A PROJECT REPORT

ON

"TOUR BUDDY(MONUMENT DETECTION USING YOLO)"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

KHAN SAHIJAD ALI SAYYED ALI KAUSAR JAHAN	Roll.18DCO06
KHAN ZAID MEHMOOD SULTANA	Roll.18DCO07
SIDDIQUI SAIF SHAMSHAD	Roll.15CO40
NATIQ ABIS HUSAIN PARVEZ TARANNUM	Roll.18DCO12

UNDER THE GUIDANCE OF
PROF. ABDUL SALAM SHAIKH

DEPARTMENT OF COMPUTER ENGINEERING Anjuman-I-Islam's Kalsekar Technical Campus SCHOOL OF ENGINEERING & TECHNOLOGY Plot No. 2 3, Sector - 16, Near Thana Naka, Khandagaon, New Panvel - 410206 2020-2021

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

TOUR BUDY(MONUMENT DETECTION USING YOLO)

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

KHAN SAHIJAD ALI SAYYED ALI KAUSAR JAHAN	Roll.18DCO06
KHAN ZAID MEHMOOD SULTANA	Roll.18DCO07
SIDDIQUI SAIF SHAMSHAD	Roll.15CO40
NATIQ ABIS HUSAIN PARVEZ TARANNUM	Roll.18DCO12

UNDER THE GUIDANCE OF
PROF. ABDUL SALAM SHAIKH

DEPARTMENT OF COMPUTER ENGINEERING Anjuman-I-Islam's Kalsekar Technical Campus SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2020-2021
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"TOUR BUDDY(MONUMENT DETECTION USING YOLO)"

submitted by

KHAN SAHIJAD ALI SAYYED ALI KAUSAR JAHAN	Roll.18DCO06
KHAN ZAID MEHMOOD SULTANA	Roll.18DCO07
SIDDIQUI SAIF SHAMSHAD	Roll.15CO40
NATIQ ABIS HUSAIN PARVEZ TARANNUM	Roll.18DCO12

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engi-neering) at Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai under the University of MUMBAI. This work is done during year 2020-2021, under our guidance.

Date: 30/ 4/ 2021

Prof.Abdul salam shaikh
Supervisor

Prof. Kalpana R. Bodke
Project Coordinator

Prof. Tabrez Khan
Head of Department

Dr. Abdul Razak Honnutagi
Director

Acknowledgements

We would like to take the opportunity to express my sincere thanks to my guide Prof.Abdul salam shaikh, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.We am grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards DR. ABDUL RAZAK HONNUTAGI, Director, AIKTC, Navi Mumbai, Prof. TABREZ KHAN, Head of Department of Computer Engineering and Prof.KALPANA R BODKE, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Natiq Abis Husain Parvez Tarannum
Khan Sahijad Ali Sayyed Ali Kausar Jahan
Khan Zaid Mehmood Sultana
Siddiqui Saif Shamshad

Project I Approval for Bachelor of Engineering

This project entitled **Tour buddy(monument detection using YOLO)¨** by Natiq Abis Husain Parvez Tarannum,Khan Sahijad Ali Sayyed Ali Kausar Jahan,Khan Zaid Mehmood Sultana,Siddiqui Saif Shamshad is approved for the degree of Bachelor of Engineering in Department of Computer Engineering.

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Student Name:
Roll Number:

Student Name:
Roll Number:

Student Name:
Roll Number:

Student Name:
Roll Number:

ABSTRACT

In this paper, we have classified the famous Indian Monuments as India is 7th largest country which has large number of monuments and Historic places where millions of tourists are visits everyday but it is not possible everyone knows everything about those monuments.The paper proposes an approach for classification of various monuments based on the features of the monument images .A well known Convolutional neural network (CNN) Algorithm and YOLO has been adopted to provide on-time and striking accuracies for classifying the Monument images.The Darknet library YOLO has been used for all the training computations as YOLO is a clever convolutional neural network (CNN) for doing object recognition in real-time.The algorithm applies a single neural network to the full image, and then divides the image into multiple regions and predicts bounding boxes and probabilities for each region.After the models gets trained the overall accuracy achieved is 98,Using CNN and YOLO , for a total 20 different monuments that have been considered in the dataset for classifications.

Index Terms—YOLO, Convolutional Neural Network ,Deep Learning, Machine Learning , Image recognition .

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Monument is a type of Building or structure that has been constructed in order to commemorate a person ,event or which has become a very important part to a social group as a part of them remembering historic times or cultural heritage, or as an example of artistic, historic and political architecture.The people belonging to a various castes , cultures and religions take pride of their culturally rich heritage which is given to them in the form of monuments.
Monuments are tourist destinations in any country and due to which the economy of that country is increased and India is the 7th largest country with a large number of monuments and historical places which attract millions of tourists every day but it is not possible everyone knows everything about those monuments and because of that they have to wander there, due to which their time is wasted.

1.1 Purpose

India is 7th largest country which has large no of monuments and Historic places where millions of tourists visits everyday But no one have exact/genuine information about any of the monument and because of that the guides are make them fool ,So we are trying to develop an application which will help the users/tourists to gain the exact information about any of the monument by using their own smart phone camera ,This application will work as a virtual guide for the tourists which will help them to get a proper information related to that monument ,So user can save his/her time , money and efforts.

1.2 Project Scope

- This project is proposed to help those people who are traveller or loves to visit historical monument places using Deep learning,Machine Learning.The system allows the visitors to get all the data related to the perticuler monument the are searching for by just capturing the image of that monument.In future we'll provide a overall tour schedule to the users such as Day , Time and Places as per thier availability and seasons.

Chapter 2

Literature Survey

2.1 Paper Title 1: Cross Platform Web-based Smart Tourism Using Deep Monument Mining..

This system architecture is completely decentralized and platform independent. The user, records the image of the landmark from any perceptive using the smart phone camera.The images are sent to the nearest web-server based on the tourist location and IP.

2.1.1 Advantages of Paper

a. Its a solution to avoid the language barriers.
b. Users can gain information easily on his own smart phones.
c. which saves his/her time.

2.1.2 Disadvantages of Paper

a. Its required large no of Data sets.

b. Images recorded by tourists are usually contain a large amount of noise which requires double filtering.

2.1.3 How to overcome the problems mentioned in Paper

a. The proposed platform uses a deep neural network for feature extraction from the images captured on a mobile phone and a classification algorithm for identification of monuments in the image.

2.2 Paper Title 2: Monument Recognition using Deep Neural Networks.

In this paper, we have classified the famous Indian Monuments across the Golden Quadrilateral.Once the model is fully trained, the model is tested on a few arbitrary images to determine the test accuracy of the model.
Data Set: There are approximately 50 images per monument

2.2.1 Advantages of Paper

a. It can act as a guide for tourists visiting at the monuments.
b. Users can gain information easily on his own smart phones which saves his/her time.

2.2.2 Disadvantages of Paper

a. Provides less Accuracy (90%) as compare other algorithms.
b. Images contains large amount of noise which causes more filtering.

2.2.3 How to overcome the problems mentioned in Paper

a. the monument or landmark can be accurately identified using this classification model.

2.3 Paper Title 3: Image-based Monument Classification Using Bag-Of-Word Architecture

There are a number of representations for shape and texture. It has been seen that the Bag-of-words architecture is found out to be better than other techniques for the purpose of representation and classification. This is seen to have achieved a result of 96.07% accuracy using SVM classifier. It is better than the other hand crafted features that have been used.

