A

PROJECT REPORT ON

IMAGE PROCESSING APPLICATIONS

By

Harshit Tarsariya	**CE136**	**18CEUBG080**
Janak Vaghasiya	**CE142**	**18CEUOS058**
Jayesh Zinzuvadia	**CE149**	**18CEUBG064**

B. Tech. CE Semester – VI

Subject: System Design Practice

Guided by:

Prof. Apurva A. Mehta

Assistant Professor

Dept. of Comp. Engg.

Faculty of Technology

Department of Computer Engineering

Dharmsinh Desai University

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 **System Design Practice** and recorded in this journal is the

bonafide work of

Harshit Tarsariya	**CE136**	**18CEUBG080**
Janak Vaghasiya	**CE142**	**18CEUOS058**
Jayesh Zinzuvadia	**CE149**	**18CEUBG064**

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2020-2021**

Prof. Apurva A. Mehta
Assistant Professor,
Dept. of Computer Engg.
Faculty of Technology
Dharmsinh Desai University, Nadiad

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

Table of Contents

List of Figures

List of Tables

Abstract

"A picture is worth a thousand words"

-Fred R. Barnard

Image Processing Applications project is itself a bunch of projects. Each sub-project is an application of image processing and computer vision. The main purpose of this project is to perform various operations on an image and to extract some useful information from it.

For example - let's say that you give an image to the application, which contains some text on it. So, the application will first pre-process the image and then it will extract the text information and convert it into the digital form, which you can further use it in some document. Thus, it saves your time to type down the text manually.

Apart from that, the application can also perform various tasks based on the type of image given to it. Like it can solve a Sudoku puzzle, solve the basic mathematical equations, read aloud the text image, translate the text to some another language and can also determine the product name by reading the barcode of a product.

1. Introduction

1.1 Brief Introduction:-

Image Processing Applications is a collection of various sub-projects. Each sub-project has a common aim i.e. to deal with image processing. So, the main focus of the study is image processing, let's define it-

"Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it."

In this system, there is only one end user i.e. one who is going to use this application. Here, the user is required to give an image as input and based on the type of image, the system can perform the following tasks:-

1. Extract text from the image
2. Solves a Sudoku puzzle
3. Solves a basic mathematical equation
4. Read aloud the text from the image
5. Translate the image-text to some other language
6. Read the barcode and determine the product name

1.2 Tools, Technology and Platform used:-

1) Programming Languages: Python and Dart

2) IDE: PyCharm Community Edition and Visual Studio Code

3) Flask Server as backend

4) Python Libraries used:-

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

5) Flutter's ML kit package

2. Software Requirements Specifications

2.1 Product Scope

The system is designed to perform various image processing operations based on the type of image. Scope of the system is global and open for all users. System provides various functionalities to the users like image to text extraction, translating into other language, solving Sudoku puzzle and basic mathematical equations, barcode to product details and image text to speech.

2.2 Types of User

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

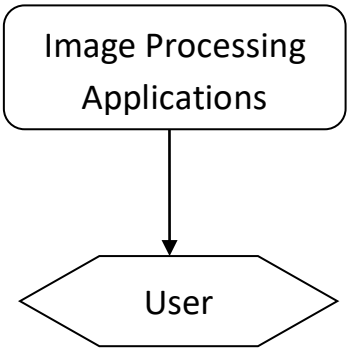


Fig. 2.1: Product Perspective Diagram

2.3 System Functional Requirements

NOTE: Here, the functional requirements are listed as module wise

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 purpose.

Input: Document or Text Image

Output: Displays the image text on the screen

R2. Sudoku Solver

Description: This requirement is for solving the Sudoku puzzle problem.

Input: Sudoku Image

Output: Show the final answer of Sudoku puzzle

R3. Math Equation Solver

Description: This requirement is for solving the mathematical equations like basic arithmetic operations such as addition, multiplication, division and subtraction.

Input: Mathematical equation image

Output: Show the answer of the given equation

R4. Translator

Description: This requirement is for translating the text from the images into some other language.

Input: Text to be translated, Language

Output: Show the translated text in the given language fonts

R5. Text Reader

Description: This requirement is for reading aloud the text from the images.

Input: Document or Text Image

Output: Machine will read the text from the image

R6. Barcode to Product Details

Description: This requirement is for reading the barcode from the image and determining the product name as output from the barcode image.

Input: Barcode Image

Output: Show the product name belonging to the barcode.

2.4 Other Non-functional Requirements

1) Performance

The application should run efficiently. It must be interactive and user friendly in nature.

2) Reliability

The application must ensure that the system is reliable in its image processing operations.

