A
Project Report On

ImageIt
(image captioning and ocr app)

by

Sarthi Kalathiya (ce092)(21CEUOS087)

B. Tech CE, Semester VI
Subject: System Design Practice

Guided By: Prof. Ashish k Gor.
Assistant Professor
Dept. of Comp. Engg.

Department of Computer Engineering
Faculty of Technology,
Dharmsinh Desai University
College Road,
Nadiad-387001

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical/term work carried out in the subject of Software Design Project and recorded in this journal is the bonafide work of Sarthi Kalathiya , ID No: 21CEUOS087 of B.Tech semester VI in the branch of Computer Engineering during the academic year 2020-2021.

Prof. Ashish K Gor
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University,
Nadiad

Dr. C.K. Bhensdadia
Head,
Dept. of Computer Engg.,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad

Acknowledgement

It is with immense pleasure that we extend our heartfelt gratitude to all those who contributed to the success of this project. Without the support and assistance of numerous individuals who provided invaluable insights and guidance, navigating through various challenges would have been considerably more arduous.

Theoretical knowledge, while indispensable, remains inert without practical application. We are deeply appreciative of our institute for affording us the platform to translate our theoretical understanding into tangible outcomes through this project. Submitting this endeavor as an integral component of our curriculum is not only a privilege but also a testament to our commitment to academic excellence.

We wish to express our sincere appreciation to our mentor, Prof. Ashish K Gor, whose unwavering guidance propelled us forward throughout this journey. His wealth of knowledge, coupled with his relentless support and exceptional testing acumen, significantly contributed to the refinement and ultimate success of our project. Without his dedication and expertise, this endeavor would not have reached its fruition.

With Sincere Regards
Sarthi Kalathiya

Table of Contents

1. Abstract.. 5
2. Introduction.. 6
2.1 Brief Introduction.. 6
2.2 Tools, Technology and Platform used...6
3. Software Requirements Specifications.. 7
3.1 Product Scope.. 7
3.2 Types of User.. 7
3.3 System Functional Requirements.. 7
3.4 Other Non-functional Requirements...8
4. Design... 9
4.1 Use Case Diagram.. 9
4.2 Class Diagram... 10
4.3 Activity diagram... 11
5. Implementation Details.. 12
5.1 Front-End Design Implementation Details...12
5.2 Sub Project Implementation Details.. 13
5.3 Flask Server as Backend.. 15
6. Testing.. 16
6.1 Testing Method Used...16
6.2 Test Cases...16
7. ScreenShots... 18
7.1 Signup & Login.. 18
7.2 Ocr (Text recognition)... 20
7.3 Image captioning... 22
8.Conclusion.. 24
9. Limitation and Future Extension.. 25
9.1 Limitation... 25
9.2 Future Extension... 25
10. Bibliography.. 26

1. Abstract

"A picture is worth a thousand words"
-Fred R. Barnard

The Image Processing Applications project comprises a collection of sub-projects, each centered around leveraging image processing and computer vision techniques. The primary objective of this endeavor is to conduct diverse operations on images and extract valuable information from them.

For instance, consider providing the application with an image containing text. Upon receiving the image, the application initiates a pre-processing phase, followed by text extraction and conversion into a digital format. This facilitates the seamless integration of textual information into documents, eliminating the need for manual transcription and thereby optimizing time utilization.

Moreover, the application exhibits versatility in its functionality, adapting its operations based on the nature of the input image.

2. Introduction

2.1 Brief Introduction

The Image Processing Applications project comprises a multitude of sub-projects, all unified by a common objective : image processing. The core focus of this study lies in defining image processing as follows:

"Image processing is a method employed to execute various operations on an image, with the aim of either enhancing the image or extracting valuable information from it."

Within this system, there exists a single end user who utilizes the application. The user's interaction with the system involves providing an image as input, whereupon the system is capable of executing the following tasks based on the type of image provided:

1. Extract text from the image
2. Generating a caption for the image

2.2 Tools, Technology and Platform used

1) Programming Languages: Python and Dart

2) IDE: Android Studio and Visual Studio Code

3) Flask Server as backend

4) Python Libraries used:-

OpenCV, NumPy, Tensorflow, Keras, Pyzbar, Scikit-Learn, Pandas

5) Flutter’s ML kit package

3. Software Requirements Specifications

3.1 Product Scope

The system is meticulously crafted to execute diverse image processing operations, tailoring its functions to the specific characteristics of the provided image. Its scope extends globally, welcoming users from all corners of the world. Within this expansive scope, the system offers users a range of functionalities as below :

1) Image-to-Text Extraction : Enabling users to effortlessly extract textual content from images, streamlining the process of converting visual information into a digital and editable format.
2) Image Caption Generation : Empowering users to generate descriptive captions for their images, enhancing the interpretability and communicative value of visual content.

3.2 Types of User

Here, there is only one end-user who is going to use this application.

3.3 System Functional Requirements

R1. Image to Text Extraction

Description: This requirement is for converting text from the images into text format/file. System will extract the text from the images and display it on the screen. This text can be copied and one can use it for other purposes.

Input: Document or Text Image

Output: Displays the image text on the screen

R2. Caption the Image

Description: This requirement entails the generation of descriptive captions for images. The system is designed to analyze the content and context of an image and produce a succinct, informative caption that encapsulates its essence. The generated caption is displayed alongside the image for user reference and interpretation.