2.3.1 Advantages of Paper

a. Accuracy :96.07%.

b. Dataset :459 monument image which have been classified under 9 different monument categories.

2.3.2 Disadvantages of Paper

a. large no of dataset.

2.3.3 How to overcome the problems mentioned in Paper

a. This has been achieved using various hand crafted features as well as Bag-of-Words architecture.The achieved accuracy of 96.07% using SVM classifier is good with regards to accuracy.

2.4 Technical Review

Advances in Backbone Engines, such as VGG, Deep Residual Networks (ResNet), DenseNet, Squeeze Excitation Networks (SENet), ResNeXt DetNet. Development of new detectors (such as CenterNet, DeepRegionLet, RefineDet, and others), and the upgradation of the earlier ones (e.g., new YOLO versions.) Innovations in feature representations (e.g., feature fusion and high-resolution feature learning with large receptive fields), and context encoding refinement. Localization improvements (e.g., bounding box refinement, and new loss functions) and anchor-free detection methods. Robustness in detection through rotation-invariant loss functions, scale adaptive detection, learning with enriched features, multi-task loss functions, and others. Other direct and indirect areas, such as adversarial learning, capsule networks, knowledge distillation, light-weight object detection, memory-efficient networks, and transfer learning.

2.4.1 Advantages of Technology

a. Deep Residual Networks (ResNet): For the same level of accuracy, deeper networks can be much more efficient in terms of computation and number of parameters. Deeper networks are able to create deep representations, at every layer, the network learns a new, more abstract representation of the input. A shallow network has less number of hidden layers

b. Transfer learning: Transfer learning has several benefits, but the main advantages are saving training time, better performance of neural networks (in most cases), and not needing a lot of data.

2.4.2 Reasons to use this Technology

a. Transfer of learning is highly related to retention of learning. In both cases, performance is assessed both during learning itself and during testing after learning is completed, with the test often occurring after a delay.

b. In the case of pure retention, the same tasks are examined during training and testing, whereas in the case of transfer different tasks are examined during training and testing.

c. Transfer of learning is the most important goal of computerized cognitive training. In practice, it reflects the ability to transfer what is learned in one context or situation to another.

d. Naturally, previous studies that have included assessments of both pure retention and transfer have typically shown better performance at test when pure retention is assessed (i.e., when the training task matches the testing task) than when transfer is assessed (i.e., when the training and testing tasks differ)

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Abis Natiq	Front and back end Collect dataset train the model
2	Khan Zaid	Front and back end Extend dataset
3	Khan Sahijad Ali	Back end,train model Literature Survey
4	Sidiqui Saif	Train model Literature Survey

Work Breakdown Structure

a. All of the members are equally important in developing the project.

b. We work on a different part of the project based on one's capability.

c. Firstly we came up with documentation, And based on the documentation we set our goal and created a blueprint.

d. We then started going hands-on with the project to develop it according to the flow as decided earlier.

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Abis Natiq	Team Leader	Front and back end Collect dataset train the model
2	Khan Zaid	Embedding System	Front and back end Extend dataset
3	Khan Sahijad Ali	Project Timeline Manager	Back end,train model Literature Survey
4	Siddiqui Saif	Project Timeline Manager	Train model Literature Survey

3.3 Assumptions and Constraints

a. People who are visiting historical monuments for guidience are using this Application.

b. User of his App will just have to click the photo.

c. App will response within a seconds.

d. The system will detect the monument and show the entire data present in the data set.

3.4 Project Management Approach

a. Planning of project.

b. Defining the scope of the project.

c. Estimation of time and It's management.

d. Creating Gantt Charts and properly assigning tasks to members.

e. Reporting the progress of project with the guide.

3.5 Ground Rules for the Project

a. Properly planning and gathering relevant information is very important.

b. Developing a Blueprint of the project and work accordingly.

c. All the members should report to the guide whenever required

d. Setting up small goals every week.

e. Achieving the small goal within that span of time.

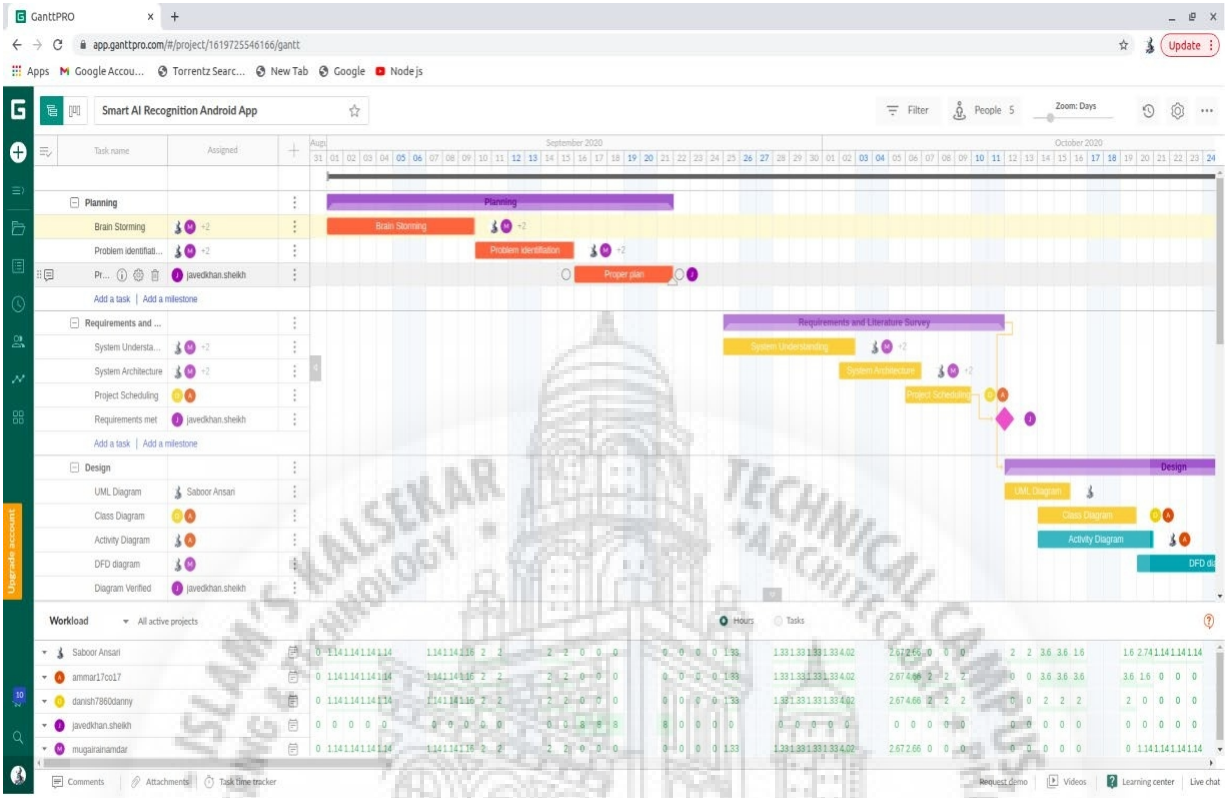
f. Keeping tracks of the progress towards project.

