Leyla Abdullayeva - 1904010038

Age & Gender Detection - Python

Final Product

Information About The Project

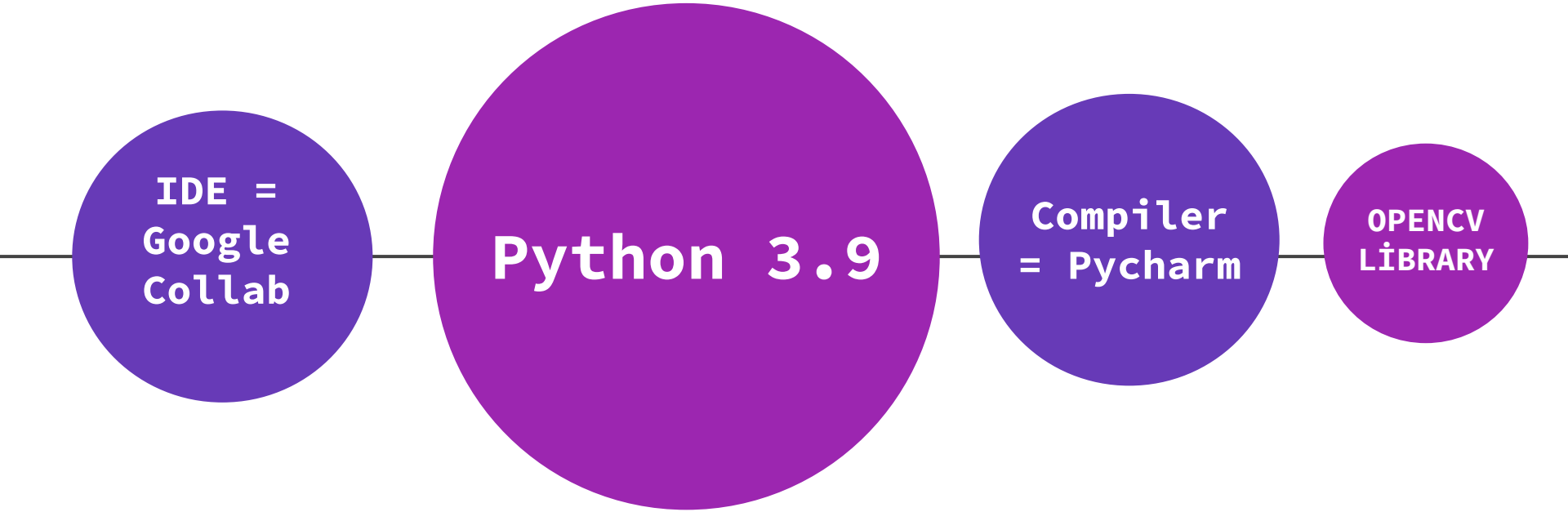
The project is written in Python.(OpenCv)

(It could be also done by CNN).

What is the aim of my project?