Input: Image file

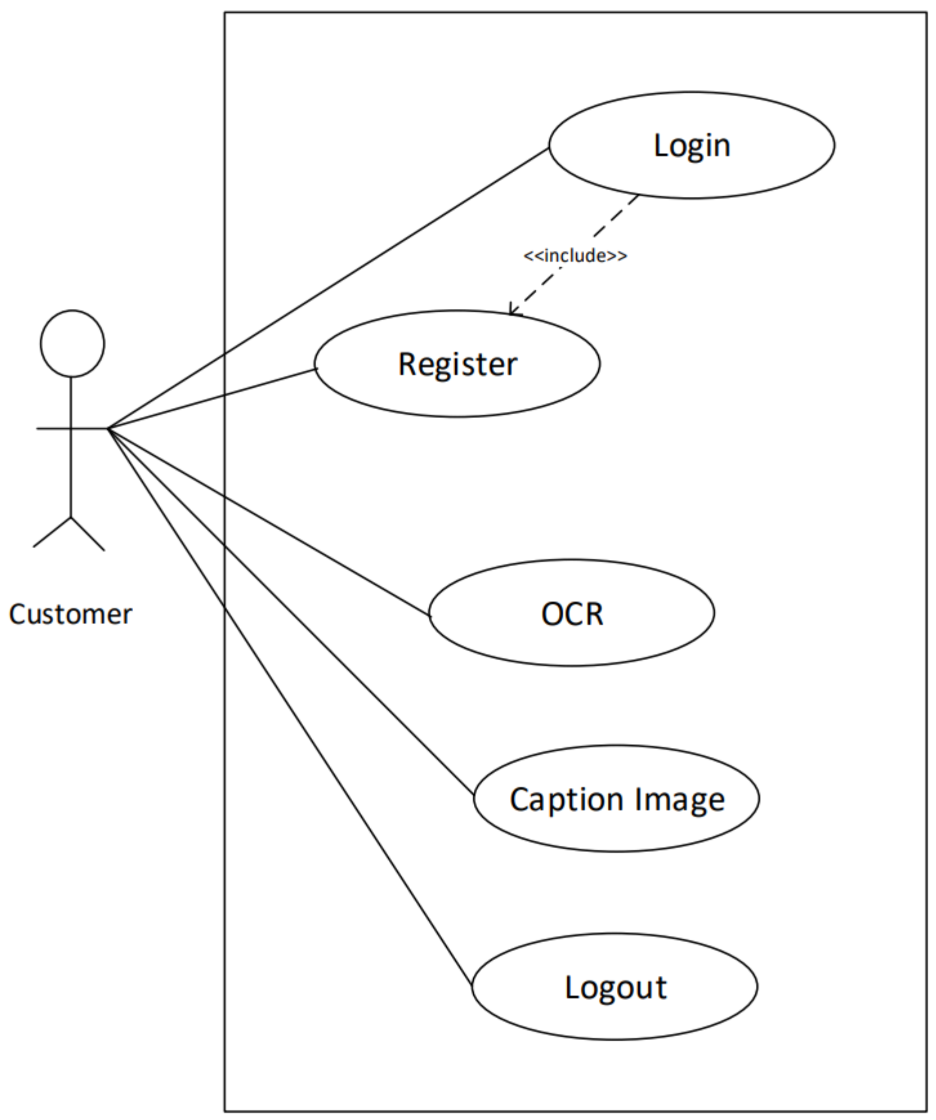
Output: Display of the generated caption accompanying the image

3.4 Other Non-functional Requirements

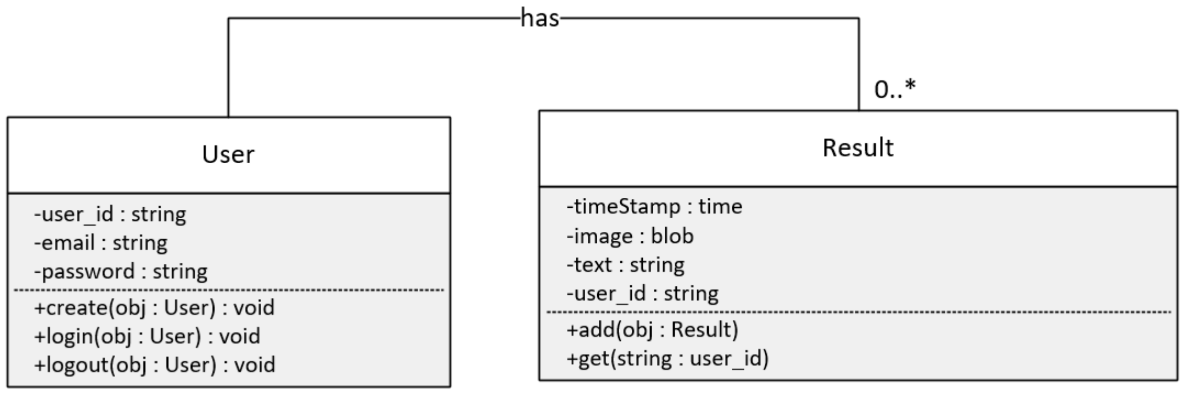
- **Performance**
  - The application should run efficiently. It must be interactive and user friendly in nature.
- **Reliability**
  - The application must ensure that the system is reliable in its image processing operations.