3. Design

3.1 Use Case Diagram

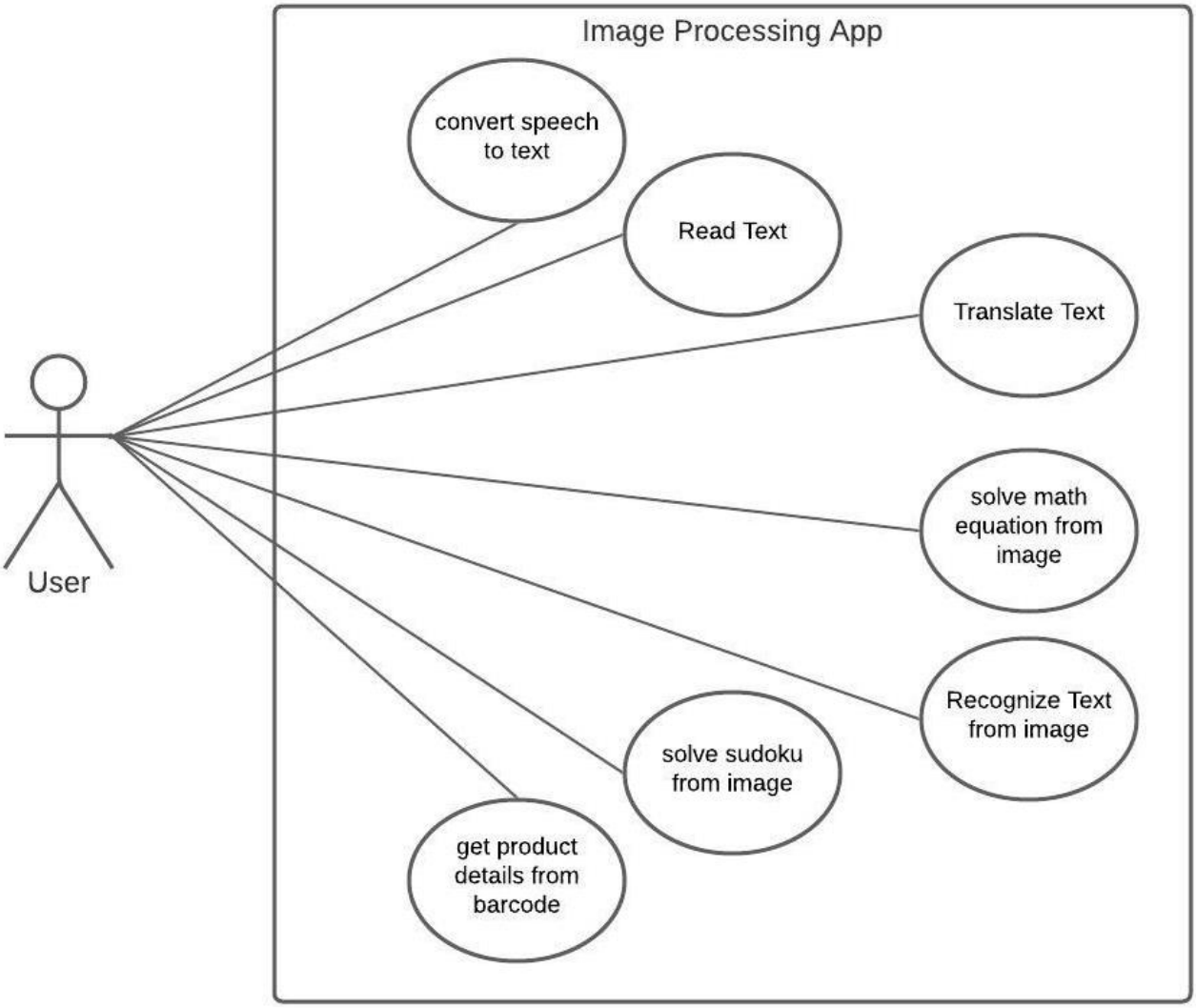


Fig. 3.1: Use Case Diagram

3.2 Class Diagram

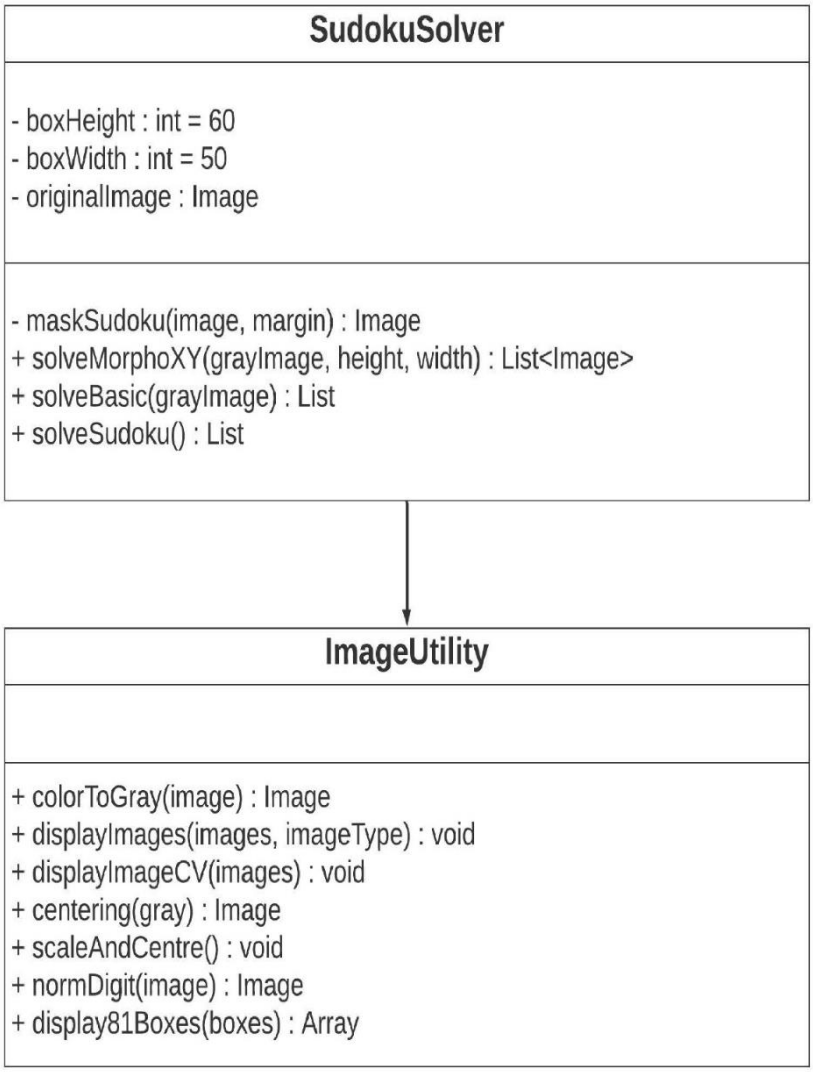


Fig. 3.2.1: Sudoku Solver Class Diagram

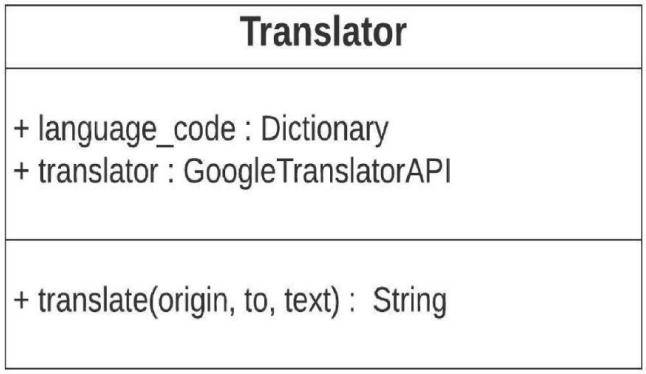


Fig. 3.2.2: Translator Class Diagram

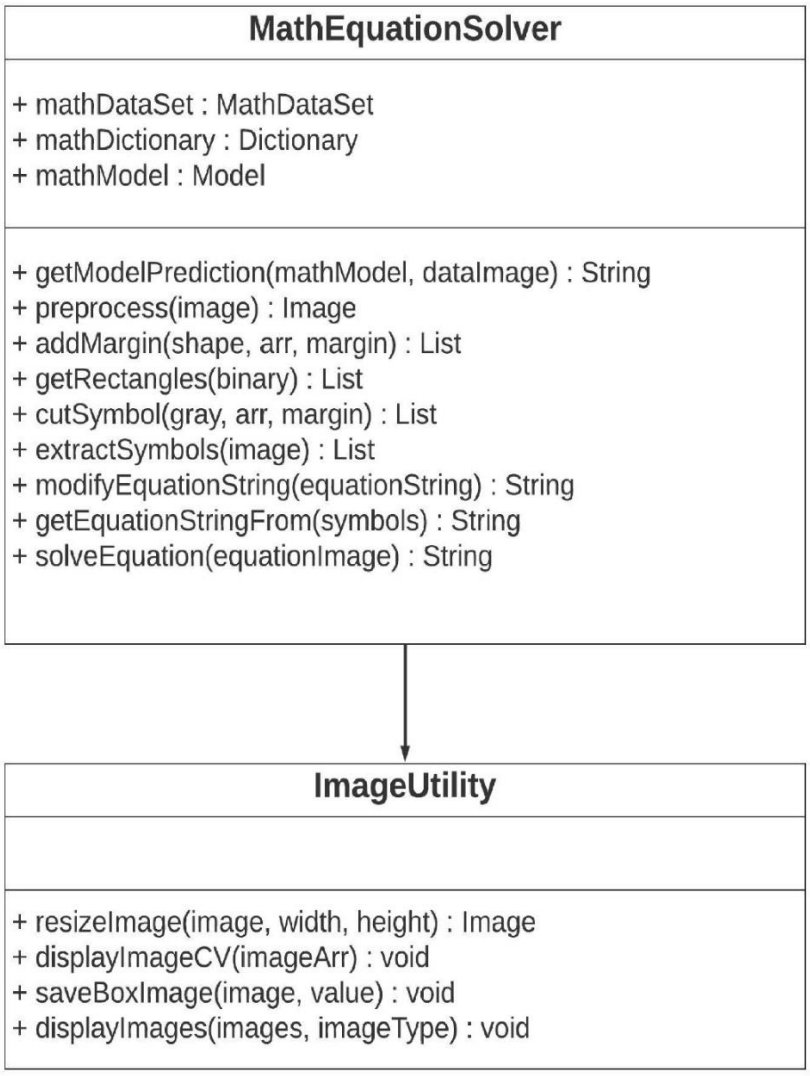


Fig. 3.2.3: Math Equation Solver Class Diagram

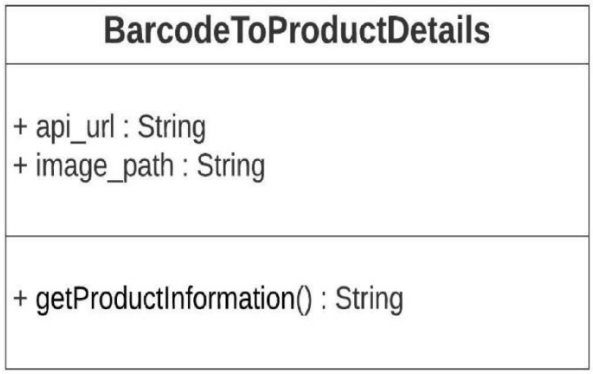


Fig. 3.2.4: Barcode to Product Details Class Diagram

Fig. 3.2: Class Diagrams

3.3 State Diagram

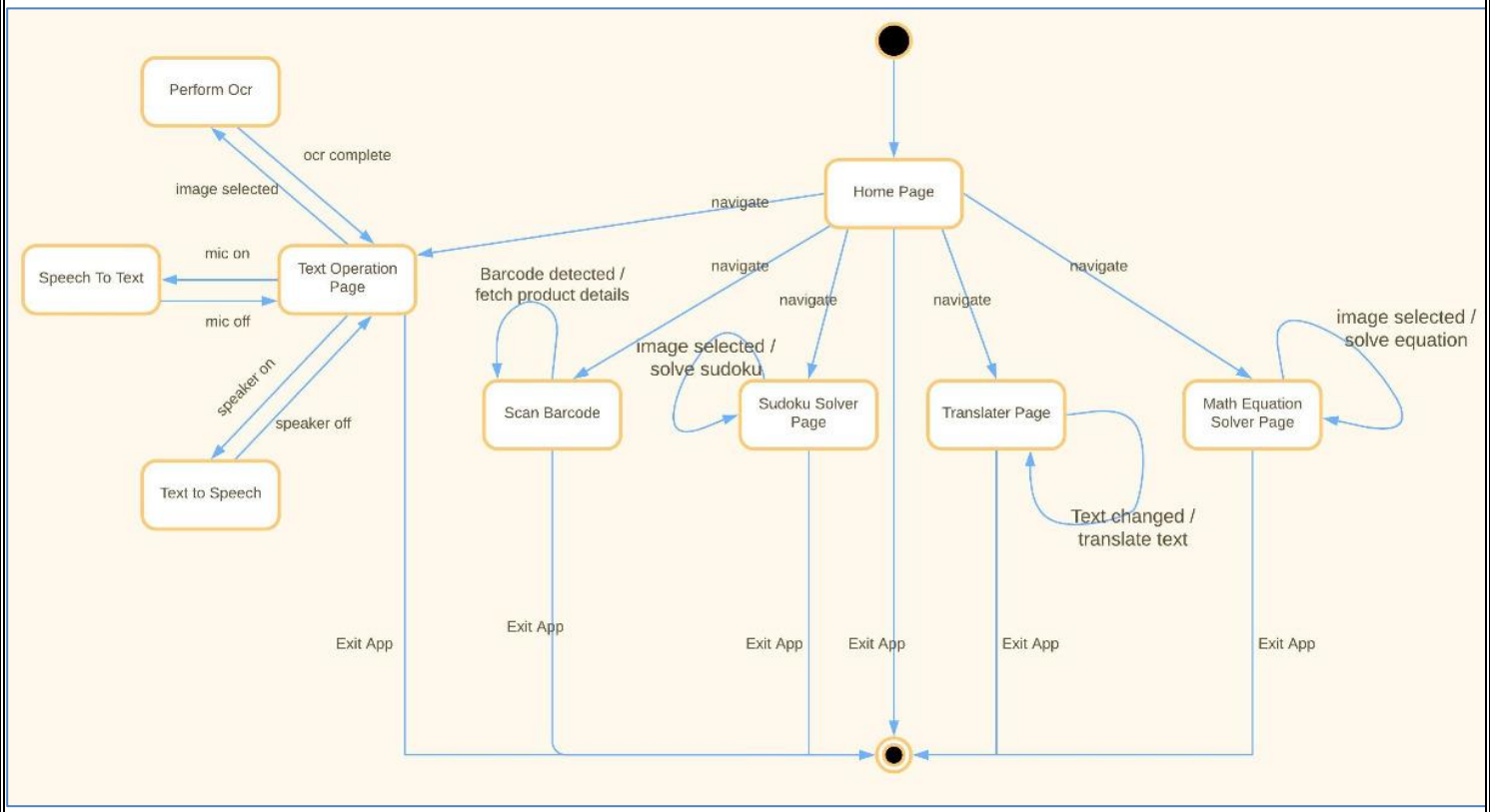


