult for users to find an image which they have saved, since unlike text documents, there is no robust method to search for images. With the users shifting towards cloud and having virtually unlimited storage capacity, this problem will be compounding rapidly in the future. To solve this problem, we are building a text-based image retrieval application, which links to a user’s photo gallery and generates text captions for the images stored in it. The users can then enter the text describing the events taking place in the image they are searching for; this application finds the closest captions stored to the query text and displays the images corresponding to them.

**Problem definition:**

To solve this problem, we are building a text-based image retrieval application, which

links to a user’s photo gallery and generates text captions for the images stored in it.

The users can then enter the text describing the events taking place in the image they

are searching for; this application finds the closest captions stored to the query text and

displays the images corresponding to them.

**Scope of the project:**

This project will include linking the developed app to the photo gallery app of the user.

In addition to this two Machine Learning models will be developed: A caption

generation model, which generates caption corresponding to the stored images and an

NLP string similarity measuring model for displaying images having similar captions to

the search query. The captions will be indexed to the images and stored on the user’s

phone. The caption generation model will be deployed on a server.

This project however does not include the ability to store images for each user.

**Hardware and Software Requirements :**

Requirements for the project include:

Ram: 8 GB DDr3

CPU: Intel Pentium 4 processor or later that's SSE3 capable

Windows 7, Windows 8, Windows 8.1, Windows 10 or later

Browser: Chrome Version 92.0.4515.159, Edge Version 92.0.902.84

HDD: 4 GB free space

Modules

Pickle module for python 3

Tensorflow module for python 3

Flask for python 3

Passlib for python 3

Psycopg2 for python 3

Minimum: Android 4.0+ 3GB ram, quad core processor

Minimum: IOS 10+ 1.5GB ram, A10 processor

**Chapter 2:**

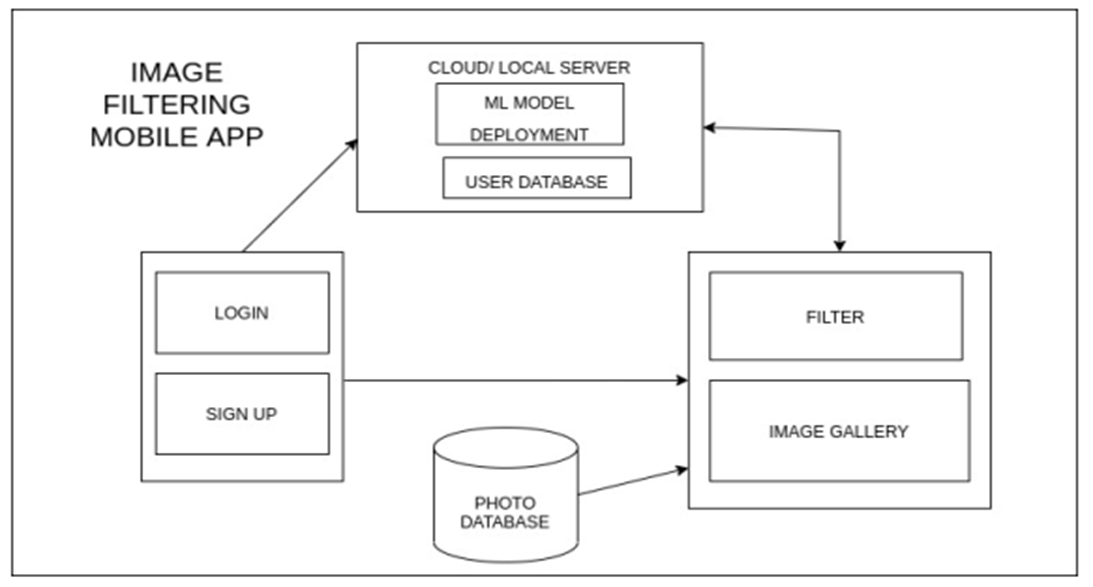
**Literature Survey**

|  |  |  |  |
| --- | --- | --- | --- |
| **Research paper Title** | **Approach/Algorithm/ Methodology** | **Findings** | **Gaps Identified.** |
| ViLBERT: Pretraining Task- Agnostic Visiolinguistic Representations for Vision-and- Language Tasks | Learns task-agnostic joint representations of image content and natural language | Seminal work in vision+ text transformer s. Many other models reviewed have cited this work. | Needs data in the form of MCQ questions and answers. Training data used comparatively small as compared to most modern DL models. |
| sentence- transformers/bert- base-nli- mean- tokens | Maps sentences & paragraphs to a 768 dimensional dense vector space and can be used for tasks like clustering or semantic search. | Gives accurate similarity between sentences even if the sentences don’t have common keywords | No methodology for finding a cutoff point beyond which the similarity is no longer relevant |
| Composing Text and Image for Image Retrieval - An Empirical  Odyssey | Studies the task of image retrieval, where the input query is specified in the form of an image plus some text that describes desired modifications to the input image. | Works by finding images similar to the input image. The difference s between the two images specified using string. | The images used in testing are that of a controlled environment. No empirical data to support its performance on images in the wild. |
| Content- Based Image Retrieval and Feature Extraction: A Comprehens ive Review | -(Review paper) | - | - |