4. Design

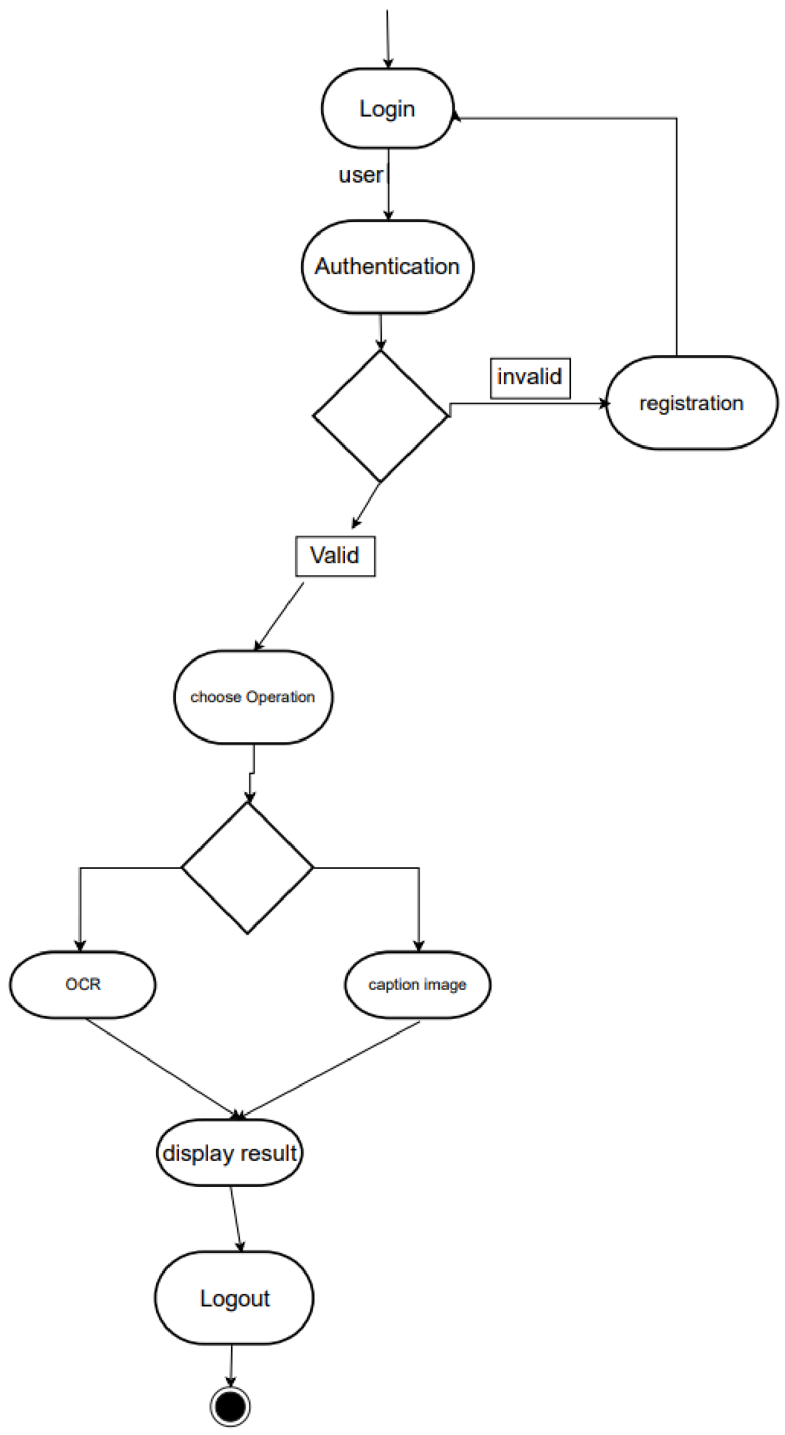
4.1 Use Case Diagram



4.2 Class Diagram



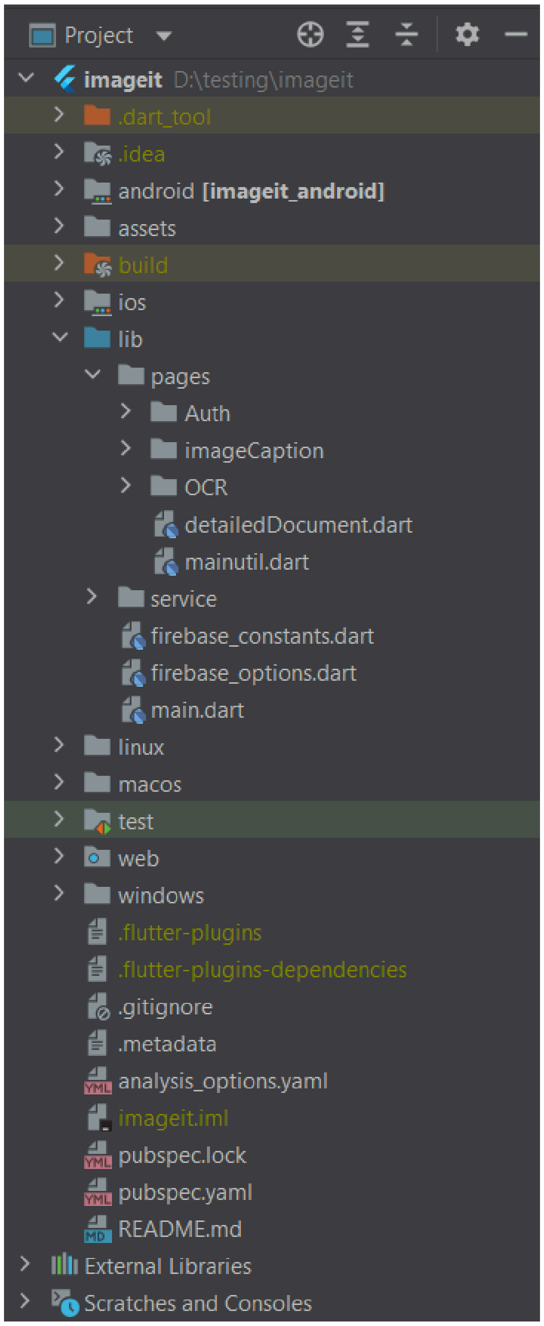
4.3 Activity diagram



5. Implementation Details

5.1 Front-End Design Implementation Details

Front end is designed using the Flutter SDK created by Google. Everything in Flutter is a widget. We have created required widgets separately and imported them on the required view. Flutter App's folder structure:



```
Project
imageit D:\testing\imageit
    .dart_tool
    .idea
    android [imageit_android]
    assets
    build
    ios
    lib
        pages
            Auth
            imageCaption
            OCR
            detailedDocument.dart
            mainutil.dart
        service
        firebase_constants.dart
        firebase_options.dart
        main.dart
    linux
    macos
    test
    web
    windows
    .flutter-plugins
    .flutter-plugins-dependencies
    .gitignore
    .metadata
    analysis_options.yaml
    imageit.iml
    pubspec.lock
    pubspec.yaml
    README.md
External Libraries
Scratches and Consoles
```

5.2 Sub Project Implementation Details

5.2.1 Text Recognition

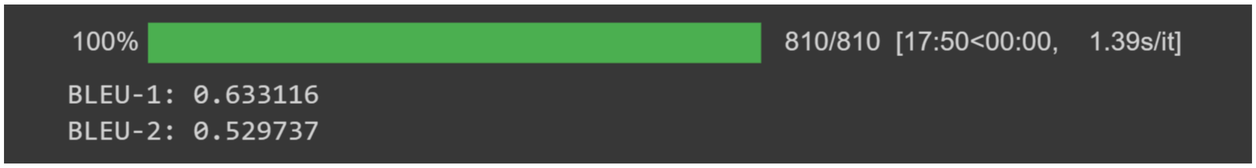
Input : Image
Output : Text that is in the image
Approach : Read image from file system. Do image preprocessing with help of Google ML Kit and Extract the text using TextRecognizer

5.2.2 Image captioning

Input: Image
Output: Caption of the image

Approach: Read image from the file system. Make a POST request to the backend Flask server and display the string response from the server to the user.

Evaluation : The system employs BLEU Score to evaluate the predicted text against a reference text, represented as a list of tokens.
A BLEU Score greater than 0.4 is considered a good result. For achieving a better score, it is recommended to increase the number of epochs accordingly.



```
100% ████████████████████ 810/810 [17:50<00:00,  1.39s/it]
BLEU-1: 0.633116
BLEU-2: 0.529737
```

architecture that is used is VGG19 :
VGG19 stands as a pivotal milestone in the realm of deep learning, renowned for its architectural simplicity yet profound impact in various machine learning tasks. Developed by the Visual Geometry Group at the University of Oxford, VGG19 belongs to the VGG family of convolutional neural networks (CNNs), originally proposed by Simonyan and Zisserman in their landmark paper.

At its core, VGG19 is characterized by its deep structure comprising 19 layers, hence its name, with 16 convolutional layers followed by 3 fully connected layers. The hallmark of VGG19 lies in its uniform architecture, where convolutional layers employ small 3x3 filters with a stride of 1 pixel, and max-pooling layers utilizing 2x2 filters applied with a stride of 2 pixels. This uniformity not only simplifies the network's design but also facilitates its interpretability and transferability across diverse datasets and tasks.

