MARIELA QUINTANAR DE LA MORA

JUNE DE 2023

USING PYHON, TENSORFLOW AND FACENET

FACE RECOGNITION

OBJECTIVE

A facial recognition machine learning project offers numerous benefits and can yield impactful results across various domains. By leveraging advanced algorithms and artificial intelligence, this cutting-edge technology can revolutionize how we interact with the world and streamline numerous processes.

Some of the key benefits and results that can be obtained from a facial recognition machine learning project are:

Enhanced Security and Safety: Facial recognition can significantly strengthen security measures. Access control systems in sensitive areas like airports, government buildings, and data centers can become more robust, thwarting unauthorized access attempts.

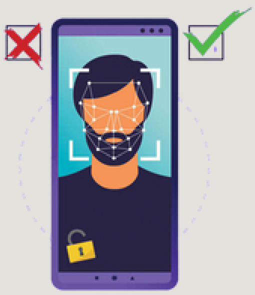
Improved User Experience: In the consumer realm, facial recognition offers a seamless and personalized experience. Smartphones, tablets, and computers can use facial recognition as a convenient method for unlocking devices, accessing apps, and making secure online transactions.

Time and Cost Savings: Automating various identification and verification processes through facial recognition can lead to substantial time and cost savings in various industries. Companies can allocate resources more efficiently and focus on core activities.

Research and Social Impact: Facial recognition projects can facilitate research in fields such as psychology, sociology, and human-computer interaction. Furthermore, it can have a positive social impact by assisting in locating missing persons, reuniting families, and aiding humanitarian efforts.

While the potential benefits of facial recognition machine learning projects are vast, it is crucial to implement and regulate these technologies responsibly. With responsible implementation, facial recognition can unlock new possibilities, making our lives safer, more efficient, and personalized.

FACIAL RECOGNITION WITH MACHINE LEARNING



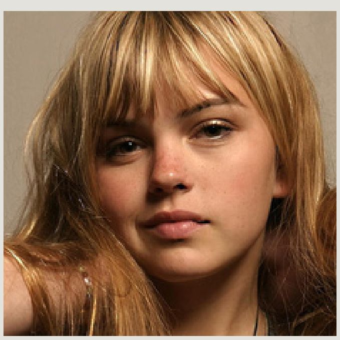
There are two applications of facial recognition:

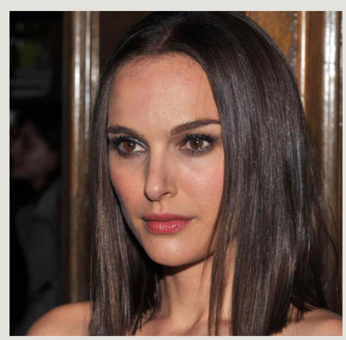
Verification:
An unknown face is compared with a reference image and the system allows to confirm or reject the identity.

Identifcation:
An unknown face is compared with a database of images containing previously stored faces, in order to determine the identity corresponding to the input face.

HOW DOES FACIAL IDENTIFICATION WORK?

Suppose we have four faces and we want to differentiate one from the other. Taking into account only the color of her hair and her eyes