Fig. 3.3: State Diagram

3.4 Activity Diagram

For Sudoku Solver	For Math Equation Solver

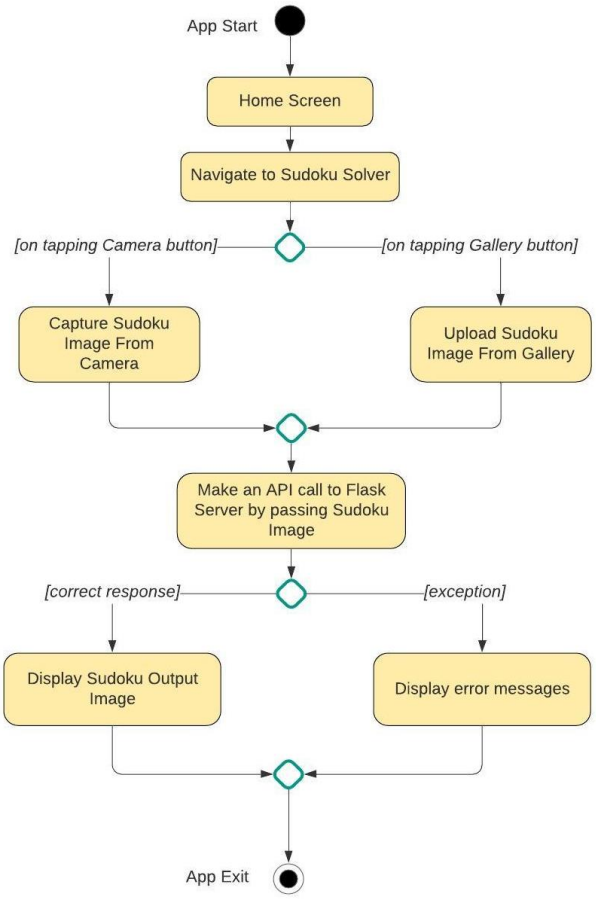
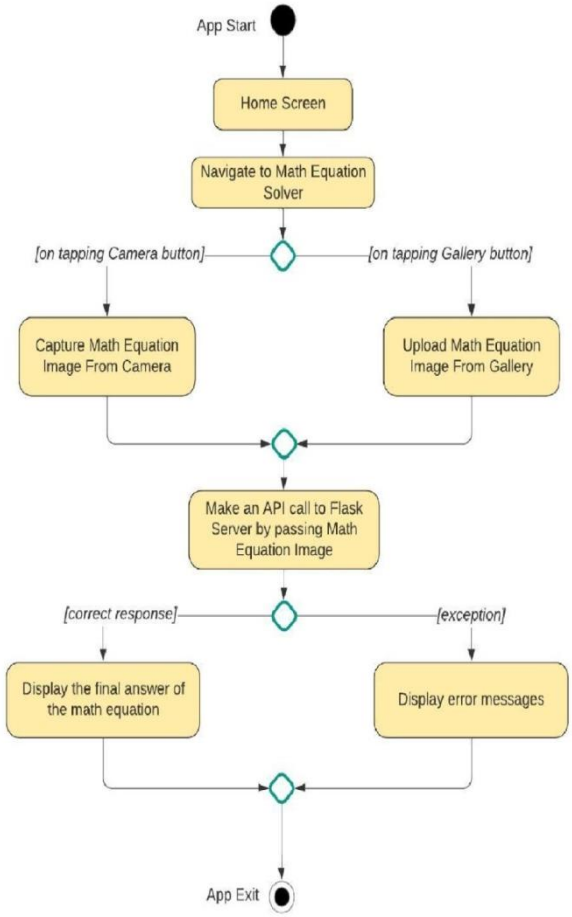

	

Fig. 3.4.1: ***Sudoku Solver Activity Diagram***	***Fig. 3.4.2:*** ***Math Equation Solver Activity Diagram***

For Translator	For Barcode to Product Details

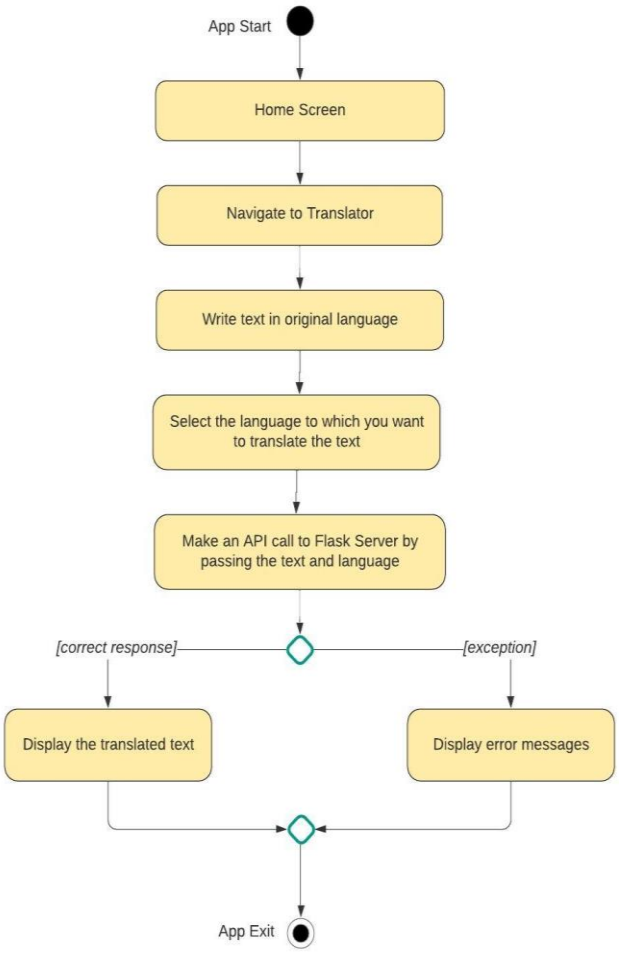
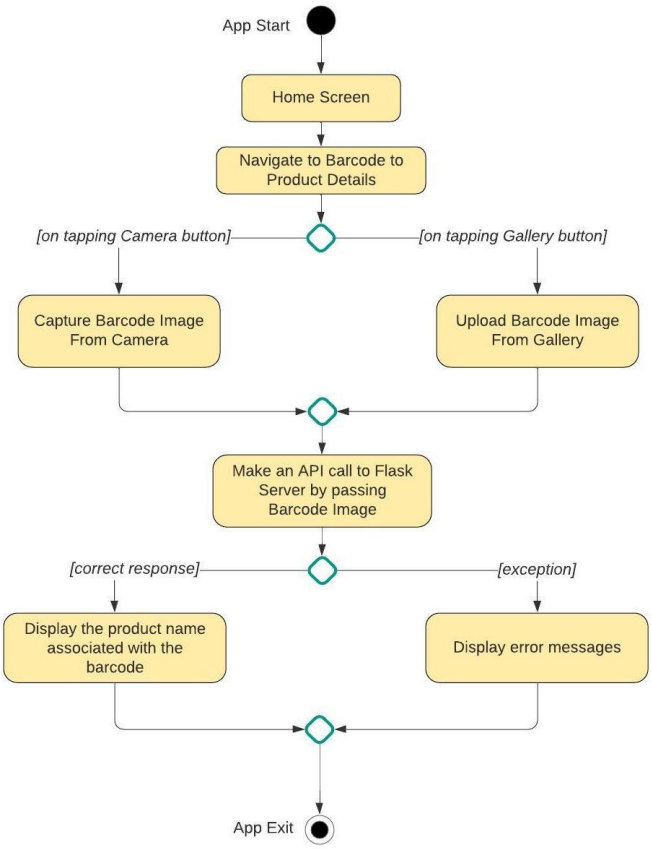

	

Fig. 3.4.3: ***Translator Activity Diagram***	***Fig. 3.4.4:*** ***Barcode to Product Details Activity Diagram***