The brilliance of VGG19 lies not only in its architecture but also in its performance. Despite its simplicity compared to more modern architectures, VGG19 exhibits remarkable capability in feature extraction and classification tasks, achieving state-of-the-art results in various image recognition challenges, including the renowned ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

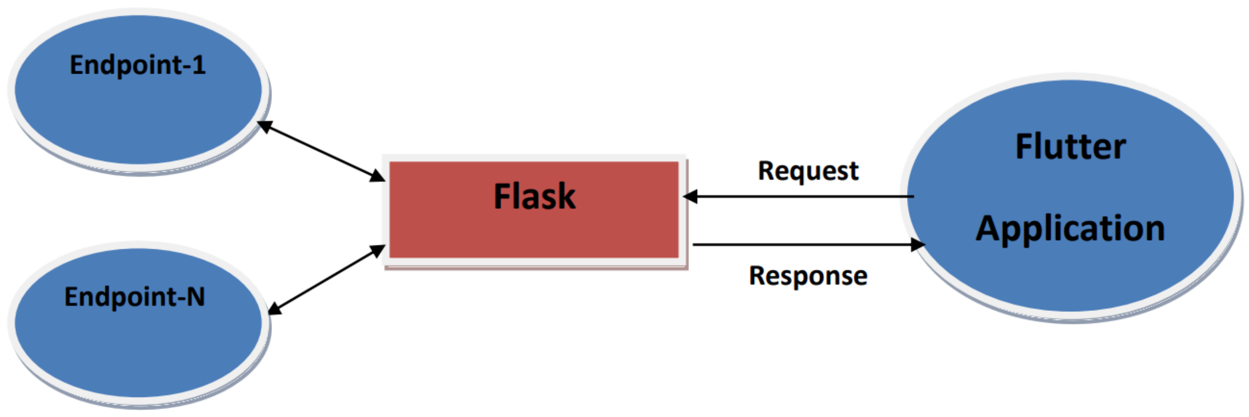
One of the key advantages of VGG19 is its versatility. While initially trained on the ImageNet dataset, VGG19's pre-trained weights serve as valuable initialization for transfer learning across a myriad of computer vision tasks, including object detection, segmentation, and even medical image analysis. This adaptability has made VGG19 a cornerstone in many deep learning projects, providing a reliable starting point for researchers and practitioners alike.

In conclusion, VGG19 stands as a testament to the enduring influence of simplicity and elegance in deep learning. Its straightforward architecture, coupled with its impressive performance and versatility, has solidified its position as a fundamental tool in the machine learning practitioner's toolkit, embodying the timeless adage that sometimes, less is indeed more in the pursuit of intelligence from data.

Dataset used here is flicker8K :
The Flickr8k dataset comprises 8,000 images with five captions per image. It's widely used in computer vision and NLP for tasks like image captioning. Each image is paired with concise, manually annotated descriptions. Ideal for training models to generate human-like captions for images. Valuable resource for developing image captioning apps with accurate and descriptive outputs.

5.3 Flask Server as Backend

- The Project is developed using python language. Functionalities involving ML and DL are done using Python. So we were required to access app endpoints using flutter. So Flask is used as a backend server from where the functionalities are exposed through URL.
- To and Fro data transfer is done using JSON



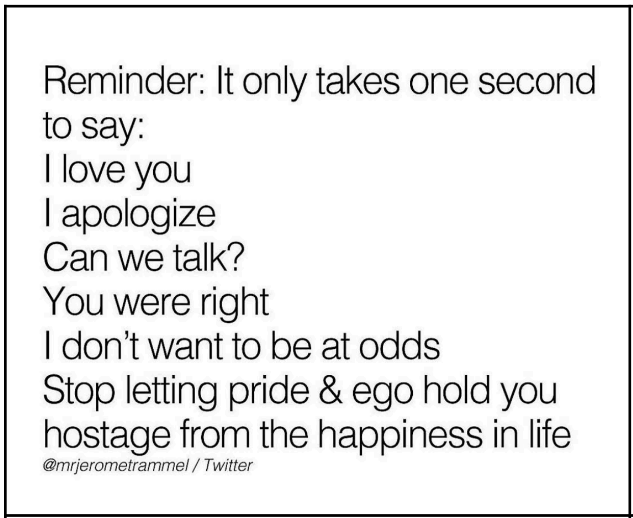
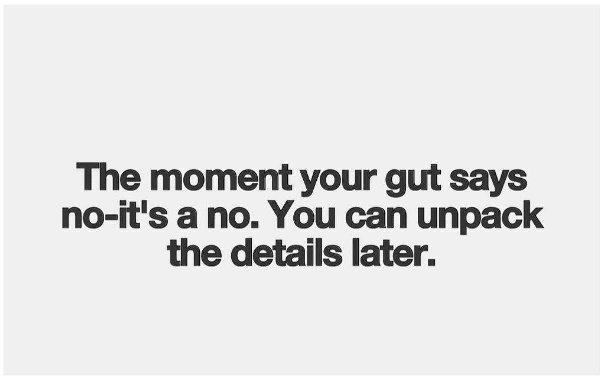
6. Testing

6.1 Testing Method Used

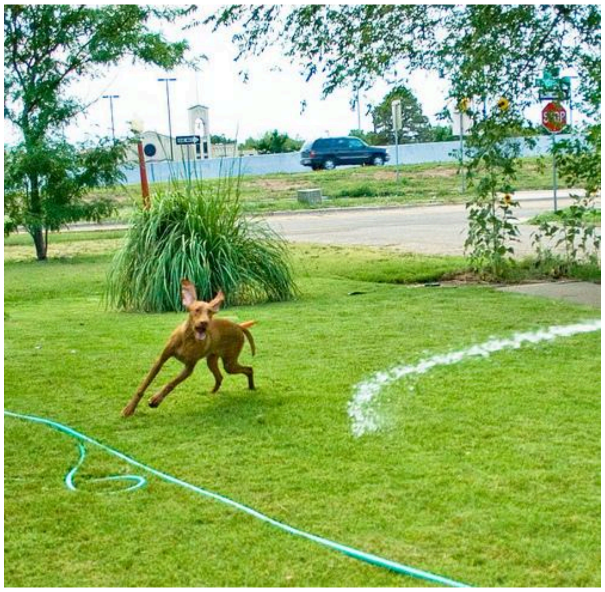
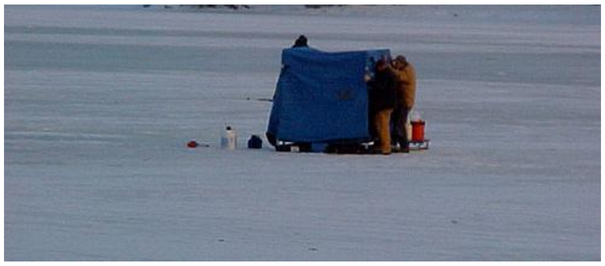
- We have performed unit testing during the development. But for testing purposes, we have used black box testing method.
- For black box testing, we have designed the test cases for each sub project and have tested it in our application. Also, we have observed the output and noted down the results in the next section.