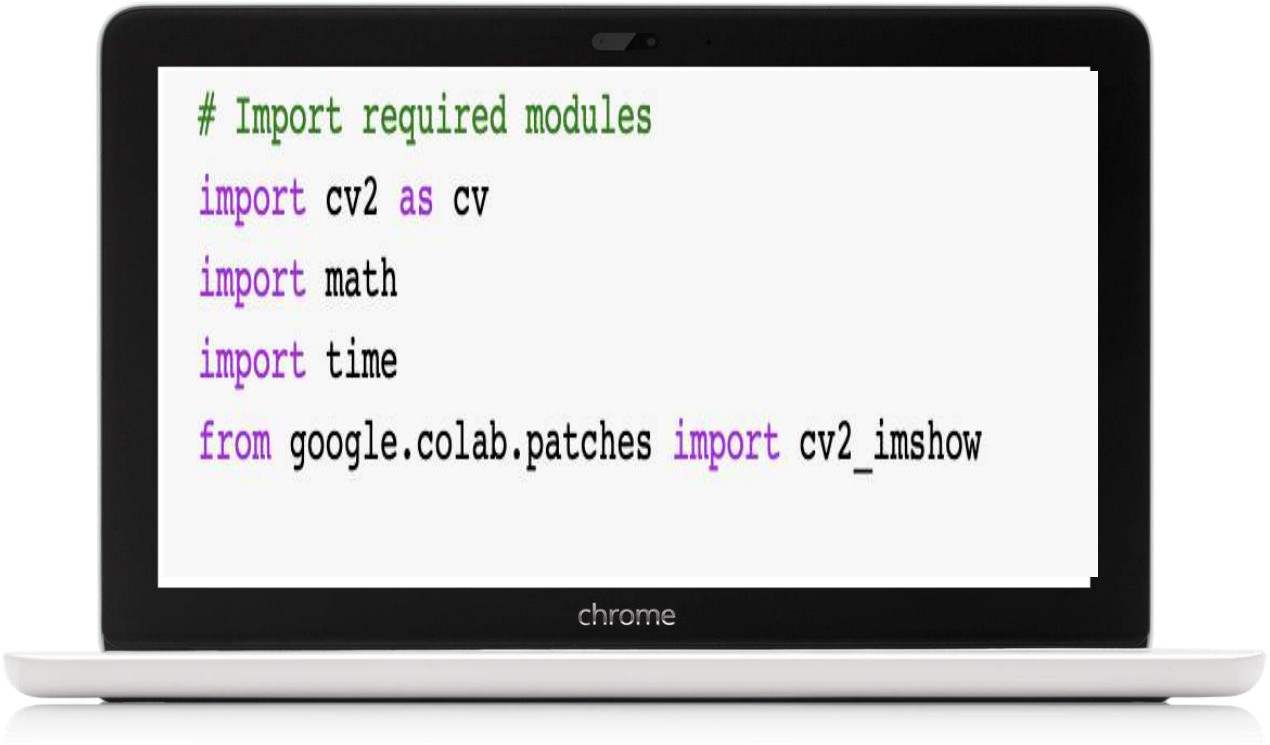
In this Python Project, we will utilize Deep Learning to precisely recognize the gender and age of an individual from a single image of face.

What development tools and software languages did I use for my project?



What I used in my project?

- PYTHON 3.9
- GOOGLE COLLAB
- OPENCV LİBRARY
- MATH & TİME (OPENCV)



```
# Downloading pretrained data and unzipping it
!gdown https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSb
# https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSbIS
!unzip modelNweight.zip
```

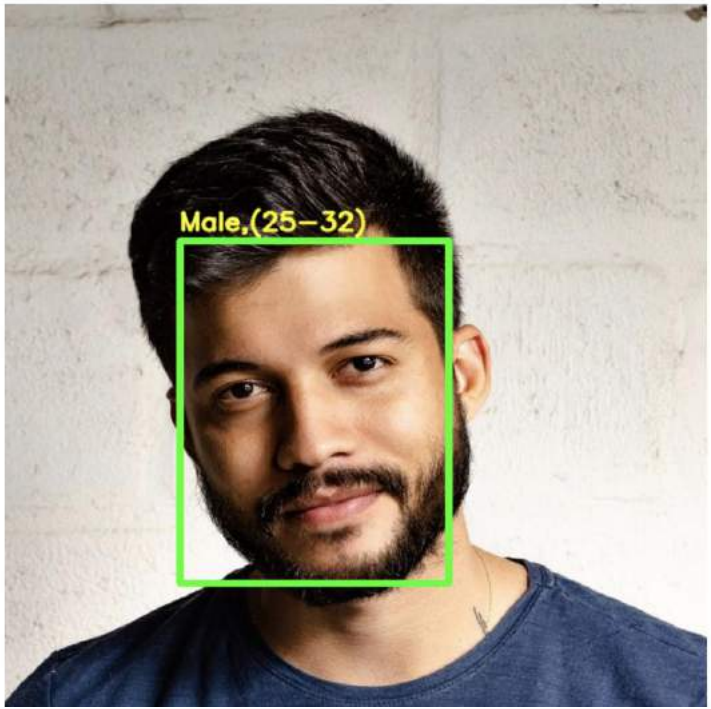
```
Downloading...
From: https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSb
To: /content/age_and_gender_detection/age_and_gender_detection/modell
86.2MB [00:00, 206MB/s]
Archive:  modelNweight.zip
   creating: modelNweight/
  inflating: modelNweight/age_deploy.prototxt
  inflating: modelNweight/age_net.caffemodel
  inflating: modelNweight/gender_deploy.prototxt
  inflating: modelNweight/gender_net.caffemodel
  inflating: modelNweight/opencv_face_detector.pbtxt
  inflating: modelNweight/opencv_face_detector_uint8.pb
```

