

FACIAL RECOGNITION



Fall 2019
Praveena Shrestha
Machine learning

Introduction

When it began

- 1960s, Bledsoe developed a system that could classify photos of faces by hand using RAND tablet
- a device that people could use to input horizontal and vertical coordinates on a grid using a stylus and manually record the coordinate locations of various facial features including the eyes, nose, hairline and mouth.

Transition

- Social Media (2011 - Present)
- OSAMA BIN LADEN IDENTIFIED (2011)
- AIRPORT (2011)
- IPHONE X (2017)
- Everywhere (present)

How does it work

- OpenCV
- Haar Cascade Classifiers
- LBPH Algorithm
- Numpy

OpenCV/2

What is OpenCV ?

- Most popular library for computer vision
- Uses Machine learning algorithms to search for faces within a picture.
- Real-time analytics of images and recognition can be done for assorted applications

Classifiers

- The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve.
- 6000 or more classifiers
- Scikit-Learn (Logistic Regression, K Nearest Neighbors)

Haar Cascade Classifier

- A machine learning object detection algorithm used to identify objects in an image or video
- breaks the problem of detecting faces into multiple stages
- algorithm have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass

Numpy in OpenCV

OpenCV-Python makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax.

All the OpenCV array structures are converted to and from Numpy arrays.

This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

LBPH Algorithm

provided by the **OpenCV** library

Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.

Neighbors: the number of sample points to build the circular local binary pattern. It is usually set to 8.

Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality feature vector, usually set to 8.

Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector, usually set to 8.

More to LBPH



3x3 pixels



200	50	50
50	90	100
160	70	210

Threshold
90



1	0	0
0		1
1	0	1

Binary
10001101



150	90	80
30	141	

Decimal
141

Implementation

FOLDERS

- FaceRecognition-master
 - .idea
 - __pycache__
 - HaarCascade
 - haarcascade_frontalface_default.xml
 - resizedTrainingImages
 - TestImages
 - trainingImages
 - .gitignore
 - faceRecognition.py
 - Readme.md
 - resizedImages.py
 - tester.py
 - trainingData.yml
 - videoTester.py
 - videotoimg.py

```

41 and on any theory of liability, whether in contract, strict liability,
42 or tort (including negligence or otherwise) arising in any way out of
43 the use of this software, even if advised of the possibility of such damage.
44 -->
45 <opencv_storage>
46 <cascade type_id="opencv-cascade-classifier"><stageType>BOOST</stageType>
47 <featureType>HAAR</featureType>
48 <height>24</height>
49 <width>24</width>
50 <stageParams>
51 <maxWeakCount>211</maxWeakCount></stageParams>
52 <featureParams>
53 <maxCatCount>0</maxCatCount></featureParams>
54 <stageNum>25</stageNum>
55 <stages>
56 <>
57 <maxWeakCount>9</maxWeakCount>
58 <stageThreshold>-5.0425500869750977e+00</stageThreshold>
59 <weakClassifiers>
60 <>
61 <internalNodes>
62 0 -1 0 -3.1511999666690826e-02</internalNodes>
63 <leafValues>
64 2.0075380039215080e+00 -2.2172100543975830e+00</leafValues></>
65 <>
66 <internalNodes>
67 0 -1 1 1.2396000325679779e-02</internalNodes>
68 <leafValues>
69 -1.8633940219879150e+00 1.3272049427032471e+00</leafValues></>
70 <>
71 <internalNodes>
72 0 -1 2 2.1927999332547188e-02</internalNodes>
73 <leafValues>
74 -1.5105249881744385e+00 1.0625729560852051e+00</leafValues></>
75 <>
76 <internalNodes>
77 0 -1 3 5.7529998011887074e-03</internalNodes>
78 <leafValues>
79 -8.7463897466659546e-01 1.1760339736938477e+00</leafValues></>
80 <>
81 <internalNodes>
82 0 -1 4 1.5014000236988068e-02</internalNodes>
83 <leafValues>
84 -7.7945697307586670e-01 1.2608419656753540e+00</leafValues></>
85 <>
86 <internalNodes>
87 0 -1 5 9.9371001124382019e-02</internalNodes>
88 <leafValues>
89 5.5751299850093262e-01 -1.8743000030517578e+00</leafValues></>
90 <>
91 <internalNodes>
92 0 -1 6 2.7340000960975885e-03</internalNodes>
93 <leafValues>
94 1.4011000000000000e+00 -1.4000000000000000e+00</leafValues></>

