**SOURCE CODE:**

**Userside views:**

from django.shortcuts import render,HttpResponse

from django.contrib import messages

from .forms import UserRegistrationForm

from .models import UserRegistrationModel

from django.core.files.storage import FileSystemStorage

# Create your views here.

def UserRegisterActions(request):

if request.method == 'POST':

form = UserRegistrationForm(request.POST)

if form.is\_valid():

print('Data is Valid')

form.save()

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

form = UserRegistrationForm()

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

else:

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

print("Invalid form")

else:

form = UserRegistrationForm()

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

def UserLoginCheck(request):

if request.method == "POST":

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

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

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

try:

check = UserRegistrationModel.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

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

return render(request, 'users/UserHome.html', {})

else:

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

return render(request, 'UserLogin.html')

except Exception as e:

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

pass

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

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

def UserHome(request):

return render(request, 'users/UserHome.html', {})

def UserSurveillance(request):

import os

import subprocess

try:

p = os.path.join(os.getcwd(), 'media', 'models', 'yolo.py')

print(p)

process = subprocess.call(['python', p, '--webcam', 'arg2'])

# output =subprocess.run(p,capture\_output=True,shell=False)

# exitstatus = process.poll()

return render(request, 'users/UserHome.html', {})

except Exception as ex:

return render(request, 'users/UserHome.html', {})

def UserViewImages(request):

return render(request, 'users/viewDataset.html',{})

def modelTraing(request):

# from .utility.trainingUtility import startModelBuilding

# h = startModelBuilding()

return render(request,'users/modelresults.html',{})

def UserImageTest(request):

if request.method == 'POST':

myfile = request.FILES['file']

fs = FileSystemStorage()

filename = fs.save(myfile.name, myfile)

uploaded\_file\_url = fs.url(filename)

from .utility import weaponr\_predictions

result = weaponr\_predictions.start\_prediction(filename)

print('Result:', result)

return render(request, "users/test\_form.html", {"result": result, "path": uploaded\_file\_url})

else:

return render(request, "users/test\_form.html", {})

**base.html**

<!doctype html>

{% load static %}

<!--[if IE 7 ]> <html lang="en-gb" class="isie ie7 oldie no-js"> <![endif]-->

<!--[if IE 8 ]> <html lang="en-gb" class="isie ie8 oldie no-js"> <![endif]-->

<!--[if IE 9 ]> <html lang="en-gb" class="isie ie9 no-js"> <![endif]-->

<!--[if (gt IE 9)|!(IE)]><!-->

<html lang="en-gb" class="no-js">

<!--<![endif]-->

<head>

<meta charset="utf-8">

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

<!--[if lt IE 9]>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<![endif]-->

<title>Weapon Detection - Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications</title>

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

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

<!--[if lt IE 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

<!--[if lte IE 8]>

<script type="text/javascript" src="http://explorercanvas.googlecode.com/svn/trunk/excanvas.js"></script>

<![endif]-->

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

<link rel="stylesheet" type="text/css" href="{%static 'css/isotope.css'%}" media="screen" />

<link rel="stylesheet" href="{%static 'js/fancybox/jquery.fancybox.css'%}" type="text/css" media="screen" />

<link rel="stylesheet" type="text/css" href="{%static 'css/da-slider.css'%}" />

<!-- Owl Carousel Assets -->

<link href="{%static 'js/owl-carousel/owl.carousel.css'%}" rel="stylesheet">

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

<!-- Font Awesome -->

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

</head>

<body>

<header class="header">

<div class="container">

<nav class="navbar navbar-inverse" role="navigation">

<div class="navbar-header">

<button type="button" id="nav-toggle" class="navbar-toggle" data-toggle="collapse" data-target="#main-nav">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a href="#" class="navbar-brand scroll-top logo"><b>Weapon Detection</b></a>

</div>

<!--/.navbar-header-->

<div id="main-nav" class="collapse navbar-collapse">

<ul class="nav navbar-nav" id="mainNav">

<li><a href="{%url 'index'%}" class="scroll-link">Home</a></li>

<li><a href="{%url 'UserLogin'%}" class="scroll-link">ML Scholar</a></li>

<li><a href="{%url 'AdminLogin'%}" class="scroll-link">Admin</a></li>

<li><a href="{%url 'UserRegister'%}" class="scroll-link">Registration</a></li>

</ul>

</div>

<!--/.navbar-collapse-->

</nav>

<!--/.navbar-->

</div>

<!--/.container-->

</header>

<!--/.header-->

<div id="#top"></div>

&nbsp;