For Text and Speech

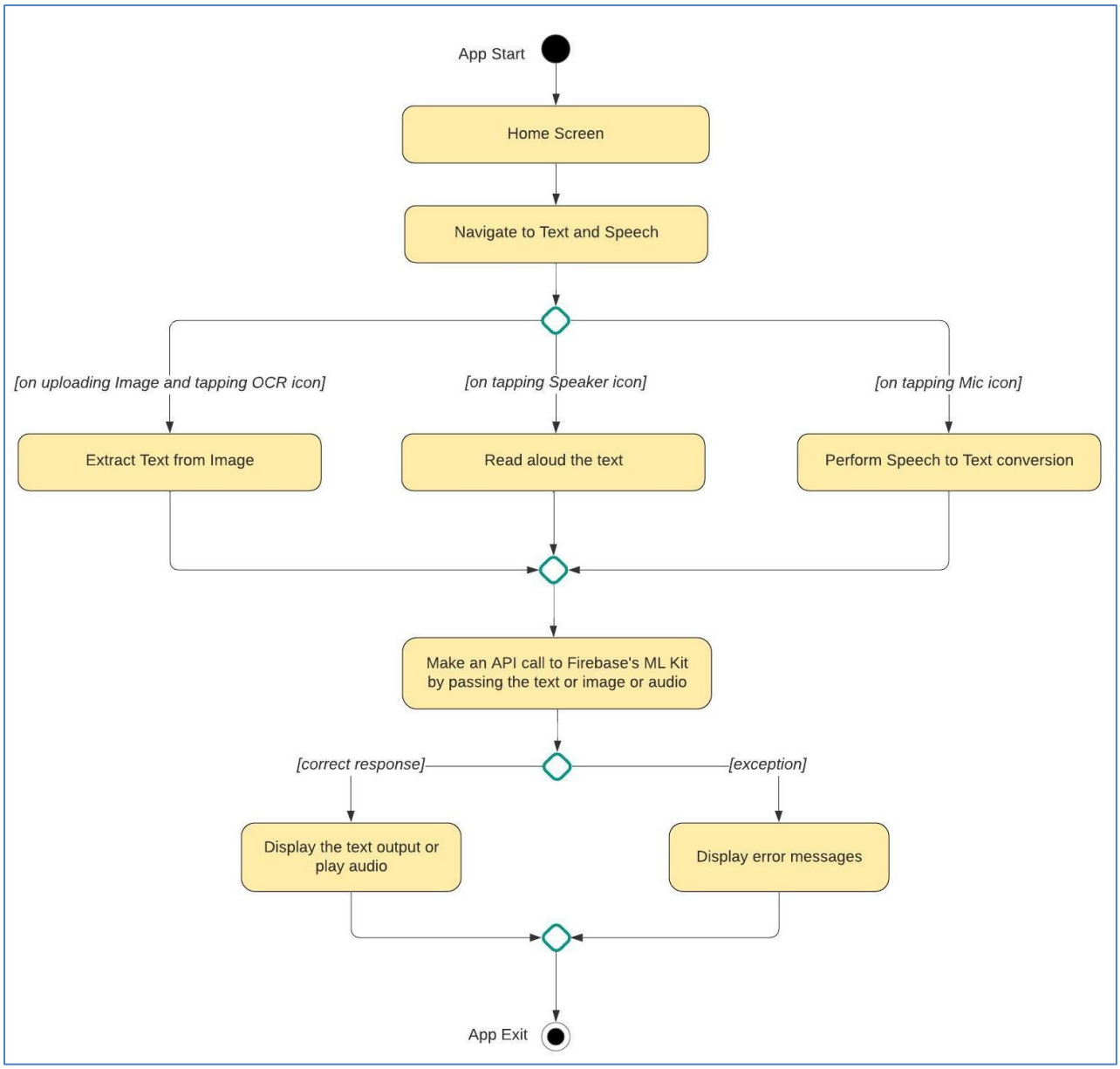


Fig. 3.4.5:
Text and Speech Activity Diagram

Fig. 3.4: Activity Diagrams

3.5 Sequence Diagram

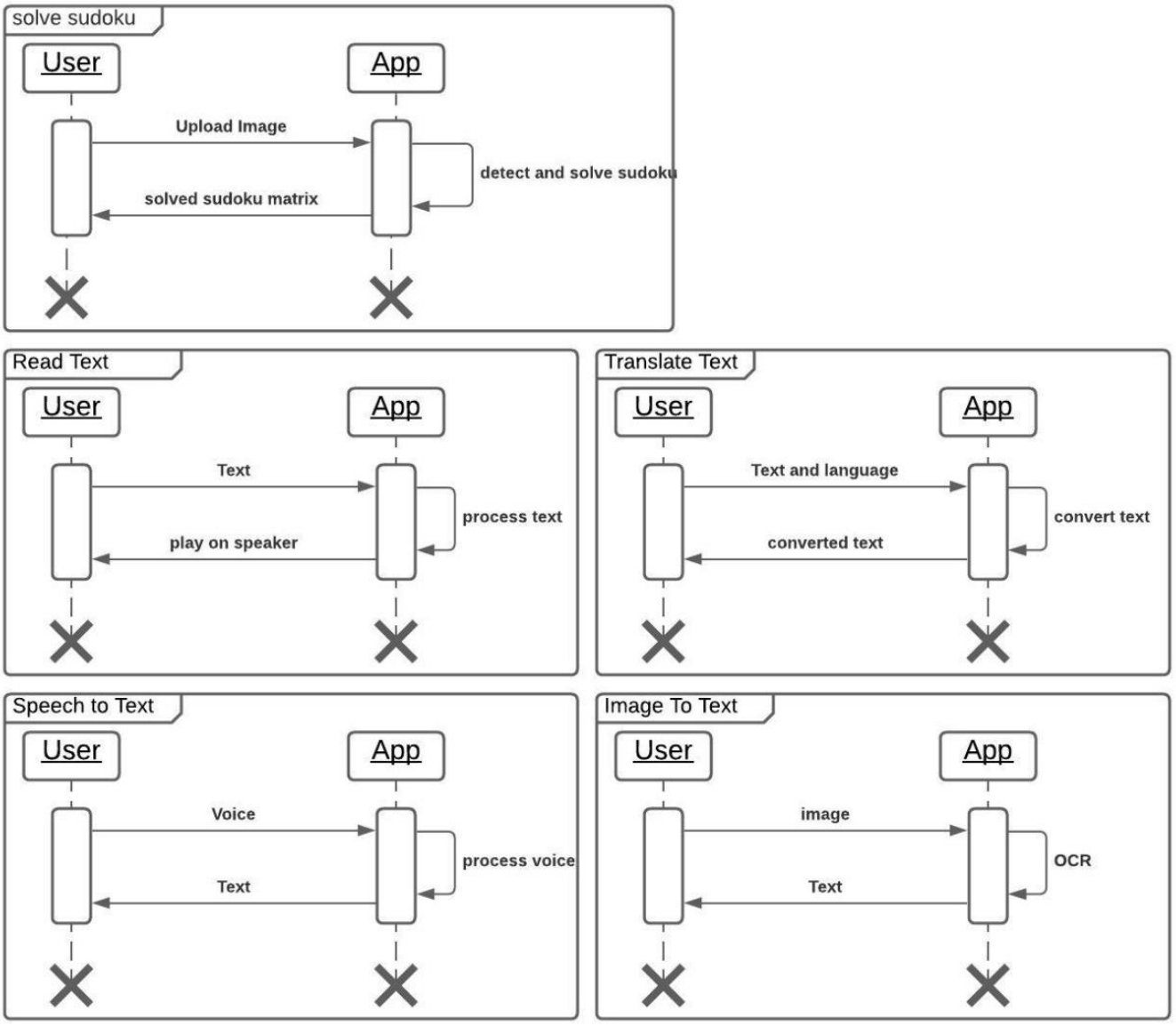


Fig. 3.5.1:
Combined Sequence Diagrams of
Sudoku Solver,
Text Reader,
Translator,
Speech to Text
and
Image to Text

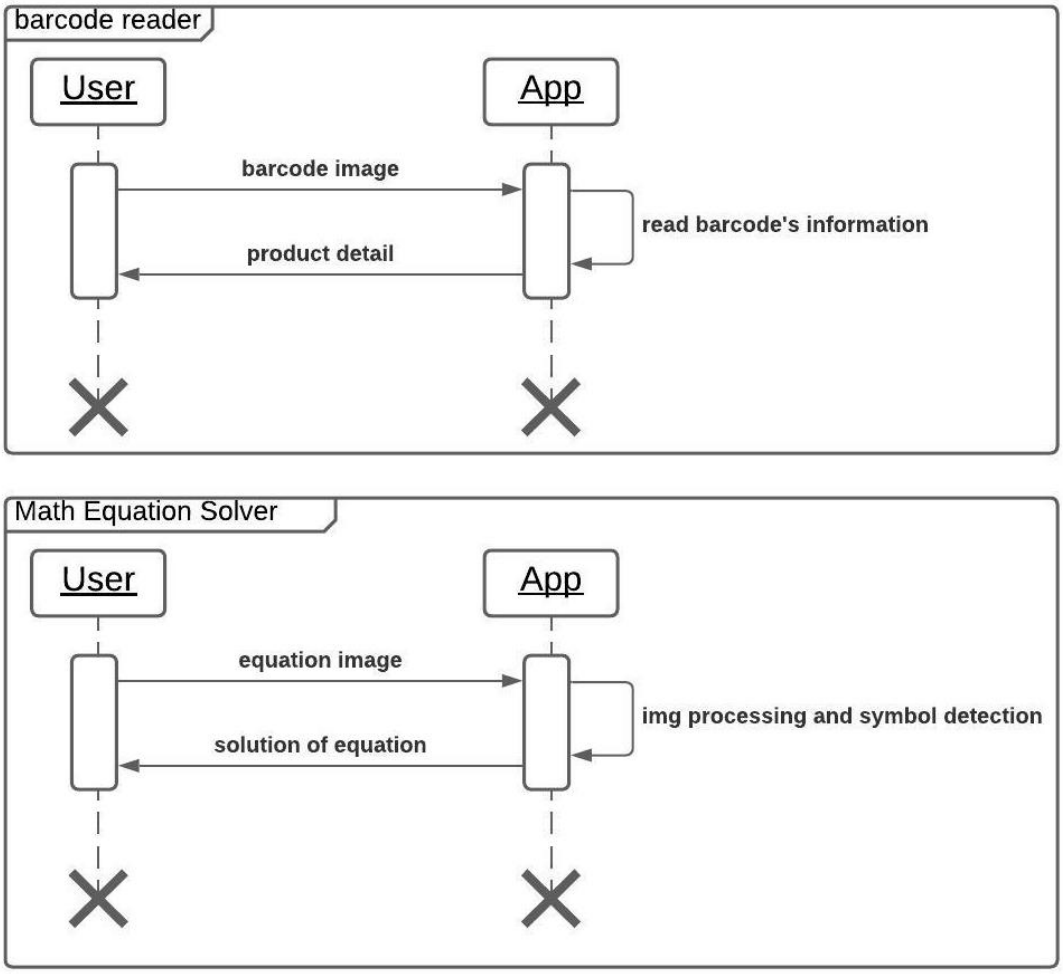


Fig. 3.5.2:
Combined Sequence Diagrams of
Barcode to Product Details
and
Math Equation Solver

Fig. 3.5: Sequence Diagrams

4. Implementation Details

4.1 Front-End Design Implementation Details

Front end is designed using the Flutter SDK created by Google. Everything in Flutter is widget. We have created required widgets separately and imported them on the required view.