**Chapter 3:**

**Project design**

**Proposed system model:**

****

The user will be asked to Login/ Sign up before using the app. The login details and the last time the user has synced the app are a part of user database which will be stored on the cloud. The ML model for generating captions will also be deployed on the cloud.

This app will request access to the photo gallery app of the user’s phone. The stored images will be passed through the ML model on the cloud and the corresponding captions will be stored in the user’s device.

When the user enters a query for searching an image in the database, the images

corresponding to the caption(s) closely matching the query will be retrieved for the

user.

**Software Project Management plan**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | R1 | R2 | | R3 | | Mentor | |
| **1. Requirement Gathering** |  |  | |  | |  | |
| 1.1 Interaction with customer | C | C | | C | | A | |
| 1.2 Preparing SRS | C | C | | C | | A | |
| **2. Design** |  |  | |  | |  | |
| 2.1 Preparing Block diagram | C | C | | C | | A | |
| 2.2 Writing Functional Requirements | C | C | | C | | A | |
| 2.3 Writing Non- Functional Requirements | C | C | | C | | A | |
| 2.4 Developing Use Case | C | C | | C | | A | |
| 2.5 Developing Test Cases | C | C | | C | | A | |
| 3. **Planning** | C | C | | C | | A | |
| 4. **Coding** |  |  | |  | |  | |
| 4.1 NLP Model | C | R | | R | | A | |
| 4.2 Front end/UI | R | | C | | C | | A |
| 4.3 Integration | R | | C | | C | | A |
| 5. **Testing** |  | |  | |  | |  |
| 5.1 Unit 1 | E | | A | | A | |  |
| 5.2 Unit 2 | A | | E | | E | |  |
| 5.3 System Testing | E | | E | | E | | A |

C: Creator, R: Reviewer, A: Approver E: Executor

R1 = Aakash  
R2 = Bhairav  
R3 = Nidhi

**Use case Diagram**

Diagram

Description automatically generated

**Functional Requirements**

1. Text Retriever should be implemented successfully such that noise and unwanted text is not included

2. The software should be able to work with pictures taken from all cameras, as different users could use different snapshot devices, it should even be able to identify text in it

3. The software designed should be easy to use and should be able to be learnt quickly by all users.

4. Make app light weight so that all types of user depending on age/device/use type can use it.

5. Have a caption bank for caption wanted to be saved.

**Non Functional Requirements**

1. Aesthetic appearance should be implemented, something that is easy to use and

the most important icons are at the forefront, it could be designed in a way that

users like working and using it in convenient.

2. Fast image recognition system for quick and fast responses of the information.

3. Good data set to include as many possible images.

**Chapter 4: Implementation and experimentation of one of the issues related to project work/ topic**

**Backend Design:**

In this backend design we have implemented login signup for authentication in React Native and have used api for login and registration of the user. We have created a dummy NOSQL database for storing of image and its corresponding label. We have integrated phone library with the app.

The user will be asked to Login/ Sign up before using the app. The login details and the last time the user has synced the app are a part of user database which will be stored. The ML model for generating captions will also be deployed on the cloud.

This app will request access to the photo gallery app of the user’s phone. The stored images will be passed through the ML model on the cloud and the corresponding captions will be stored in the user’s device.

When the user enters a query for searching an image in the database, the images corresponding to the caption(s) closely matching the query will be retrieved for the user.

