

Cins Face Recognition using Deep Learning

Machine Learning Homework3

Code Explanation and User Manual

Erdoğan Abacı
150315025

erdoganabaci97@gmail.com

Computer Science Engineering
Celal Bayar University – Manisa/Turkey

June 01, 2019

Firstly; I wrote the code with python and my compiler is spyder.I Imported these libraries.

```
1 #to encode image pixels 1 to 0
2 from sklearn.preprocessing import LabelEncoder
3 #k fold split test and validation set to classify data is close result
4 from sklearn.model_selection import train_test_split
5 #to shuffle arrays or sparse matrices in a consistent way
6 from sklearn.utils import shuffle
7 #to draw loss and accuracy
8 import matplotlib.pyplot as plt
9
10 #to change learning rate.gradient descent using.
11 from keras import layers, optimizers, models
12 #to use l2 penalty regularization
13 from keras.constraints import maxnorm
14 #to classifying more than one person so using categorical
15 from keras.utils import to_categorical
16 #to prevent overfitting data.Close to real image
17 from keras.preprocessing.image import ImageDataGenerator
18 #very usefull l2 penalty adaptive learning rate.
19 from keras.optimizers import RMSprop,Adam
20 #collection of algorithms for image processing.
21 from skimage import io
22 #to use array
23 import numpy as np
24 #to draw line resize ,crop image
25 import cv2
26 #to acces os system for example mkdir to create folder.
27 import os
28 #to play voice
29 from pygame import mixer
30 #to sleep program
31 import time
```

Take first %70 sample as train data and last %30 sample as test data. Train images pixel and get save models

```

43 def train():
44     global y
45     x = None
46
47     if not os.path.exists('dataset.npy'):
48
49         index = 1
50         imgs = []
51         for dir in dirs:
52             #access images
53             images = os.listdir(os.path.join("img", dir))
54             for image in images:
55                 #check file extension.
56                 if image.endswith(".png"):
57                     #convert image to pixel with prepare_image (50, 50, 1)
58                     tmp_img = prepare_image(os.path.join("img", dir, image))
59                     #print(tmp_img.shape)
60                     #to desired image size and change shape to give cnn filters and ann(artificall neural network)
61                     tmp_img = resize(width_height, width_height, tmp_img)
62                     #reshape to desired pixel
63                     tmp_img = tmp_img.reshape((tmp_img.shape[0] * tmp_img.shape[1],))
64                     b = np.zeros((tmp_img.shape[0] + 1,))
65                     b[0] = y[index - 1]
66                     b[1:] = tmp_img
67                     imgs.append(b)
68                 index += 1
69
70         x = np.array(imgs)
71         #shuffle pixels learn different images to prevent overfitting
72         x = shuffle(x)
73         #to save learning pixel which means that every pixel coded in machine
74         np.save("dataset", x)
75     else:
76         x = np.load("dataset.npy")
77         #pixels for cnn convert shape.
78         y = x[:, [0]]
79         y = y.reshape(y.shape[0], )
80         x = x[:, 1:]
81         #my classifed size. which means that if 1 5 person ,i clasifed 5 perosn
82         total = len(dirs)
83         #our pixels
84         x = x.astype('float64')
85         x = x.reshape(x.shape[0], width_height, width_height, 1)
86         #people corresponding to pixels
87         y = to_categorical(y, total)
88
89         model = models.Sequential()
90         #this is first neuron it helps to 32 filter 5x5 dimension matris ,stride 1 default , maxnorm(m) will, if the L2-Norm of your weights exceed
91         #input shape our data shape 50x50 and 1 means is we use gray scale to machine easily train data
92         model.add(layers.Conv2D(32, (5, 5), padding='same', kernel_constraint=maxnorm(3), input_shape=(width_height, width_height, 1)))
93         #activation function is relu because easy to derivate function and if you pass greater than 0 you pass the activation and second layer.
94         model.add(layers.Activation('relu'))
95         #pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next
96         model.add(layers.MaxPooling2D(pool_size=(2, 2)))
97
98         #same layer adding. More layers means more detail.
99         model.add(layers.Conv2D(64, (5, 5), padding='same', kernel_constraint=maxnorm(3)))
100         model.add(layers.Activation('relu'))

```

Resize and Crop images to scale image to prepare train

```

184 def resize(max_height: int, max_width: int, frame: np.ndarray) -> np.ndarray:
185     height, width = frame.shape[:2]
186
187     if max_height < height or max_width < width:
188         if width < height:
189             scaling_factor = max_width / float(width)
190         else:
191             scaling_factor = max_height / float(height)
192
193     frame = cv2.resize(frame, None, fx=scaling_factor, fy=scaling_factor, interpolation=cv2.INTER_AREA)
194     return frame
195
196 #crop face(images) in center
197 def crop_center(frame: np.ndarray) -> np.ndarray:
198     short_edge = min(frame.shape[:2])
199     yy = int((frame.shape[0] - short_edge) / 2)
200     xx = int((frame.shape[1] - short_edge) / 2)
201     crop_img = frame[yy: yy + short_edge, xx: xx + short_edge]
202     return crop_img
203
204     start = int((frame.shape[1] - frame.shape[0]) / 2)
205     end = int(frame.shape[1] - (frame.shape[1] - frame.shape[0]) / 2)
206     return frame[:, start:end]
207