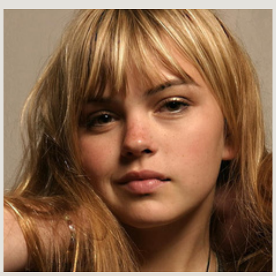


STEP: 1

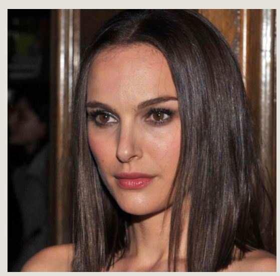
MAKE A COMPACT REPRESENTATION OF THE FEATURES OF THE FACES

LIGHT EYES
LIGHT HAIR

LIGHT EYES
DARK HAIR



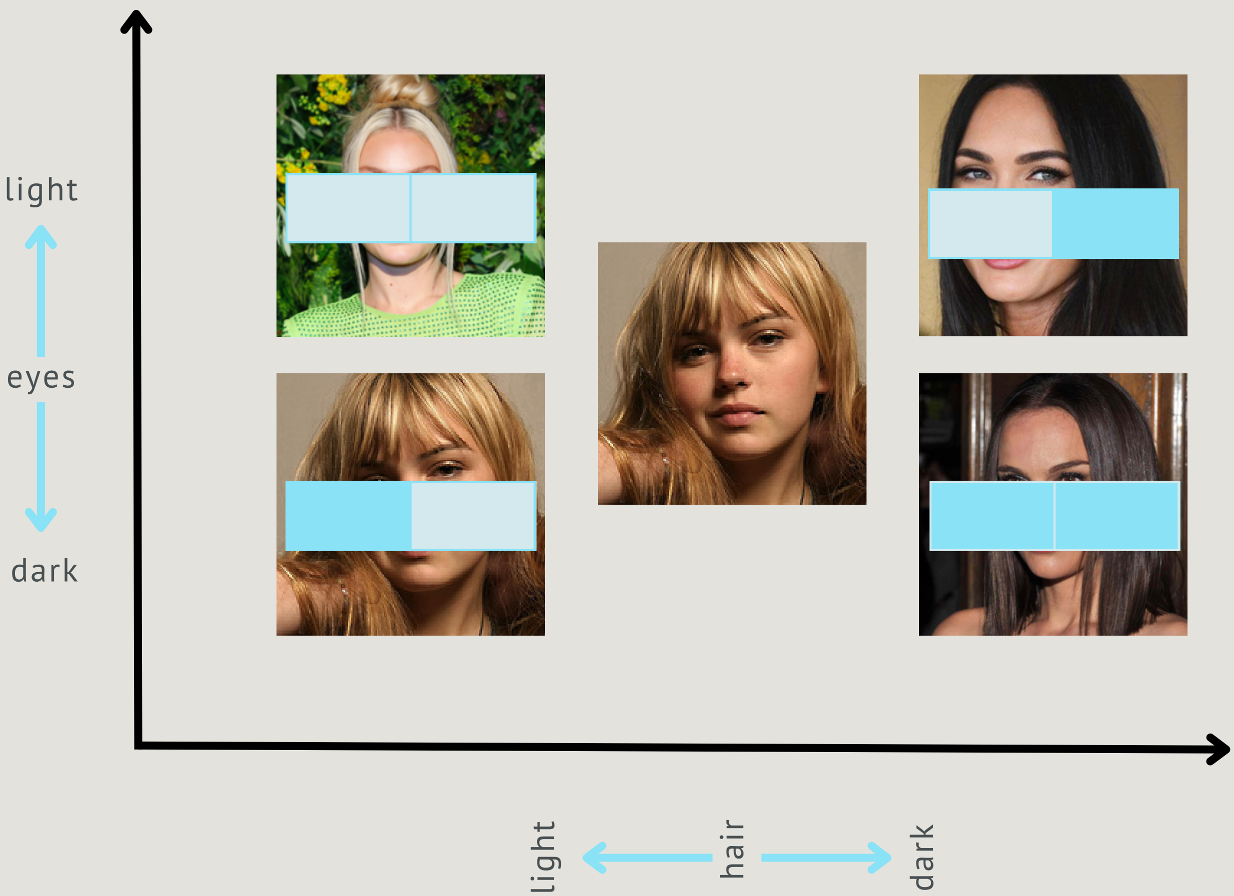
DARK EYES
LIGHT HAIR



DARK EYES
DARK HAIR

STEP: 2

WE LOCATE EACH COMPACT REPRESENTATION IN A COORDINATE SYSTEM
WE INTRODUCE AN IMAGE OF AN UNKNOWN FACE THAT WE WANT TO IDENTIFY