{%block contents%}

{%endblock%}

<footer>

<div class="container">

<div class="clear"></div>

<!--CLEAR FLOATS-->

</div>

</footer>

<!--/.page-section-->

<section class="copyright">

<div class="container">

<div class="row">

<div class="col-sm-12 text-center">

<span>&copy; Weapon Detection 2023 All right reserved. By </span><a

href="#" target="\_blank">Alex Corporations</a>

</div>

</div>

<!-- / .row -->

</div>

</section>

<a href="#top" class="topHome"><i class="fa fa-chevron-up fa-2x"></i></a>

<!--[if lte IE 8]><script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script><![endif]-->

<script src="{%static 'js/modernizr-latest.js'%}"></script>

<script src="{%static 'js/jquery-1.8.2.min.js'%}" type="text/javascript"></script>

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

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

<script src="{%static 'js/fancybox/jquery.fancybox.pack.js'%}" type="text/javascript"></script>

<script src="{%static 'js/jquery.nav.js'%}" type="text/javascript"></script>

<script src="{%static 'js/jquery.cslider.js'%}" type="text/javascript"></script>

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

<script src="{%static 'js/owl-carousel/owl.carousel.js'%}"></script>

</body>

</html>

**Admin side views:**

from django.shortcuts import render,HttpResponse

from django.contrib import messages

from users.models import UserRegistrationModel

# Create your views here.

def AdminLoginCheck(request):

if request.method == 'POST':

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

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

print("User ID is = ", usrid)

if usrid == 'admin' and pswd == 'admin':

return render(request, 'admins/AdminHome.html')

elif usrid == 'Admin' and pswd == 'Admin':

return render(request, 'admins/AdminHome.html')

else:

messages.success(request, 'Please Check Your Login Details')

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

def AdminHome(request):

return render(request, 'admins/AdminHome.html')

def ViewRegisteredUsers(request):

data = UserRegistrationModel.objects.all()

return render(request, 'admins/RegisteredUsers.html', {'data': data})

def AdminActivaUsers(request):

if request.method == 'GET':

id = request.GET.get('uid')

status = 'activated'

print("PID = ", id, status)

UserRegistrationModel.objects.filter(id=id).update(status=status)

data = UserRegistrationModel.objects.all()

return render(request, 'admins/RegisteredUsers.html', {'data': data})

def adminViewModelResults(request):

return render(request, 'admins/modelResult.html',{})

**index.html**:

{%extends 'base.html'%}

{%load static%}

{%block contents%}

<section id="aboutUs" class="page-section darkBg pDark pdingBtm30">

<div class="container">

<div class="heading text-center">

<!-- Heading -->

<h2>Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications</h2>

<p>Real-time object detection to improve surveillance methods is one of the sought-after applications of Convolutional Neural Networks (CNNs).</p>

</div>

<div class="row">

<div class="col-md-4">

<img src="{%static 'images/weapon.png'%}" class="fitImage" alt="img" />

</div>

<div class="col-md-8">

<h3>Machine Learning</h3>

<p>Security is always a main concern in every domain, due to a rise in crime rate in a crowded event or suspicious lonely areas. Abnormal detection and monitoring have major applications of computer vision to tackle various problems. Due to growing demand in the protection of safety, security and personal properties, needs and deployment of video surveillance systems can recognize and interpret the scene and anomaly events play a vital role in intelligence monitoring. This paper implements automatic gun (or) weapon detection using a convolution neural network (CNN) based SSD and Faster RCNN algorithms. Proposed implementation uses two types of datasets. One dataset, which had pre-labelled images and the other one is a set of images, which were labelled manually. Results are tabulated, both algorithms achieve good accuracy, but their application in real situations can be based on the trade-off between speed and accuracy. </p>

</div>

</div>

<div class="row">

<div class="col-md-4 col-sm-4">

<h3><i class="fa fa-desktop color"></i>&nbsp;Convolutional</h3>

<!-- Paragraph -->

<p>In deep learning, a convolutional neural network (CNN or ConvNet) is a class of deep neural networks, that are typically used to recognize patterns present in images but they are also used for spatial data analysis, computer vision, natural language processing, signal processing, and various other purposes</p>

</div>

<div class="col-md-4 col-sm-4">

<!-- Heading -->

<h3><i class="fa fa-cloud color"></i>&nbsp;Pooling</h3>

<!-- Paragraph -->

<p>Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarises the features present in a region of the feature map generated by a convolution laye</p>

</div>

<div class="col-md-4 col-sm-4">

<!-- Heading -->

<h3><i class="fa fa-home color"></i>&nbspFlattening</h3>

<!-- Paragraph -->

<p>Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer</p>

</div>

</div>

</div>

<!--/.container-->

</section>

{%endblock%}