```

Detect Face with Model

```
208
209 def detect_my_face():
210     #face recognition with voice
211     voice_active = input("Do you want to voice activate ? [Y]/n ")
212     voice_active.lower()
213     if(voice_active == 'y'):
214         print("Voice Activated.")
215     else:
216         print("Voice Deactivated")
217     # capture frames from a camera
218     cap = cv2.VideoCapture(0)
219     #creation dense in this part we create dense model to detect images(faces)
220     model = models.Sequential()
221     #our faces
222     with open('face.json', 'r') as f:
223         model = models.model_from_json(f.read())
224     #our w which means that weights model
225     model.load_weights('face.h5')
226
227     while (1):
228         #read camera frames
229         ret, real_frame = cap.read()
230         #convert rgb to grayscale
231         frame = cv2.cvtColor(real_frame, cv2.COLOR_BGR2GRAY)
232         #resize and crop image to recognize spesification
233         frame = resize(width_height, width_height, frame)
234         frame = crop_center(frame)
235         #convert binary to normalize data
236         predict = frame / 255
237         predict = predict.astype('float64')
238         #convert rgb to gray
239         predict = predict.reshape(width_height, width_height, 1)
240         #predict the frame(image captured) according to model
241         prediction = model.predict(np.array([predict])).tolist()
242         #if prediction result greater than our trashold result then prediction is correct
243         prediction_result = np.max(prediction)
244         #to get accurate person(trashold level=limit value)
245         if prediction_result > 0.5:
246             #directory name which means that predicted names
247             text = dirs[np.argmax(prediction)]
248             #convert prediction result
249             prediction_result = "%.2f" % prediction_result
250             prediction_result = int(float(prediction_result)*100)
251             #to draw line we define start line and end line
252             start = int((real_frame.shape[1] - real_frame.shape[0]) / 1.5)
253             end = int(real_frame.shape[1] - (real_frame.shape[1] - real_frame.shape[0]) / 1.5)
254             #position of the prediction text
255             cv2.putText(real_frame, text, (0, 160), cv2.FONT_HERSHEY_DUPLEX, 1.6, (0, 0, 255))
256             cv2.putText(real_frame, "%"+str(prediction_result), (0, 55), cv2.FONT_HERSHEY_DUPLEX, 1.6, (0, 0, 255))
257             #if erdogan program says i recognize you if not access is denied
258             if text == "erdogan":
259                 cv2.putText(real_frame, "i recognize you", (0, 100), cv2.FONT_HERSHEY_DUPLEX, 1.6, (0, 0, 0))
260                 print("Ok.. I recognize you.. Welcome")
261                 print("To close camera push ESC ")
262             else:
263                 cv2.putText(real_frame, "Access is denied", (0, 100), cv2.FONT_HERSHEY_DUPLEX, 1.6, (0, 0, 0))
264                 print("Who you are? Access is denied")
265                 print("To close camera push ESC ")
266     ..
---
```

Prepare image to convert image to pixel **normalize pixels**

```
323 #to convert image to pixel
324 def prepare_image(image_path: str) -> np.ndarray:
325     tmp_img = io.imread(image_path)
326     #I chose grayscale to avoid fatigue.
327     tmp_img = cv2.cvtColor(tmp_img, cv2.COLOR_BGR2GRAY)
328     tmp_img = resize(width_height, width_height, tmp_img)
329     tmp_img = crop_center(tmp_img)
330     #all image normalize with encode 0 to 1
331     tmp_img = tmp_img / 255
332     tmp_img = tmp_img.astype('float64')
333
334     tmp_img = tmp_img.reshape(width_height, width_height, 1)
335     return tmp_img
336
```

Save image to create image data my own datas with capture the camera

```
300 def save_images(path):
301     print('Loading ' + path + "image")
302     cap = cv2.VideoCapture(0)
303     i = 0
304     while True:
305         print(str(i))
306         i = i + 1
307
308         ret, frame = cap.read()
309         frame = resize(width_height, width_height, frame)
310         frame = crop_center(frame)
311
312         cv2.imshow("Shrunked image", frame)
313         cv2.imwrite('img/' + path + '/' + str(i) + '.png', frame)
314         #exit with esc
315         cv2.waitKey(5) & 0xFF
316         #when take 1000 images break
317         if i >= 1001:
318             break
319
320     cap.release()
321     cv2.destroyAllWindows()
322
```

User Manual

Download Libraries if you use anaconda with anaconda prompt

- conda install -c conda-forge keras
- python -m pip install matplotlib.pyplot
- python -m pip install numpy

If you want to train you have to use gpu provide tensorflow-gpu.It help to parallel cuda gpu train data.Otherwise use it cpu.

