Mobile Application for Vehicle Information Extraction using AI :Vinfo

Internship report submitted in partial fulfilment of the requirements

for the degree of B.Tech. (and M.Tech (for DD))

by

Maria Philna Aruja
(Roll No: CED17I028)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING, KANCHEEPURAM

December 2021

Certificate

I, **Maria Philna Aruja**, with Roll No: **CED17I028** hereby declare that the material presented in the Internship Report titled **Mobile Application for Vehicle Information extraction using AI : Vinfo** represents original work carried out by me in the **Department of Computer Science and Engineering** at the **Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram** during the year **2021**. With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: December 9, 2021 Student's Signature

In my capacity as supervisor of the above-mentioned work, I certify that the work presented in this Report is carried out under my supervision, and is worthy of consideration for the requirements of internship work during the period May 2021 to October 2021.

Advisor's Name: Advisor's Signature

Abstract

Vehicular data is used by different agencies/departments of the government to track the vehicles and to ensure that the vehicles on the road abide by the rules framed by the government. Certificates provided by the government ensures the eligibility of automobiles to run on the road. Insurance and fitness are two such certificates provided by the government. These certificates have to be renewed from time to time and are indispensable to run the vehicle on the road. It is an offense to not have these certificates updated. Missing the renewal date of the certificates leads to a lot of consequences.

Having an application that tells the user about the fitness and insurance status would help in such a situation. Government can also make use of this application.

This project uses a combination of yolov4, OpenCV techniques, and TPS-ResNet-BiLSTM-Attn for number plate recognition and expo for front-end development. It takes the image of the vehicle as input. The number plate in the image is recognized, and all relevant information is displayed.

Acknowledgements

I like to express my sincere gratitude to Dr. Rahul Raman, Assistant professor IIITDM Kancheepuram, for the constant guidance throughout the internship.

I express my heartfelt thanks to my family members, especially my brother, for their prayers and motivation.

I would also like to thank all my friends, especially Mr. Venkat Ramanan, for their constant help and support.

Contents

List of Figures

List of Tables

Abbreviations

TPS	**T**hin - **P**late **S**pline
ResNet	**Res**idual **Net**work
BiLSTM	**B**idirectional **LSTM**
YOLO	**Y**ou **O**nly **L**ook **O**nce

Chapter 1

Introduction

A vehicle registration number is a sequence of alphanumeric characters that uniquely identify a vehicle. This number appears on the number plate, and it can provide relevant information associated with the vehicle, including the fitness and insurance status. Hence an automated system to extract this number would find many applications, especially for the motor vehicle department.

Number plate recognition comprises of three main stages:

- Number plate Extraction: Extracting the number plate in the image provided.
- Character segmentation: Segmenting the characters present in the number plate.
- Character recognition: Character recognition recognizes the segmented characters. Combining the recognized characters gives us the registration number of the vehicle.

Integrating the above features into a mobile application would increase the acceptance of the solution.

1.1 Background

Background here

1.2 Motivation

Long queues of vehicles are common in major check posts in India. The authorities spend a lot of time checking the validity of the certificates, including the fitness certificate. A large portion of the queue is by the vehicles that are part of the logistics network. Hence a lot of time is wasted in these check posts for the logistics companies. These queues also indirectly affect the citizen in terms of delays in goods delivery. A technology that automates checking the validity of the certificates would save a lot of time. It also allows the recording of the vehicular data passing through a check post digitally to a centralized system. It would make the process more transparent and lucid.

1.3 Objectives of the work

To build a mobile application that :

- Takes an image as input from the user and identifies the vehicle registration number.
- Process the registration number to get vehicle details.
- Save the vehicle details as a pdf file.

Chapter 2

Methodology

2.1 Front end

Expo and react native are platforms that helps in developing, building and deploying web, ios, and Android applications. This project uses these platform to create the front end, and it uses the packages provided by them to build individual components present in the application.