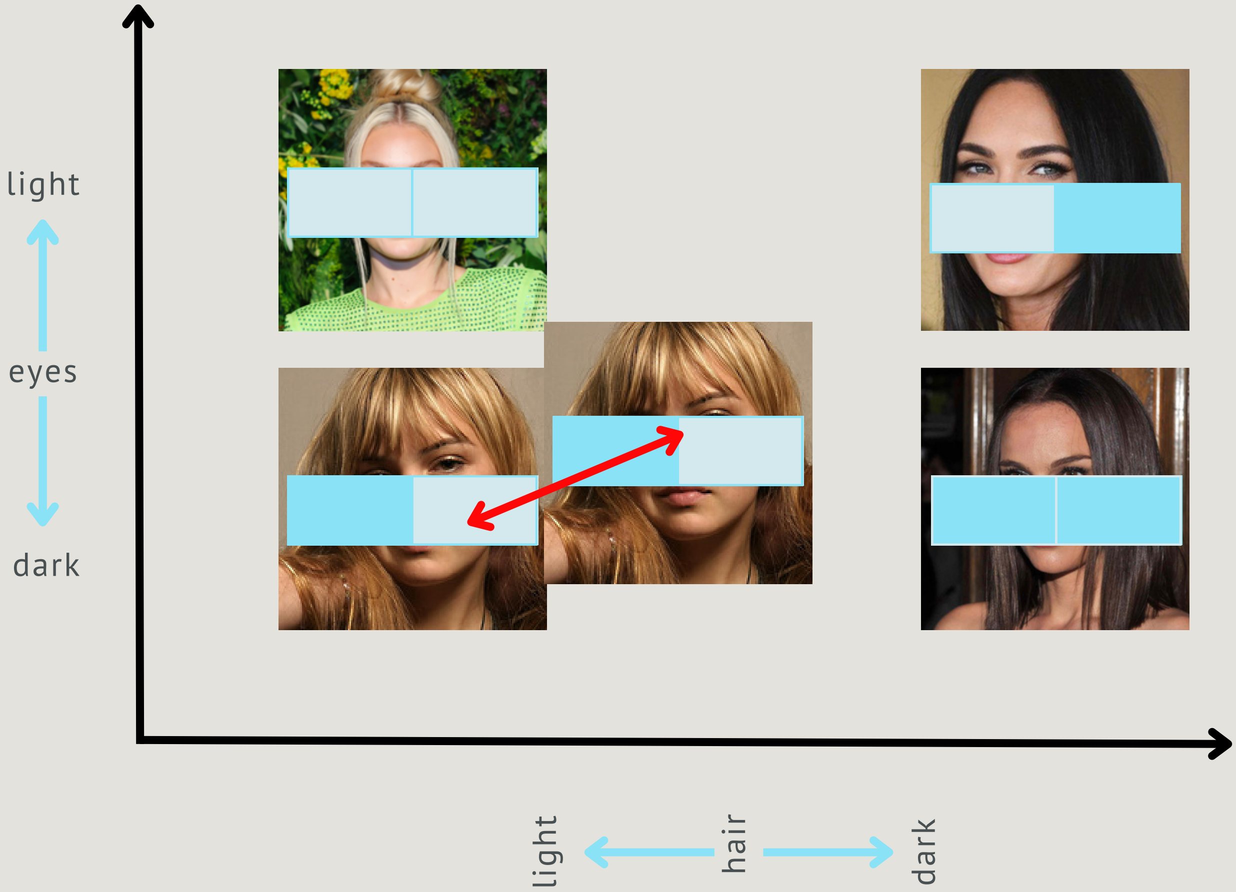


STEP: 3

WE OBTAIN THE CHARACTERISTICS OF THIS NEW FACE.
WE THEN LOCATE THIS COMPACT REPRESENTATION IN THE SAME COORDINATE SYSTEM AS THE REFERENCE IMAGES.

STEP: 4

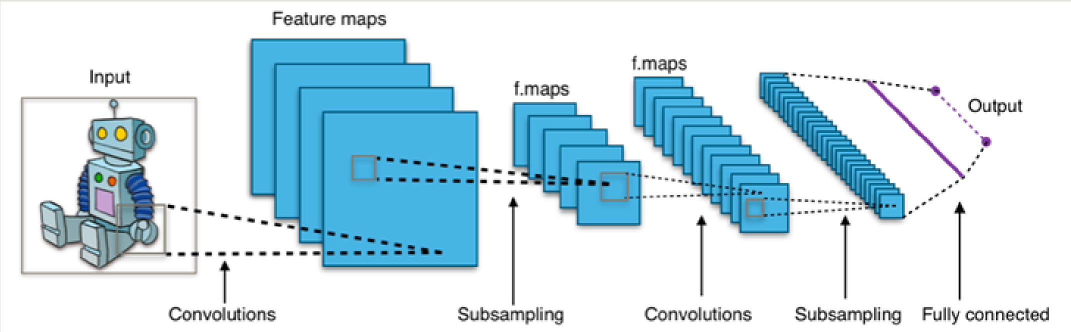
WE IDENTIFY WHICH OF THE FOUR DIFFERENCE GROUPINGS THE REPRESENTATION OF THE NEW FACE IS CLOSEST TO.
THE DISTANCE BETWEEN THE REPRESENTATION OBTAINED OF THE UNKNOWN FACE AND THOSE OF THE REFERENCE IMAGES IS CALCULATED.
THE CLOSEST DISTANCE INDICATES WHICH PERSON THE UNKNOWN FACE CORRESPONDS TO



COMO SE LORGRA HACER EL RECONOCIMIENTO FACIAL CON IA?