```



```
1  #YAML:1.0
2
3  opencv_lbpHfaces:
4    threshold: 1.7976931348623157e+308
5    radius: 1
6    neighbors: 8
7    grid_x: 8
8    grid_y: 8
9    histograms:
10     - !!opencv-matrix
11       rows: 1
12       cols: 16384
13       dt: f
14       data: [ 6.31487891e-02, 3.71011160e-02, 1.63398695e-03,
15              5.09419432e-03, 1.11495573e-02, 3.65244132e-03,
16              4.32525948e-03, 1.59554016e-02, 2.59515573e-03,
17              1.44175312e-03, 0., 3.84467508e-04, 7.01653212e-03,
18              3.36409081e-03, 1.08612068e-02, 4.00807373e-02,
19              4.47904654e-02, 1.20146098e-02, 5.76701248e-04,
20              2.11457140e-03, 2.21068808e-03, 9.61168771e-05,
21              1.82622066e-03, 2.21068808e-03, 9.80392192e-03,
22              1.92233757e-03, 4.80584393e-04, 1.05728570e-03,
23              1.13417916e-02, 2.21068808e-03, 2.57593226e-02,
24              2.11457126e-02, 9.61168786e-04, 9.61168786e-04,
25              1.92233754e-04, 2.88350624e-04, 2.88350624e-04,
26              9.61168771e-05, 4.80584393e-04, 5.76701248e-04,
27              1.92233754e-04, 0., 0., 0., 9.61168771e-05, 9.61168771e-05,
28              4.80584393e-04, 3.46020749e-03, 8.26605130e-03,
29              1.73010374e-03, 6.72818162e-04, 9.61168786e-04,
30              6.72818162e-04, 0., 7.68935017e-04, 4.80584393e-04,
31              4.70972713e-03, 3.84467508e-04, 5.76701248e-04,
32              3.84467508e-04, 4.32525948e-03, 4.80584393e-04,
33              8.84275232e-03, 9.03498661e-03, 8.93886946e-03,
34              2.49903882e-03, 1.92233754e-04, 1.92233754e-04,
35              1.73010374e-03, 0., 8.65051872e-04, 4.80584393e-04,
36              4.80584393e-04, 9.61168771e-05, 0., 9.61168771e-05,
37              5.76701248e-04, 9.61168771e-05, 8.65051872e-04,
38              1.05728570e-03, 3.65244132e-03, 3.84467508e-04,
39              9.61168771e-05, 0., 0., 0., 0., 0., 1.82622066e-03, 0.,
40              9.61168771e-05, 2.88350624e-04, 6.72818162e-04, 0.,
41              1.24951941e-03, 5.76701248e-04, 4.13302565e-03,
42              2.97962315e-03, 9.61168771e-05, 2.88350624e-04,
43              8.65051872e-04, 9.61168771e-05, 3.84467508e-04,
44              6.72818162e-04, 9.61168771e-05, 9.61168771e-05, 0.,
45              9.61168771e-05, 5.76701248e-04, 9.61168771e-05,
46              1.05728570e-03, 2.59515573e-03, 1.98961943e-02,
```

```

1 import cv2
2 import os
3 import numpy as np
4 import faceRecognition as fr
5
6
7 #This module takes images stored in disk and performs face recognition
8 test_img=cv2.imread('TestImages/kangana.jpg')#test_img path
9 faces_detected,gray_img=fr.faceDetection(test_img)
10 print("faces_detected:",faces_detected)
11
12
13 #Comment belows lines when running this program second time.Since it saves
14 faces,faceID=fr.labels_for_training_data('trainingImages')
15 face_recognizer=fr.train_classifier(faces,faceID)
16 face_recognizer.write('trainingData.yml')
17
18
19 #Uncomment below line for subsequent runs
20 # face_recognizer=cv2.face.LBPHFaceRecognizer_create()
21 # face_recognizer.read('trainingData.yml')#use this to load training data
22
23 name={0:"Priyanka",1:"Kangana"}#creating dictionary containing names for e
24
25 for face in faces_detected:
26     (x,y,w,h)=face
27     roi_gray=gray_img[y:y+h,x:x+h]
28     label,confidence=face_recognizer.predict(roi_gray)#predicting the labe
29     print("confidence:",confidence)
30     print("label:",label)
31     fr.draw_rect(test_img,face)
32     predicted_name=name[label]
33     if(confidence>37):#If confidence more than 37 then don't print predict
34         continue
35     fr.put_text(test_img,predicted_name,x,y)
36
37 resized_img=cv2.resize(test_img,(1000,1000))
38 cv2.imshow("face detection tutorial",resized_img)
39 cv2.waitKey(0)#Waits indefinitely until a key is pressed
40 cv2.destroyAllWindows
41
42