The list of packages and their purposes are:

- ”expo-image-picker” : To request permission to access the gallery and upload an image to the application.
- ”react-native” : Provides components like view, text, button, touchableOpacity, and image that helps to construct each screen of the application. The stylesheet provided by this module helps in styling the application.
- ”expo-print” and ”expo-sharing” : To create and save the vehicle details as a pdf.

It also uses additional functions to compare the fitness and insurance expiry date with the current date and to submit post requests to the back end.

2.2 Back end

Back end has three main components. They are:

2.2.1 Number Plate Extraction

2.2.1.1 Detecting region of interest:

: The first step in number plate extraction is to detect the number plate from the image. Yolov4 is a model known for object detection tasks. We use a pre-trained yolov4 model to detect number plate present in the image.

2.2.1.2 Post-processing

: The detected number plate goes through a series of post-processing stages:

- Crop - After the number plate is detected it is cropped and stored. We use the coordinates of the number plate that yolov4 returns for number plate to cropping and opencv function cv2.imwrite to store the image.
- Gaussian blur - This step uses the Gaussian filter to remove noises present in the number plate. Gaussian filter is a low pass filter that filters out the high-frequency pixels. We use the opencv function ”cv2.GaussianBlur” for this step.
- Ostu thresholding - It is a technique that converts an image to a binary image. This technique chooses the value of the threshold automatically.
- Dilation - This step increases the area of the character region, making the characters more recognizable. It increases the character recognition accuracy. We use the ”cv2.dilate” function to carry out this step.

2.2.2 Character Segmentation

During the character segmentation, the individual characters are extracted from the number plate. The first step in character segmentation is to identify the various contours in the image. The relevant character images can be obtained by sorting and filtering the list of contours.

The steps involved in filtering of the contour images are as follows:

- The total height of the image should be less than six times that of the contour considered.
- The height to width ratio of the contour image should be less than 1.5.
- The total width of the image should be less than 30 times that of the contour considered.
- The total area of the contour should be less than 100 pixels.

2.2.3 Character Recognition

Four major stages associated with character recognition are:

- Transformation: Normalizing the input image for downstream stages using thin-plate spline transformation.
- Feature extraction: Feature extraction maps the input image to a representation with the help of ResNet. ResNet focuses on relevant features in the image and suppresses irrelevant ones such as font, background, etc.
- Sequence modeling: Captures the contextual information with the help of BILSTM.
- Prediction: Recognizes the characters from the feature of the images using attention-based sequence prediction.

In other words, it can be said that the TPS-ResNet-BiLSTM-Attn model [1] is used for character recognition

The table 2.1 shows the performance parameters of the TPS-ResNet-BiLSTM-Attn model, by Jeonghun Baek et. al [1]. Hence it is convincing to use this model in our usecase.

Trans	**Feat.**	**Seq.**	**Pred.**	**Accuracy**	**Time**	**Params**
TPS	**ResNet**	**BiLSTM**	**Attn**	**84.0**	**27.6**	**49.6**

Table 2.1: Performance parameters of the TPS-ResNet-BiLSTM-Attn model

Chapter 3

Work Done

3.1 Front end

The Expo platform [2] is used to build the front end of the solution. The front enables the user to use the application engagingly. The front end of the application takes the user through four windows, to arrive at the final solution. The first window, also called the home screen, lets the user discover what action he/she needs to do to initiate the process. In this window, the user is asked to click a button to upload an image. The home screen also displays a small description of the use of the app. On clicking the upload button, the user is taken to another window from which he can choose to upload the image. Once the image is selected, the user is taken to a third window where a preview of the selected image is shown. A confirmation is asked to the user through a button. On confirmation, the image is processed and a message is displayed about the progress of the same. The result of the processed image is then shown in a tabular form, which contains the details about the vehicle. The application also has a feature to save the pdf and check for other vehicle details.

