

Faces detector app

Data Analytics bootcamp final project

Friday, 22 May 2020

(a.k.a. Adience Benchmark) Database

/ 23000 faces of different nationalities

/ Male, Female

/ 8 different ages groups:

'0' : '(0,3)',
'1' : '(4,7)',
'2' : '(8,14)',
'3' : '(15,21)',
'4' : '(22,33)',
'5' : '(34,43)',
'6' : '(44,53)',
'7' : '(54,100)'



<https://talhassner.github.io/home/projects/Adience/Adience-data.html>

database-image processing

/ CV2 module

/ face detection

/gray color map

/normalization

/Resize 64x64 px

/ stack



```
train_images=[]
for image in X['path']:
    im = cv2.imread(image)
    im_bw = cv2.cvtColor(im, cv2.COLOR_RGB2GRAY)
    try:
        faces = face_cascade.detectMultiScale(im_bw, 2, 4)
        if len(faces) > 1:
            faces=faces[0]
        x,y,w,h = faces
        cara = im_bw[y:y+h,x:x+h]
    except:
        cara=im_bw
    resized = cv2.resize(cara, (64,64))
    resized = resized.astype('float32')
    mean, sd = cv2.meanStdDev(resized)
    imageProc = (resized - mean) / sd
    train_images.append(imageProc)

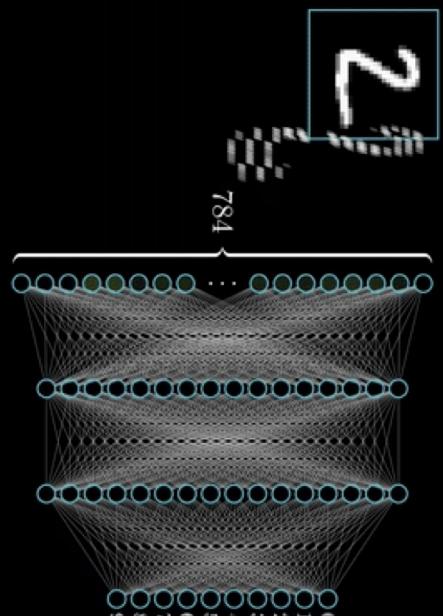
train_x=np.stack(train_images)#provare a togliere la med
```

colab-age & gender model

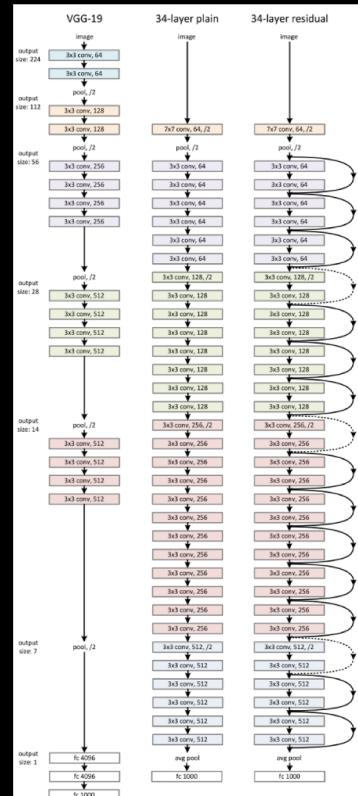
/input: 4-dimension array

/test – train split

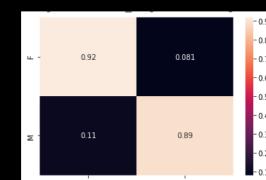
/Colab GUI runtime



Personal model [Age]
/accuracy: 0.8201



Resnet50 [Gender]
/accuracy: 0.8071



1. More layers is better but because of the vanishing gradient problem, weights of the first layers can not be updated correctly. That is the objective of Resnet : preserve the gradient.