6.2 Test Cases

6.2.1 Ocr (Text recognition)

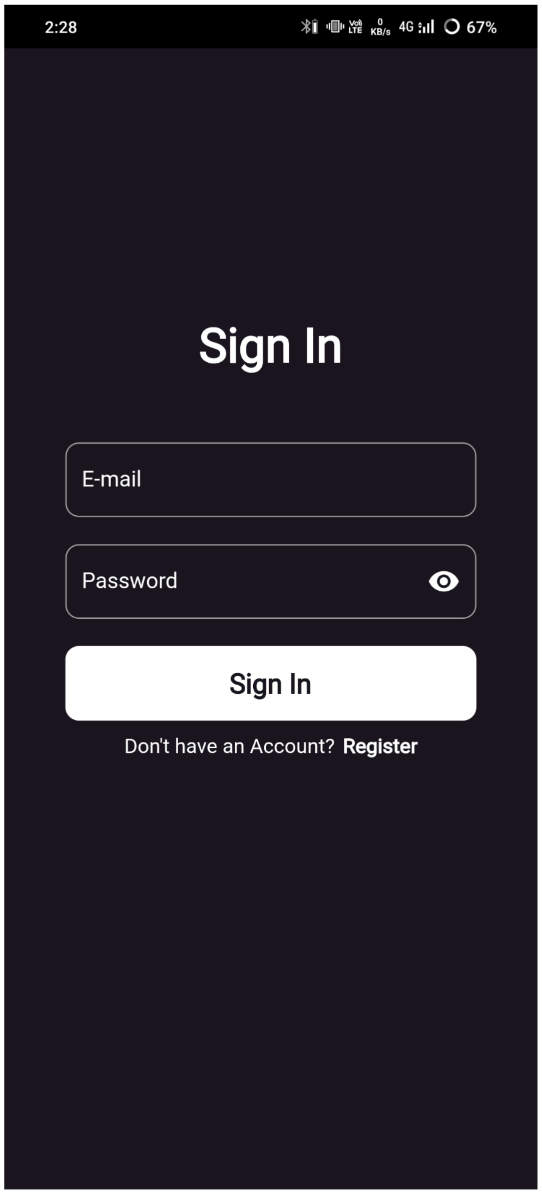
 Reminder: It only takes one second to say: I love you I apologize Can we talk? You were right I don’t want to be at odds Stop letting pride & ego hold you hostage from the happiness in life @mrjerometrammel / Twitter	Reminder: It only takes one second to say: I love you I apologize Can we talk? You were right I don't want to be at odds Stop letting pride & ego hold you hostage from the happiness in life @mrjerometrammel / Twitter
 The moment your gut says no-it's a no. You can unpack the details later.	The moment your gut says no-it's a no. You can unpack the details later

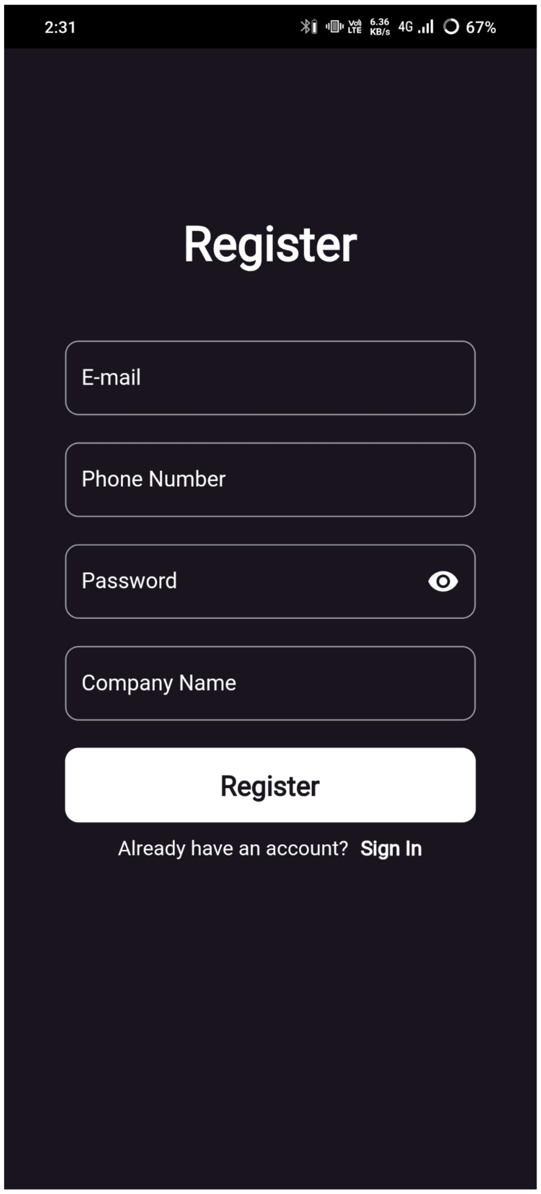
6.2.2 Image captioning

	A dog is playing with a hose .
	Two people standing outside a blue tent structure on a snowy surface .