3.2 Back end

The back end is powered by the flask framework [3]. The microframework helps in the processing of the image. The image goes through the following stages:

- Number plate Extraction: A yolov4 model is used to detect the number plate. The detected number plate is then extracted from the image,post-processed and stored.
- Character segmentation: This step involves cropping all the characters on the number plate. Various opencv techniques help in achieving this step.
- Character recognition: The scene text recognition model that uses a combination of thin-plate spline transformation, ResNet for feature extraction, Bidirectional LSTM for
- sequence modeling, and attention-based sequence prediction helps in character recognition.

The entire workflow diagram is shown in figure 3.1.

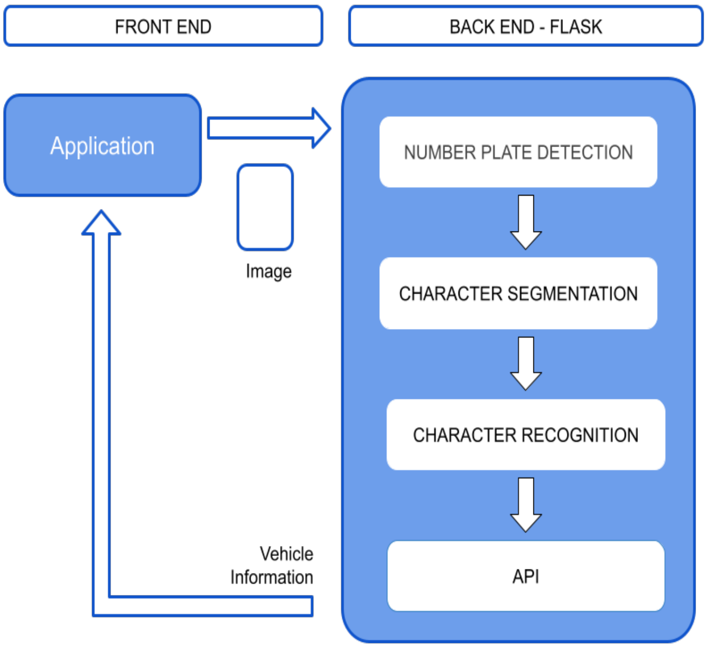


FIGURE 3.1: Worflow diagram

Chapter 4

Results and Discussions

4.1 Front end

Different screens in the application are:

4.1.1 Splash screen

Splash screen of the application is shown in figure 4.1. It is the first screen and contains the logo of the application. The logo represents the application for its meaning. The logo contains the representation of a vehicle and a representation of a document. Both the representations are circumscribed in a circle. It represents the view through a lens, apparently through a binocular. The name VINFO is taken from two words - Vehicular INFOrmation.

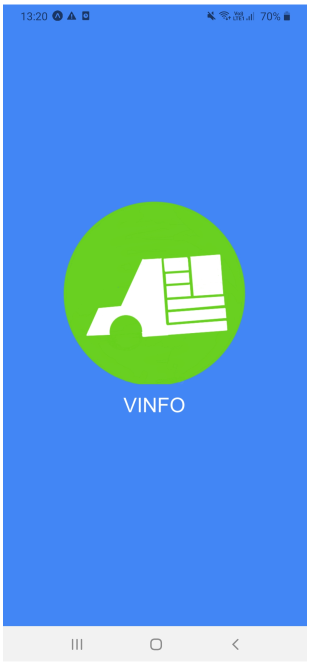


FIGURE 4.1: Splash screen

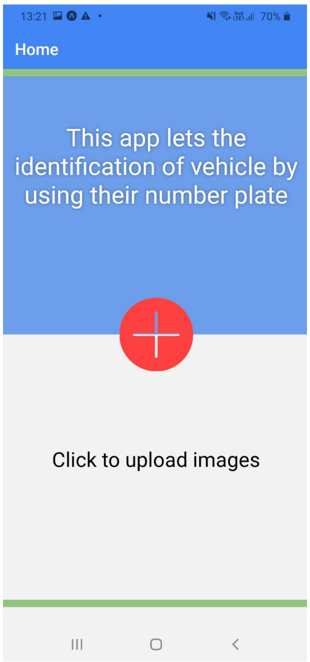
Figure 4.2 shows the logo of the application

FIGURE 4.2: logo

4.1.2 Home screen

This screen contains the description of the application. It centers a button to select an image from the gallery. It also displays a message that says ”click to upload images” which indicates the use of the button.