Flutter App's folder structure:

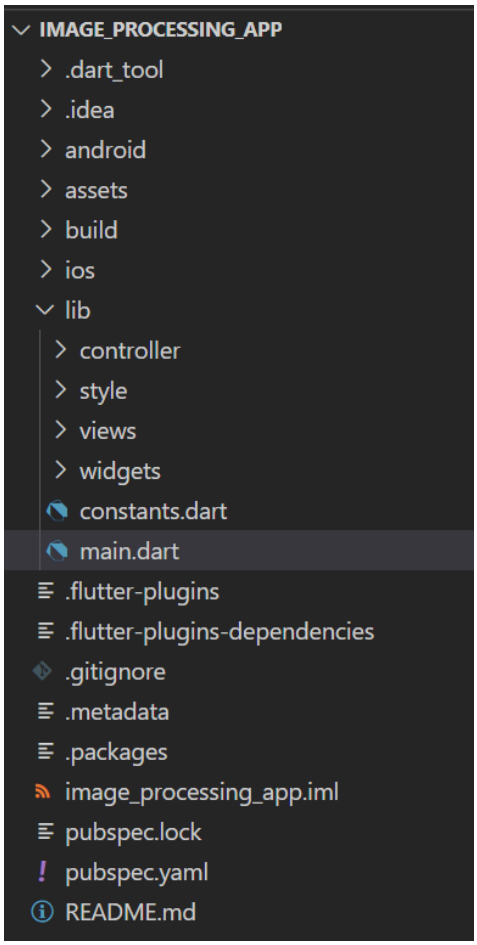


Fig. 4.1: Flutter App's Folder Structure

Folder Structure:-

1) controller – For controlling the state

2) style – Defines the style used in this app like background wave animation

3) views – Defines a complete screen of the app

4) widgets – Defines the widgets used in this app. It is reusable.

5) constants.dart – Defines the constants

6) main.dart – Entry point of our app

7) pubspec.yaml – Contains the dependencies

8) assets – contains the fonts and images

We have basically six views defined inside views folder:

- **Home View:**
  View includes card navigation to all functionalities developed.

- **Sudoku Solver View:**
  View contains two buttons to select image from camera or gallery. After selecting image, there will be solution of that Sudoku.

- **Math Equation View:**
  View contains two buttons to select image from camera or gallery. After selecting image, there will be solution of the simple math equation provided.

- **Barcode View:**
  Similar to above view it contains two buttons, after selecting image the barcode is detected in image and if product details of barcode is present then it is shown in view else will show “NOT PRESENT” message.

- **Text and Speech View:**
  View is divided into three parts as Image View, Text View and Navigation panel with different buttons like mic, speaker, gallery and camera

- **Translator View:**
  View contains text box and two language selection options. On selection of language the text is translated to destination language.

4.2 Sub Project Implementation Details

4.2.1 Sudoku Solver

Input: Sudoku Puzzle Image

Output: 9 x 9 matrix of solved Sudoku

Approach:

- Read image from http request and convert it into numpy array
- Do image preprocessing (blur, filtering) and convert it into grayscale then apply thresholding to convert it to binary image
- Now find contours (outline of object) of image and select only those which are approximately rectangle and assume such biggest one is our Sudoku's boundary.
- Now cut Sudoku from image and apply warpPerspective (flatten it)
- Detect all horizontal and vertical lines present in Sudoku and find intersection points of those lines , now we have endpoints of 81 boxes of Sudoku extract those boxes from image
- Model predicts the value present in each box and fill 9 x 9 matrix
- Solve Sudoku using backtracking technique
- Return solved matrix

Sudoku Solver's model training and validation accuracy:-

```
model=the_model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, verbose=1,
                    validation_split=validation_split)

Epoch 1/10
121/121 [==============================] - 22s 185ms/step - loss: 0.8090 - accuracy: 0.7331 - val_loss: 0.7200 - val_accuracy: 0.8448
Epoch 2/10
121/121 [==============================] - 23s 187ms/step - loss: 0.0567 - accuracy: 0.9824 - val_loss: 0.4202 - val_accuracy: 0.8910
Epoch 3/10
121/121 [==============================] - 22s 183ms/step - loss: 0.0333 - accuracy: 0.9914 - val_loss: 0.2386 - val_accuracy: 0.9701
Epoch 4/10
121/121 [==============================] - 22s 185ms/step - loss: 0.0203 - accuracy: 0.9955 - val_loss: 0.2742 - val_accuracy: 0.9597
Epoch 5/10
121/121 [==============================] - 24s 200ms/step - loss: 0.0158 - accuracy: 0.9954 - val_loss: 0.2845 - val_accuracy: 0.9716
Epoch 6/10
121/121 [==============================] - 22s 183ms/step - loss: 0.0094 - accuracy: 0.9982 - val_loss: 0.2381 - val_accuracy: 0.9672
Epoch 7/10
121/121 [==============================] - 22s 185ms/step - loss: 0.0495 - accuracy: 0.9934 - val_loss: 0.4273 - val_accuracy: 0.9657
Epoch 8/10
121/121 [==============================] - 23s 190ms/step - loss: 0.0089 - accuracy: 0.9975 - val_loss: 0.2491 - val_accuracy: 0.9672
Epoch 9/10
121/121 [==============================] - 23s 190ms/step - loss: 0.0083 - accuracy: 0.9990 - val_loss: 0.2475 - val_accuracy: 0.9731
Epoch 10/10
121/121 [==============================] - 22s 184ms/step - loss: 0.0091 - accuracy: 0.9993 - val_loss: 0.2839 - val_accuracy: 0.9716
```

Fig. 4.2: Sudoku Solver's Model Accuracy

4.2.2 Math Equation Solver

Input: Math equation image

Output: Solution of the mathematical equation

Approach:

- Read image from http request
- Do image preprocessing (blur, filtering) and convert it into grayscale then apply thresholding to convert it to binary image
- Now find contours (outline of object) of image and select only those which are approximately rectangle. Image will have white foreground and black background.
- So, it will extract the symbols from the equation image i.e. if the equation is 5 + 3, then it will extract 3 symbols viz. 5, + and 3.
- Now, each extracted symbol is passed to the model for prediction.
- Model predicts the value of each symbol and equation string is formed from it.
- Now, the equation string is passed to the eval(), which is a built-in python function for solving the mathematical expressions in string format.
- eval() function returns the answer of the equation.

Math Equation Solver's model training and validation accuracy:-

```
1174/1174 - 15s - loss: 0.0097 - accuracy: 0.9973 - val_loss: 0.1535 - val_accuracy: 0.9825
Epoch 196/200
1174/1174 - 16s - loss: 0.0057 - accuracy: 0.9981 - val_loss: 0.1498 - val_accuracy: 0.9839
Epoch 197/200
1174/1174 - 15s - loss: 0.0031 - accuracy: 0.9992 - val_loss: 0.1374 - val_accuracy: 0.9833
Epoch 198/200
1174/1174 - 15s - loss: 0.0088 - accuracy: 0.9980 - val_loss: 0.1531 - val_accuracy: 0.9815
Epoch 199/200
1174/1174 - 14s - loss: 0.0071 - accuracy: 0.9979 - val_loss: 0.1672 - val_accuracy: 0.9817
Epoch 200/200
1174/1174 - 15s - loss: 0.0063 - accuracy: 0.9981 - val_loss: 0.1525 - val_accuracy: 0.9827
```

Fig. 4.3: Math Equation Solver's Model Accuracy

4.2.3 Barcode to Product Details

Input: Barcode Image

Output: Name of the product

Approach:

- Read image from http request
- Pass the image to the decode() function of pyzbar python package
- decode() function returns the barcode number from the given image. It will internally do image processing for barcode image. We don't have to do it.
- Now, we have the barcode number. So, we can now make an API call to monster API by passing the barcode number as argument.
- API will return json object of the product details as response
- From the json object, we can obtain the product name.

4.2.4 Translator

Input: Text in Original Language, Language to which you want to translate

Output: Translated text

Approach:

- Take a text string from the user
- User will type the string in some language say L1
- User will select the language, L2 to which he/she wants to translate
- Now, we will make an API call to google_translator by passing the text string and L2
- API will return the translated text as response
- So, the translation text is displayed on the screen

4.2.5 Image to Text

Input: Image containing text

Output: Text in string format

Approach:

- Take image as input from the user
- Recognize text using flutter ML kit's OCR
- Return detected text

4.2.6 Text Reader

Input: Text string

Output: Play the text on mobile speaker

Approach:

- Read text using flutter's tts (text to speech) module for the user

4.3 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 backend server from where the functionalities are exposed through URL.
- To and Fro data transfer is done using JSON.

