



Detecting Faces and Analyzing Them with Computer Vision

Project focuses on detecting faces and analyzing them to estimate age, emotion, and gender. Let's get into details



by Müdafer Kaymak

Real Time Face Detection

After opening the camera, we get instant frame and store it into a variable

Load pre-trained Haar Cascade classifier called haarcascade_frontalface_default.xml

Perform the face detection

```
face =  
face_classifier.detectMultiScale(  
gray_image, scaleFactor=1.1,  
minNeighbors=5, minSize=(40, 40))
```

Draw a rectangle around the face

```
for (x, y, w, h) in face:  
cv2.rectangle(img, (x, y), (x + w, y +  
h), (0, 255, 0), 4)
```

Pros and Cons of Face Detection Technology

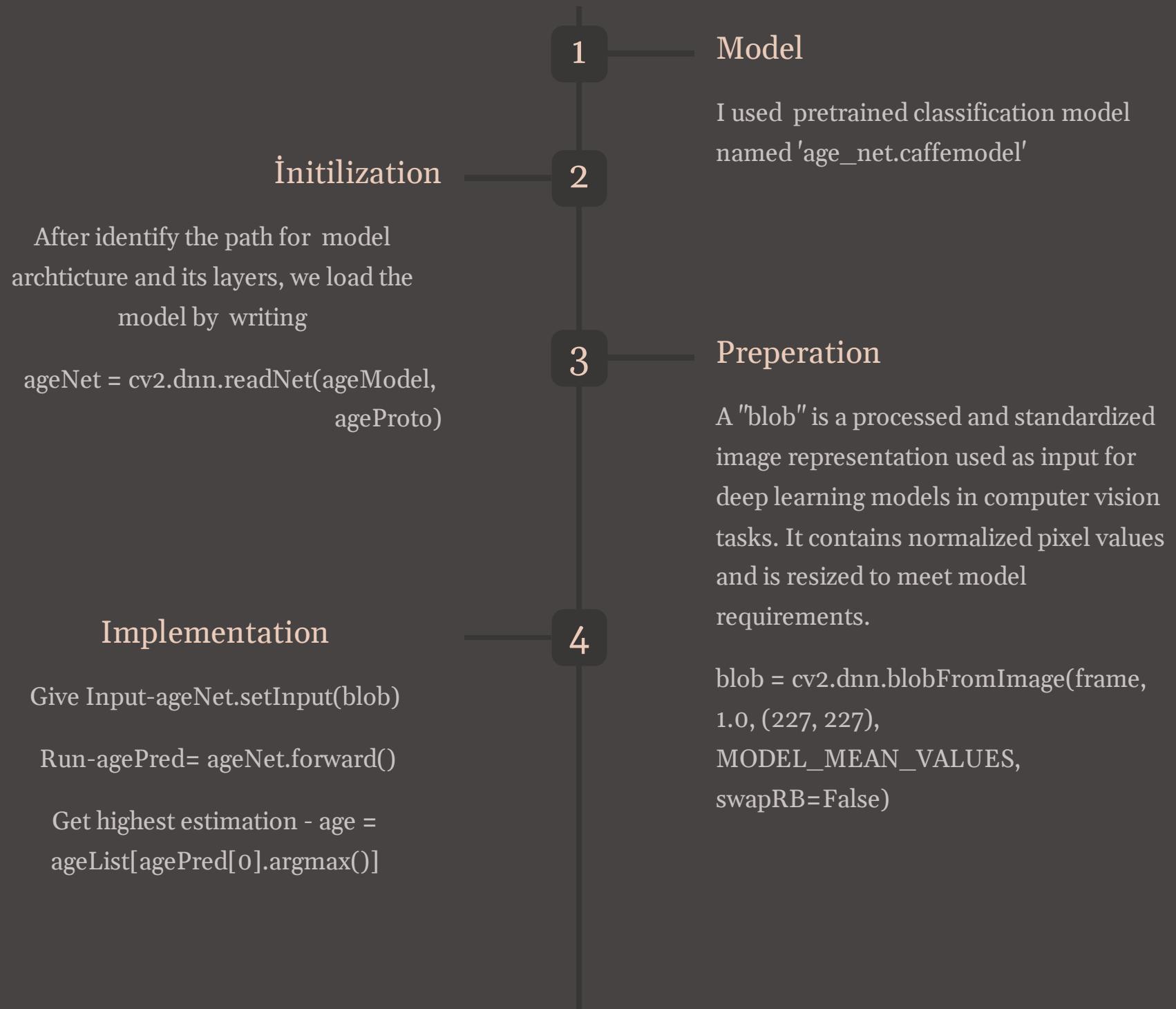
1 Pros:

Face detection technology has many useful applications, including security monitoring, photo organization, and biometric identification.

