**SAMPLE CODE**

**User side views.py**

from django.shortcuts import render,HttpResponse

from django.contrib import messages

from .forms import DriverRegistrationForm

from .models import DriverRegistrationModel,FattigueInfoModel

# Create your views here.

def DriverRegisterActions(request):

if request.method == 'POST':

form = DriverRegistrationForm(request.POST)

if form.is\_valid():

print('Data is Valid')

form.save()

messages.success(request, 'You have been successfully registered')

form = DriverRegistrationForm()

return render(request, 'AutoistRegister.html', {'form': form})

else:

messages.success(request, 'Email or Mobile Already Existed')

print("Invalid form")

else:

form = DriverRegistrationForm()

return render(request, 'AutoistRegister.html', {'form': form})

def AutoistLoginCheck(request):

if request.method == "POST":

loginid = request.POST.get('loginid')

pswd = request.POST.get('pswd')

print("Login ID = ", loginid, ' Password = ', pswd)

try:

check = DriverRegistrationModel.objects.get(loginid=loginid, password=pswd)

status = check.status

print('Status is = ', status)

if status == "activated":

request.session['id'] = check.id

request.session['loggeduser'] = check.name

request.session['loginid'] = loginid

request.session['email'] = check.email

request.session['vehiclenumber'] = check.vehiclenumber

print("User id At", check.id, status)

return render(request, 'autoist/AutoistHome.html', {})

else:

messages.success(request, 'Your Account Not at activated')

return render(request, 'AutoistLogin.html')

except Exception as e:

print('Exception is ', str(e))

pass

messages.success(request, 'Invalid Login id and password')

return render(request, 'AutoistLogin.html', {})

def AutoistHome(request):

return render(request, 'autoist/AutoistHome.html', {})

def DetectFatigueDriver(request):

from users.utility.detections import FatigueDetections

obj = FatigueDetections()

flag = obj.start\_process()

import geocoder

g = geocoder.ip('me')

import datetime

l = g.latlng

lattitude = l[0]

longitude = l[1]

if flag:

print('Fatigue Detetcted')

user\_name = request.session['loggeduser']

logged\_user = request.session['loginid']

email = request.session['email']

vehiclenumber = request.session['vehiclenumber']

c\_date = datetime.datetime.now()

rslt\_dict = {

'user\_name': user\_name,

'login\_user': logged\_user,

'email': email,

'vehiclenumber': vehiclenumber,

'lattitude': lattitude,

'longitude': longitude,

'fatigue': 'Fatigue',

'c\_date': c\_date

}

FattigueInfoModel.objects.create(user\_name=user\_name, login\_user=logged\_user, email=email,vehiclenumber=vehiclenumber,lattitude=lattitude,longitude = longitude,fatigue='Fatigue',c\_date=c\_date)

else:

print('No Fatigue')

return render(request, 'autoist/DetectionImage.html', rslt\_dict)

def StartTraining(request):

from users.utility import model

return render(request, 'autoist/TrainingComplete.html', {})

def Autoisthistory(request):

logged\_user = request.session['loginid']

qs = FattigueInfoModel.objects.filter(login\_user=logged\_user)

return render(request, 'autoist/FastAutoistHistory.html',{'data':qs})

building model

import os

from keras.preprocessing import image

import matplotlib.pyplot as plt

import numpy as np

from keras.utils.np\_utils import to\_categorical

import random, shutil

from keras.models import Sequential

from keras.layers import Dropout, Conv2D, Flatten, Dense, MaxPooling2D, BatchNormalization

from keras.models import load\_model

from django.conf import settings

def generator(dir, gen=image.ImageDataGenerator(rescale=1. / 255), shuffle=True, batch\_size=1, target\_size=(24, 24),

class\_mode='categorical'):

return gen.flow\_from\_directory(dir, batch\_size=batch\_size, shuffle=shuffle, color\_mode='grayscale',

class\_mode=class\_mode, target\_size=target\_size)

BS = 32

TS = (24, 24)

train\_batch = generator(os.path.join(settings.MEDIA\_ROOT, 'data', 'train'), shuffle=True, batch\_size=BS, target\_size=TS)

valid\_batch = generator(os.path.join(settings.MEDIA\_ROOT, 'data', 'valid'), shuffle=True, batch\_size=BS, target\_size=TS)

