**SOURCE CODE**

1. **User side views.py**
2. import tensorflow as tf
3. from django.shortcuts import render, HttpResponse
4. from django.contrib import messages
5. from .forms import UserRegistrationForm
6. from .models import UserRegistrationModel,UserImageMaskModel
7. from django.core.files.storage import FileSystemStorage
8. from tensorflow import keras
9. from tensorflow.keras import layers
10. from tensorflow.keras.models import Sequential
11. from tensorflow import keras
12. import matplotlib.pyplot as plt
13. from tensorflow.keras.preprocessing.image import ImageDataGenerator
14. import numpy as np
15. import keras.backend as k
16. from keras.layers import Conv2D, MaxPooling2D, SpatialDropout2D, Flatten, Dropout, Dense
17. from keras.models import Sequential, load\_model
18. from keras.optimizers import Adam
19. # from tensorflow.keras.preprocessing.image import img\_to\_array
20. from tensorflow.keras.utils import img\_to\_array
21. import cv2
22. import datetime
23. from keras.models import Sequential
24. from keras.models import Model
25. # Create your views here.
26. def UserRegisterActions(request):
27. if request.method == 'POST':
28. form = UserRegistrationForm(request.POST)
29. if form.is\_valid():
30. print('Data is Valid')
31. form.save()
32. messages.success(request, 'You have been successfully registered')
33. form = UserRegistrationForm()
34. return render(request, 'UserRegistrations.html', {'form': form})
35. else:
36. messages.success(request, 'Email or Mobile Already Existed')
37. print("Invalid form")
38. else:
39. form = UserRegistrationForm()
40. return render(request, 'UserRegistrations.html', {'form': form})
41. def UserLoginCheck(request):
42. if request.method == "POST":
43. loginid = request.POST.get('loginname')
44. pswd = request.POST.get('pswd')
45. print("Login ID = ", loginid, ' Password = ', pswd)
46. try:
47. check = UserRegistrationModel.objects.get(
48. loginid=loginid, password=pswd)
49. status = check.status
50. print('Status is = ', status)
51. if status == "activated":
52. request.session['id'] = check.id
53. request.session['loggeduser'] = check.name
54. request.session['loginid'] = loginid
55. request.session['email'] = check.email
56. print("User id At", check.id, status)
57. return render(request, 'users/UserHome.html', {})
58. else:
59. messages.success(request, 'Your Account Not at activated')
60. return render(request, 'UserLogin.html')
61. except Exception as e:
62. print('Exception is ', str(e))
63. pass
64. messages.success(request, 'Invalid Login id and password')
65. return render(request, 'UserLogin.html', {})
66. def UserHome(request):
67. return render(request, 'users/UserHome.html', {})
68. def train(request):
69. model = Sequential()
70. model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(150, 150, 3)))
71. model.add(MaxPooling2D())
72. model.add(Conv2D(32, (3, 3), activation='relu'))
73. model.add(MaxPooling2D())
74. model.add(Conv2D(32, (3, 3), activation='relu'))
75. model.add(MaxPooling2D())
76. model.add(Flatten())
77. model.add(Dense(100, activation='relu'))
78. model.add(Dense(1, activation='sigmoid'))
79. model.compile(optimizer='adam', loss='binary\_crossentropy',metrics=['acc'])
80. train\_datagen = ImageDataGenerator(
81. rescale=1./255,
82. shear\_range=0.2,
83. zoom\_range=0.2,
84. horizontal\_flip=True)
85. test\_datagen = ImageDataGenerator(rescale=1./255)
86. training\_set = train\_datagen.flow\_from\_directory(
87. 'train',
88. target\_size=(150, 150),
89. batch\_size=16,
90. class\_mode='binary')
91. test\_set = test\_datagen.flow\_from\_directory(
92. 'test',
93. target\_size=(150, 150),
94. batch\_size=16,
95. class\_mode='binary')
96. history = model.fit\_generator(
97. training\_set,
98. epochs=2,
99. validation\_data=test\_set,)
100. print(history.history.keys())
101. # summarize model\_saved for accuracy
102. plt.plot(history.history['acc'])
103. plt.plot(history.history['val\_acc'])
104. plt.title('model acc')