Implementing pretrained datas to Age & Gender Detection project

In this function,I'm downloading my pretrained data from drive link, then unzipping it.
By doing this, we had implement our data sets into the project.
İt contains gender (gender_deploy.prototxt) & age (age_deploy) caffemodels and txt files.
Then, datasets have implemented to our project, after running the code, it will give us the final outputs.

First Prototype Outputs:

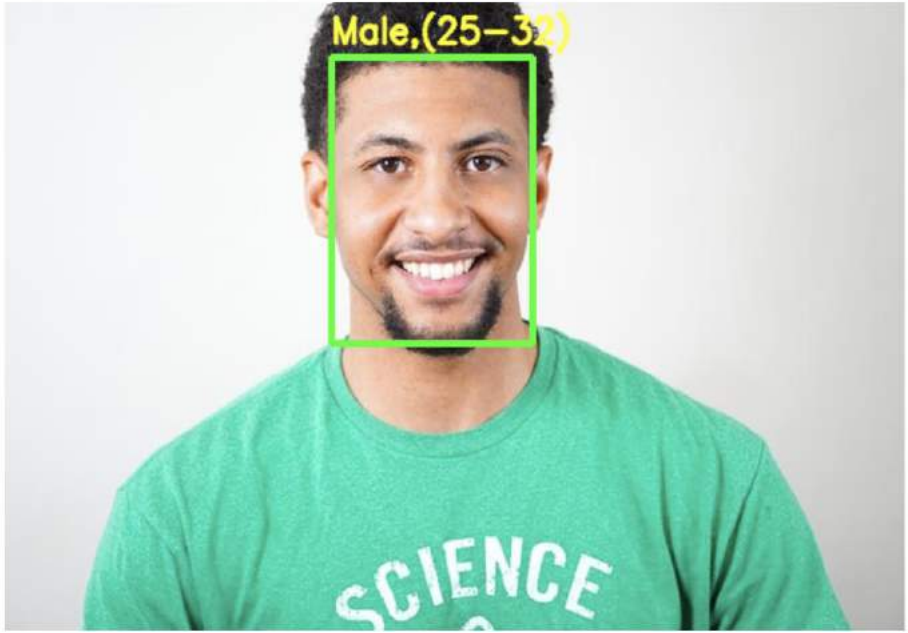
```
input = cv.imread("image.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
```



1st output

As you can see this is first output of this project.

```
input = cv.imread("image1.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
```



2nd output

The 2nd output is also defined male from dataset. (random).And we can see the age prediction also.

Second Prototype Outputs:

```
def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      // Resize the output to fit the video element.
      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.
      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
  display(js)
  data = eval_js('takePhoto({})'.format(quality))
  binary = b64decode(data.split(',')[1])
  with open(filename, 'wb') as f:
    f.write(binary)
  return filename
```

In this part, we are inserting new function, which is for taking photo in order to save the data in my Project.
It's javascript code & python extension.

```
image_file = take_photo()
```

With this function it allows us to take photo, and then it will be saving automatically into Project.

```
#image = cv2.imread(image_file, cv2.IMREAD_UNCHANGED)
image = cv2.imread(image_file)

# resize it to have a maximum width of 400 pixels
image = imutils.resize(image, width=400)
(h, w) = image.shape[:2]
print(w,h)
cv2_imshow(image)
```