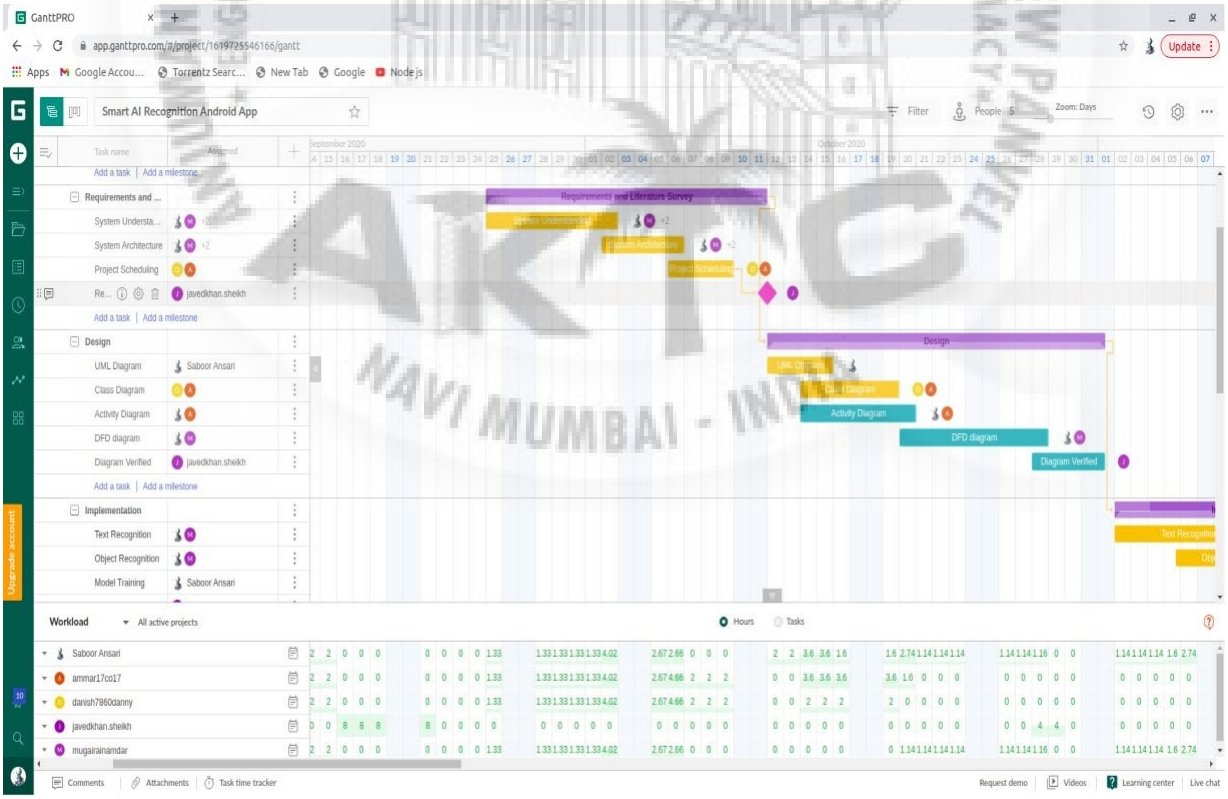
3.6 Project Budget

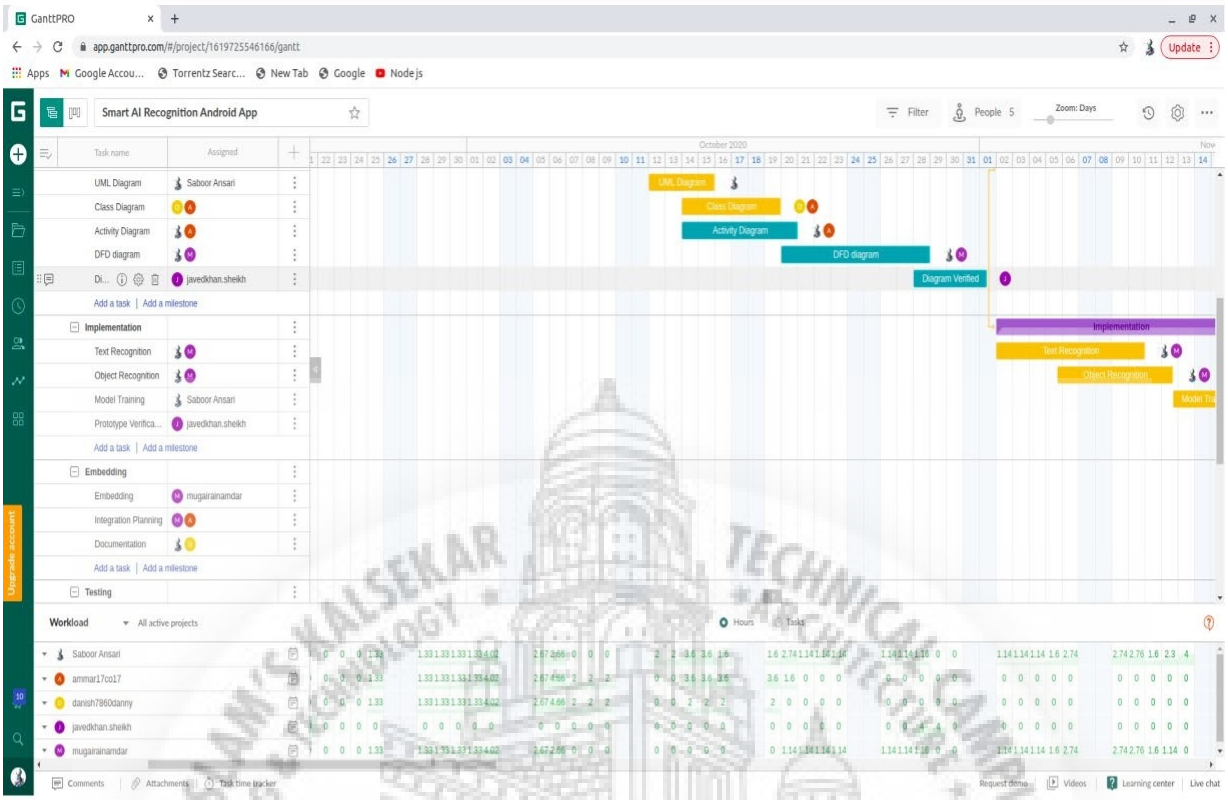
a. It is a light project.