- pip install tensorflow-gpu

To run command

python face.py

If you want to exit from camera you can push 'esc'

Code output

X train images (I commented 48 and 62 code lines)

```
[[[0.48235294]
 [0.46666667]
 [0.45882353]
 ...
 [0.43921569]
 [0.44313725]
 [0.44705882]]

 [[0.49803922]
 [0.4627451 ]
 [0.44705882]
```

Figure 1X train

Y train values

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

Figure 2 Y train

Number of Iteration(epoch) neural network start to learn

I commented because I have already trained my img folders data

```
Epoch 10/50
59/59 [=====] - 1s 19ms/step - loss: 0.4724 - acc: 0.8539 - val_loss: 0.1684 -
val_acc: 0.9593
Epoch 11/50
59/59 [=====] - 1s 19ms/step - loss: 0.4088 - acc: 0.8649 - val_loss: 0.1966 -
val_acc: 0.9395
Epoch 12/50
59/59 [=====] - 1s 18ms/step - loss: 0.5024 - acc: 0.8503 - val_loss: 0.1219 -
val_acc: 0.9716
Epoch 13/50
59/59 [=====] - 1s 19ms/step - loss: 0.3358 - acc: 0.8893 - val_loss: 0.0947 -
val_acc: 0.9704
Epoch 14/50
59/59 [=====] - 1s 19ms/step - loss: 0.3351 - acc: 0.8813 - val_loss: 0.1413 -
val_acc: 0.9457
Epoch 15/50
59/59 [=====] - 1s 18ms/step - loss: 0.3279 - acc: 0.8935 - val_loss: 0.0804 -
val_acc: 0.9741
Epoch 16/50
59/59 [=====] - 1s 19ms/step - loss: 0.4051 - acc: 0.8677 - val_loss: 0.1256 -
val_acc: 0.9568
Epoch 17/50
5/59 [=>.....] - ETA: 0s - loss: 0.4258 - acc: 0.8625
```

Figure 3 Epoch

Loss decrease Model accuracy increase and W size and minimized w values

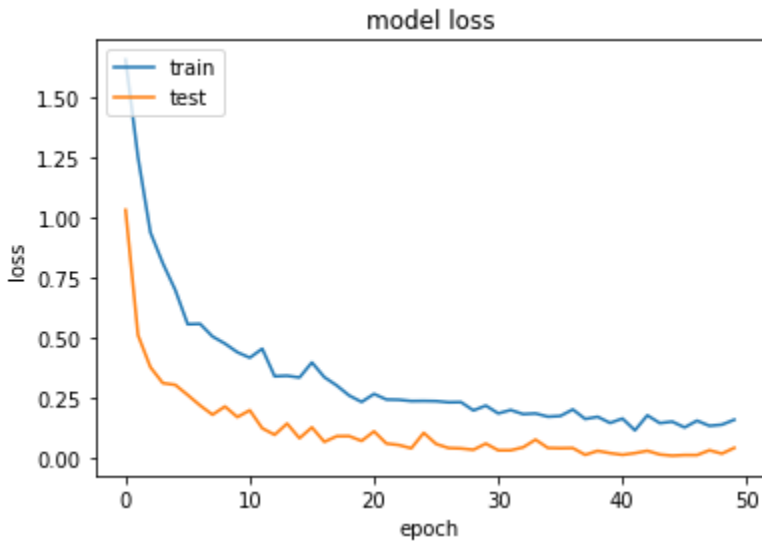


Figure 4 Model Loss

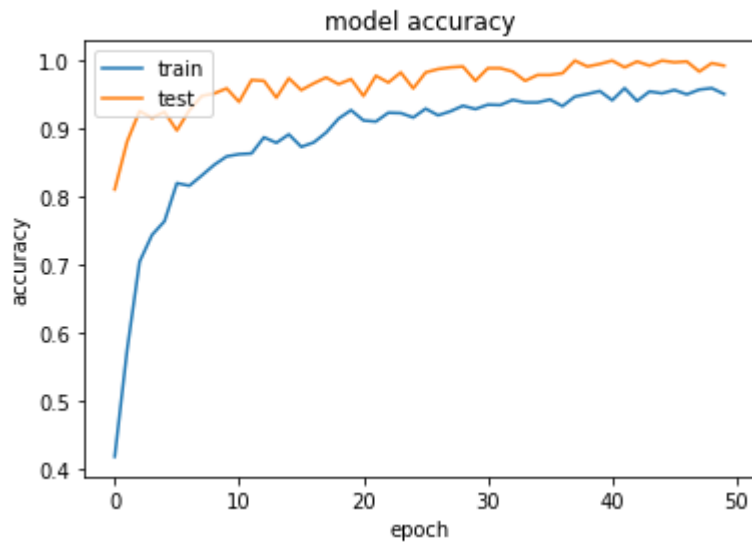


Figure 5 Model Accuracy

If recognize me then say , I recognize you otherwise , acces denied

```
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
Ok.. I recognize you.. Welcome
To close camera push ESC
```

Figure 6 Recognize You

If you open voice time.sleep(5) which means that every 5 second check who am I? with take a frame.

If you want to exit from camera you can push 'esc'



Figure 7 My Real Time Capture Camera :)