Figure 4.3 shows the Home screen.



13:21 70%

Home

This app lets the identification of vehicle by using their number plate

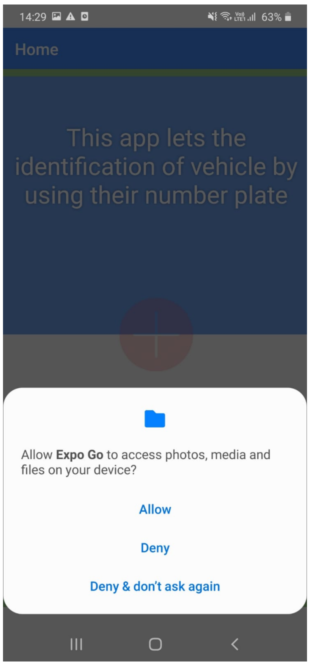
Click to upload images

FIGURE 4.3: Home screen

Various components in this screen are built with the help of the ”react-native” package. Before an application access the storage of a device it should request permission from the user to do so. The ”expo-image-picker” package is used to request this permission. This package is triggered by the upload button.

On clicking the upload button, a pop up screen shown in figure 4.4 appears requesting for permission. The image gallery shown in figure 4.6 opens up when the request is accepted , while an alert message shown in figure 4.5 pops up on declining the request.

Accepting the request lets the selection of images from the gallery.



14:29

63%

Home

This app lets the identification of vehicle by using their number plate

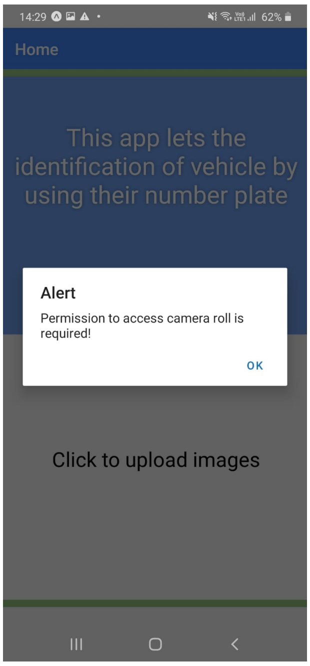
Allow **Expo Go** to access photos, media and files on your device?

Allow

Deny

Deny & don't ask again

FIGURE 4.4: Screen requesting permission



14:29

62%

Home

This app lets the identification of vehicle by using their number plate

Alert

Permission to access camera roll is required!

OK

Click to upload images

FIGURE 4.5: Request denied screen

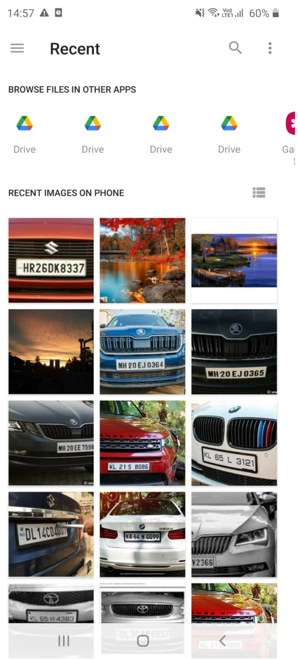


FIGURE 4.6: Gallery

4.1.3 Image screen

This screen, shown in figure 4.7, displays the selected image and a button to confirm further processing. On clicking the button, it returns an ”image is being processed” feedback, and also it passes the image to the backend.