2. How ? Thanks to the identity matrix because “*what if we were to backpropagate through the identity function? Then the gradient would simply be multiplied by 1 and nothing would happen to it!*”.

different faces detection

/main tools

1. cv2

2. dlib



```
dlib.face_recognition_model_v1.compute_face_descriptor(frame, face_pose, 1)
```

128 face measurements

128 face measurements

```
np.linalg.norm(known_faces - face, axis=1) <= TOLERANCE
```

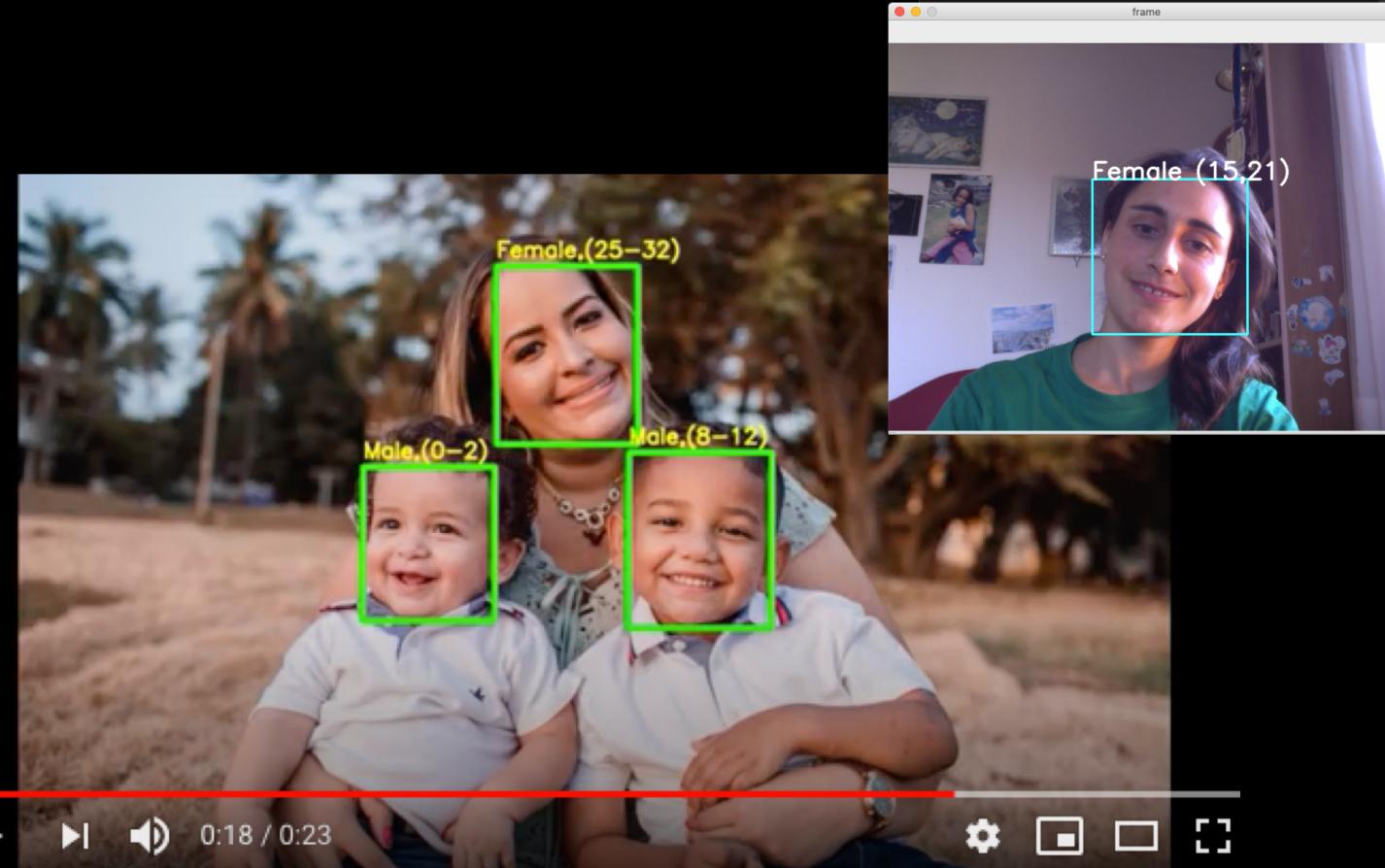
main development

/main tools

1. cv2 - webcam connection
2. pafy module – youtube video connection

/main operations:

1. loading of the models
2. call to video detector object
3. faces detection for each frame
4. show of the frame
5. age and gender prediction
6. SQL database registration of all the new faces id, age and gender