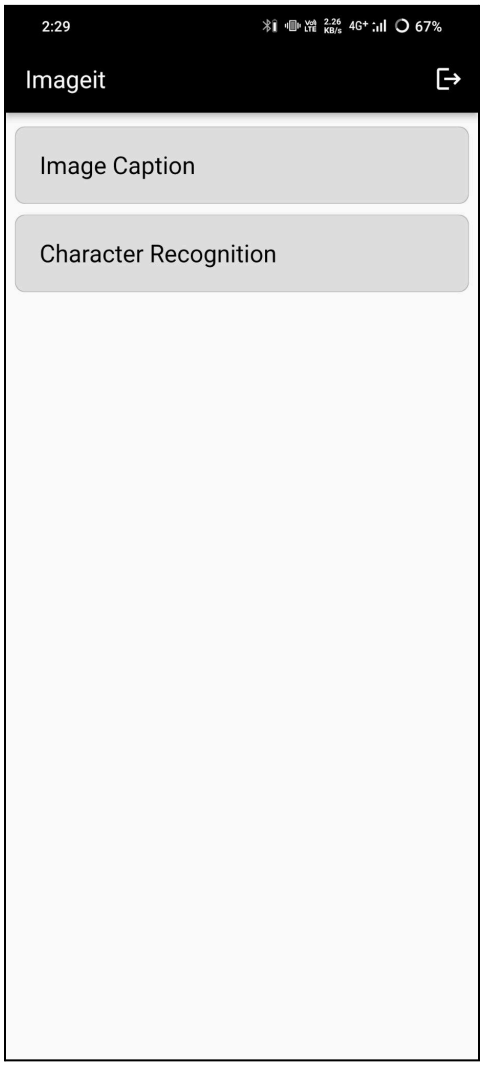
7. ScreenShots

7.1 Signup & Login

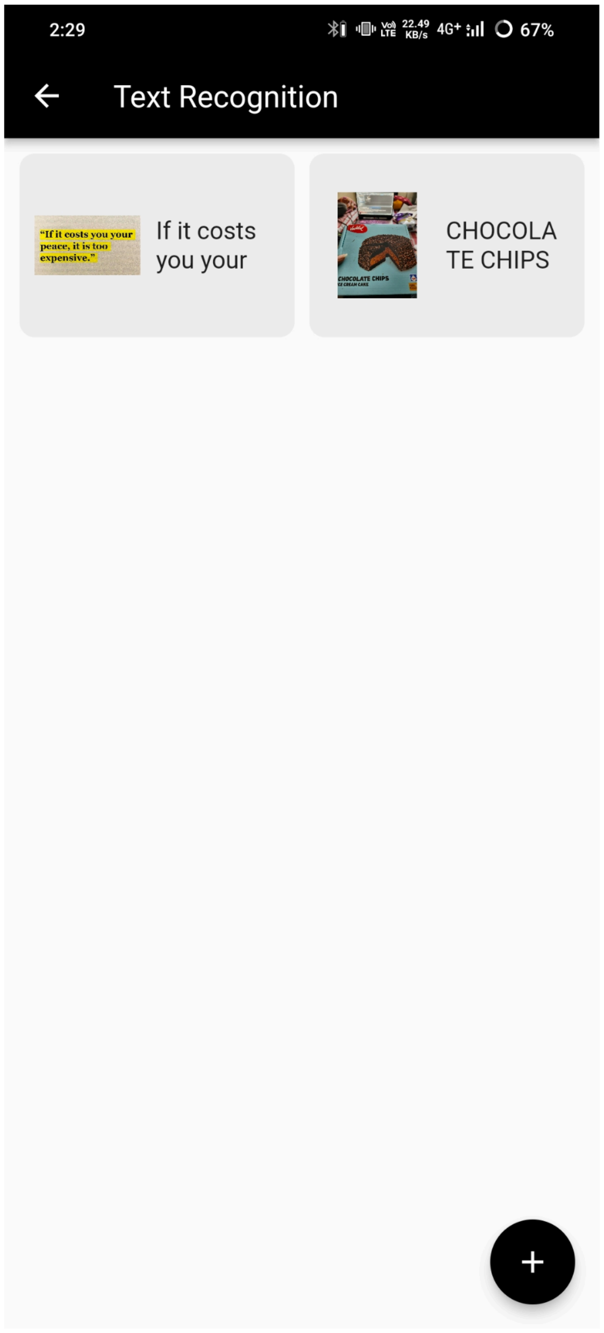


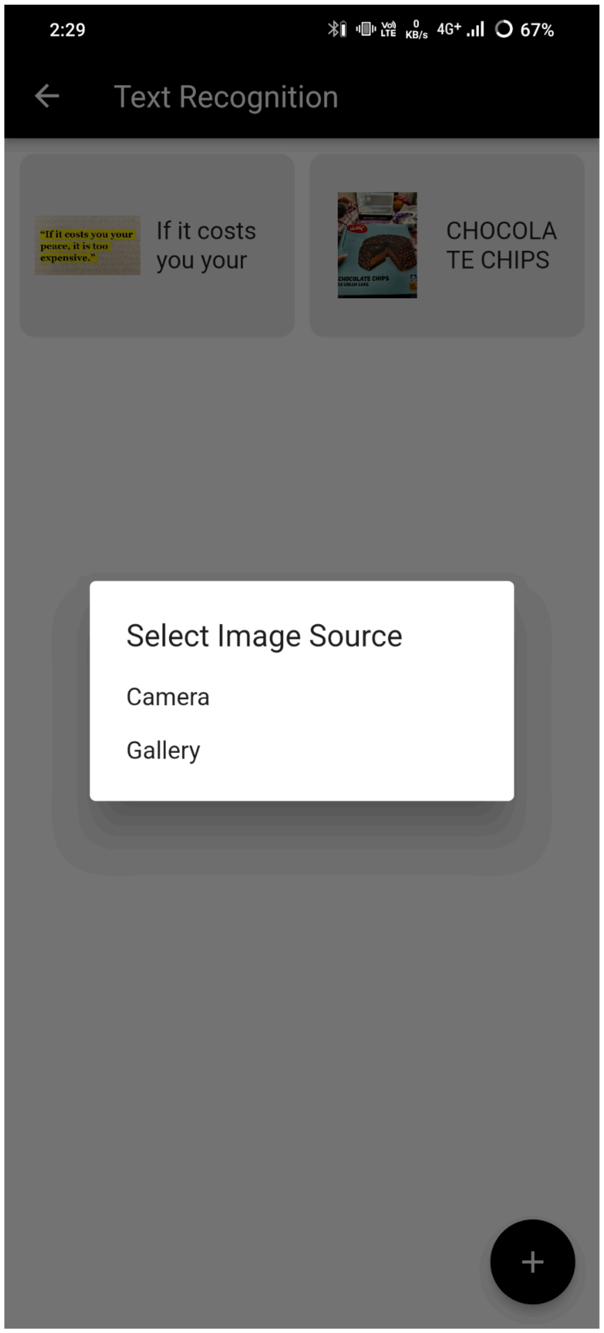


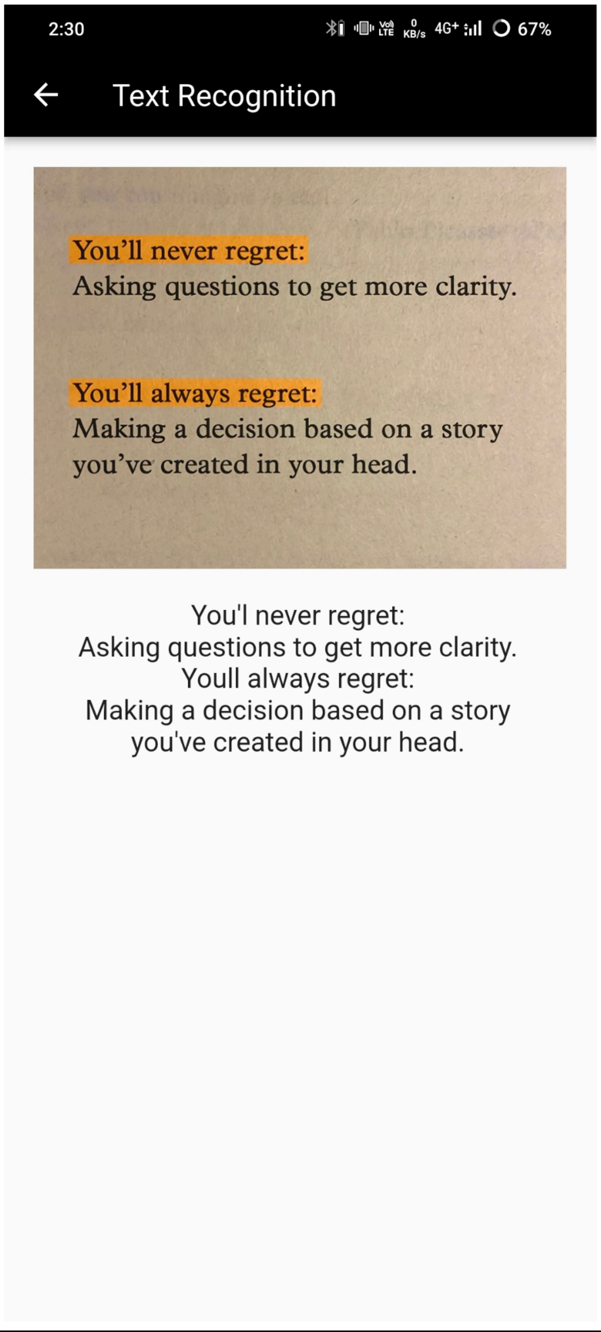
Home screen



7.2 Ocr (Text recognition)







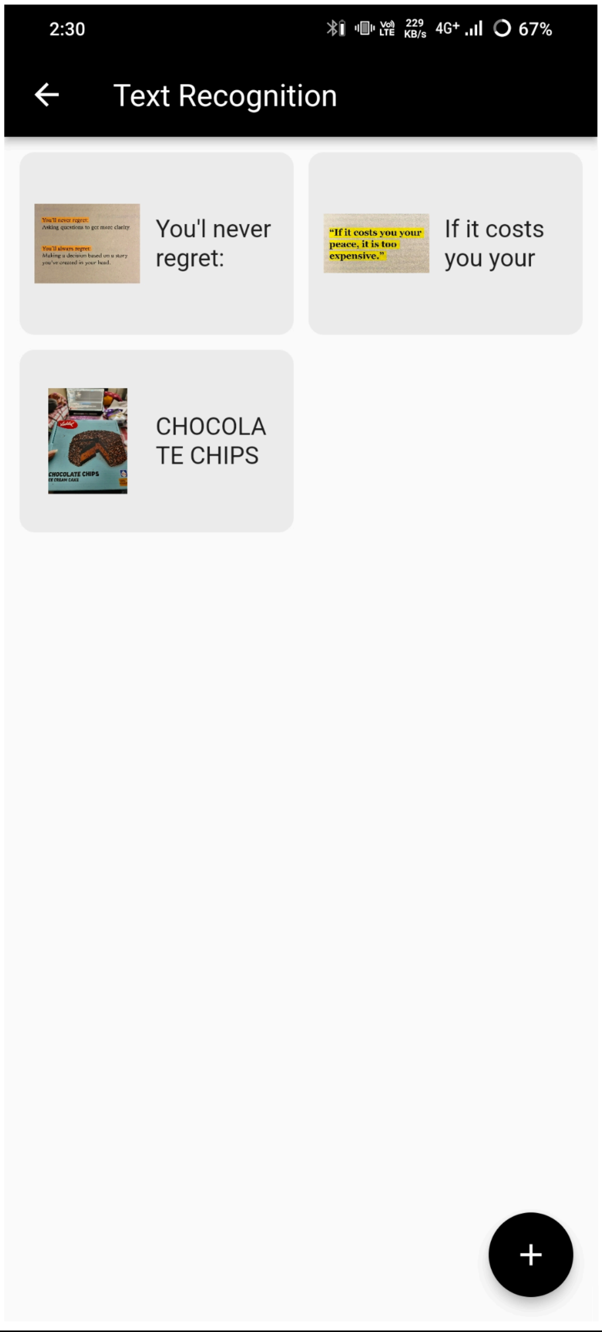
2:30 67%

Text Recognition

You'll never regret:
Asking questions to get more clarity.