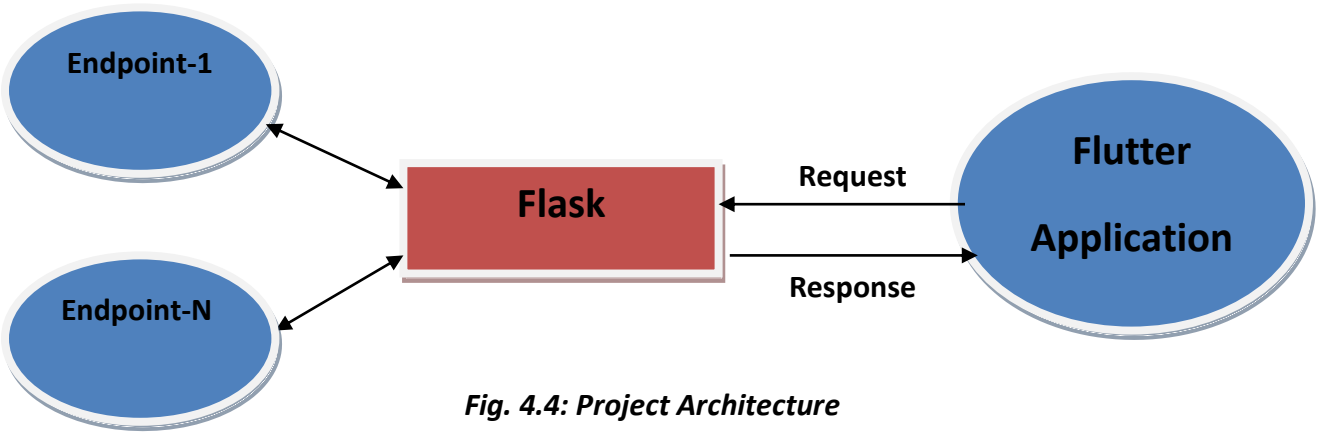


Fig. 4.4: Project Architecture

5. Testing

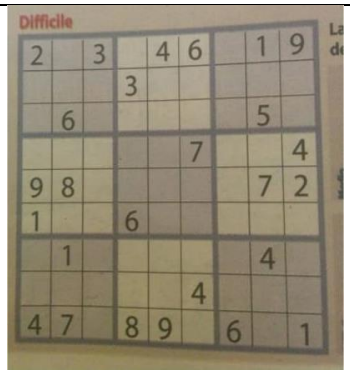
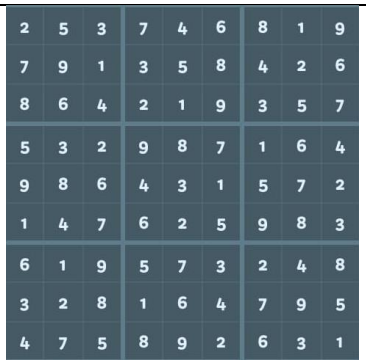
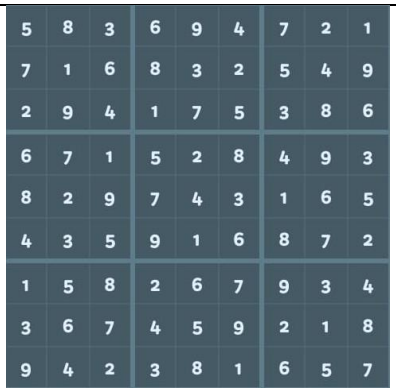
5.1 Testing Method Used

We have performed unit testing during the development. But for testing purpose, 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 note down the results in the next section.

5.2 Test Cases

5.2.1 Sudoku Solver

Test Case ID	Test Data (Unsolved Sudoku)	Actual Output (Solved Sudoku)	Pass/Fail
T1			Pass
T2			Pass

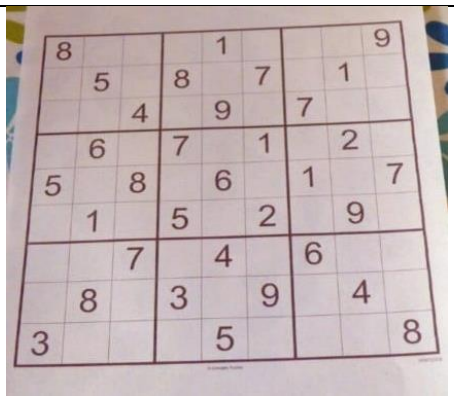
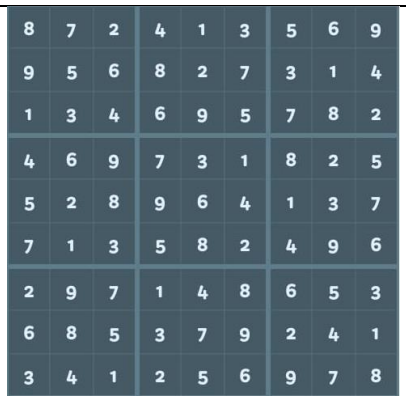
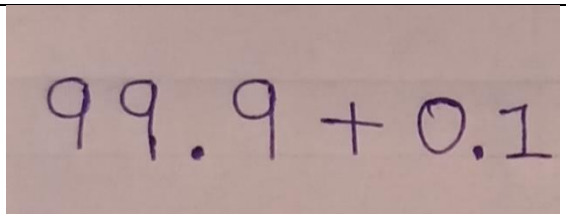
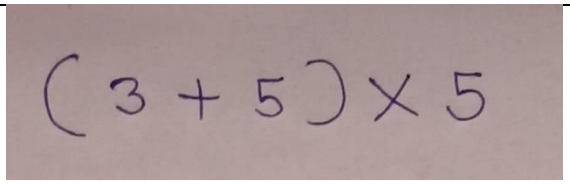
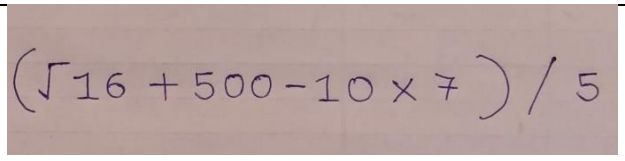
T3			**Pass**

Table 5.1: Test Cases for Sudoku Solver Project

5.2.2 Math Equation Solver

Test Case ID	Test Data (inside the Image)	Expected Output	Actual Output	Pass/Fail
T1	 99.9 + 0.1	100	100.0	**Pass**
T2	 (3 + 5) X 5	40	40	**Pass**
T3	 (\/16+500-10X7)/5	86.8	86.8	**Pass**

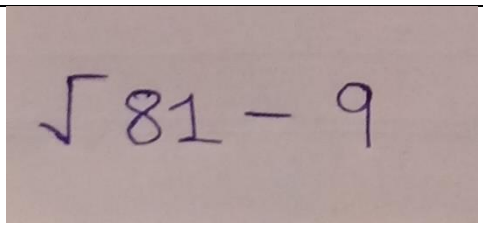
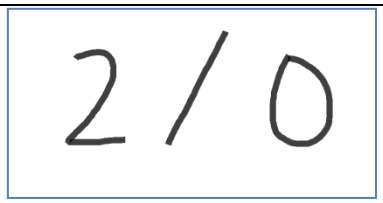
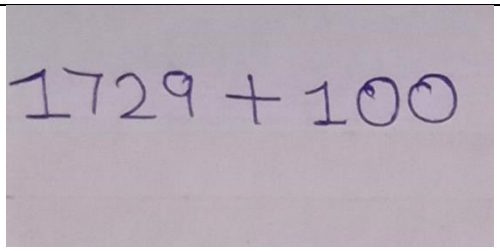
T4	 \/81 – 9	0	0.0	**Pass**
T5	 2 / 0	Can't divide by zero	You cannot perform division by zero!	**Pass**
T6	 1729 + 100	1829	1629	**Fail** Model predicted 7 as 6

Table 5.2: Test Cases for Math Equation Solver Project

5.2.3 Barcode to Product Details

Test Case ID	Test Data (Barcode Number)	Expected Output	Actual Output	Pass/Fail
T1	9789350293478	Turning Points	Turning Points: A Journey Through Challenges: Kalam, A.P.J.	**Pass**
T2	9781259028564	CAO by John P. Hayes	Computer Architecture & Org.: Hayes:	**Pass**
T3	9789339219666	Programming in ANSI C	School Textbooks & Study Guides	**Fail** partially

T4	9788120321410	Intro. to Algo.	Introduction to Algorithms (International Edition)	**Pass**

Table 5.3: Test Cases for Barcode to Product Details Project

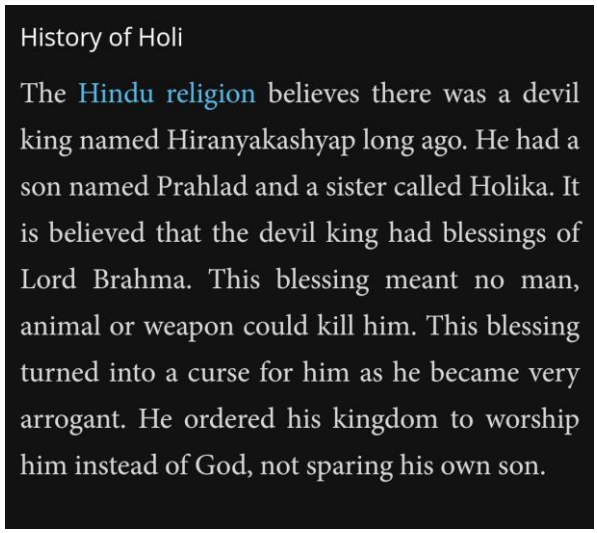
5.2.4 Translator

Test Case ID	Test Data	Expected Output	Actual Output	Pass/Fail
T1	Text : “hello” Origin : “English” Translate Into : “Gujarati”	નમસ્તે	નમસ્તે	**Pass**
T2	Text : “how are you?” Origin : “English” Translate Into : “Gujarati”	તમે કેમ છો?	તમે કેમ છો?	**Pass**
T3	Text : “Never say goodbye!” Origin : “English” Translate Into : “Hindi”	कभी अलविदा न कहना	कभी अलविदा न कहना	**Pass**

Table 5.4: Test Cases for Translator Project

5.2.5 Image to Text

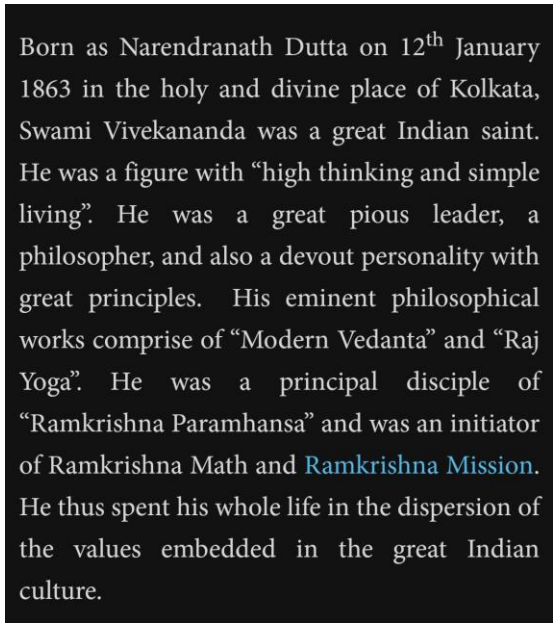
Test Case 1: (Status: **Pass** **100%)**



Text After OCR..

History of Holi The Hindu religion believes there was a devil king named Hiranyakashyap long ago. He had a son named Prahlad and a sister called Holika. It is believed that the devil king had blessings of Lord Brahma. This blessing meant no man, animal or weapon could kill him. This blessing turned into a curse for him as he became very arrogant. He ordered his kingdom to worship him instead of God, not sparing his own son.