```

```

1 import os
2 import cv2
3 import numpy as np
4 import faceRecognition as fr
5
6
7 #This module captures images via webcam and performs face recognition
8 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
9 face_recognizer.read('trainingData.yml')#Load saved training data
10
11 name = {0 : "Praveena Shrestha",1 : "Erik Grielmanan"}
12
13
14 cap=cv2.VideoCapture(0)
15
16 while True:
17     ret,test_img=cap.read()# captures frame and returns boolean value and
18     faces_detected,gray_img=fr.faceDetection(test_img)
19
20
21
22     for (x,y,w,h) in faces_detected:
23         cv2.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)
24
25     resized_img = cv2.resize(test_img, (1000, 700))
26     cv2.imshow('face detection Tutorial ',resized_img)
27     cv2.waitKey(10)
28
29
30     for face in faces_detected:
31         (x,y,w,h)=face
32         roi_gray=gray_img[y:y+w, x:x+h]
33         label,confidence=face_recognizer.predict(roi_gray)#predicting the
34         print("confidence:",confidence)
35         print("label:",label)
36         fr.draw_rect(test_img,face)
37         predicted_name=name[label]
38         if confidence < 39:#If confidence less than 37 then don't print pr
39             fr.put_text(test_img,predicted_name,x,y)
40
41
42     resized_img = cv2.resize(test_img, (1000, 700))
43     cv2.imshow('face recognition tutorial ',resized_img)

```

8071ea_Kanagana_ranaut.jpg
id: 1
img_path: trainingImages/1/kangana_ranaut_1553246965.jpeg
id: 1
img_path: trainingImages/1/indian_bollywood_actress_kangana_ranaut_showcas
reation_by_at_picture_id1092782148.jpg
id: 1
img_path: trainingImages/1/Manikarnika_bg_1.jpg
id: 1
img_path: trainingImages/1/session_paliwal_actress_kangana_ranaut_mastercl
gnature_6791aa46_4ca2_11e9_aca9_eac9e517f545.jpg
id: 1
img_path: trainingImages/1/MV5BMTA2NTg0ODA1ODJJeQTJeQWpwZ15BbWU4MDM0NDkxNTM
UY317_CR5_0_214_317_AL_.jpg
id: 1
img_path: trainingImages/1/58567_tksenxnhbd_1495192424.jpg
id: 1
img_path: trainingImages/1/have_been_harassed_by_actors_on_sets__says_kang
naut_2019_01_21.jpg
id: 1
img_path: trainingImages/1/435069_kangana.jpg
id: 1
img_path: trainingImages/1/kangna_ranaut_20150417103630_164.jpg
id: 1
img_path: trainingImages/1/kangana_ranaut_1505378148.jpeg
id: 1
img_path: trainingImages/1/kangana_saree_look_anita_dongre.jpg
id: 1
img_path: trainingImages/1/Kan.jpg
id: 1
img_path: trainingImages/1/Kangana.jpg
id: 1
img_path: trainingImages/1/Kangana_Ranaut_g_d.jpg
id: 1
img_path: trainingImages/1/aa_Cover_567vnisio43sb4ngroftgfljf1_20180619225
di.jpeg