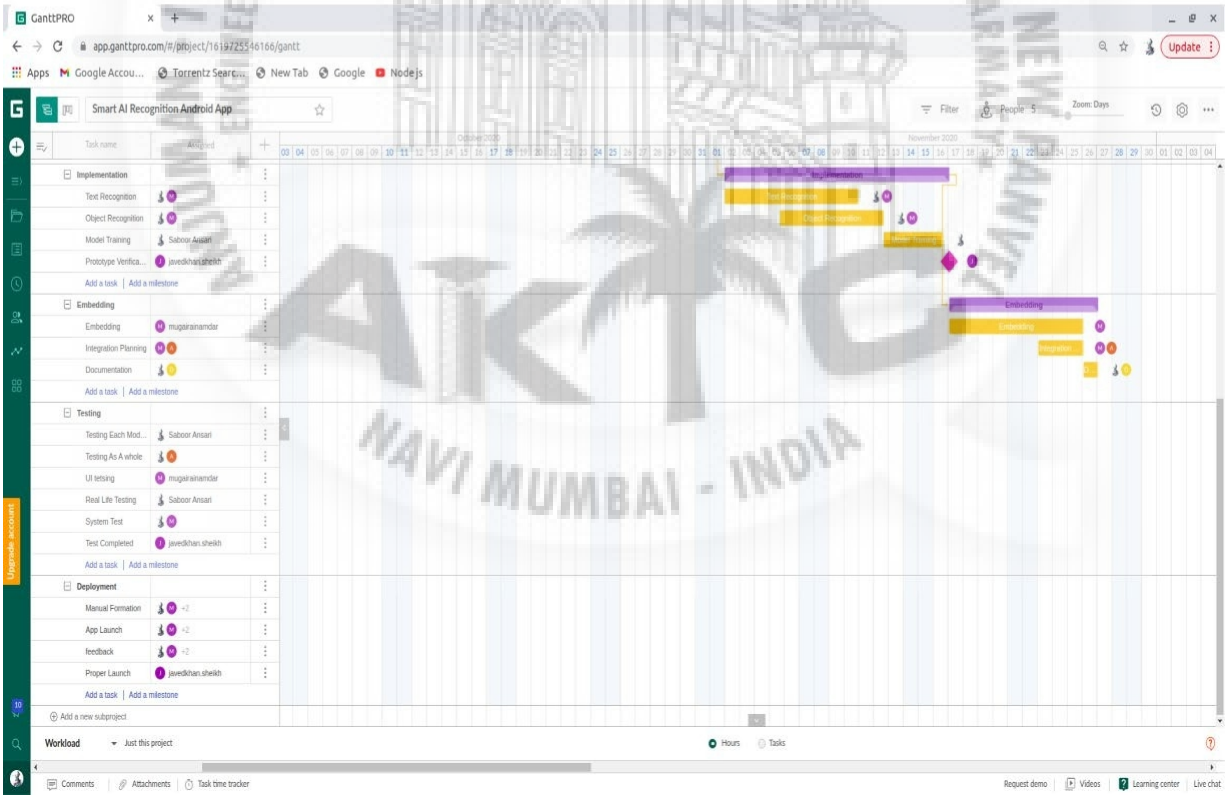
b. Cost of the project is very low and efficient.

3.7 Project Timeline






GanttPRO
app.ganttpro.com/#/project/1619725546166/gantt
Update
Smart AI Recognition Android App
Filter
People 5
Zoom: Days
October 2020
Task name
Assigned
UML Diagram
Saboor Ansari
Class Diagram
Activity Diagram
DFD diagram
Diagram Verified
javedkhan.sheikh
Add a task | Add a milestone
Implementation
Text Recognition
Object Recognition
Model Training
Prototype Verifica...
Embedding
mugairainamdar
Integration Planning
Documentation
Testing
Workload
All active projects
Hours
Tasks
Saboor Ansari
ammar17co17
danish7860danny
javedkhan.sheikh
mugairainamdar
Comments
Attachments
Task time tracker
Request demo
Videos
Learning center
Live chat


GanttPRO
app.ganttpro.com/#/project/1619725546166/gantt
Update
Smart AI Recognition Android App
Filter
People 5
Zoom: Days
November 2020
Implementation
Text Recognition
Object Recognition
Model Training
Saboor Ansari
Prototype Verifica...
javedkhan.sheikh
Add a task | Add a milestone
Embedding
mugairainamdar
Integration Planning
Documentation
Testing
Testing Each Mod...
Testing As A whole
UI telsing
Real Life Testing
System Test
Test Completed
Deployment
Manual Formation
App Launch
feedback
Proper Launch
Add a new subproject
Workload
Just this project
Hours
Tasks
Comments
Attachments
Task time tracker
Request demo
Videos
Learning center
Live chat

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

Keeping in mind the challenges faced by the tourists who loves to visit historical places we had come with an mobile application. In this paper we present a mobile application dedicated to the aid of tourists.The main aim of this application is to reduce problems faced by the tourists in single application.The goal of this project is creating monument detection application which provides services like monument detection,provide all the information about monument, show all the data to provide a proper guidance.

4.1.2 Product Features

Tour buddy provides proper guidance.user just have to capture the image of the monument.user will have all the data releted to that perticular monument.tour buddy will guide the tourists as a professional guide.

4.1.3 User Classes and Characteristics

This is project is a social project.The Users of this system are mostly those are the tourists and specially they are historical monuments lover and want to know everything about monuments.

.

4.1.4 Operating Environment

a. Python 3.7

b. React Native

c. JSON (Java Script Notation)

d. YOLO and Machine learning Algorithms (CNN) for Image recognition / Detection.

4.1.5 Design and Implementation Constraints

This system focuses one of the features at time. It is not able to provide one or more services at a time.At any instant only one of the services is accessible which is image recognition.

4.2 System Features

Image Recognition provides an information of the image given from camera modle and It can even detect and recognize every monuments present on an image.Object Recognition classifies between the known and unknown Objects. Text-to-speech this helps to read out the text and convert it into an audio file.With the help of this feature recognized text are converted into audio file and the names of objects are spoken out Speech-to-text this helps to convert speech into text.

4.2.1 System Feature

Text recognition provides detection and recognition of texts that are present on any background.

Description and Priority

This is one of the main feature of the system .This helps in extraction of text from the background. It firstly, detect the text from background.Then it recognize it eventually and convert them into text output.And pass it onto Text-to-speech module.

Stimulus/Response Sequences

Stimulus:User tells the app to open text recognition.
Response:Text recognition activity page is opened.
Stimulus:User places the text on the camera.
Response:App will automatically take picture in 5 secs .
Response:Audio of extracted texts is played on an default player of a user.

Functional Requirements

REQ-1: Authorization from camera module

4.2.2 System Feature

Image recognition provides detection and recognition of monuments images.

Description and Priority

This is one of the main feature of the system .This helps in detection of Object. It firstly, captures the image from the main window.Then it recognize and compare with the known monuments in the data set.Accordingly it classifies it and give a output as name of a monuments and its information .

Stimulus/Response Sequences

Stimulus: User tells the app to open obeject recognition.
Response: Object recognition activity page is opened.
Stimulus: User places the text on the camera.
Response: App will automatically take picture in 5 secs .
Response: The name of the object is displayed at the bottom of the screen
Response: Audio of extracted texts is played on an default player of a user.

Functional Requirements

REQ-1: Authorization from camera module

4.2.3 User Interfaces

It is very light app, so the GUI is very simple.Home pages provides one button to provide image capturing service to the users. This button is of text capturing the image.This button connects to a new window with respective services like return the lable of that particular monument.

4.2.4 Hardware Interfaces

This app requires permission of some of the hardware commodities. One need to give camera access in-order to capture the pic. One should also give storage access to save the images and the audio files. Microphone access should also given to talk back with system.

4.2.5 Software Interfaces

This software uses different libraries.Numpy library is use to compute and evaluate the images. Tensorflow Lite model is used to deal with the computer vision part. Tf lite Classifier is used to recognize text. speech recognition is used to interact with system. And also use it to match while performing the task.OS support also needed to manage the audio files.

4.3 Nonfunctional Requirements

4.3.1 Performance Requirements

Performance of overall system is very efficient and well optimize. From the time taken to capture and process it everything is well organized.While processing an image it take same time for other operations.

4.3.2 Safety Requirements

This system does not contain any critical data. Still it provide. The databases that are accessed are locally executed.In case of any updates in libraries used can lead to the failure in systems.

4.3.3 Security Requirements

All the libraries used are certified and standard.Also camera access is until the process is done completely.After that is released.

Chapter 5

System Design

5.1 System Requirements Definition

System requirement definitions specify [1] what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement:

5.1.1 Functional requirements

The basic functions that the system must provide and focus on the needs and goals of the end users are :-

a. Capture Image - Application should be able to capture image as per user instruction.

b. Detect image - Application should be able to detect the monument as per the dataset.

c. Provide proper information - Application should be able to provide provide proper information related to monument..