Test Case 2: (Status: **Pass** 99%)



Text After OCR..

Born as Narendranath Dutta on 12n January 1863 in the holy and divine place of Kolkata, Swami Vivekananda was a great Indian saint. He was a figure with "high thinking and simple living He was a great pious leader, a philosopher, and also a devout personality with great principles. His eminent philosophical works comprise of "Modern Vedanta" and "Raj Yoga'. He was a principal disciple of "Ramkrishna Paramhansa" and was an initiator of Ramkrishna Math and Ramkrishna Mission. He thus spent his whole life in the dispersion of the values embedded in the great Indian culture.

5.2.6 Text Reader

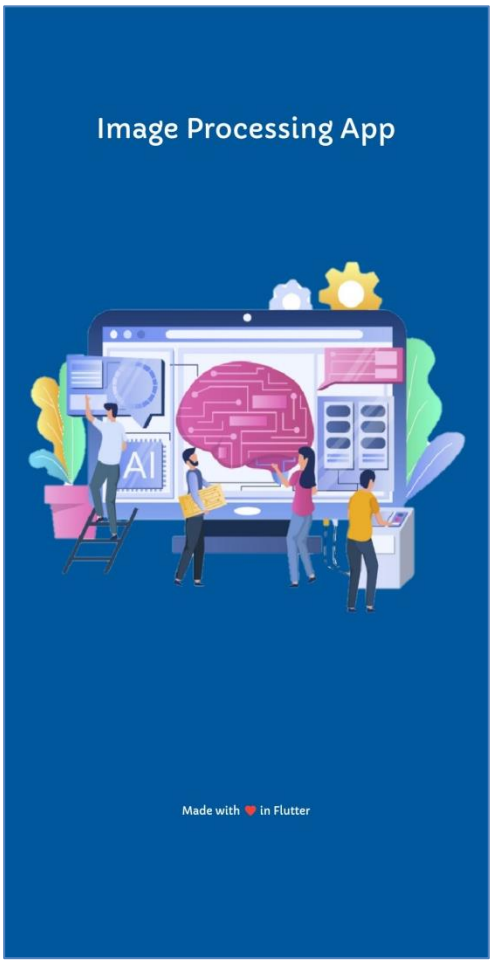
Test Case ID	Test Data	Output(read)	Pass/Fail
T1	**Diwali** is a festival of lights. It is one of the biggest and grandest festivals celebrated mainly in India. **Diwali** is celebrated in the honour of Lord Ramchandra.	**Diwali** is a festival of lights. It is one of the biggest and grandest festivals celebrated mainly in India. **Diwali** is celebrated in the honour of Lord Ramchandra.	**Pass**
T2	**Holi** is known as the festival of colours. It is one of the most important festivals in India. **Holi** is celebrated each year with zeal and enthusiasm in the month of March by followers of the Hindu religion.	**Holi** is known as the festival of colours. It is one of the most important festivals in India. **Holi** is celebrated each year with zeal and enthusiasm in the month of March by followers of the Hindu religion.	**Pass**
T3	**Christmas** is celebrated on 25th December every year. This festival commemorates the birth anniversary of Jesus Christ	**Christmas** is celebrated on 25th December every year. This festival commemorates the birth anniversary of Jesus Christ	**Pass**

Table 5.5: Test Cases for Text Reader Project

6. Screenshots

6.1 Output Screenshots for each sub project:-

6.1.1 Home Screen



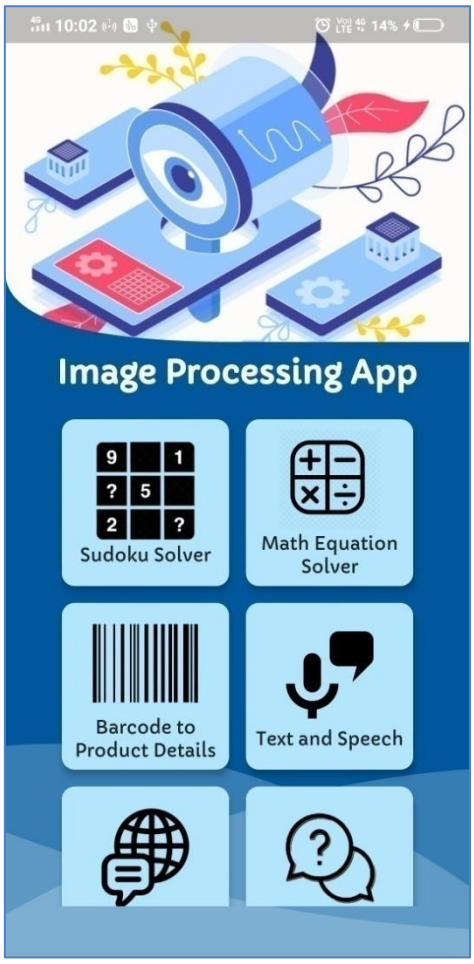


Fig. 6.1: Loading Screen ***Fig. 6.2: Home Screen***

- When a user opens the app for the first time, it displays the Loading screen for 3 seconds and then it shows the Home screen, having wave animation.
- Now, from the Home screen, user can navigate to any of the projects and can perform the image processing tasks.

6.1.2 Sudoku Solver

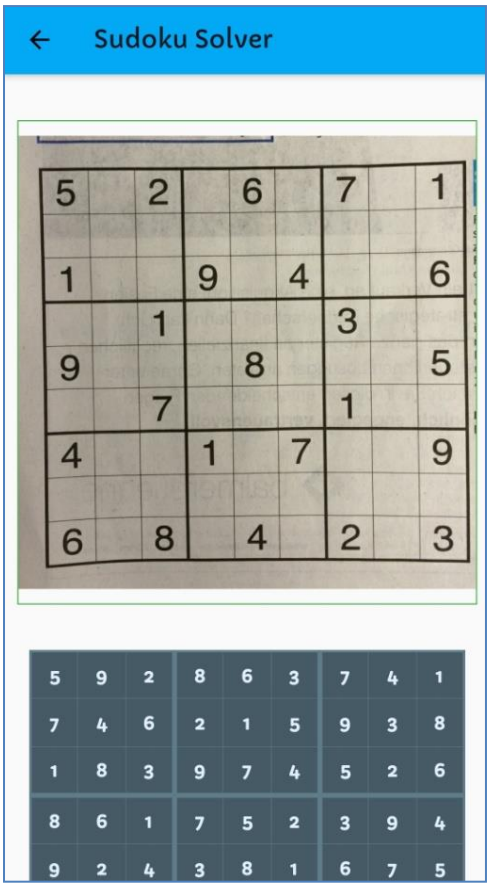


Fig. 6.3: Preview of the uploaded Sudoku Image

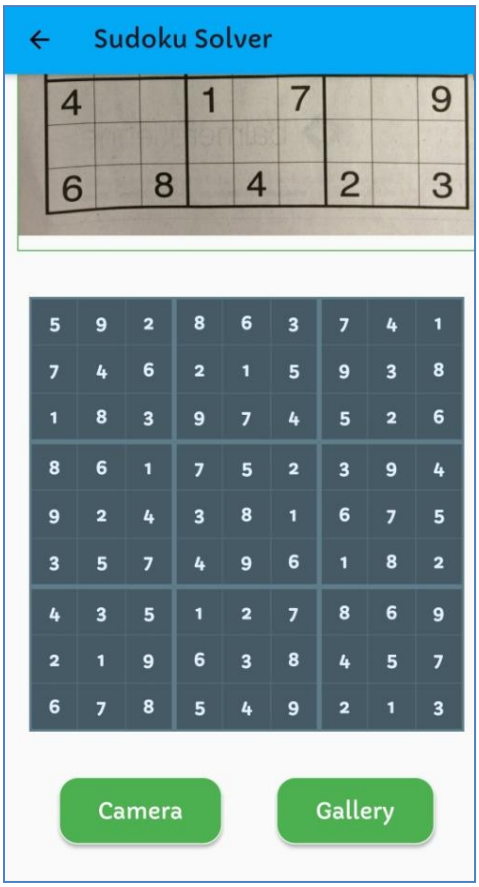
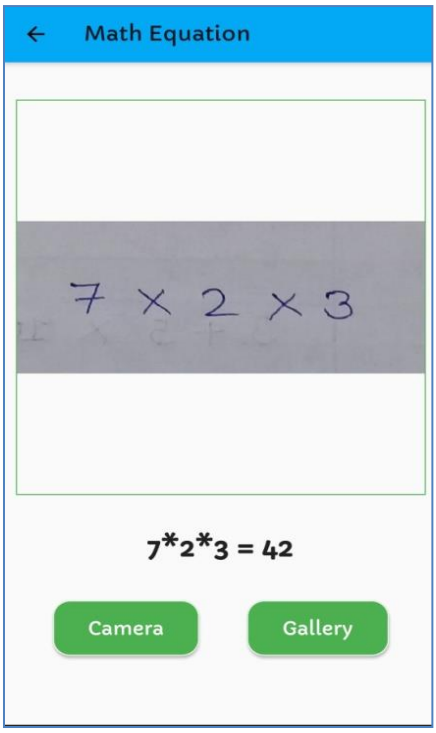


Fig. 6.4: Solved Sudoku Output

- User uploads Sudoku image from either Camera or Gallery
- Preview of the uploaded image
- After uploading the image, the final output of Solved Sudoku is displayed to the user.