105. plt.ylabel('acc')
106. plt.xlabel('epoch')
107. plt.legend(['train', 'test'], loc='upper left')
108. plt.savefig('model\_acc.png')
109. plt.show()
110. # summarize model\_saved for loss
111. plt.plot(history.history['loss'])
112. plt.plot(history.history['val\_loss'])
113. plt.title('model loss')
114. plt.ylabel('loss')
115. plt.xlabel('epoch')
116. plt.legend(['train', 'test'], loc='upper left')
117. plt.savefig('model\_loss.png')
118. plt.show()
119. return render(request, 'users/train.html')
120. def detect(request):
121. import subprocess
122. runvalue = "yolov5\_model.py"
123. subprocess.call("python " + runvalue)
124. return render(request, 'users/UserHome.html')
125. def upload\_image(request):
126. if request.method == 'POST':
127. myfile = request.FILES['file']
128. fs = FileSystemStorage()
129. filename = fs.save(myfile.name, myfile)
130. uploaded\_file\_url = fs.url(filename)
131. from .utility.yolomodels import face\_mask\_detection
132. results\_caption = face\_mask\_detection(filename)
133. loginid = request.session['loginid']
134. messages.success(request, 'Image Processed Success')
135. print("File Image Name " + uploaded\_file\_url)
136. return render(request, "users/upload\_image.html", {'path': uploaded\_file\_url})
137. else:
138. return render(request, 'users/upload\_image.html', {})
139. def UploadImageAction(request):
140. if request.method=='POST':
141. image\_file = request.FILES['file']
142. print('if....')
143. fs = FileSystemStorage(location="media/actions/")
144. filename = fs.save(image\_file.name, image\_file)
145. print('filename:',filename)
146. # detect\_filename = fs.save(image\_file.name, image\_file)
147. uploaded\_file\_url = "/media/actions/" + filename # fs.url(filename)
148. from .utility.ClassifyImages import image\_face\_mask\_detection
149. results\_caption = image\_face\_mask\_detection(filename)
150. loginid = request.session['loginid']
151. email = request.session['email']
152. UserImageMaskModel.objects.create(username=loginid, email=email, filename=filename, file=uploaded\_file\_url)
153. messages.success(request, 'Image Processed Success')
154. print("Image path ", uploaded\_file\_url)
155. return render(request,"users/upload\_image.html",{'path':uploaded\_file\_url})
156. else:
157. return render(request,'users/upload\_image.html',{})
158. # from
159. # obj = ClassifyUserPics()
160. # result = obj.startProcess(filename)
161. # print("Result=", result)
162. # return render(request, "users/UploadPicForm.html", {'result': result})
163. **Base.html:**
164. {%load static%}
165. <!DOCTYPE html>
166. <html lang="en">
167. <head>
168. <!-- basic -->
169. <meta charset="utf-8">
170. <meta http-equiv="X-UA-Compatible" content="IE=edge">
171. <!-- mobile metas -->
172. <meta name="viewport" content="width=device-width, initial-scale=1">
173. <meta name="viewport" content="initial-scale=1, maximum-scale=1">
174. <!-- site metas -->
175. <title>webwing</title>
176. <meta name="keywords" content="">
177. <meta name="description" content="">
178. <meta name="author" content="">
179. <!-- bootstrap css -->
180. <link rel="stylesheet" href="{%static 'css/bootstrap.min.css'%}">
181. <!-- style css -->
182. <link rel="stylesheet" href="{%static 'css/style.css'%}">
183. <!-- Responsive-->
184. <link rel="stylesheet" href="{%static 'css/responsive.css'%}">
185. <link rel="stylesheet" href="{%static 'css/owl.carousel.min.css'%}">
186. <!-- fevicon -->
187. <link rel="icon" href="{%static 'images/fevicon.png'%}" type="image/gif" />
188. <!-- Scrollbar Custom CSS -->
189. <link rel="stylesheet" href="{%static 'css/jquery.mCustomScrollbar.min.css'%}">
190. <!-- Tweaks for older IEs-->
191. <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css">
192. <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css" media="screen">
193. <!--[if lt IE 9]>
194. <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
195. <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-->