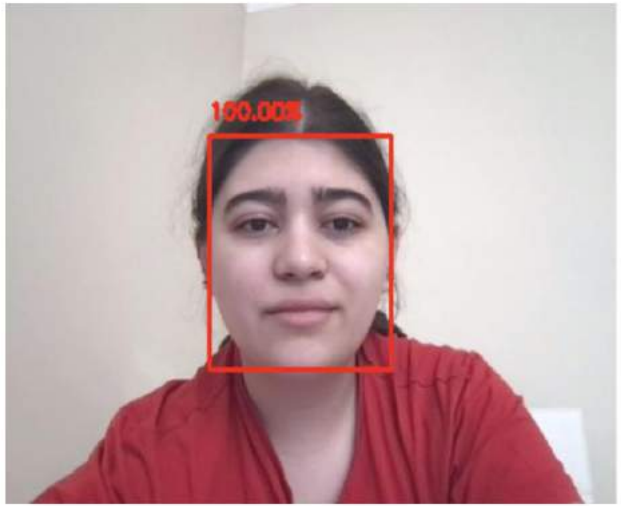
```
400 300
```

In this first example we'll learn how to apply face detection with OpenCV to single input images.
In the next section we'll learn how to modify this code and apply face detection with OpenCV to videos, video streams, and webcams.

```
# function to convert the JavaScript object into an OpenCV image
def js_to_image(js_reply):
  """
  Params:
          js_reply: JavaScript object containing image from webcam
  Returns:
          img: OpenCV BGR image
  """
  # decode base64 image
  image_bytes = b64decode(js_reply.split(',')[1])
  # convert bytes to numpy array
  jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
  # decode numpy array into OpenCV BGR image
  img = cv2.imdecode(jpg_as_np, flags=1)

  return img

# function to convert OpenCV Rectangle bounding box image into base64 byte string to be overlayed on video stream
def bbox_to_bytes(bbox_array):
  """
  Params:
          bbox_array: Numpy array (pixels) containing rectangle to overlay on video stream.
  Returns:
        bytes: Base64 image byte string
  """
  # convert array into PIL image
  bbox_PIL = PIL.Image.fromarray(bbox_array, 'RGBA')
  iobuf = io.BytesIO()
  # format bbox into png for return
  bbox_PIL.save(iobuf, format='png')
  # format return string
  bbox_bytes = '{}'.format((str(b64encode(iobuf.getvalue()), 'utf-8')))

  return bbox_bytes
```

We have to insert the real time object detection into a Project. This function helps us to convert the JavaScript object into an OpenCv image.

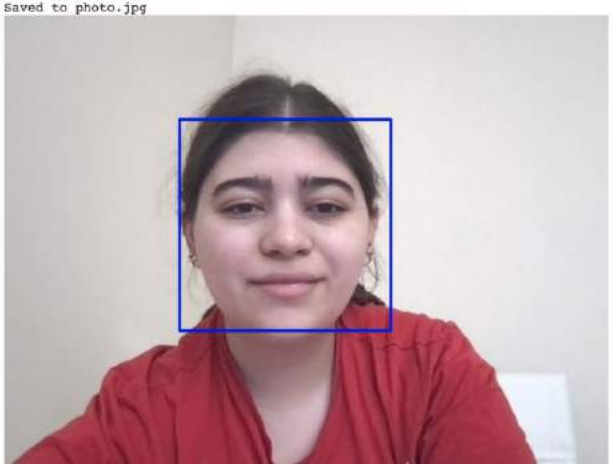
```
cv2_imshow(image)
```



After running the code, we can see the final result. As you can see, our image has saved and object detection feature works 100%.

```
try:
  filename = take_photo('photo.jpg')
  print('Saved to {}'.format(filename))

  # Show the image which was just taken.
  display(Image(filename))
except Exception as err:
  # Errors will be thrown if the user does not have a webcam or if they do not
  # grant the page permission to access it.
  print(str(err))
```

```
(480, 640)
Saved to photo.jpg
```