6.1.3 Math Equation Solver



- User uploads math equation image from either Camera or Gallery.
- Preview of the uploaded image
- After uploading the Image, the final output of the equation string and the answer is displayed

Fig. 6.5:
Math Equation Solver Output

6.1.4 Translator

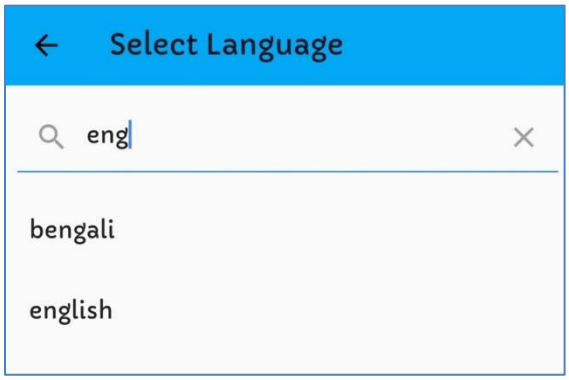


Fig. 6.6: Select your Source language

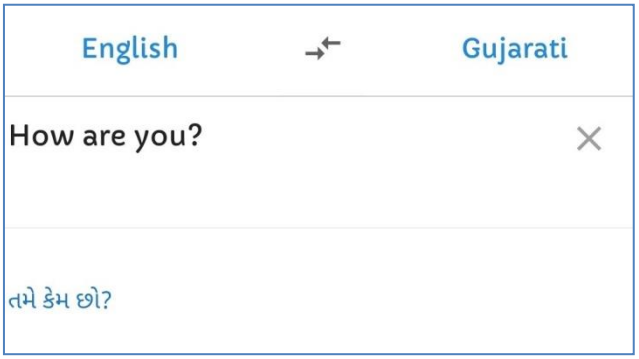
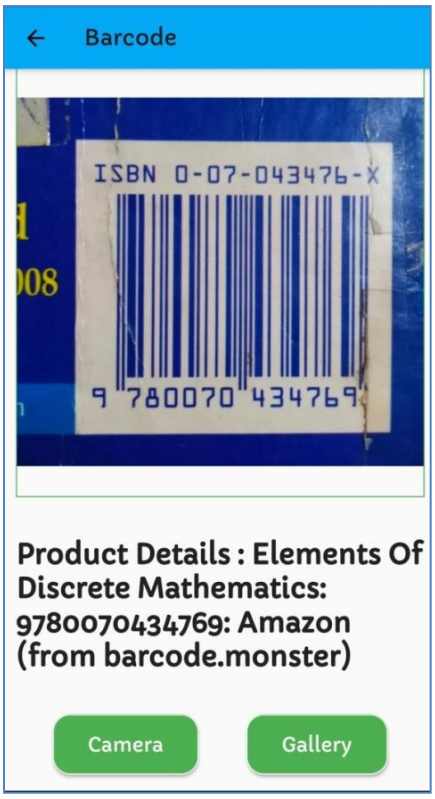


Fig. 6.7: Translated Text Output

- Select your source (from left side) and destination (from right side)
- Type the text in source language
- Text is translated (on the right side)

6.1.5 Barcode to Product Details



- User uploads Barcode image from Camera or Gallery
- Preview of the uploaded image
- After uploading the image, it displays the book name from Barcode Image

Fig. 6.8:
Barcode to Product Details Output

6.1.6 Image to Text and Text Reader

Fig. 6.9: Text and Speech

- User uploads a text image from Camera or Gallery
- Preview of the uploaded image
- After uploading the image, user can tap on the following icons, to do the following:-

1) OCR - Extract text from image
2) Mic - Do speech to text
3) Speakers - Play the text extracted
4) Gallery - Upload image from gallery
5) Cameras - Capture image
6) Copy - Copying the text on clipboard
7) Trash - Clear the text

6.1.7 Help

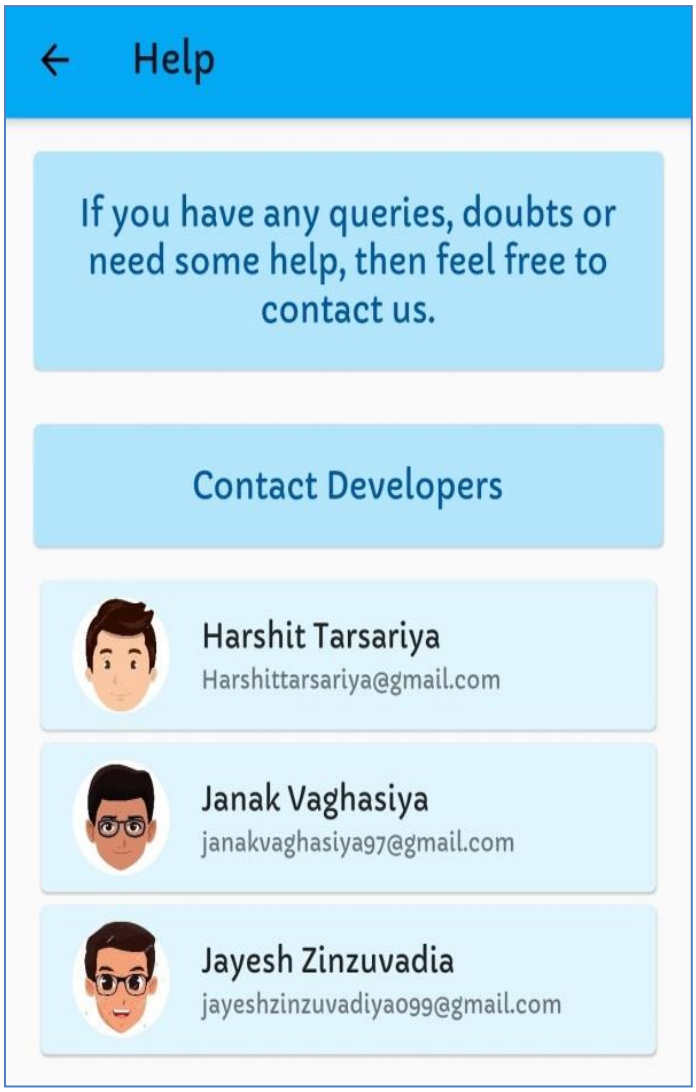


Fig. 6.10: Help Screen

- The purpose of Help page is to provide support and to take feedback from the users about the app functioning and what new features they want to add.
- Also, developer's email addresses are provided; through which users can directly interact with the developers and can ask queries or doubts.

7. 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, we have selected 6 project definitions which are related to image processing. Among this we have implemented 2 projects i.e. Sudoku Solver and Math Equation Solver from the scratch, where we have done the image processing and train the model using machine learning and deep learning by providing images to the dataset. For the remaining 4 projects, we are making API calls to the respective endpoints.

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

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

"A picture is a poem without words"

-Horace

8. Limitation and Future Extension

8.1 Limitation

- For all the sub projects - It will not work with images which are unclear or some blurriness and has less intensity and brightness.
- Sudoku Solver works well with printed images only but not working for handwritten Sudoku images.
- For inverted Sudoku image and with some change in the rotation/orientation of the Sudoku image, it is not working.
- Also, for thin bordered Sudoku image, it will not be able to detect the Box accurately.
- In Math Equation Solver project, it will work with only those input images which have black foreground and white background color. It is not working with equations written on ruled pages of a notebook.

8.2 Future Extension

- Currently, in the application, first we upload/capture the image, and then it makes an API call and process the result and returns the output. So, next thing that we can extend from here is to give live or frame by frame response to the user i.e. the user needs not to upload/capture the image. All the tasks will happen automatically behind the scene.
- Improvement in the orientation limitation of the Sudoku Solver project
- Improvement in the preprocessing part of Math Equation Solver so as to work well with the equations written on ruled pages.
- Improving the model accuracy and adding new operations like Integration, Differentiation and Trigonometry to Math Equation Solver project.

9. Bibliography

9.1 Following links and websites are referred during the development of this project for:-

9.1.1 Sudoku Solver Project

Description	References
Sudoku dataset	http://bit.ly/3coVwhI
Image processing and Sudoku detection	http://bit.ly/3viV5hG http://bit.ly/38xN0fk http://bit.ly/38xh7U7 http://bit.ly/38xNguO

Table 9.1: References for Sudoku Solver Project

9.1.2 Math Equation Solver Project

Description	References
Handwritten Math Symbols Datasets	https://www.kaggle.com/xainano/handwrittenmathsymbols https://www.kaggle.com/clarencezhao/handwritten-math-symbol-dataset
GFG Article on Handwritten Equation Solver in Python	https://www.geeksforgeeks.org/handwritten-equation-solver-in-python/
OpenCV Python Tutorial for Beginners playlist by Programming Knowledge	https://youtube.com/playlist?list=PLS1QulWo1RIa7D1O6skqDQ-JZ1GGHKK-K

YouTube channel	
Handwritten Equation Solver by Vaibhavi Malik, DSC USICT YouTube channel	Part 1: https://youtu.be/uMKF3bpH9Ec Part 2: https://youtu.be/1VBuDk21J44
Keras – Python Deep Learning Neural Network API playlist by deeplizard YouTube channel	https://youtube.com/playlist?list=PLZbbT5o_s2xrwRnXk_yCPtnqqo4_u2YGL
StackOverflow	https://stackoverflow.com/

Table 9.2: References for Math Equation Solver Project

9.1.3 Translator Project

Description	References
Google translator API	https://translate.google.com/?hl=en&tab=TT

Table 9.3: References for Translator Project

9.1.4 Barcode to Product Details Project

Description	References
How to detect QRCode and Barcode using OpenCV in Python by Murataza's Workshop YouTube channel	https://youtu.be/SrZuwM705yE
Barcode Monster API for Fetching product details from Barcode Number	https://barcode.monster/api/

Table 9.4: References for Barcode to Product Details Project

9.1.5 Image to Text Project

Description	References
OCR	https://pub.dev/packages/firebase_ml_vision
Flutter learning	https://github.com/Solido/awesome-flutter

Table 9.5: References for Image to Text Project

9.1.6 Text Reader Project

Description	References
Text to speech	https://pub.dev/packages/flutter_tts

Table 9.6: References for Text Reader Project

9.2 Following links and websites are referred for creating this project report:-

Description	References
Information about Image Processing – Used for abstract and introduction	https://sisu.ut.ee/imageprocessing/book/1
For project report format	From project report writing guidelines and some of the senior's reports
For any queries, Google it	www.google.com

Table 9.7: References for Project Report