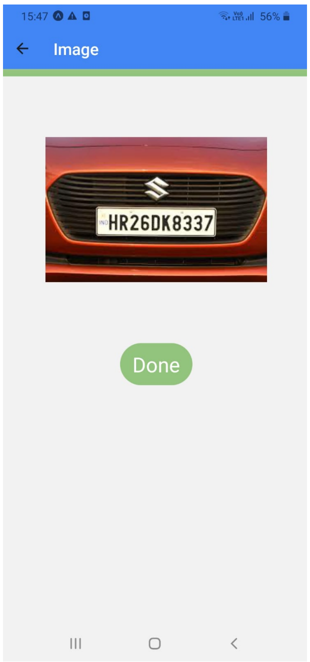


FIGURE 4.7: Image screen

The screen which displays the feedback is shown in figure 4.8.

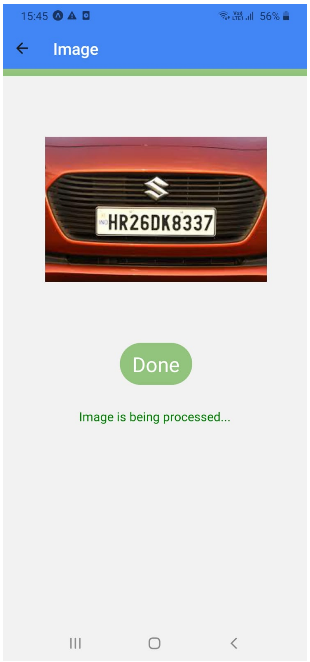
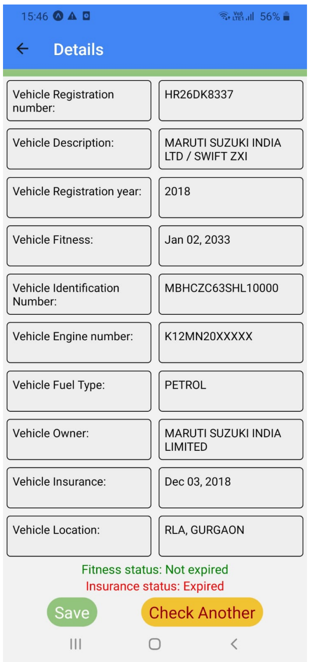


Image is being processed...

FIGURE 4.8: Image screen showing feedback

4.1.4 Details screen

The details screen displays details returned from the backend. These details include registration number, registration year, description, fitness status, insurance status, engine number, fuel type, location, and identification number of the vehicle. It also contains two buttons; “save” and “check Another”, as shown in figure 4.10. The save button creates a pdf of the details and saves it on the device. Whereas The ”Check Another” button lets the user check the details of another vehicle. It also displays the status of the vehicle’s fitness and insurance status.

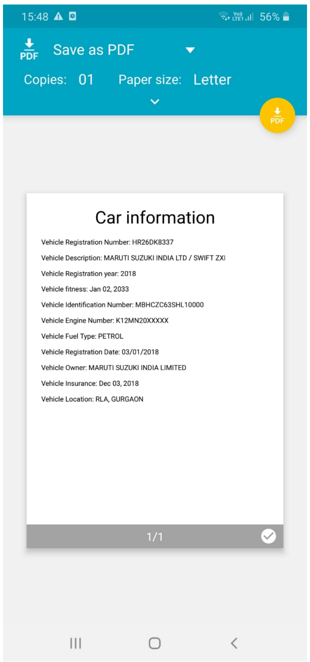


Vehicle Registration number:	HR26DK8337
Vehicle Description:	MARUTI SUZUKI INDIA LTD / SWIFT ZXI
Vehicle Registration year:	2018
Vehicle Fitness:	Jan 02, 2033
Vehicle Identification Number:	MBHCZC63SHL10000
Vehicle Engine number:	K12MN20XXXXX
Vehicle Fuel Type:	PETROL
Vehicle Owner:	MARUTI SUZUKI INDIA LIMITED
Vehicle Insurance:	Dec 03, 2018
Vehicle Location:	RLA, GURGAON

Fitness status: Not expired
Insurance status: Expired

Figure 4.9: Details screen

Figure 4.9 shows the pdf saved in the device.