Level 0

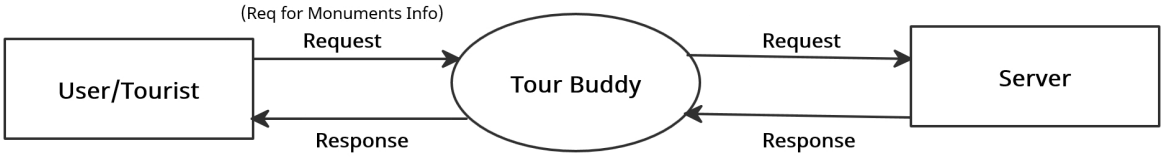


Fig : DFD level 0 for Tour buddy

Level 1

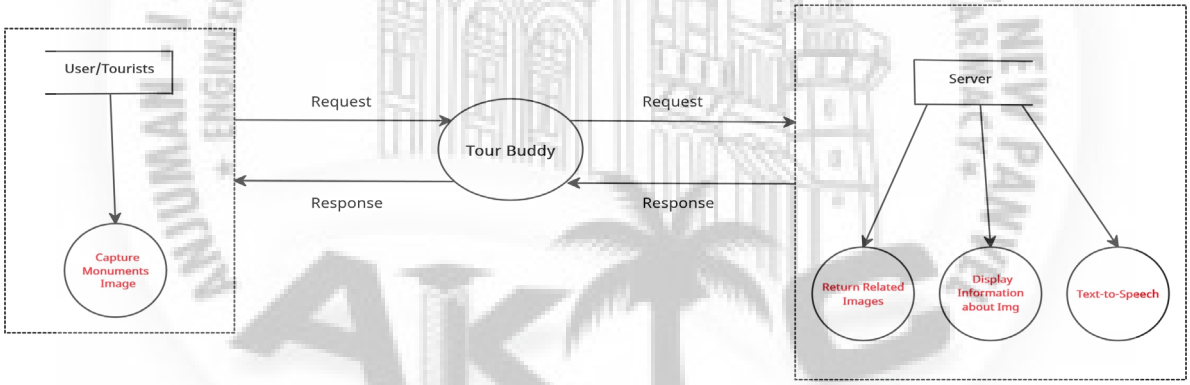


Fig : DFD level 1 for Tour buddy

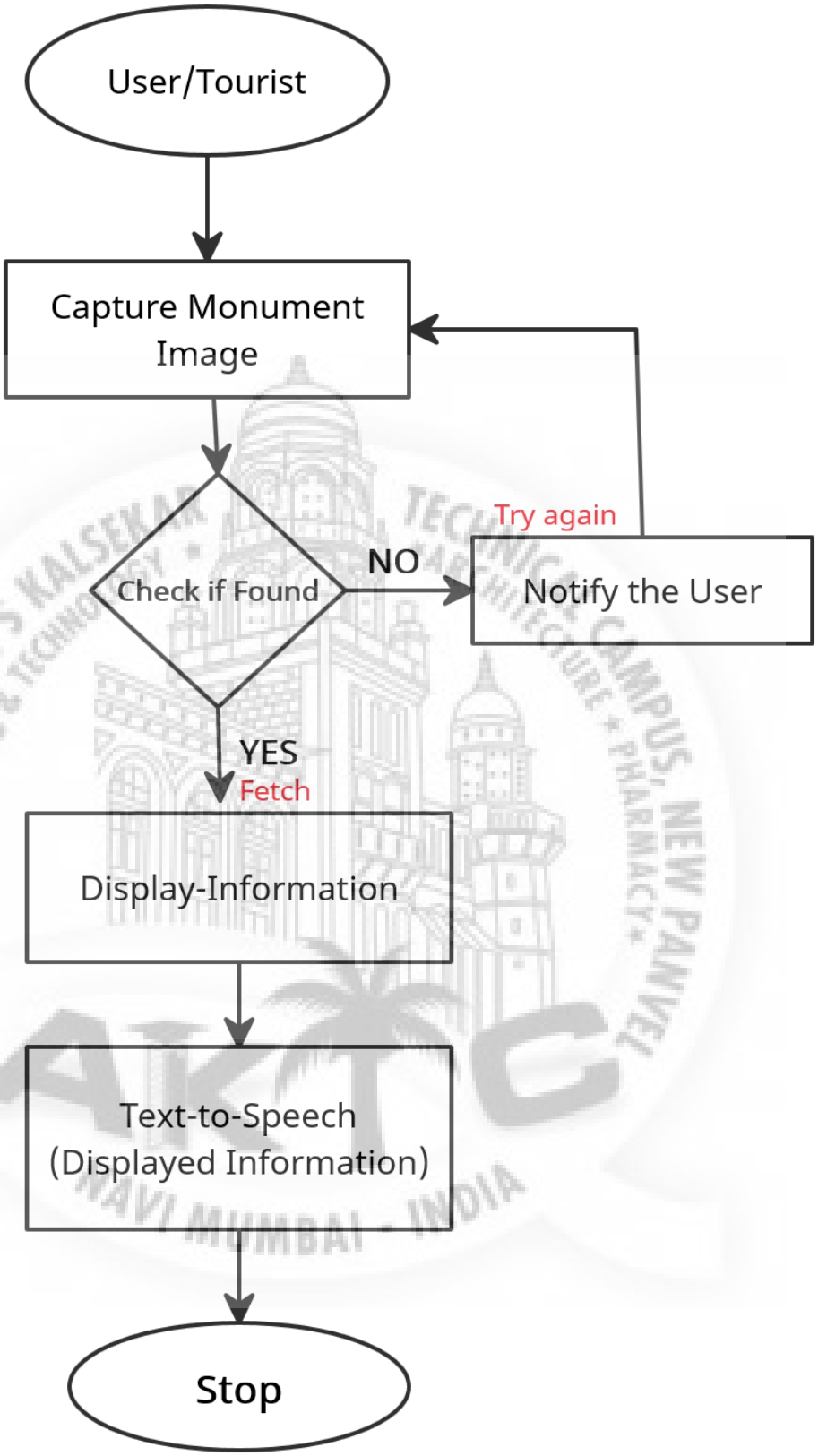


Fig : Flowchart of Tour Buddy

5.1.2 System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

a. Usability - Application implementation is feasible using technologies that are accessible to the end-users.

b. Portability - The interfaces are compatible with Android.

c. Performance Efficiency -Application is able to perform well in a proper time constraint.

d. Multi User System -Application is able to consider the presence of more than one user in the same environment. All the features of the system operates properly for all users and provides proper transparency.

e. Time Efficiency - Time taken for the executing of system is less.