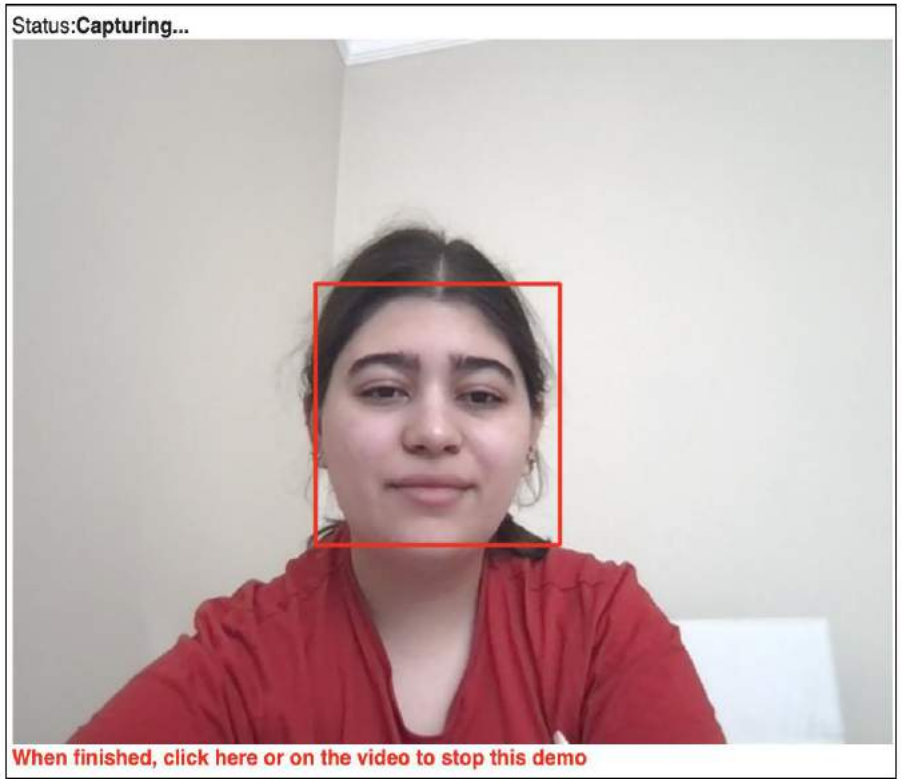


First part of video streaming is again using our webcam, taking and saving our photo into Project. We also can see dimensions of detected area.

```
# start streaming video from webcam
video_stream()
# label for video
label_html = 'Capturing...'
# initialze bounding box to empty
bbox = ''
count = 0
while True:
    js_reply = video_frame(label_html, bbox)
    if not js_reply:
        break

    # convert JS response to OpenCV Image
    img = js_to_image(js_reply["img"])

    # create transparent overlay for bounding box
    bbox_array = np.zeros([480,640,4], dtype=np.uint8)

    # grayscale image for face detection
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

    # get face region coordinates
    faces = face_cascade.detectMultiScale(gray)
    # get face bounding box for overlay
    for (x,y,w,h) in faces:
      bbox_array = cv2.rectangle(bbox_array,(x,y),(x+w,y+h),(255,0,0),2)

    bbox_array[:,:,3] = (bbox_array.max(axis = 2) > 0 ).astype(int) * 255
    # convert overlay of bbox into bytes
    bbox_bytes = bbox_to_bytes(bbox_array)
    # update bbox so next frame gets new overlay
    bbox = bbox_bytes
```

Finally, we achieved to make object detection. After this, we gotta make video streaming video from webcam.

```
bbox_array[:,:,3] = (bbox_array.max(axis = 2) > 0 ).astype(int) * 255
# convert overlay of bbox into bytes
bbox_bytes = bbox_to_bytes(bbox_array)
# update bbox so next frame gets new overlay
bbox = bbox_bytes
```



Here, in this part, program allows us to print demo video after detecting our face and reaching to our webcam.
Then this demo video will be automatically saved between Project files.