SPE = len(train\_batch.classes) // BS

VS = len(valid\_batch.classes) // BS

print(SPE, VS)

# img,labels= next(train\_batch)

# print(img.shape)

model = Sequential([

Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(24, 24, 1)),

MaxPooling2D(pool\_size=(1, 1)),

Conv2D(32, (3, 3), activation='relu'),

MaxPooling2D(pool\_size=(1, 1)),

# 32 convolution filters used each of size 3x3

# again

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D(pool\_size=(1, 1)),

# 64 convolution filters used each of size 3x3

# choose the best features via pooling

# randomly turn neurons on and off to improve convergence

Dropout(0.25),

# flatten since too many dimensions, we only want a classification output

Flatten(),

# fully connected to get all relevant data

Dense(128, activation='relu'),

# one more dropout for convergence' sake :)

Dropout(0.5),

# output a softmax to squash the matrix into output probabilities

Dense(4, activation='softmax')

])

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.fit\_generator(train\_batch, validation\_data=valid\_batch, epochs=15, steps\_per\_epoch=SPE, validation\_steps=VS)

model.save(os.path.join(settings.MEDIA\_ROOT, 'models', 'Dp\_cnnCat2.h5'), overwrite=True)

detetction Model

from comtypes.tools.typedesc import SAFEARRAYType

from django.conf import settings

import os

class FatigueDetections:

def start\_process(self):

import cv2

import os

from keras.models import load\_model

import numpy as np

from pygame import mixer

import time

mixer.init()

sound = mixer.Sound(os.path.join(settings.MEDIA\_ROOT, 'alarm.wav'))

face = cv2.CascadeClassifier(

os.path.join(settings.MEDIA\_ROOT, 'haar cascade files', 'haarcascade\_frontalface\_alt.xml'))

leye = cv2.CascadeClassifier(

os.path.join(settings.MEDIA\_ROOT, 'haar cascade files', 'haarcascade\_lefteye\_2splits.xml'))

reye = cv2.CascadeClassifier(

os.path.join(settings.MEDIA\_ROOT, 'haar cascade files\haarcascade\_righteye\_2splits.xml'))

lbl = ['Close', 'Open']

model = load\_model(os.path.join(settings.MEDIA\_ROOT, 'models', 'cnncat2.h5'))

path = os.getcwd()

cap = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_COMPLEX\_SMALL

count = 0

score = 0

thicc = 2

rpred = [99]

lpred = [99]

flag = False

while (True):

ret, frame = cap.read()

height, width = frame.shape[:2]

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face.detectMultiScale(gray, minNeighbors=5, scaleFactor=1.1, minSize=(25, 25))

left\_eye = leye.detectMultiScale(gray)

right\_eye = reye.detectMultiScale(gray)

cv2.rectangle(frame, (0, height - 50), (200, height), (0, 0, 0), thickness=cv2.FILLED)

for (x, y, w, h) in faces:

cv2.rectangle(frame, (x, y), (x + w, y + h), (100, 100, 100), 1)

for (x, y, w, h) in right\_eye:

r\_eye = frame[y:y + h, x:x + w]

count = count + 1

r\_eye = cv2.cvtColor(r\_eye, cv2.COLOR\_BGR2GRAY)

r\_eye = cv2.resize(r\_eye, (24, 24))

r\_eye = r\_eye / 255

r\_eye = r\_eye.reshape(24, 24, -1)

r\_eye = np.expand\_dims(r\_eye, axis=0)

rpred = model.predict\_classes(r\_eye)

if (rpred[0] == 1):

lbl = 'Open'

if (rpred[0] == 0):

lbl = 'Closed'

break

for (x, y, w, h) in left\_eye:

l\_eye = frame[y:y + h, x:x + w]

count = count + 1

l\_eye = cv2.cvtColor(l\_eye, cv2.COLOR\_BGR2GRAY)

l\_eye = cv2.resize(l\_eye, (24, 24))

l\_eye = l\_eye / 255

l\_eye = l\_eye.reshape(24, 24, -1)

l\_eye = np.expand\_dims(l\_eye, axis=0)

lpred = model.predict\_classes(l\_eye)

if (lpred[0] == 1):

lbl = 'Open'

if (lpred[0] == 0):

lbl = 'Closed'

break

if (rpred[0] == 0 and lpred[0] == 0):

score = score + 1

cv2.putText(frame, "Closed", (10, height - 20), font, 1, (255, 255, 255), 1, cv2.LINE\_AA)