Save as PDF

Copies: 01 Paper size: Letter

Car information

Vehicle Registration Number: HR26DK8337

Vehicle Description: MARUTI SUZUKI INDIA LTD / SWIFT ZXI

Vehicle Registration year: 2018

Vehicle fitness: Jan 02, 2033

Vehicle Identification Number: MBHCZC63SHL10000

Vehicle Engine Number: K12MN20XXXXX

Vehicle Fuel Type: PETROL

Vehicle Registration Date: 03/01/2018

Vehicle Owner: MARUTI SUZUKI INDIA LIMITED

Vehicle Insurance: Dec 03, 2018

Vehicle Location: RLA, GURGAON

Figure 4.10: Pdf image

The home, image, and details screen are stacked using ”react-navigation/stack” package.

4.2 Back end

The below figure 4.11 shows an example of the input image. This image is received from the front end of the application. It goes through various stages:

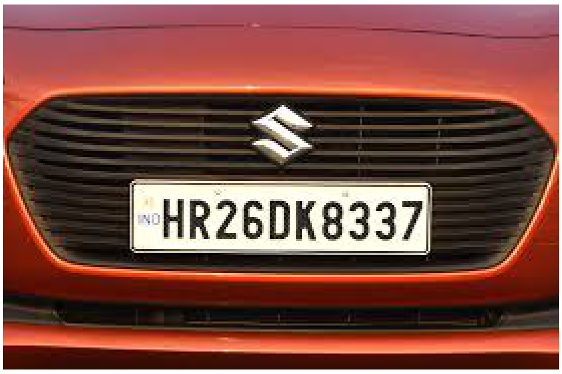


Figure 4.11: Input image

4.2.1 Number Plate Extraction

4.2.1.1 Detecting region of interest

The yolov4 model predicts the coordinates of the location of the number plate present in the image. The figure 4.12 below depicts the same.

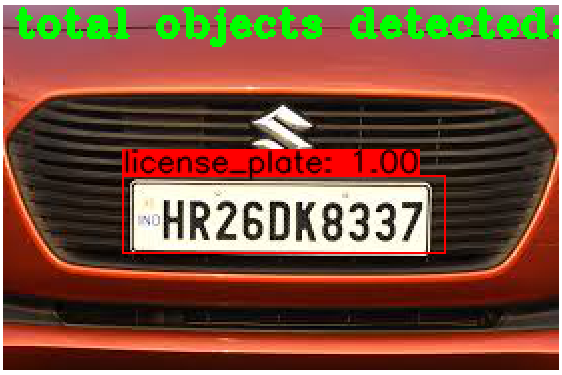


FIGURE 4.12: Detected number plate

4.2.1.2 Post-Processing

The number plate goes through a series of post-processing stages before reaching the state required for character segmentation. The output of each stage is given below:

- Crop: The area of interest is cropped from the image. The area of interest is cropped from the image. The figure 4.13 shows the result after cropping.

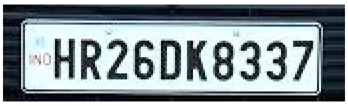


FIGURE 4.13: Result after cropping

- Gaussian blur : After cropping, the image is smoothened for better results. The figure 4.14 shows the result after smoothening:

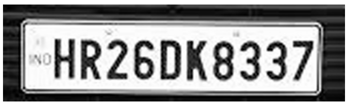


FIGURE 4.14: Result after smoothening

- Ostu thresholding : This step helps in distinguishing between the foreground and background. The smoothened image goes through ostu thresholding to aid further stages in number plate recognition. The result after ostu thresholding is shown in figure 4.15

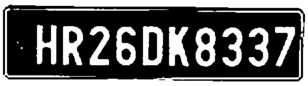


Figure 4.15: Result after thresholding

- Dilation : Dilation widens the character region and helps in better character recognition. The below figure 4.16 shows the result after dilation.