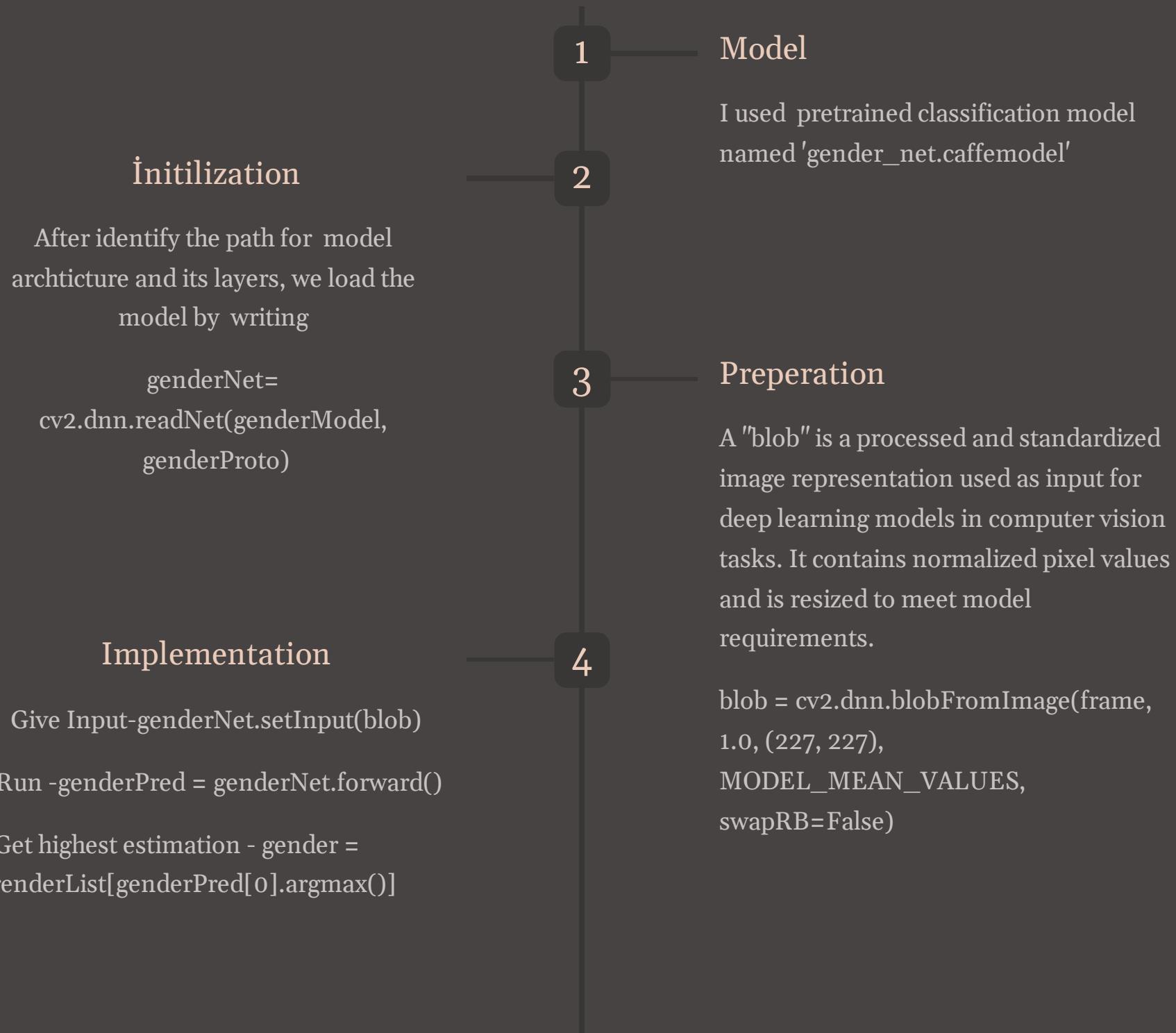
2 Cons:

However, the technology also raises concerns about privacy and potential misuse, as well as issues with accuracy and bias in certain situations.