You'll always regret:
Making a decision based on a story you've created in your head.

You'l never regret:
Asking questions to get more clarity.
Youll always regret:
Making a decision based on a story
you've created in your head.



2:30 67%

Text Recognition

You'l never regret:

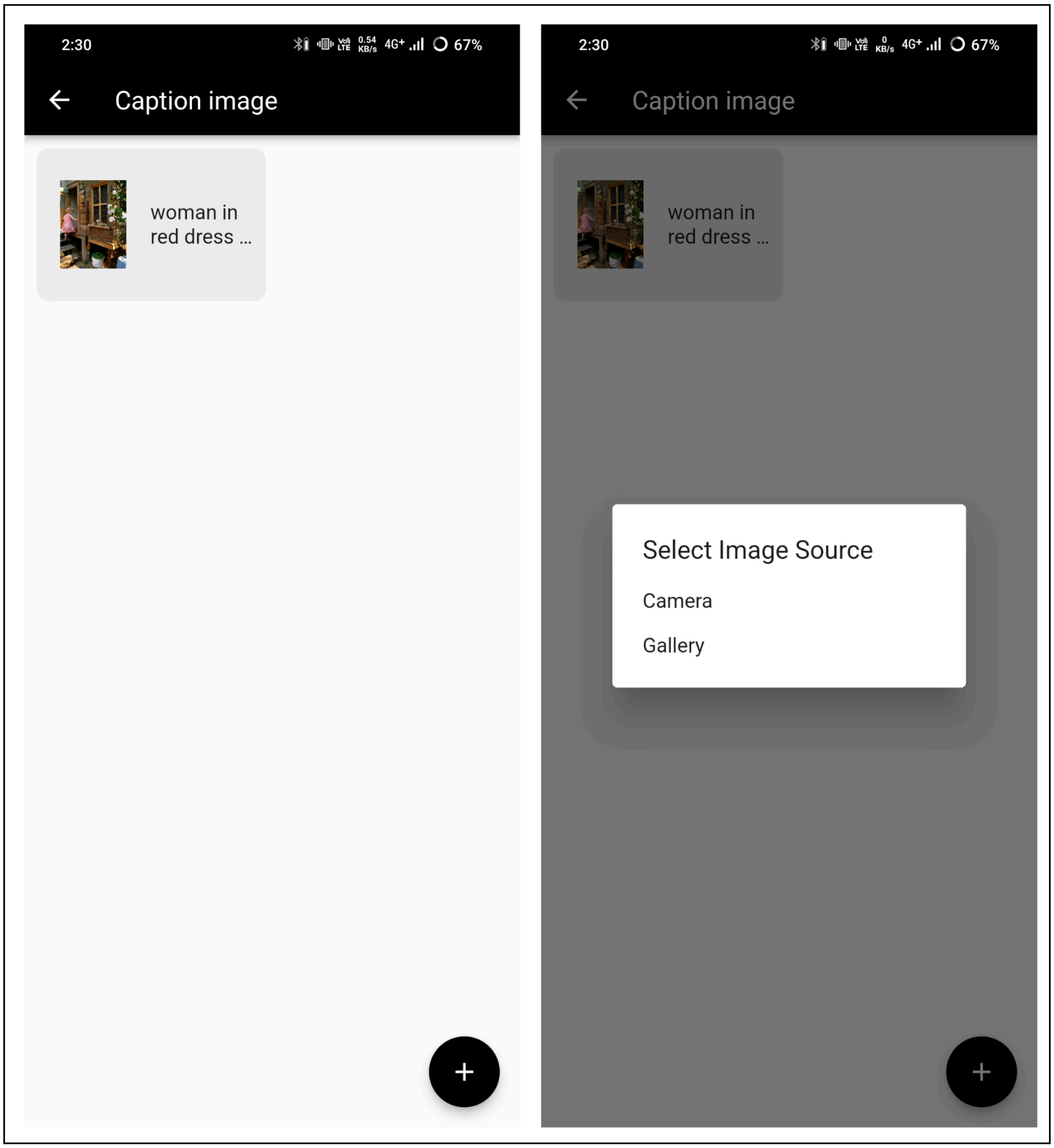
"If it costs you your peace, it is too expensive."