CONVOLUTIONAL NETWORK

How does the convolutional network work?



Convulsion layers: they are in charge of extracting relevant characteristics from the images.

Fully-connected layers: neural networks that are responsible for obtaining a compact representation of the characteristics obtained

FACENET

Artificial Intelligence developed by Google. It is one of the most accurate facial recognition systems that exist today.
In this system there is an input of a color image and its output will be a vector of 128 elements. These elements (embeddings) are the compact representation of the entered face.

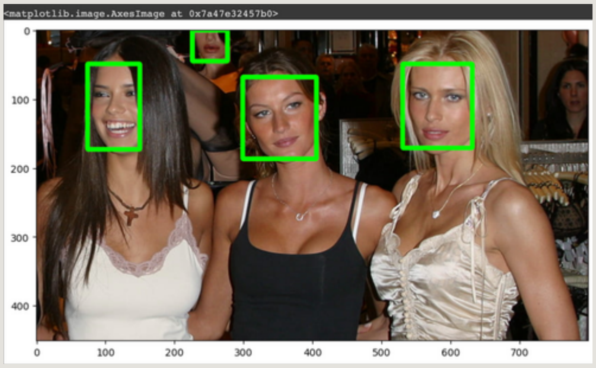
This AI is trained with the function called triplet loss, which helps the AI to be more accurate.
In this function, triples are entered, that is, a reference image, a positive image (a face that belongs to the same reference image) and a negative image (face different from that of the reference image).

With this training, the network will progressively learn to place the embeddings of similar faces closer each time in the space of 128 dimensions and at the same time the embeddings of different faces will be further and further away from each other.

By using more than 500 million images for this training, FaceNet achieves 99.63% accuracy in face verification. This means that out of every 10,000 faces it erroneously identifies 37.

FACE DETECTION

BOUNDING BOXES



To achieve recognition, it is necessary to isolate the portions of the image that contain faces.
Face detection consists of locating one or more faces in an image. This location consists of finding the coordinates of the faces within that image. These "little boxes" that delimit each face are known as bounding boxes.

Covolutional networks are trained to classify the content of the scene by determining whether or not it contains a human face.
If the image contains a face, the network learns to generate the starting coordinates of the bounding box as well as its width and height.

MOBILE NET

Convolutional network capable of detecting objects, which can be trained for face detection.
The advantage of this network is the speed of its processing, which is achieved by making the seizures more efficient.

How does mobileNet work?
The input image is divided into its 3 planes and each one is convolved with a 3x3x1 filter.
In this process only 432 multiplications are required.
Then the resulting volume is convolved with four 1x1x3 filters.

FACE RECOGNITION WITH PYTHON

IMPORTED LIBRARIES

```
import tensorflow as tf # mobile net system
from keras.models import load_model # to load facenet
import cv2 # allows us to read images from the hard drive and store them
import numpy as np # to make calculations on faces that are detected
import matplotlib.pyplot as plt # to draw the images that we obtained
import os # to be able to read the data on the hard drive
!pip install keras_facenet
from keras_facenet import FaceNet
from tensorflow.keras.models import load_model
```

FOLDERS

```
# Folders to store the known and unknown images and the results
DIR_KNOWNS = 'knowns'
DIR_UNKNOWNS = 'unknowns'
DIR_RESULTS = 'results'
```

READ MOBILENET

```
# read mobilenet_graph.pb
with tf.io.gfile.GFile('frozen_inference_graph_face.pb','rb') as f:
    graph_def = tf.compat.v1.GraphDef()
    graph_def.ParseFromString(f.read())

with tf.Graph().as_default() as mobilenet:
    tf.import_graph_def(graph_def,name='')
```

```
[ ] print(mobilenet)

<tensorflow.python.framework.ops.Graph object at 0x7a4895219210>
```

FACE RECOGNITION WITH PYTHON

LOAD IMAGES

```
[35] # DIR -> directorio
     # NAME -> nombre del archivo
     def load_image(DIR, NAME):
         return cv2.cvtColor(cv2.imread(f'{DIR}/{name}'), cv2.COLOR_BGR2RGB)
```