Difficulties that I've had:

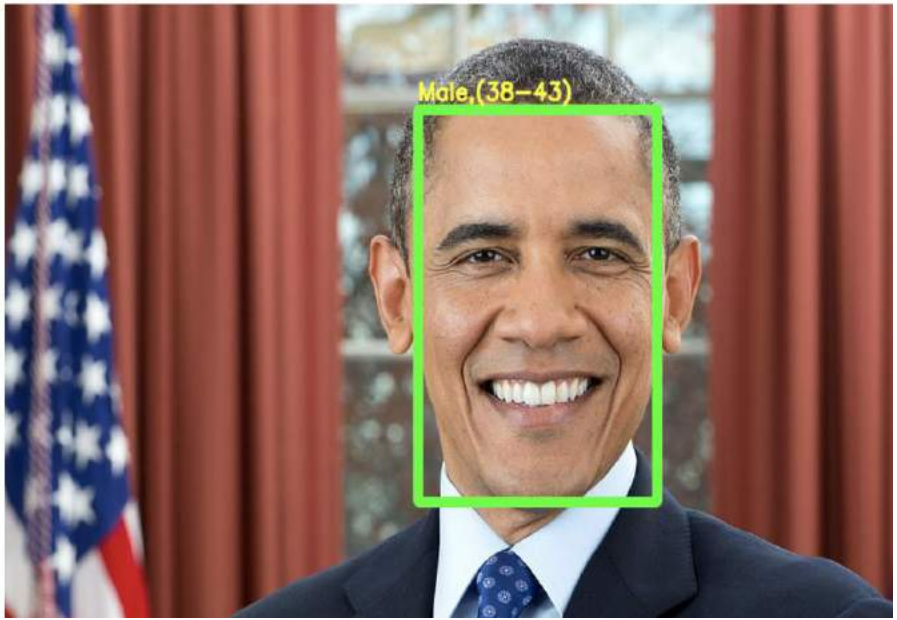
Unfortunately, I'm not able to insert my own data.

I've solved the opening webcam part, but it's not detecting my age & gender.
It only allow to detect my face.

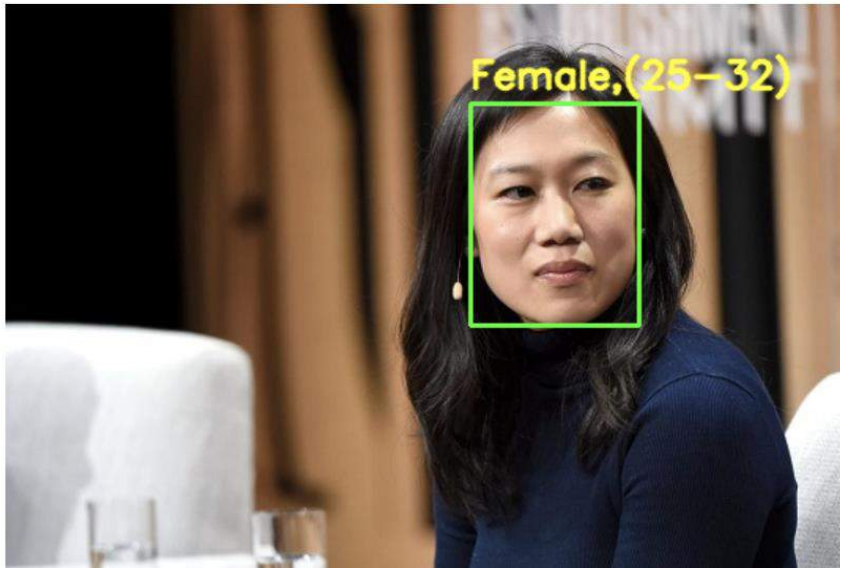
Final Product Outputs:

Final Products outputs:

```
input = cv.imread("6.jpeg")
output = age_gender_detector(input)
cv2_imshow(output)
```

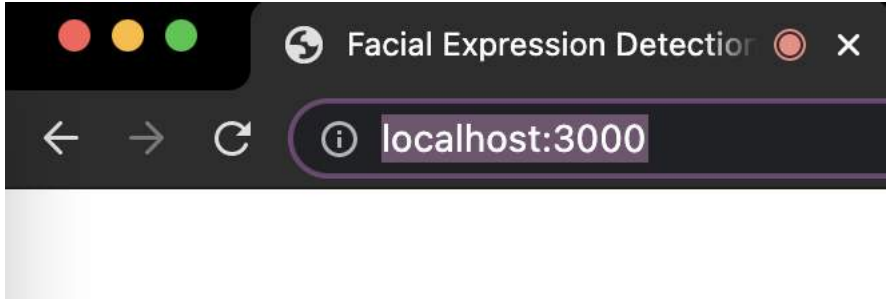



```
input = cv.imread("8.jpeg")
output = age_gender_detector(input)
cv2_imshow(output)
```