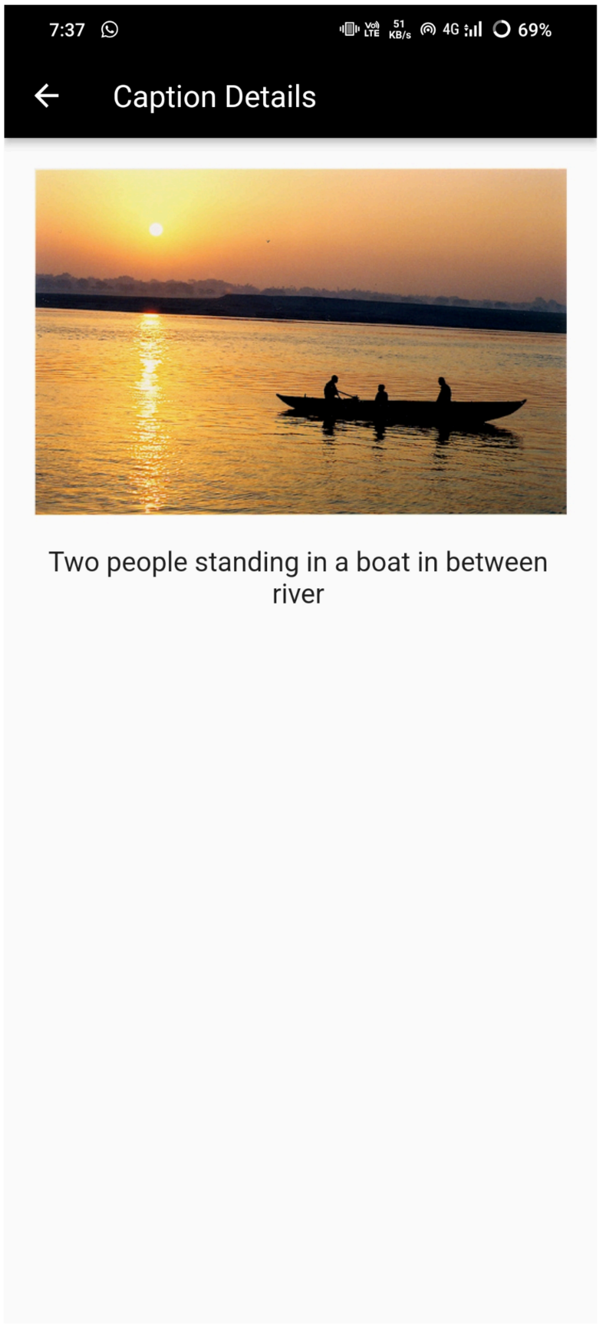
If it costs you your

CHOCOLATE CHIPS

CHOCOLA TE CHIPS

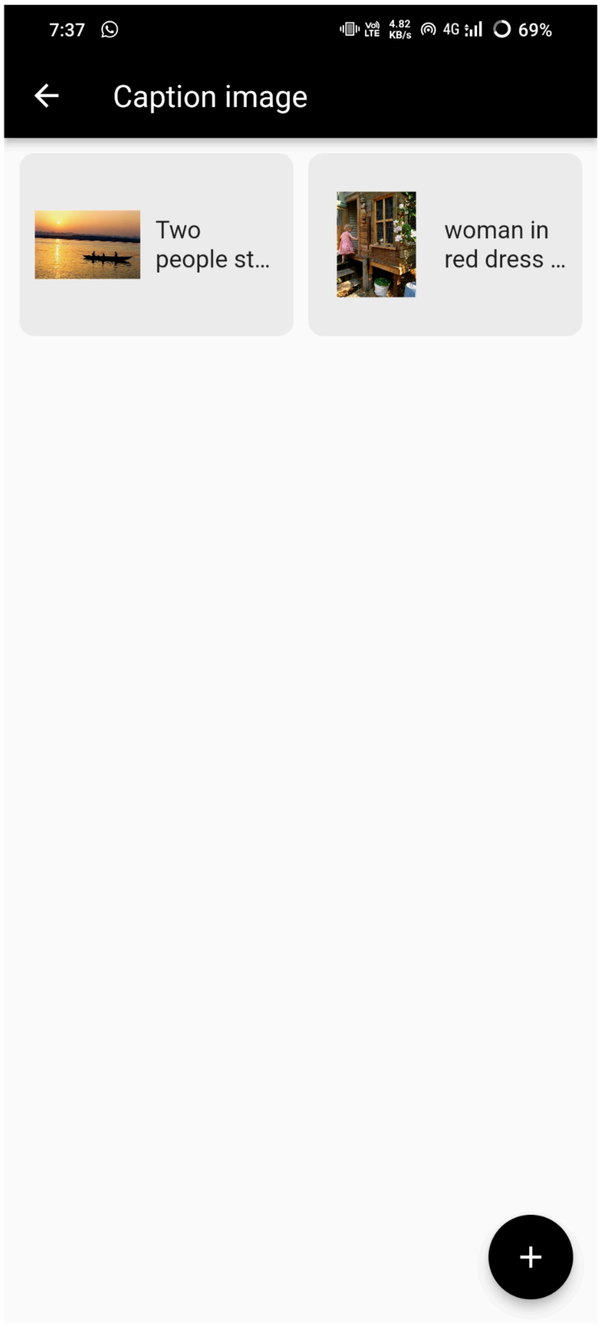
7.3 Image captioning




7:37

Caption Details

Two people standing in a boat in between river


7:37

Caption image

Two people st...

woman in red dress ...

+

8.Conclusion

This project is aimed at developing Image Processing Applications using the concepts of image processing, computer vision, machine learning and deep learning. The idea was to process the image, to obtain some useful information from it and to add some new information to it.

For this project, I have selected 2 project definitions which are related to image processing. i.e. Ocr (Text recognition) and Image captioning from scratch, where I have done the image processing and trained the model using machine learning and deep learning by providing images to the dataset.

The project’s backend is developed using the python language and Flask server. The front end of the mobile application is developed using the Flutter SDK, created by Google.

So, after performing various comprehensive tests, I conclude that our project is working successfully. But of course, there is always a scope for improvement and learning. This was my first AI project but yes in the end, I did learn something new from it. I am now looking forward to overcoming the existing limitations and to add new possible extensions which are discussed in the next section.

“A picture is a poem without words”
-Horace

9. Limitation and Future Extension

9.1 Limitation

- The functionality is limited when processing images that are unclear, blurry, or lack sufficient intensity and brightness.
- Optical Character Recognition (OCR) is optimized for printed images and may not accurately recognize handwritten content.

9.2 Future Extension

- Presently, the application operates by uploading or capturing an image, followed by an API call for processing and returning results. A potential future enhancement involves providing live or frame-by-frame responses to users, eliminating the need for manual image uploading or capturing. This streamlined process would occur seamlessly in the background.

10. Bibliography

- Keras – Python Deep Learning Neural Network API playlist by deeplizard
- OCR
- Information about Image Processing – Used for abstract and introduction
- StackOverflow