ANACONDA NAVIGATOR

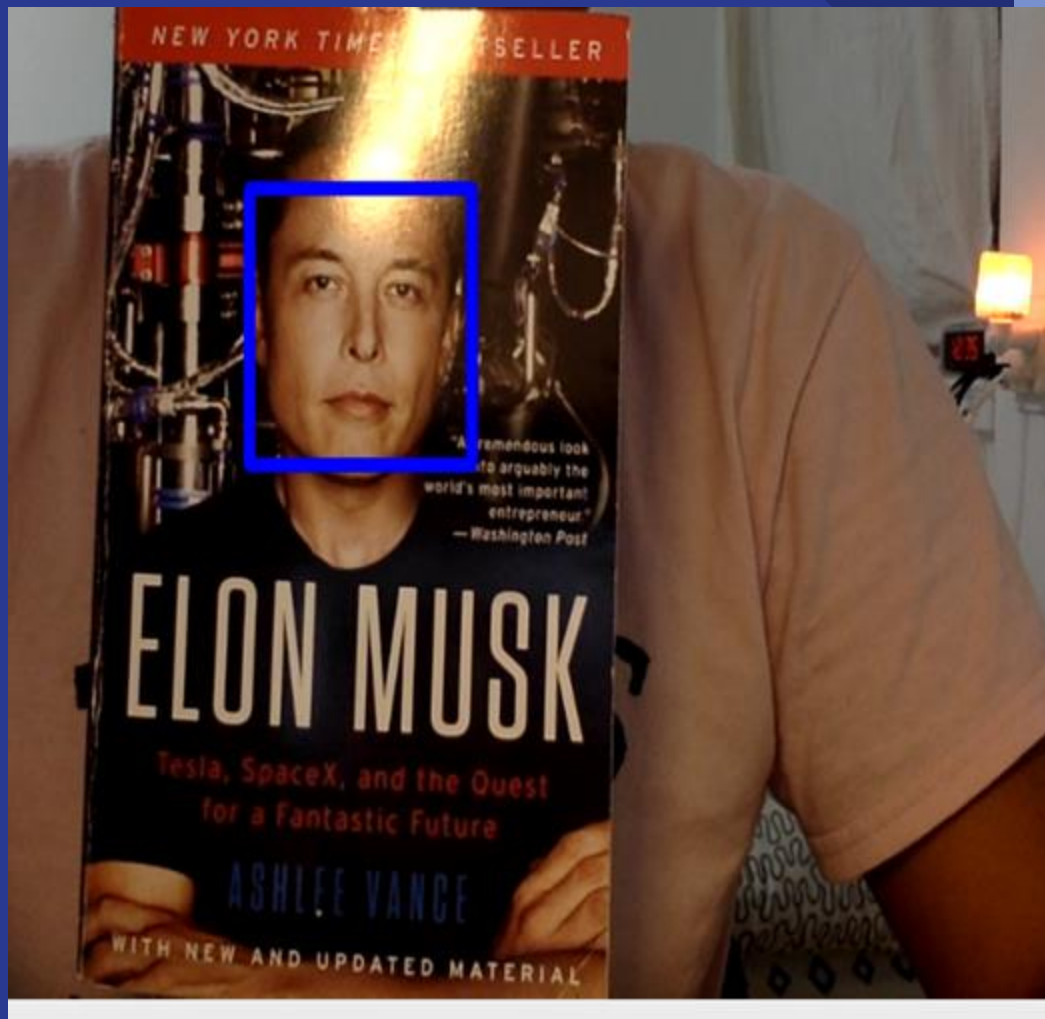
```
Application on: here (root)
FaceRecognition-master — python test
img_path: trainingImages/1/111508_dajqqajfw_1548486
id: 1
img_path: trainingImages/1/kangana_ranaut (1).jpg
id: 1
img_path: trainingImages/1/67896457.jpg
id: 1
img_path: trainingImages/1/Kangana_Ranaut.gif
id: 1
Image not loaded properly
img_path: trainingImages/1/mumbai_actress_kangana_ra
51d3edc_187e_11e9_9a47_fd04d5270281.jpg
id: 1
img_path: trainingImages/1/05_07_2018_kng_shdi_1_181
id: 1
img_path: trainingImages/1/kangana_ranaut_i_want_to_
hood_19_1482123573.jpg
id: 1
img_path: trainingImages/1/220px_Kangana_Ranaut_at_F
id: 1
img_path: trainingImages/1/Inside_Kangana_Ranaut_Sta
rty.jpg
id: 1
img_path: trainingImages/1/gs0tZiOd_400x400.jpg
id: 1
img_path: trainingImages/1/kangana_ranaut_21.jpg
id: 1
img_path: trainingImages/1/kangana_ranaut_slams_to_k
them_if_they_continue_to_harass_her_1547805217_725x
id: 1
img_path: trainingImages/1/kangana_ranaut_biopic_155
id: 1
img_path: trainingImages/1/Kangana_Ranaut_Tomatohear
id: 1
```

```
videoTester.py — FaceRecognition-master
videoTester.py x videotimg.py x resizeImages.py x trainingData.yml x
1 import os
2 import cv2
3 import numpy as np
4 import faceRecognition as fr
5
6
7 #This module captures images via webcam and performs face recognition
8 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
9 face_recognizer.read('trainingData.yml') #Load saved training data
10
```

