ADHIYAMAAN COLLEGE OF ENGINEERING
(AUTONOMOUS) ,HOSUR
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Signs with Smart Connectivity for Better Road Safety

Team-id:PNT2022TMID08042

GUIDED BY:
Mrs.B.Revathi M.E
AP/ CSE

TEAM MEMBERS:
Swithin Asir S - AC19UCS125
Sneha P - AC19UCS109
Swapna V - AC19UCS123
Thejeshkumar S - AC19UCS129

OVERVIEW

- ABSTRACT
- OBJECTIVE
- LITERATURE SURVEY
- EXISTING SYSTEM
- PROPOSED SYSTEM
- SOLUTION ARCHITECTURE
- MODULES
- SYSTEM CONFIGURATION
- OUTPUT
- CONCLUSION AND FUTURE WORK
- REFERENCES

OBJECTIVE:

- The main objective of our project is to reduce the speed the
- speed limit of the vehicle.
- By using the weather API we will change the speed limit in the sign board according to the current weather.
- We will generate the fine who are crossing the speed limit in digital sign board.
- By using this sign board we will avoid the unwanted noise from the vehicles.

LITERATURE SURVEY:

- Ashish Dhar: **Traffic and road condition monitoring system** Indian Institute of Technology, Mumbai. - 2008.
  - Reports severity, intensity and dimension of a damaged road segment.
  - Proposed a different solution using AMR Magnetic Sensor
- Pooja Pawar, Suvarna Langade, Mohini Bandgar: **IOT Based digital Notice Board using Arduino ATMega 328**. International Research Journal of Engineering and Technology(IRJET).- 2019.
  - Circulates notice regularly & reduce physical efforts.
  - Send message at any distant location within a second.

- Sandeep Chaware, Trushitha Chaware: **Proposed Algorithm for Smart Traffic Control using Ultrasonic Sensor.** International Journal of Engineering and Advanced Technology (IJEAT).- 2019.
  - The outcome of the project is to learn insights of the traffic controlling and management at the signal with the dynamically changing in timing of timer as per need.
- KamnaSingh , Deepa Bura: **IOT distinct algorithms for the Sensor Connectivity with Comparative Study between node MCU and Arduino MCU**. NVEO Journal– 2021
  - Presents different algorithms for the connection between different types of sensors.
  - Brief description of node MCU & Arduino MCU.
  - Step by step solution to provide connectivity with IOT technology.

- Jack Greenhaigh : **Recognizing Text Based Traffic Signs**. IEEE – 2015 • Detect all possible Road sign candidates. 4
  - Reduce total regions based on contextual constraints.
  - A Novel System for the automatic detection and recognition of text in traffic sign based on MSER & MSV.
- 6. Bhumika.R , Harshita. S.A, Meena. D, Asha. N: Accident Prevention and Road Safety in Hilly Region using IOT Module International Research Journal of Engineering and Technology (IRJET) – 2021
  - Stay away from mishap & forestall clog in sloping region & hair clip twist.
  - As a significant part of street mathematical plan bended street portion

EXISTING SYSTEM:

The Existing road system and connectivity, emphasis on the traffic and route reckoning features which cordially provisions the user acceptability to have better connectivity management. But, this often results in non-parallel road conditions and high noise ratios through the calibrations. It reiterates various subjections in its compilation and leading to segmentation error throughout. It penetrates the various unit cases in order to subsequently manifest the output. This alternatively symbolizes the ineffectively programmed web user interface. The IOT based model of our project complies of the verdict to specify the safe zone in the path. It manually ask the user to turn off the horn, which invariably decreases the decibel level of the power output. Illustratively, it confides the work schematics of the precedent evaluation under the system and allows the user to access the terminals of the app nodes variably.

DISADVANTAGE:

- This finding is contrary to previous research that suggests differences in crash counts exist in the presence of static roadside advertising.
- The quantity and quality of available evidence limit our conclusion.
- Fixed object, side swipe and rear end crashes are the most common types of crashes in the presence of roadside advertising signs.