# if(rpred[0]==1 or lpred[0]==1):

else:

score = score - 1

cv2.putText(frame, "Open", (10, height - 20), font, 1, (255, 255, 255), 1, cv2.LINE\_AA)

if (score < 0):

score = 0

cv2.putText(frame, 'Score:' + str(score), (100, height - 20), font, 1, (255, 255, 255), 1, cv2.LINE\_AA)

if (score > 15):

# person is feeling sleepy so we beep the alarm

cv2.imwrite(os.path.join(path,'assets','static','image.jpg'), frame)

try:

sound.play()

flag = True

except: # isplaying = False

pass

if (thicc < 16):

thicc = thicc + 2

else:

thicc = thicc - 2

if (thicc < 2):

thicc = 2

cv2.rectangle(frame, (0, 0), (width, height), (0, 0, 255), thicc)

cv2.imshow('frame press Q to Exit', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

return flag

base.html

{% load static%}

<!DOCTYPE html>

<html lang="en">

<head>

<!-- basic -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- mobile metas -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="viewport" content="initial-scale=1, maximum-scale=1">

<!-- site metas -->

<title>Driver Drowsiness</title>

<meta name="keywords" content="">

<meta name="description" content="">

<meta name="author" content="">

<!-- bootstrap css -->

<link rel="stylesheet" href="{%static 'css/bootstrap.min.css'%}">

<!-- style css -->

<link rel="stylesheet" href="{%static 'css/style.css'%}">

<!-- Responsive-->

<link rel="stylesheet" href="{%static 'css/responsive.css'%}">

<!-- fevicon -->

<link rel="icon" href="" type="{%static 'image/gif'%}" />

<!-- Scrollbar Custom CSS -->

<link rel="stylesheet" href="{%static 'css/jquery.mCustomScrollbar.min.css'%}">

<!-- awesome fontfamily -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<!-- Tweaks for older IEs-->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-->

</head>

<!-- body -->

<body class="main-layout">

<!-- loader -->

<div class="loader\_bg">

<div class="loader"><img src="{%static 'images/loading.gif'%}" alt="" /></div>

</div>

<!-- end loader -->

<div class="wrapper">

<div class="sidebar">

<!-- Sidebar -->

<nav id="sidebar">

<div id="dismiss">

<i class="fa fa-arrow-left"></i>

</div>

<ul class="list-unstyled components">

<li><a href="{%url 'index'%}">Home</a></li>

<li><a href="{%url 'AutoistLogin'%}">Autoist</a></li>

<li><a href="{%url 'AdminLogin'%}">Admin</a></li>

<li><a href="{%url 'AutoistRegister'%}">Autoist Register</a></li>

</ul>

</nav>

</div>

<div id="content">

<!-- section -->

<section id="home" class="top\_section">

<div class="row">

<div class="col-lg-12">

<!-- header -->

<header>

<!-- header inner -->

<div class="container">

<div class="row">

<div class="col-lg-3 logo\_section">

<div class="full">

<div class="center-desk">

<div class="logo"> <a href="index.html"><img src="{%static 'images/logo.png'%}" alt="#"></a> </div>

<button type="button" id="sidebarCollapse">

<img src="{%static 'images/menu\_icon.png'%}" alt="#" />

</button>

</div>

</div>

</div>

<div class="col-lg-9">

<div class="right\_header\_info">

<h3 style="color:Tomato;">Driver Drowsiness Monitoring using Convolutional Neural Networks</h3>

<!-- <ul>-->

<!-- <li><img style="margin-right: 15px;" src="" alt="#" /><a href="#"></a></li>-->