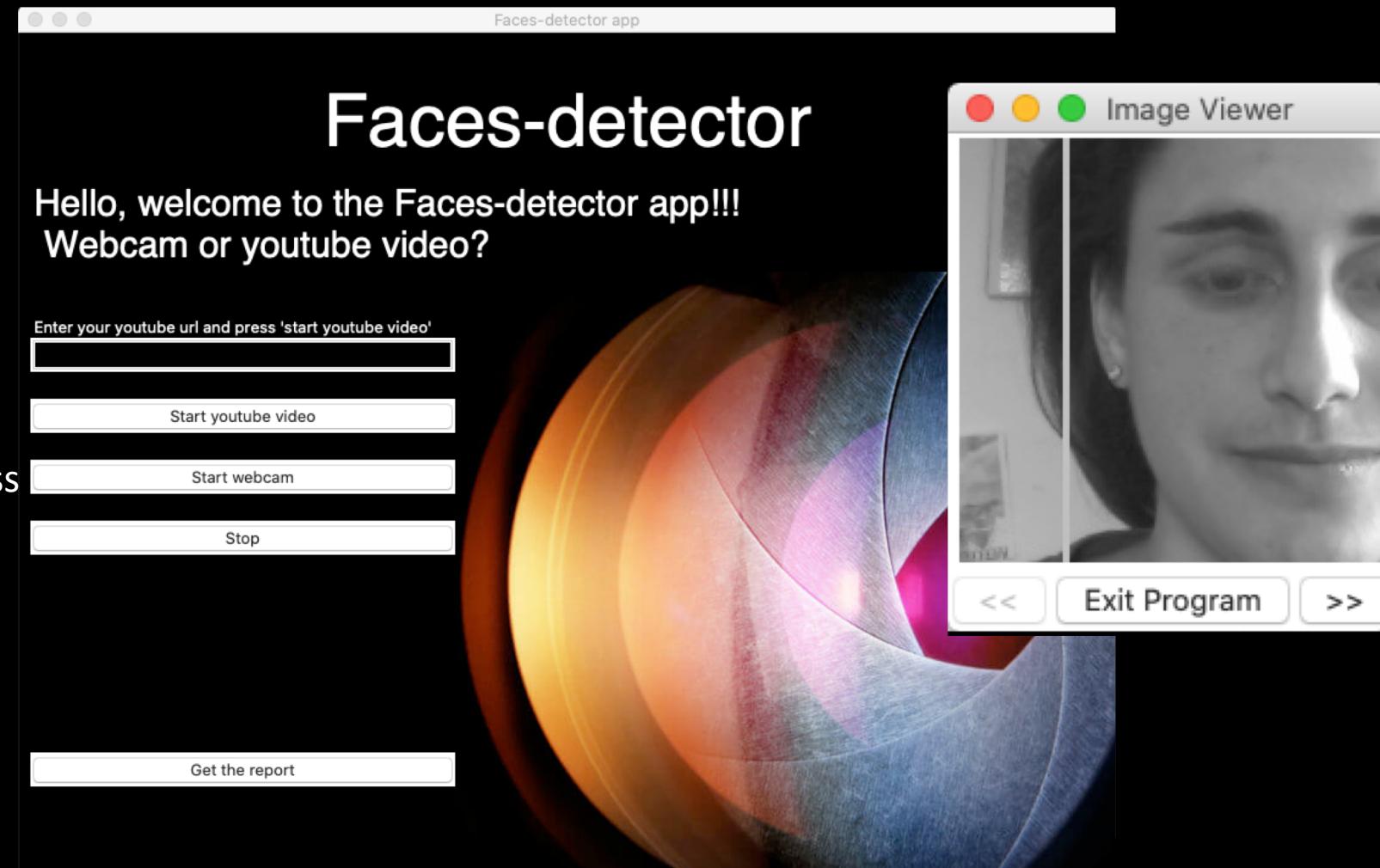
GUI development

/main tools

1. tkinter+PIL – graphic interface
2. pafy module – youtube video connection

/workflow

- 1a. Insert a youtube url and press “Start youtube video”
- 1b. Press “Start webcam”
2. Connection
3. Get the report



Thank you-Demo time!

Silvia Serafini

serafinisilvi@gmail.com

www.linkedin.com/in/silvia-serafini-madrid