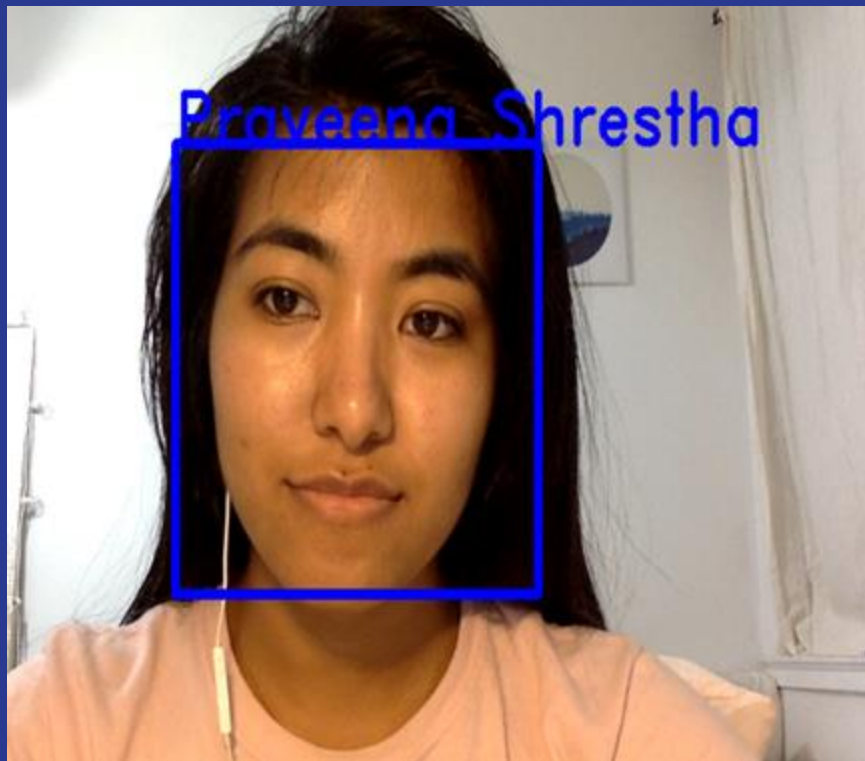
face detection tutorial

Kangana





Praveena Shrestha





THANK YOU