DETECT FACES FUNCTION

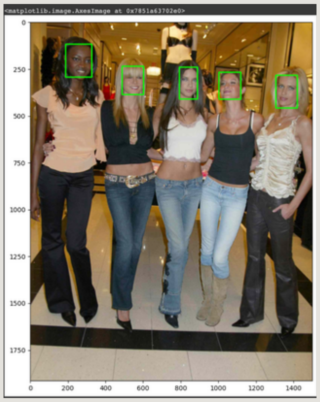
```
[41] # bounding boxes from 0.7 up
     def detect_faces(image, score_threshold=0.7):
         global boxes, scores
         (imh, imw) = image.shape[:-1]
         img = np.expand_dims(image,axis=0)

       # at first our image only has 3 dimensions but since
       # we want to add several images at the same time we must create another dimension

       # initialize mobilenet
       # with 3 containers
       # container 1: we indicate where the input image is stored
       # container 2 and 3: output containers (bounding boxes)

         sess = tf.compat.v1.Session(graph=mobilenet)
         image_tensor = mobilenet.get_tensor_by_name('image_tensor:0')
         boxes = mobilenet.get_tensor_by_name('detection_boxes:0')
         # probabilities
         scores = mobilenet.get_tensor_by_name('detection_scores:0')

         # prediction (detection)
         (boxes, scores) = sess.run([boxes, scores], feed_dict={image_tensor:img})

         # Readjust sizes of boxes and scores (we must remove a dimension)
         boxes = np.squeeze(boxes,axis=0)
         scores = np.squeeze(scores,axis=0)

         # Only bounding boxes that have a face
         idx = np.where(scores>=score_threshold)[0]
```

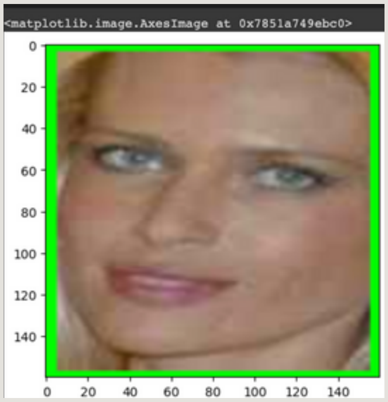
FACE RECOGNITION WITH PYTHON

EXTRACT FACES FUNCTION

```
[79] # extract faces
     def extract_faces(image,bboxes,new_size=(160,160)):
       # Store faces in a list
         cropped_faces = []
         for box in bboxes:
             left, right, top, bottom = box
            # coordinates from top to bottom and from left to right
             face = image[top:bottom,left:right]
             cropped_faces.append(cv2.resize(face,dsize=new_size))
         return cropped_faces
```

```
[68] faces = extract_faces(image,bboxes)
     plt.imshow(faces[2])
```





FACENET AND COMPUTE EMBEDDINGS FUNCTIONS

```
[69] # Facenet
     facenet = FaceNet()
```

```
[70] # Generate 128 embeddings
     def compute_embedding(model,face):
       # tenemos que pasar la imagen a tipo punto flotante
       # porque tenemos que normalizar la entrada
       face = face.astype('float32')
       #detections = model.extract(face,threshold=0.95)

       # nuestra image tiene que tener 4 dimensiones asi que le agregamos una
       face = np.expand_dims(face,axis=0)

       # predicción
       embedding = model.embeddings(face)
       return embedding
```

FACE RECOGNITION WITH PYTHON

CALCULATION OF EMBEDDINGS OF KNOWN FACES

```
[82] # calculate the reference embeddings (known faces)
     known_embeddings = []

     print('Processing known faces...')
     for name in os.listdir(DIR_KNOWNS):
         if name.endswith('.jpg'):
             print(f'   {name}')
             image = load_image(DIR_KNOWNS,name)
             bboxes = detect_faces(image)
             face = extract_faces(image,bboxes)
             known_embeddings.append(compute_embedding(facenet,face[0]))
```

```
Processing familiar faces...
   daniela.jpg
1/1 [==============================] - 0s 173ms/step
   daniela3.jpg
1/1 [==============================] - 0s 113ms/step
   daniela2.jpg
1/1 [==============================] - 0s 123ms/step
```

```
[75] print(known_embeddings)
```

```
[array([[-0.0326693 , -0.01694929,  0.01387257,  0.0260126 ,  0.02320362,
          0.06294798,  0.06079357, -0.15522939, -0.01436119,  0.05834669,
          0.03622599, -0.05092921, -0.00274189, -0.04103192,  0.01440275,
         -0.06728374,  0.04572108,  0.06936757,  0.08967696,  0.03876172,
         -0.04792026, -0.04653615,  0.06811258,  0.07889079, -0.01900661,
          0.01218933, -0.03188233, -0.00715941, -0.0827156 ,  0.07326867,
          0.01346683,  0.06457892, -0.09815217, -0.02614421, -0.02470938,
         -0.06774891,  0.03124091, -0.02007668, -0.00194665,  0.02764112,
         -0.02391333,  0.01012488, -0.03699211,  0.06477404,  0.04989131,
         -0.01164973,  0.0338331 ,  0.02840735,  0.00772133, -0.01658448,
          0.00959601, -0.01556257, -0.00390857,  0.10872427, -0.00226008,
```