5.2 System Architecture Design

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

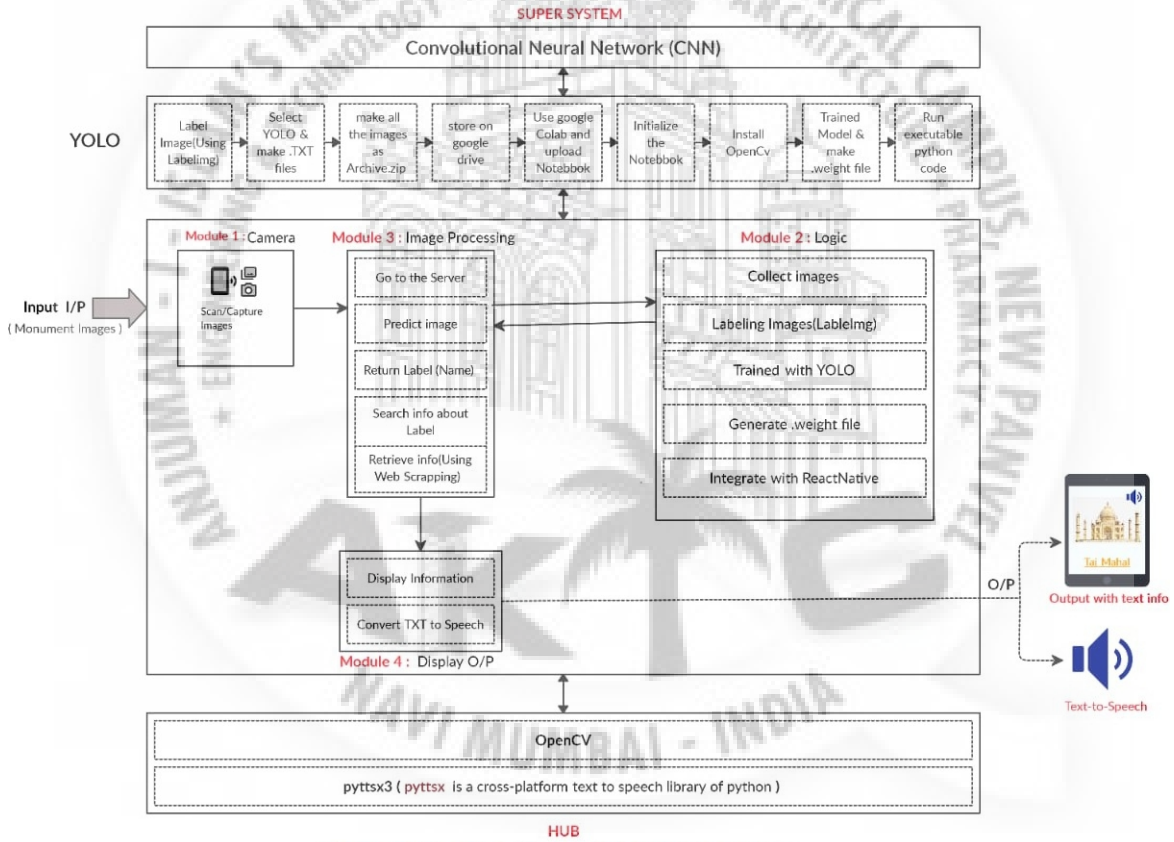


Figure : Architecture of Monument Recognition system

Figure 5.6: System Architecture.

5.3 Sub-system Development

This system consist of one module image recognition and Object.The input to the image recognition module is an image from camera containing monument image and the output will be displayed with the information of that particular monument image.The next module i.e object recognition module takes image as an input and provides name of the object as an output.

5.3.1 Image Recognition

Classifying a monument is very difficult as many images of a single monument are to be used to train the system which are very different from each other in their orientations.

Moreover, the noise present in the images in the form of trees, people, animals, decorations etc.often leads to less accuracy. These variations make monument recognition a challenging problem and The noise which comes in monument images has been taken care of already, as manually better cropped images have been used.The handcrafted features gave accuracies but when CNN was used, better accuracy was achieved in monument recognition. CNN features take the best out of monument images and rest of the work is done by the neural network which feeds better features into it. Moreover this method can be thought of as a new idea to recognize different cultural monuments to promote work in Indian culture too.

5.3.2 Information Recognition

In this, In the developed software tool for Android smart phone, we propose the three following modules:-

a. Training module-Object, Scene.

CNN-based object detectors require large amounts of anno- tated data for training, due to the large number of parameters that need to be learned. For object instance detection the training data should also cover the variations in the object's viewpoint and other nuisance parameters such as lighting, occlusion and clutter. Manually collecting and annotating scenes with the aforementioned properties is time-consuming and costly. Another factor in annotation is the sometimes low generalization capability of trained models across different environments and backgrounds

b. Detection module-Object, Scene, Motion.

Traditional methods for object de- tection in cluttered scenes follow the sliding window based pipeline with hand designed flat feature representations (e.g. HOG) along with discriminative classifiers.

Figure 5.8: Flow Chart for object Recognition

5.4 Systems Integration

System integration (SI) is an engineering process or phase concerned with joining different subsystems or components as one large system. It ensures that each integrated subsystem functions as required. SI is also used to add value to a system through new functionalities provided by

connecting functions of different systems.

27

Chapter 6

Implementation

6.1 Monument detection Module

This module is implemented using React native and flask.

This module takes an input image from camera using the phone camera of the android phone.the input images od them passed to Firebase image recognition module.Then the Firebase module further processs the image and gives out put.