<!--&lt;!&ndash; <li><img style="margin-right: 15px;" src="{%static 'images/mail\_icon.png'%}" alt="#" /><a href="#">demo@gmail.com</a></li>&ndash;&gt;-->

<!--&lt;!&ndash; <li><img src="{%static 'images/search\_icon.png'%}" alt="#" /></li>&ndash;&gt;-->

<!--&lt;!&ndash; <li>&ndash;&gt;-->

<!-- </li>-->

<!-- </ul>-->

</div>

</div>

</div>

</div>

<!-- end header inner -->

</header>

{%block contents%}

{%endblock%}

<div class="cpy\_right">

<div class="container">

<div class="row">

<div class="col-md-12">

<div class="full">

<p>© 2022 All Rights Reserved. Design by <a href="https://html.design">Alex Corporations Templates</a></p>

</div>

</div>

</div>

</div>

</div>

<!-- right copyright -->

</div>

<div class="overlay"></div>

<!-- Javascript files-->

<script src="{%static 'js/jquery.min.js'%}"></script>

<script src="{%static 'js/popper.min.js'%}"></script>

<script src="{%static 'js/bootstrap.bundle.min.js'%}"></script>

<!-- Scrollbar Js Files -->

<script src="{%static 'js/jquery.mCustomScrollbar.concat.min.js'%}"></script>

<script src="{%static 'js/custom.js'%}"></script>

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA8eaHt9Dh5H57Zh0xVTqxVdBFCvFMqFjQ&callback=initMap"></script>

<!-- end google map js -->

<script type="text/javascript">

$(document).ready(function () {

$("#sidebar").mCustomScrollbar({

theme: "minimal"

});

$('#dismiss, .overlay').on('click', function () {

$('#sidebar').removeClass('active');

$('.overlay').removeClass('active');

});

$('#sidebarCollapse').on('click', function () {

$('#sidebar').addClass('active');

$('.overlay').addClass('active');

$('.collapse.in').toggleClass('in');

$('a[aria-expanded=true]').attr('aria-expanded', 'false');

});

});

</script>

</body>

</html>

Autoistregister.html

{%extends 'base.html'%}

{%load static%}

{%block contents%}

<section>

<div class="container-fluid">

<div class="row">

<div class="col-md-5">

<div class="full slider\_cont\_section">

<p class="large">Autoist Registration Form</p>

<form action="{%url 'DriverRegisterActions'%}" method="POST" style="width:100%;color:red">

{% csrf\_token %}

<table>

<tr><td></td>

<td>Driver Name</td>

<td>{{form.name}}</td>

</tr>

<tr><td></td>

<td>Login ID</td>

<td>{{form.loginid}}</td>

</tr>

<tr><td></td>

<td>Password</td>

<td>{{form.password}}</td>

</tr>

<tr><td></td>

<td>Mobile</td>

<td>{{form.mobile}}</td>

</tr>

<tr><td></td>

<td>Email</td>

<td>{{form.email}}</td>

</tr>

<tr><td></td>

<td>Vehicle Number</td>

<td>{{form.vehiclenumber}}</td>

</tr>

<tr><td></td>

<td>Address</td>

<td>{{form.address}}</td>

</tr>

<tr><td></td>

<td>City</td>

<td>{{form.city}}</td>

</tr>

<tr><td></td>

<td>State</td>

<td>{{form.state}}</td>

</tr>

<tr><td></td>

<td></td>

<td>{{form.status}}</td>

</tr>

<tr><td></td>

<td></td>

<td><button type="submit" value="Register" class="button\_section">Register</button></td>

</tr>

<tr>

<td>

<div class="form-group mt-3">

<span >&nbsp;</span>

</div>

</td>

</tr>

{% if messages %}

{% for message in messages %}

<font color='white'> {{ message }}</font>

{% endfor %}

{% endif %}

</table>

</form>

</div>

</div>

<div class="col-md-7">

<div id="slider\_main" class="carousel slide" data-ride="carousel">

<!-- The slideshow -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="{%static 'images/slider\_1.png'%}" alt="#" />

</div>

<div class="carousel-item">

<img src="{%static 'images/slider\_2.png'%}" alt="#" />

</div>

</div>

</div>

</div>

</div>

</div>

</section>

{%endblock%}