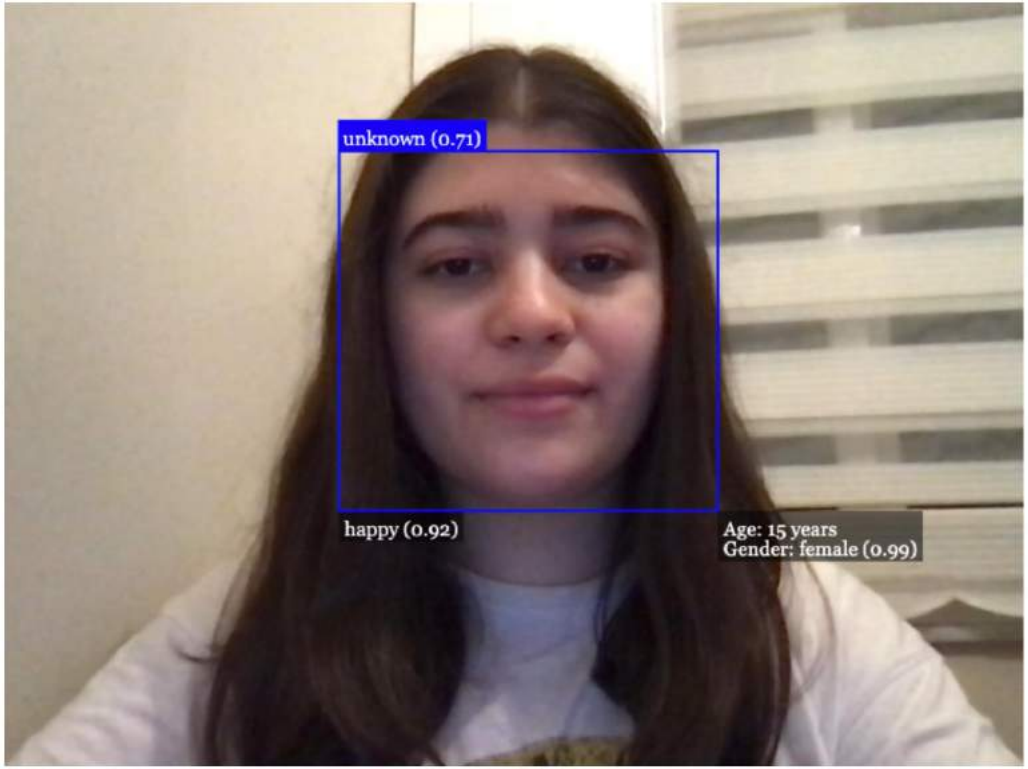


Realtime web application

- I'll be using OpenCv's harcascade to run face detection on the images and video streams.
-
- So the way we are going to be connecting our machine's webcam is through some javascript code and it is going to da the connection your local webcam



After typing localhost:3000 to a browser, we can run the web application and the result



Here, you can see the live demo video of my Project.

PREREQUİSİTES

You'll need to install OpenCV (cv2) to be able to run this project. You can do this with pip-

```
pip install opencv-python
```

Other packages you'll be needing are math and argparse, but those come as part of the standard Python library.

Then we have to insert our pre trained dataset which is I found from internet...

https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSbIs

https://colab.research.google.com/drive/1QnC7lV7oVFk5OZCm75fqbLAfD9qBy9bw?usp=sharing#scrollTo=1nkSnkbkk4cC

Resources:

---

https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

https://sefiks.com/2020/09/07/age-and-gender-prediction-with-deep-learning-in-opencv/

https://www.youtube.com/channel/UCSY7giAIObL7RkR6r-numiQ/about

https://learnopencv.com/

THANK YOU FOR LİSTENİNG!