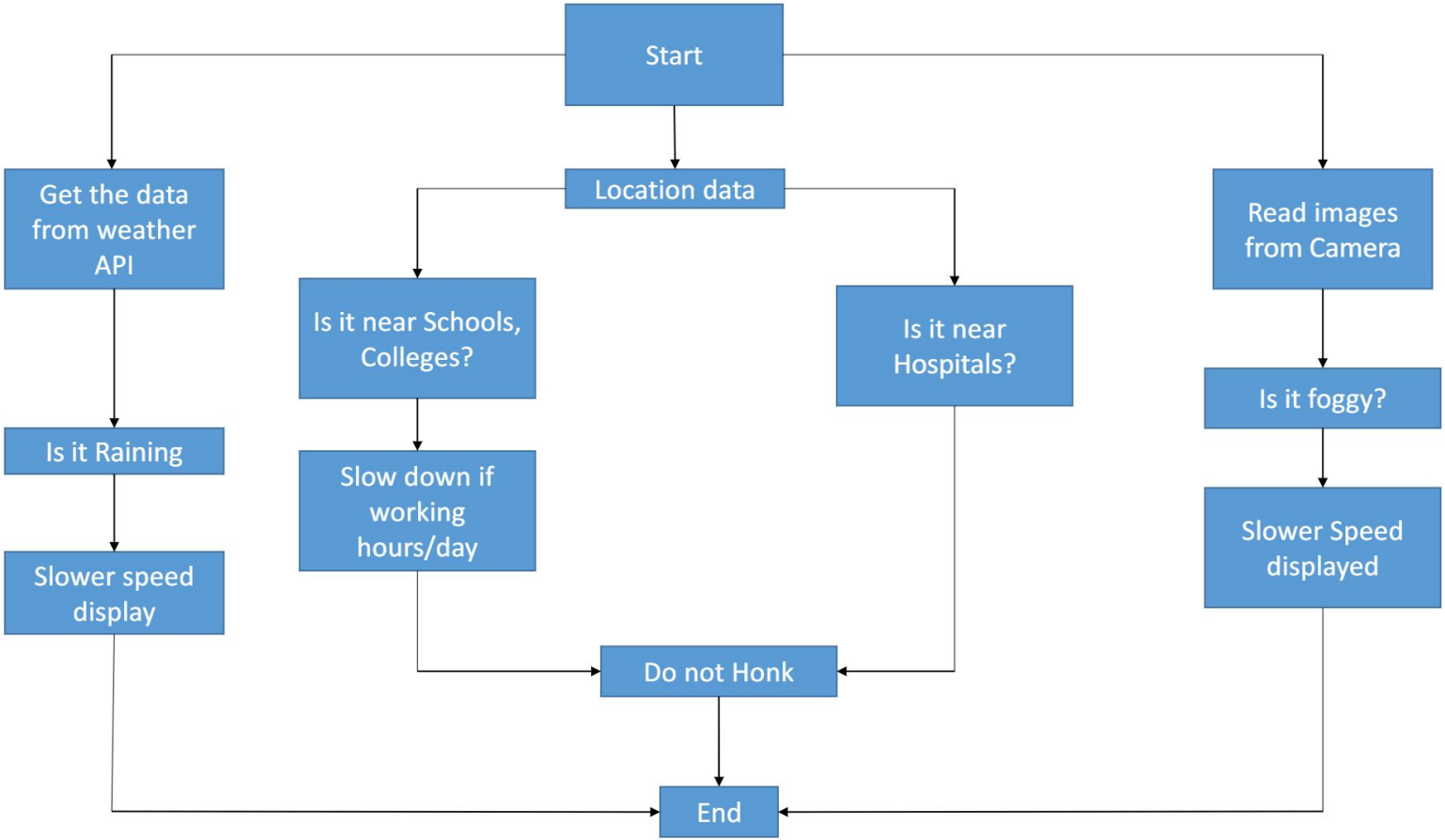
PROPOSED SOLUTION:

- In this project an assistive sign board with smart connectivity for better road safety .
- To replace the static signboard , smart connected sign board are used.
- These smart connectivity sign boards get the speed limitation from a web app using weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Based on the traffic and fatal situations the diversion signs are displayed.
- Guide(Schools),warning and service(Hospitals , Restaurant)signs are also displayed accordingly.