196. </head>
197. <!-- body -->
198. <body class="main-layout">
199. <!-- loader -->
200. <div class="loader\_bg">
201. <div class="loader"><img src="{%static 'images/loading.gif'%}" alt="#" /></div>
202. </div>
203. <!-- end loader -->
204. <!-- header -->
205. <header>
206. <!-- header inner -->
207. <div class="header">
208. <div class="header\_bo">
209. <div class="container">
210. <div class="row">
211. <div class="col-md-9 col-sm-7">
212. <nav class="navigation navbar navbar-expand-md navbar-dark ">
213. <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExample04" aria-controls="navbarsExample04" aria-expanded="false" aria-label="Toggle navigation">
214. <span class="navbar-toggler-icon"></span>
215. </button>
216. <div class="collapse navbar-collapse" id="navbarsExample04">
217. <ul class="navbar-nav mr-auto">
218. <li class="nav-item"><a class="nav-link" href="{%url 'index'%}"> Home </a></li>
219. <li class="nav-item"><a class="nav-link" href="{%url 'UserLogin'%}">Users</a> </li>
220. <li class="nav-item"><a class="nav-link" href="{%url 'AdminLogin'%}">Admin</a></li>
221. <li class="nav-item"><a class="nav-link" href="{%url 'UserRegister'%}">Registrations </a></li>
222. </ul>
223. </div>
224. </nav>
225. </div>
226. </div>
227. </div>
228. </div>
229. </div>
230. </header>
231. <!-- end header inner -->
232. <!-- end header -->
233. <!-- banner -->
234. <section class="banner\_main">
235. <div class="container">
236. <div class="row">
237. {%block contents%}
238. {%endblock%}
239. </div>
240. <div class="col-md-5 col-lg-5">
241. <div class="ban\_img">
242. <figure>
243. <img src="{% static 'images/ba\_ing.png' %}" alt="width=100% ,height=100%" />
244. </figure>
245. </div>
246. </div>
247. </div>
248. </div>
249. </section>
250. <!-- footer -->
251. <footer>
252. <div class="footer">
253. <div class="copyright">
254. <div class="container">
255. <div class="row">
256. <div class="col-md-12">
257. <p>© 2023 All Rights Reserved.<a href="#"> Alec Corporations</a></p>
258. </div>
259. </div>
260. </div>
261. </div>
262. </div>
263. </footer>
264. <!-- end footer -->
265. <!-- Javascript files-->
266. <script src="{%static 'js/jquery.min.js'%}"></script>
267. <script src="{%static 'js/popper.min.js'%}"></script>
268. <script src="{%static 'js/bootstrap.bundle.min.js'%}"></script>
269. <script src="{%static 'js/jquery-3.0.0.min.js'%}"></script>
270. <script src="{%static 'js/owl.carousel.min.js'%}"></script>
271. <!-- sidebar -->
272. <script src="{%static 'js/jquery.mCustomScrollbar.concat.min.js'%}"></script>
273. <script src="{%static 'js/custom.js'%}"></script>
274. </body>
275. </html>
276. **Admin side views.py:**
277. from django.shortcuts import render,HttpResponse
278. from django.contrib import messages
279. from users.models import UserRegistrationModel
280. # Create your views here.
281. def AdminLoginCheck(request):
282. if request.method == 'POST':
283. usrid = request.POST.get('loginid')
284. pswd = request.POST.get('pswd')
285. print("User ID is = ", usrid)
286. if usrid == 'admin' and pswd == 'admin':
287. return render(request, 'admins/AdminHome.html')
288. elif usrid == 'Admin' and pswd == 'Admin':
289. return render(request, 'admins/AdminHome.html')
290. else:
291. messages.success(request, 'Please Check Your Login Details')
292. return render(request, 'AdminLogin.html', {})
293. def AdminHome(request):
294. return render(request, 'admins/AdminHome.html')
295. def ViewRegisteredUsers(request):
296. data = UserRegistrationModel.objects.all()
297. return render(request, 'admins/RegisteredUsers.html', {'data': data})
298. def AdminActivaUsers(request):
299. if request.method == 'GET':
300. id = request.GET.get('uid')
301. status = 'activated'
302. print("PID = ", id, status)
303. UserRegistrationModel.objects.filter(id=id).update(status=status)
304. data = UserRegistrationModel.objects.all()
305. return render(request, 'admins/RegisteredUsers.html', {'data': data})