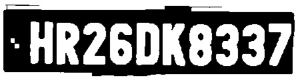


Figure 4.16: Result after dilation

4.2.2 Character Segmentation

Separating individual characters helps in better character recognition. Hence, it is an important step. Only relevant characters are considered from the regions with the help of a set of rules mentioned in 2. The result of character segmentation is shown in figure 4.17:

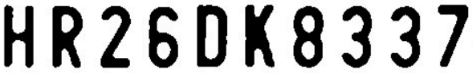


Figure 4.17: Result after character segmentation

4.2.3 Character Recognition

Character recognition predicts the individual characters present in the segmented regions with the help of a deep learning model. This model is trained for scene text recognition, and it uses a combination of TPS-ResNet-BiLSTM-Attn modules. The output of this step is the vehicle registration number. In this case, it is "HR26DK8337"

To get the vehicle details, we use an API that takes the registration number as the parameter. Then these details are returned to the front end of the application.

The time taken for various stages in the implementation are shown in the table 4.1 below:

Task	**Time Taken**
Number plate extraction	16.2s
Character segmentation	1.4s
Character recognition	6.8s

Table 4.1: Time of execution for various components in the backend

4.3 User-Interaction

Certain rules of human-computer interaction are followed while designing the application.

4.3.1 Golden rules of interface design

- Offer informative feedback: "The image is being processed" message on the image screen gives the user feedback that the action is being performed in the backend.
- Strive for consistency: Throughout the application, we use a combination of blue and green colors. All the buttons present in the app are of the same shape.
- Permit easy reversal of options : The application uses stack navigation and hence it helps in reverting to the previous stage at any point in time.

4.3.2 Fitt's law

Fitts' law states that the amount of time required for a person to move a pointer (e.g., mouse cursor) to a target area is a function of the distance to the target divided by the size of the target.

The buttons present in the application are quite big and are placed close to its description.

4.3.3 Hick and Hyman law

It says that the time taken for a user to make a decision is directly proportional to the number of choices present.

In our application, the number of choices provided is as minimum as possible.

4.3.4 Familiarity

The principles refer to the conventional practices and customs, a customer is used to. In the app, alert messages in the final solution are shown in red and successful messages/safer are displayed in green. It is used to represent the expiry status of various certificates that are relevant to the vehicle. The "+" sign on the button in the home screen clearly indicates the action that it is supposed to perform.

Chapter 5

Conclusions and Extensions

- The accuracy of the application depends on the accuracy of the character recognition model. Hence, identifying the image processing steps which improve the accuracy of the model is very important.

- The application can be improved to extract the details of a moving vehicle through video. Thus it can be integrated into a live traffic monitoring system.

- A feature to add the image through the phone camera can also be added to the application. This will increase the ease of use of the app and will allow the users to stay in the app for a longer time.

Bibliography

[1] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," *CoRR*, vol. abs/1904.01906, 2019. [Online]. Available: http://arxiv.org/abs/1904.01906

[2] Microsoft, "Discover and manage shadow it in your network," https://docs.expo.dev/.

[3] ——, "Discover and manage shadow it in your network," https://flask.palletsprojects.com/en/2.0.x/.

[4] Lubna, N. Mufti, and S. A. A. Shah, "Automatic number plate recognition:a detailed survey of relevant algorithms," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3028

[5] X. Shi, W. Zhao, and Y. Shen, "Automatic license plate recognition system based on color image processing," in *ICCSA*, 2005.

[6] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," *CoRR*, vol. abs/1802.09567, 2018. [Online]. Available: http://arxiv.org/abs/1802.09567