ADVANTAGES:

- Low-cost, low-requirement micro controllers can be employed as processing is mostly handled by Node RED servers, resulting in lower battery usage.
- Systems with longer lifespans.
- Dynamic sign-up updates.
- Alerts for the School/Hospital Zone.

SOLUTION ARCHITECTURE:



MODULE:

- Collecting the weather details and storing in the IBM cloud
- Changing the speed limit According to the weather
- Dashboard
- Fine generation

MODULE DESCRIPTION :

- **Collecting the weather details and storing in the IBM cloud:**
  - By using the open weather api we are collecting the weather report and we are storing it in IBM cloud.
- **Changing the speed limit According to the weather:**
  - In this module we will change the speed limit according to the weather condition which is taken by the open api and temp 36 sensor.
- **Dashboard:**
  - By using this dashboard we can analysis the data of overspeed vechiles which is crossed through the kit.
- **Fine generation:**
  - If the vehicle cross the speed limit the fine will be automatically generator and it will send to the vehicle owner

COLLECTING THE WEATHER DETAILS:

```
let weather = require("./API_modules/weather")
//let airpollution = require("./API_modules/airpollution")
let nearby = require("./API_modules/nearby")
let { app } = require("./API_modules/requiredFiles.js") //multiple objects can be imported

app.use(weather)
//app.use(airpollution)
app.use(nearby)

app.use("/", (req, res, next) => {
  res.status(404).send("<h1>Result Not Found! Please append in URL</h1>")
})
```

OUTPUT:


```
{
    "coord": {
        "lon": 77.1025,
        "lat": 28.7041
    },
    "weather": [
        {
            "id": 711,
            "main": "Smoke",
            "description": "smoke",
            "icon": "50d"
        }
    ],
    "base": "stations",
    "main": {
        "temp": 30.06,
        "feels_like": 29.64,
        "temp_min": 30.06,
        "temp_max": 30.06,
        "pressure": 1011,
        "humidity": 39
    },
    "visibility": 1600,
    "wind": {
        "speed": 3.09,
        "deg": 290
    },
    "clouds": {
        "all": 0
    },
    "dt": 1667380847,
    "sys": {
        "type": 1,
        "id": 9165,
        "country": "IN",
        "sunrise": 1667351046,
        "sunset": 1667390767
    },
```

COLLECTING THE CAR NO:

In [1]:
```
import cv2
import numpy as np
```

In [2]:
```
plat_detector =  cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_russian_plate_number.xml")
video = cv2.VideoCapture('Data/vid.mp4')

if(video.isOpened()==False):
    print('Error Reading Video')

while True:
    ret,frame = video.read()
    gray_video = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    plate = plat_detector.detectMultiScale(gray_video,scaleFactor=1.2,minNeighbors=5,minSize=(25,25))

    for (x,y,w,h) in plate:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0),2)
        frame[y:y+h,x:x+w] = cv2.blur(frame[y:y+h,x:x+w],ksize=(10,10))
        cv2.putText(frame,text='License Plate',org=(x-3,y-3),fontFace=cv2.FONT_HERSHEY_COMPLEX,color=(0,0,255),thickness=1,fc


    if ret == True:
        cv2.imshow('Video', frame)

        if cv2.waitKey(25) & 0xFF == ord("q"):
            break
    else:
        break

video.release()
cv2.destroyAllWindows()
```

OUTPUT:

DASHBOARD:

```
import { Routes, Route } from "react-router-dom";
import Topbar from "./scenes/global/Topbar";
import Sidebar from "./scenes/global/Sidebar";
import Dashboard from "./scenes/dashboard";
import Team from "./scenes/team";
import Invoices from "./scenes/invoices";
import Contacts from "./scenes/contacts";
import Bar from "./scenes/bar";
import Form from "./scenes/form";
import Line from "./scenes/line";
import Pie from "./scenes/pie";
import FAQ from "./scenes/faq";
import Geography from "./scenes/geography";
import { CssBaseline, ThemeProvider } from "@mui/material";
import { ColorModeContext, useMode } from "./theme";
import Calendar from "./scenes/calendar/calendar";

function App() {
```