**Diagram

Description automatically generated**

We have created an app in react native which would handle user authentication and smart gallery manager. Below are the screenshots of the pages of the app and the phone photo gallery integrated.

For initial design we had created this design. We have achieved similar to this UI screen and is intended to implement the same UI in our app.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedA picture containing text

Description automatically generated

  Machine Learning Model prediction example

A football player kicking a ball

Description automatically generated with medium confidence

A giraffe and zebras walking across a road

Description automatically generated with low confidence

**Architecture overview**

* The environment in which the app works is currently ideal but it is intended that app will work normally in all the modern devices. Performance of the app might hamper due to usage in old mobile.
* We intend our app to be used by all type of mobile users. Since this app reduces the time in search of image in phone.
* we are building a text-based image retrieval application, which links to a user’s photo gallery and generates text captions for the images stored in it. The users can then enter the text describing the events taking place in the image they are searching for; this application finds the closest captions stored to the query text and displays the images corresponding to them.

**Software testing:**

Unit testing is the first level of functional testing in order to test any software. In this, the test engineer will test the module of an application independently or test all the module functionality is called unit testing.

The primary objective of executing the unit testing is to confirm the unit components with their performance. We used it for various units of our projects.

we are successfully implementing the unit testing. It is the second level of functional testing, where we test the data flow between dependent modules or interface between two features is called integration testing.

The purpose of executing the integration testing is to test the statement's accuracy between each module. We combined the all units of project to perform it.

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Output** | **Expected Output** |
| If email id entered has valid signature | Accepted | Accepted |
| If email id entered has invalid signature | Not accepted | Not accepted |
| If email id not entered | Not accepted | Not accepted |
| If entered email id already exists in the database | Not accepted | Not accepted |
| If entered password length less than 6 | Not accepted | Not accepted |
| If entered password length greater than 25 | Not accepted | Not accepted |
| If password not entered | Not accepted | Not accepted |
| If entered password doesn’t contains capital alphabet | Not accepted | Not accepted |
| If entered password doesn’t contains small case alphabet | Not accepted | Not accepted |
| If entered password doesn’t contains any number | Not accepted | Not accepted |
| If entered password doesn’t contains any special character | Not accepted | Not accepted |
| If entered password contains capital alphabet, small case alphabet, numbers, special character and length of password is between 6 and 25 | Accepted | accepted |
| If Username not entered | Not accepted | Not accepted |
| If Username entered and not in database | accepted | Accepted |
| FirstName, LastName is entered | accepted | Accepted |
| FirstName, LastName is not entered | Not accepted | Not accepted |
| Machine learning model label generated | accepted | Accepted |

During testing it is found that UI of app is working efficiently without any bugs. And working of app is lag free tested on Iphone 12 mini , iphone SE 2017, Samsung grand 2 (2015) and mi note 4.

**Chapter 5 :**

**Conclusion and future work**

**Conclusion:**

Hence an app is developed which accesses a user’s photos stored on their device and

generates captions corresponding to the photos and stores those captions.

This enables the user to search for an image present on their device by describing the

photo in the form of a query. This app searches the photos with captions closely

matching the query and displays the corresponding photos to the user.

**Future Scope:**

The functionality of this app can be further improved in the following ways:

1. Filtering the images based on the date on which they were captured.

2. Filtering images based on the location at which they were captured.

3. Filtering the photos by the people present in the photos by performing facial

recognition.

**Bibliography**

* [Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering by Peter Anderson Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, Lei Zhang](https://arxiv.org/pdf/1707.07998.pdf)
* [Composing Text and Image for Image Retrieval by Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, James Hays](https://arxiv.org/abs/1812.07119)
* [ViLBERT by Jiasen Lu, Dhruv Batra, Devi Parikh, Stefan Lee](https://arxiv.org/abs/1908.02265)
* [React by Meta](https://reactjs.org/)
* [Object Detection by Gaudenz Boesch](https://viso.ai/deep-learning/object-detection/)

**Acknowledgements**

We would like to thank our mentors Ms Sheetal Pereira and Mrs Mansi Kambli for guiding us in the work we had taken as a part of this project.

We would also like to express our gratitude to K. J. Somaiya College of Engineering for the opportunity to work on this project.