Age Estimation Model



Gender Estimation Model



Emotion Estimation Model and Training

Initialization

To initialize the model we use keras from tensorflow library

```
model =  
tf.keras.applications.MobileNetV2()
```

Implementation

-Compile the model by giving the loss function, Optimizer and metrics to learn

how correct the model is during the training - new_model.compile(loss = "sparse_categorical_crossentropy",
optimizer = "adam", metrics =
["accuracy"])

-To traing the model with specified parameters

```
new_model.fit(X,y, epochs=10,  
batch_size = 8)
```

-To save the model

```
new_model.save('onbes_epoch.h5')
```

1

Model

I used a pretrained classification model named mobileNetV2. It is a convolutional neural network architecture that performs well on mobile devices.

2

Preperation

I have used FER 2013 dataset which includes nearly 26k labeled image but since my Ram ccapacity is not enough to train all of them I reduced the number of image to nearly 12k. I categorized emotions by giving their name numbers and after that I created the training set. After I extracted the last layer of the model I added 3 more layer which have 128,64,7 neurons(2 relu 1 softmax).

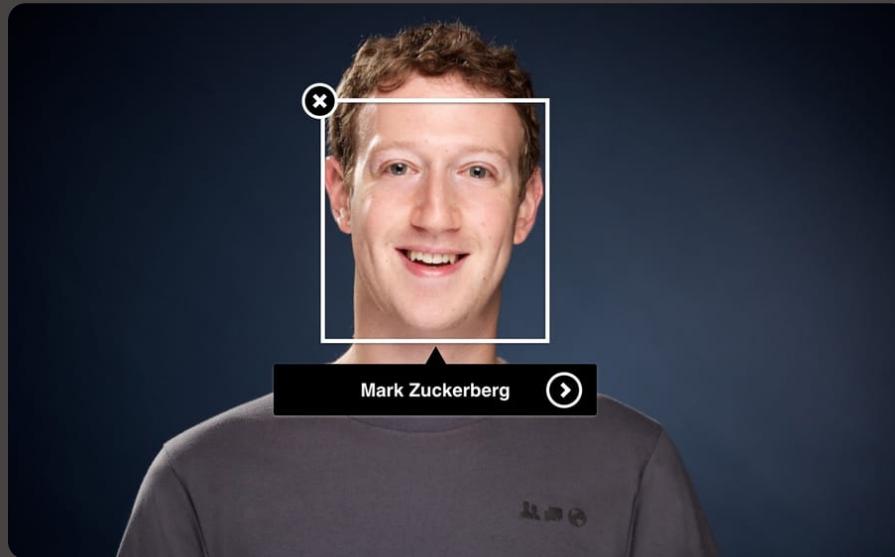
3

4

Testing

Facial Expression	Emotion	Accuracy rate
Wide eyes, raised eyebrows, and an open mouth.	Surprise	%51.74
Smiling mouth and crinkled eyes.	Happiness	%90.70
Furrowed brow, downturned mouth, and squinted eyes.	Sadness	%20.72
Tightly sealed lips, raised chin, and furrowed brow.	Anger	%80.04
Fear face emotion: Wide eyes, raised eyebrows, open mouth; indicates perceived threat or danger.	Fear	%45.14
OVERALL ACCURACY	%49.39	

Alternative Models and Techniques



DeepFace

DeepFace is a facial recognition library developed by Facebook AI Research (FAIR) for accurate face recognition using deep learning techniques.

DeepFace Implementation

1 Preparation

Just import deepface library

2 Implementation

```
result =  
DeepFace.analyze(im  
g, actions =  
['emotion'],  
enforce_detection =  
False)
```

3 Explanation

It can estimate age, race, emotion also it can detect the face and compare with other faces if they are same or not.

Application of Face Analyze

This project can be utilized in security cameras to log and record the entry time of employees entering the company, enabling employee monitoring. This can prevent any inaccuracies during the entry time of employees, ensuring reliability and accuracy in the system. It can also be used to assess the average emotional state of office employees.





Limitations of this project

- Since MobileNetV2 designed for mobile devices it would be better to use latest YOLO library to classify these features.
- Due to the positioning of security cameras, many of them being fixed on the upper walls, there might be errors in face detection. It can detect a person for many times as a different person. It might not detect at all sometimes. Especially it would really hard to detect emotions through this cameras. But this problem can be fixed with better models