App.js

```
import React, { useState, useEffect } from 'react';
import { StyleSheet, Text, View, TouchableOpacity, SafeAreaView ,Image}
from 'react-native';
import { Camera } from 'expo-camera';
import { MaterialIcons } from '@expo/vector-icons';
import { Entypo } from '@expo/vector-icons';
import { MaterialCommunityIcons } from '@expo/vector-icons';
import Animated from 'react-native-reanimated';
import BottomSheet from 'reanimated-bottom-sheet';
import * as ImagePicker from 'expo-image-picker';
import axios from 'axios';
import { useFonts, Bangers_400Regular } from '@expo-google-fonts/bangers';

var fall= new Animated.Value(1);

export default function App() {
  const [hasPermission, setHasPermission] = useState(null);
  const [type, setType] = useState(Camera.Constants.Type.back);
  const [focus, setFocus]= useState(Camera.Constants.AutoFocus.off);
  const [cameraRef, setCameraRef] = useState(null);
  const [img, setImg] = useState(null);
  const sheetRef = React.useRef(null);

  let [fontsLoaded] =
    useFonts({ Bangers_400Regula
    r,
});
```

```
64
  useEffect(() => {
      (async () => {
        const { status } =
  await
  Camera.requestPermissionsA
  sync();

  setHasPermission(status
  === 'granted');

        if (Platform.OS !==
  'web') {
          const { status } =
  await
  ImagePicker.requestMediaLi
  braryPermissionsAsync();
          if (status !==
  'granted') {
            alert('Sorry, we
  need camera roll
  permissions to make this
  work!');
          }
        }

      })();
    }, []);

    if (hasPermission ===
  null) {
      return <View />;
    }
    if (hasPermission ===
  false) {
      return <Text>No access
  to camera</Text>;
    }


65
  useEffect(() => {
      (async () => {
        const { status } =
  await
  Camera.requestPermissionsA
  sync();

  setHasPermission(status
  === 'granted');

        if (Platform.OS !==
  'web') {
          const { status } =
  await
  ImagePicker.requestMediaLi
  braryPermissionsAsync();
          if (status !==
  'granted') {
            alert('Sorry, we
  need camera roll
  permissions to make this
  work!');
66        }
```

```
    })();
  }, []);

  if (hasPermission ===
null) {
    return <View />;
  }
  if (hasPermission ===
false) {
    return <Text>No access
to camera</Text>;
  }
 const snap= async ()=> {
   // console.log('Button
Pressed');
    if (cameraRef) {
      //
console.log('Taking
photo');
       const options =
{ quality: 1, base64:
true, fixOrientation:
true,
       exif: true,
aspect:[4,3]};
       await
cameraRef.takePictureAsync
(options).then(photo => {

photo.exif.Orientation =
1;

setImg(photo.uri)

sheetRef.current.snapTo(0)
;
// console.log(photo);
          });
     }
    }

    const pickImage =
async () => {
      let result = await
ImagePicker.launchImageLib
raryAsync({
        mediaTypes:
ImagePicker.MediaTypeOptio
ns.Images,
        allowsEditing:
true,
        aspect: [4, 3],
        quality: 1,
      });
```

```
                                  )

  ;
```

```
// console.log(result);

      if (!result.cancelled)
        { setImg(result.uri);
        sheetRef.current.snapTo(0);
      }
    };

    if (!fontsLoaded)
      { return(
      <SafeAreaView>
       <Text>app not loaded properly</Text>
       </SafeAreaView>);
    } else {

      const renderContent = () => (
        <View
          style={{ backgroundColor:'
            black', paddingTop:5,
            padding: 16,
            height: 600,
            borderRadius:30
          }}
        >
<View style={{flex:
0.15,marginBottom:0,flexDirection:'row',justifyContent:'space-
between',backgroundColor:'transparent'}}>
          <Text style={{flex:1,fontSize:16,fontFamily:
'Bangers_400Regular',color:'white',backgroundColor:'transparent'}}>TOURBUD
DY</Text>
         </View>

           <Image
              source={{
                 uri:
                 img,
              }}
              style={{ width: "100%", marginTop:5,height: 180,
borderRadius:15 }}
            />
          <Text
style={{fontSize:35,fontWeight:'700',color:'white',paddingTop:10}}>Hawa
Mahal</Text>
          <Text
style={{fontSize:20,fontWeight:'400',color:'white',paddingTop:10,
paddingBottom:10}}>Hawa Mahal is situated at Badi Choupad, Pink City of
Jaipur, Hawa Mahal was built in 1799. It has 953 windows on the outside
walls. The honeycomb shaped and beautifully carved windows allow breeze to
blow through the palace and makes it a perfect summer palace. It was built
as an extension to the City Palace nearby.</Text>
        </View>
      );

     const renderHeader = () => (
        <View style={styles.header}>
          <View style={styles.panelHeader}>
            <View style={styles.panelHandle} />
          </View>
        </View>
      );
```

```
return (
    <SafeAreaView style={styles.container}>
      <Camera style={styles.camera}   type={type} ref={(ref) =>
{ setCameraRef(ref)}
    }>
          <Text
style={{flex:1,padding:10,marginTop:10,fontSize:24,fontFamily:
'Bangers_400Regular',color:'white',backgroundColor:'transparent'}}>TOURBUD
DY</Text>

<View style={styles.focusButtonContainer}>
        <TouchableOpacity
            onPress={() => {
              setFocus(
                focus === Camera.Constants.AutoFocus.off
                  ? Camera.Constants.AutoFocus.on
                  : Camera.Constants.AutoFocus.off
              );
            }}>
        <MaterialCommunityIcons name="image-filter-center-focus-strong-
outline" size={100}  color="white" />
        </TouchableOpacity>
        </View>


      <View style={styles.buttonContainer}>
          <TouchableOpacity
            style={styles.button}
            onPress={() => {
              setType(
                type === Camera.Constants.Type.back
                  ? Camera.Constants.Type.front
                  : Camera.Constants.Type.back
              );
            }}>
<MaterialIcons name="flip-camera-android" size={40} color="white" />
                      </TouchableOpacity>

          <TouchableOpacity style={styles.button} onPress={snap}>
              <MaterialIcons name="camera" size={40} color="white" />
          </TouchableOpacity>

          <TouchableOpacity style={styles.button} onPress={pickImage}>
          <Entypo name="image" size={40} color="white" />
          </TouchableOpacity>


        </View>
        </Camera>

    <BottomSheet
        ref={sheetRef}
        snapPoints={[600,0]}
        initialSnap={1}
        callbackNode={fall}
        enabledGestureInteraction={true}
        renderHeader={renderHeader}
        renderContent={renderContent}
        enabledInnerScrolling={false}
    />
    </SafeAreaView>
```

```
    );
}
}
const styles =
  StyleSheet.create({ container: {
    flex: 1,
  },
  camera:
    { flex:
    1,
    opacity: 1
  },


  onCamClick:
    { flex: 1,
    backgroundColor: 'transparent'
  },

focusButtonContainer:{
 flex:1,
marginBottom:'50%',
marginLeft:'auto',
marginRight:'auto',
textAlign: 'center',
backgroundColor: 'transparent',
alignItems:'center',
justifyContent:'center'
},
  buttonContainer:
    { flex: 0.5,
    width:'100%',
    height:'20%',
    //marginBottom:'5%',
    backgroundColor: 'transparent',
    flexDirection: 'row', alignItems:
    'center', justifyContent:
    'space-between',
    //margin: 20,
  },
  button:  { flex:
    1,
    width:'100%',
    paddingTop:'10%',
    paddingBottom:'10%',
    backgroundColor: '#000',
    alignSelf: 'flex-end',
    alignItems: 'center',
    justifyContent: 'center',
    flexDirection: 'row',
  },

text: {
    fontSize: 18,
    color: 'white',
  },
   header: {
    backgroundColor: 'transparent',
    shadowColor: '#333333',
    shadowOffset: {width: -1, height: -3},
    shadowRadius: 2,
    shadowOpacity: 0.4,
    // elevation: 5,
    paddingBottom:5,
  },
  pinkButton:{ alignItems:
    'center', justifyContent:
    'center'
```

```
  },
panelHandle:
    { width:
    60,
    height: 5,
    borderRadius: 4,
    backgroundColor: 'silver',
    marginBottom: 3,
  },
});

import os
import base64
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np
from flask import request
from flask import Flask
from werkzeug.utils import secure_filename
from flask_ngrok import run_with_ngrok


app = Flask(_name_)
run_with_ngrok(app)



@app.route('/api')
def find_monument():

f = request.files['file']
    f.save(os.path.join("./upload", secure_filename("test.jpg")))
    np.set_printoptions(suppress=True)


    classes = ["BIBI KA MAQBARA", "CHARMINAR", "GOL GUMBAZ", "HAWA
MAHAL","KANCH MAHAL", "LOTUS TEMPLE", "PARLIAMENT", "RED FORT", "TAJ
MAHAL"]

    # Load the model
    model = tensorflow.keras.models.load_model('./keras_model.h5')

    # Create the array of the right shape to feed into the keras model
    # The 'length' or number of images you can put into the array is
    # determined by the first position in the shape tuple, in this case 1.
    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

    # Replace this with the path to your image
    image = Image.open('./upload/test.jpg')

    # resize the image to a 224x224 with the same strategy as in TM2:
    # resizing the image to be at least 224x224 and then cropping from the
center
    size = (224, 224)
    image = ImageOps.fit(image, size, Image.ANTIALIAS)

    # turn the image into a numpy array
    image_array = np.asarray(image)
# display the resized image
    image.show()

    # Normalize the image
    normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

    # Load the image into the array
    data[0] = normalized_image_array

    # run the inference
```

```
    prediction = model.predict(data)
    print(prediction[0][0])
    val = str(prediction[0][0])
    print(val)
    res =  val[:1] print(res)
    print(classes[int(res)])
    return classes[int(res)]

app.run()


//////////////////////////.cfg////////////////////////////////////////////
//////////
```

```
[net]
# Testing
batch=8
subdivisions=1
# Training
# batch=64
# subdivisions=166
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 4000
policy=steps
steps=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

# Downsample

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=32
size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=128
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=256
```

```
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
```

```
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=512
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
```

```
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky
```



```
[convolutional]
```

```
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
```

```
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
```

```
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

######################

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
```

```
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 6,7,8
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=1
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1


[route]
layers = -4

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 61



[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 3,4,5
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=1
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1



[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2
```

```
[route]
layers = -1, 36



[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=1
num=9
```

```
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Chapter 7

System Testing

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements are as follows.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Recognize image	Input monument Image	image Recognized Successfully	Output of the recognized image
T02	Recognize information	Input information image	info Recognized Successfully	multiple information should recognize

7.2 Test Cases

Title: Recognize Image

Description: A System should be able to successfully recognize .

Assumption: a mobile phone with supported requirements is being used.

Test Steps:

1. open app
2. Tell the app to go for image recognition
3. Image Recognition activity will be opened.
4. app will ask weather to upload image or click image.
5. place the camera over the image from which you want to extract image or upload a image which contain data.
6. The app will take a pic in 5 secs automatically.

Expected Result: The app will capture image and recognize image.It will Display the data at the bottom of the screen.

Actual Result: Recognize image successfully.

Chapter 8

Screenshots of Project

8.1 Image Recognition

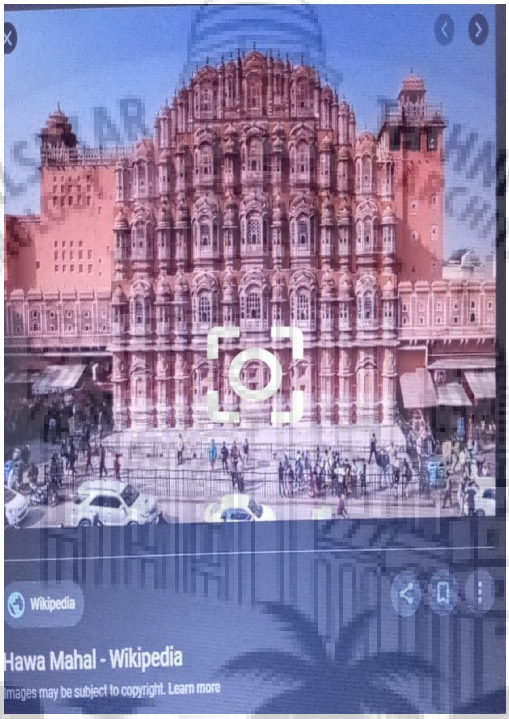
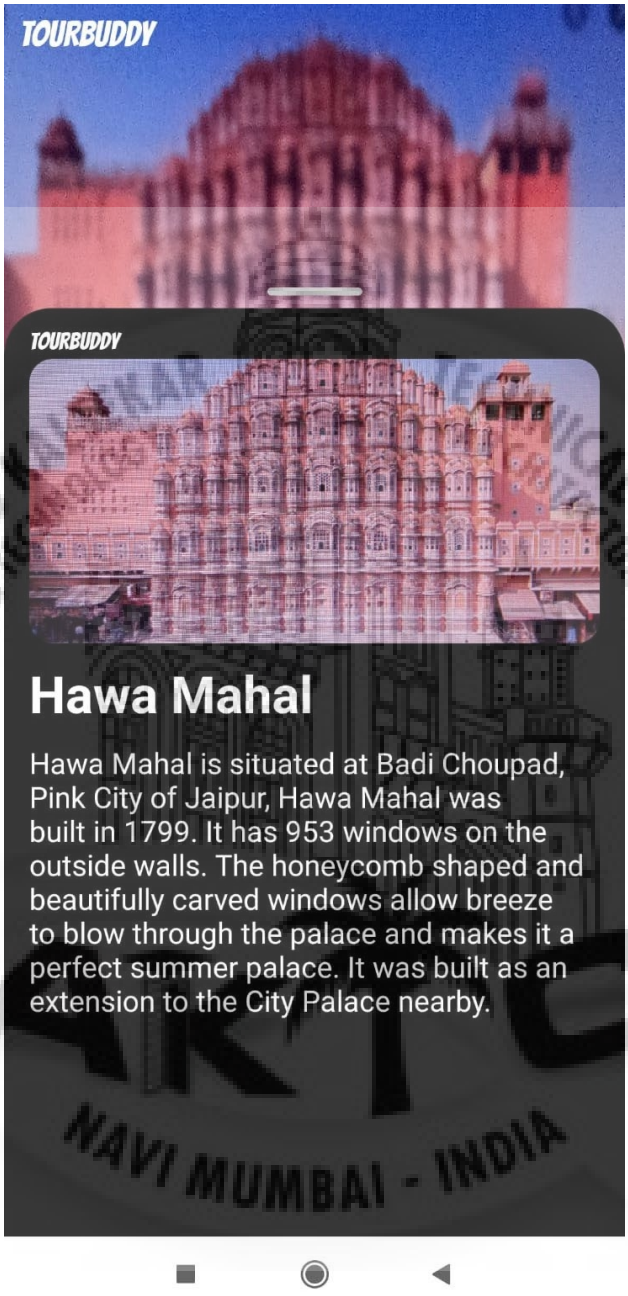


Figure 8.1: Image detection



The Output is displayed in the above image

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The Tour Buddy application provides various features to the users.The main focus of the project is to reduce users/tourists efforts and saves users time and money because India is 7th largest country which has large no of monuments and Historic places where millions of tourists visits everyday But no one have exact/genuine information about any of the monument and because of that the guides are make them fool ,So we are trying to develop an application which will help the users/tourists to gain the exact information about any of the monument by using their own smart phone camera ,This application (Monument Detection Using YOLO) will work as a virtual guide for the tourists which will help them to get a proper information related to that monument ,So user can save his/her time , money and efforts.

9.2 Future Scope

- In future we'll provide a overall tour schedule to the users such as Day , Time and Places as per thier availability and seasons.
- For Examle :
- We'll provide a overall schedule like on which Season / Day user should go there and enjoy the tour insort we'll provide a proper plan with some guidelines to the users/tourists.

References

[1] .Yaligar, S.,Sannakki,S.and Yaligar, N., 2013.Identification andRetrieval of Archaeological Monuments Using Visual Features.

[2] .He, D.C. and Wang, L., 1990. Texture unit, texture spectrum, and texture analysis. IEEE transactions on Geoscience and Remote Sensing, 28(4), pp.509-512.

[3] .Wang, L. and He, D.C., 1990. Texture classification using texture spectrum.Pattern Recognition, 23(8), pp.905-910.

[4] .Hays, J. and Efros, A.A., 2007, August. Scene completion using millions of photographs. In ACM Transactions on Graphics (TOG)(Vol. 26, No. 3, p. 4).ACM.

[5] .Ojala, T., Pietikainen, M. and Harwood, D., 1994, October. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Pattern Recognition, 1994. Vol. 1- Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on (Vol. 1, pp. 582-585). IEEE.

[6] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”, arXiv:1603.04467, 2016.

[7] .Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald,Rodney J Douglas, and H Sebastian Seung.,” Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit.”, Nature, 405(6789): 947, 2000.

[8] .Ojala, T., Pietikainen, M. and Harwood, D., 1994, October. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Pattern Recognition, 1994. Vol. 1- Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on (Vol. 1, pp. 582-585). IEEE.

[9] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”, arXiv:1603.04467, 2016.

[10] .Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald,Rodney J Douglas, and H Sebastian Seung.,” Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit.”, Nature, 405(6789): 947, 2000.

ANJUMAN - I - ISLAM'S KALSEKAR
TECHNICAL CAMPUS, NEW PANVEL
ENGINEERING & TECHNOLOGY
ARCHITECTURE
PHARMACY
AKTC
NAVI MUMBAI - INDIA