FUNCTION COMPARE FACES

```
# function to compare faces
# calculate the distance between the 128 elements
def compare_faces(embs_ref, emb_desc, umbral=11):
    distancias = []
    for emb_ref in embs_ref:
        distancias.append(np.linalg.norm(emb_ref-emb_desc))
    distancias = np.array(distancias)
    return distancias, list(distancias<=umbral)
```

FUNCTION COMPARE FACES

```
print('Processing unknown images...')
for name in os.listdir(DIR_UNKNOWNS):
    if name.endswith('.jpg'):
        print(f'   {name}')
        image = load_image(DIR_UNKNOWNS,name)
        bboxes = detect_faces(image)
        faces = extract_faces(image,bboxes)

        # For each face calculate embedding
        img_with_boxes = image.copy()
        for face, box in zip(faces,bboxes):
            emb = compute_embedding(facenet,face)

            _, recognition = compare_faces(known_embeddings,emb)

            if any(recognition):
                print('     match!')
                img_with_boxes = draw_box(img_with_boxes,box,(255,0,0))
            else:
                img_with_boxes = draw_box(img_with_boxes,box,(0,255,0))

        cv2.imwrite(f'{DIR_RESULTS}/{name}',cv2.cvtColor(img_with_boxes,cv2.COLOR_RGB2BGR))
print('End!')
```

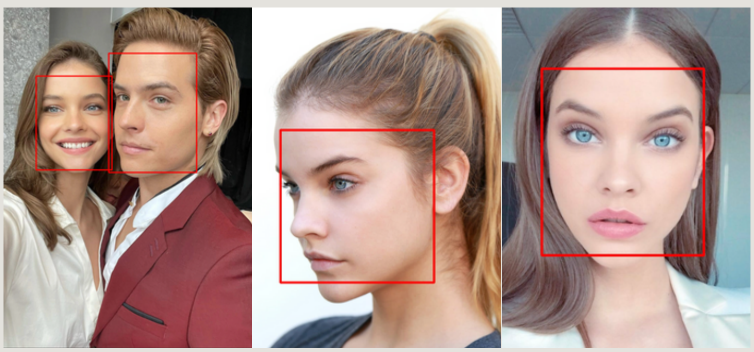
FACE RECOGNITION WITH PYTHON

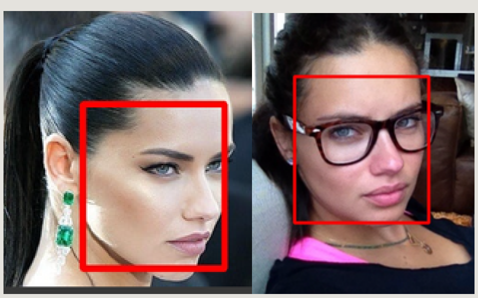
PROCESSING UNKNNOWN IMAGES

```
Processing unknown images...
   a_li.jpg
1/1 [==============================] - 0s 162ms/step
     match!
1/1 [==============================] - 0s 159ms/step
     match!
1/1 [==============================] - 0s 162ms/step
     match!
1/1 [==============================] - 0s 179ms/step
     match!
1/1 [==============================] - 0s 162ms/step
     match!
   barbara3.jpg
1/1 [==============================] - 0s 156ms/step
     match!
1/1 [==============================] - 0s 122ms/step
     match!
   zendaya2.jpg
1/1 [==============================] - 0s 214ms/step
     match!
1/1 [==============================] - 0s 262ms/step
     match!
1/1 [==============================] - 0s 114ms/step
     match!
   adriana2.jpg
1/1 [==============================] - 0s 102ms/step
     match!
   adriana_li.jpg
   barbara.jpg
1/1 [==============================] - 0s 94ms/step
     match!
   zendaya.jpg
1/1 [==============================] - 0s 95ms/step
     match!
1/1 [==============================] - 0s 107ms/step
     match!
   barbara2.jpg
1/1 [==============================] - 0s 162ms/step
     match!
   adriana3.jpg
1/1 [==============================] - 0s 160ms/step
     match!
End!
```