OUTPUT:


ADMINIS
Search
Ed Roh
VP Fancy Admin
Dashboard
Data
Manage Team
Contacts Information
Invoices Balances
Pages
Profile Form
Calendar
FAQ Page
DASHBOARD
Welcome to your dashboard
DOWNLOAD REPORTS
12,361
Emails Sent
+14%
431,225
Fine obtained
+21%
1,325,134
Traffic violation
+43%
Revenue Generated
$59,342.32
500
400
300
200
100
plane
helicopter
boat
train
subway
bus
car
moto
bicycle
horse
skateboard
others
us
france
japan
Recent Transactions
01e4dsa
johndoe
2021-09-01
$43.95
0315dsaa
jackdower
2022-04-01
$133.45
01e4dsa
aberdohnny
2021-09-01
$43.95

FINE GENERATION:

In [5]:
```
import smtplib
```

In [6]:
```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
mail_content = 'Hello,This is a simple mail. There is only text, no attachments are there The mail is sent using Python SMTP libr
sender_address = 'swithinasir.cse2019@adhiyamaan.in'
sender_pass = 'Sinola123'
receiver_address = 'vaishnaviamalkumar@gmail.com'
#Setup the MIME
message = MIMEMultipart()
message['From'] = sender_address
message['To'] = receiver_address
message['Subject'] = 'A test mail sent by Python. It has an attachment.'   #The subject line
#The body and the attachments for the mail
message.attach(MIMEText(mail_content, 'plain'))
#Create SMTP session for sending the mail
session = smtplib.SMTP('smtp.gmail.com', 587) #use gmail with port
session.starttls() #enable security
session.login(sender_address, sender_pass) #login with mail_id and password
text = message.as_string()
session.sendmail(sender_address, receiver_address, text)
session.quit()
print('Mail Sent')
```

CONFIGURATIONS:

SOFTWARE REQUIREMENTS:

- Python 3.6
- OpenWeather Api
- Node-Red

HARDWARE REQUIREMENTS:

- Arduino
- Ultrasonic Sensor
- Motion Sensor
- Camera
- ESP 32

CONCLUSION:

- Smart connectivity signs for road safety , These conclusions and guideline are addressed to policy makers and private companies that are willing to use innovative solution to decrease road-related facilities and injuries amidst populations.
- Both chapters take into account the potential users of connected technologies:
- Individual drivers , commercial drives , pedestrians , cyclist and motorcyclists.
- The task force decided to study first the potential of connected technologies in high- and middle-income countries.
- Indeed middle-income countries represent 72% of the world population,80% of road traffic deaths and 47% of registered motorized vehicles , while high income countries are leaders in development of connected vehicles.

FUTURE SCOPE:

- Data that helps travellers plan their routes. Sensors lining highways monitor traffic flow and weight load, warn drivers of traffic jams, and automatically alert the authorities about accidents.
- Fibre-optic cables embedded in the road detect wear and tear, and communication between vehicles and roads can improve traffic management.
- The smart notice board can be used in various applications such as banks, schools, restaurants, colleges, hospital etc.

REFERENCES :

https://www.researchgate.net/publication/328130229_Smart_Board

https://www.researchgate.net/publication/305365822_Poster_A_Step_Towards_Smart_Traffic_Sign_Board_by_Smart_Devices

https://www.researchgate.net/publication/336355855_Smart_Traffic_Sign_Boards_STSB_for_Smart_Cities

https://drive.google.com/file/d/1Gzr5TUW5Jb5VOSXWdFZqpFOqMo_b3V3k/v iew?usp=share_link

https://drive.google.com/file/d/1SroGSQaz9K0m3HOaj1ofkohoFaelSfZI/view?usp=share_link

https://docs.oracle.com/en/cloud/paas/app-container-cloud/python-oracle-accs/

https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-getting-started-python

THANK YOU