RESULTS

UNIDENTIFIED PEOPLE WITH REFERENCE IMAGE





IDENTIFIED PERSON: DANIELA PEŠTOVÁ

CONCLUSION

In conclusion, there are a number of important factors that make the research and comprehension of facial recognition produced by AI extremely important in the modern world.

First and foremost, facial recognition technology has ingrained itself into many facets of our everyday lives, including social media, law enforcement, healthcare, and security. It is vital for people, legislators, and enterprises to understand this technology's strengths and limitations as it develops. Understanding AI-generated facial recognition equips us to make wise choices and negotiate the moral ramifications of its widespread adoption.

Second, in the digital age, privacy and data protection have become major problems. There are legitimate concerns regarding potential abuse and intrusion into personal lives of facial recognition technologies. By being knowledgeable about the complexities of AI-generated facial recognition, we can actively contribute to the development of rules and regulations that will protect individual rights and prevent the abuse of this potent technology.

Understanding AI-generated facial recognition is also essential for promoting accountability and transparency. The "black box" aspect of some algorithms can be troubling as AI systems get more advanced. We can encourage transparency in AI development and raise people's trust in the technology and its applications by learning more about how these models operate.

Learning about facial recognition is also important. With the help of AI, bias and discrimination in algorithmic decision-making can be found and addressed. Because they are trained on such large datasets, AI systems have the potential to reinforce and exacerbate societal injustices. By researching this technology, we may endeavor to fairly benefit all societal members by making it fairer and more inclusive.

Finally, by staying knowledgeable on AI-generated facial recognition, we can maximize its potential for societal benefits. If used safely and ethically, this technology has the potential to lead to enormous breakthroughs in everything from helping law enforcement solve crimes to improving medical diagnosis and furthering research.

In conclusion, learning about facial recognition produced by AI is important for shaping both its responsible development and deployment as well as for comprehending the impact it has on numerous facets of our life. By remaining educated, we can take advantage of the potential of new technology while ensuring that it complies with our moral principles and upholds individual rights, making the world a safer, more just, and more advanced place.

BIBLIOGRAPHY

¿Qué es el Bounding Box? (n.d.). https://es.locationiq.com/glossary/bounding-box

I can't load model FaceNet it said ValueError: bad marshal data (unknown type code). (n.d.). Stack Overflow. https://stackoverflow.com/questions/67768822/i-cant-load-model-facenet-it-said-valueerror-bad-marshal-data-unknown-type-co

How Shan. (2023, March 26). FaceNet Model Not Working | facenet_keras.h5 not working | bad marshal data [Video]. YouTube. https://www.youtube.com/watch?v=15D7nmLnQcs

Codificando Bits. (2020, June 8). Reconocimiento Facial con Machine Learning [Video]. YouTube. https://www.youtube.com/watch?v=xjCrQ_XEJhI

Codificando Bits. (2020b, June 15). Detección de Rostros con Machine Learning [Video]. YouTube. https://www.youtube.com/watch?v=Y4aLYeLXAbY

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. https://doi.org/10.1109/cvpr.2015.7298682

KeepCoding, R. (2022, November 30). ¿Qué es Matplotlib y cómo funciona? | KeepCoding Bootcamps. KeepCoding Bootcamps. https://keepcoding.io/blog/que-es-matplotlib-y-como-funciona/

Zyprian, F. (2023). CV2 – Master Guide OpenCV made for Python Developers. Konfuzio.
https://konfuzio.com/es/cv2/#:~:text=CV2%20es%20una%20potente%20librer%C3%ADa,y%20el%20filtrado%20de%20im%C3%A1genes.

Red neuronal convolucional MobileNet-v2 - MATLAB mobilenetv2 - MathWorks España. (n.d.). https://es.mathworks.com/help/deeplearning/ref/mobilenetv2.html