

Prototipo para la detección de rostros utilizando el lenguaje de programación Python

Prototype for face detection using the Python programming language

Herrera Ardila, Juan Camilo¹ . Roncancio Peña, Cristian Camilo²³

Citar este documento:

Herrera-Ardila, Juan Camilo; Roncancio-Peña, Cristian Camilo. Tecnología de registro distribuido (DLT): Características y escenarios de aplicación. Revista Technol.Investig.Academia TIA, ISSN: 2344-8288, Volumen 10, Número , pp. 200-210. Bogotá-Colombia.

¹ Ingeniero de sistemas, Universidad Distrital Francisco José de Caldas, juankmiloh@hotmail.com <https://orcid.org/0000-00034804-3387> . Bogotá-Colombia

² Ingeniero de sistemas – Fundación Universitaria Konrad Lorenz, cristancho93@gmail.com. <https://orcid.org/0000-0002-8958->

³ . Bogotá-Colombia

Resumen

El objetivo principal de este artículo es la implementación en Python de un prototipo de un sistema de reconocimiento facial usando *machine learning*, para lograrlo se deberán tratar algunos temas relacionados con la inteligencia artificial y el impacto que ha tenido el desarrollo de estas tecnologías a nivel mundial, siendo implementadas en muchos campos como la medicina, la seguridad en los aeropuertos entre otras aplicaciones que han mejorado la calidad de vida de las personas.

Palabras Clave: Inteligencia artificial (IA), aprendizaje automático, reconocimiento facial, Python, keras

Abstract

The main objective of this article is the implementation in python of a prototype of a facial recognition system using machine learning, to achieve this, some issues related to artificial intelligence and the impact that the development of these technologies have had worldwide must be addressed, being implemented in many fields such as medicine, airport security among other applications that have improved the quality of life of people.

Key Words: Artificial intelligence (AI), automatic learning, facial recognition, Python, keras

I. Introducción

Este documento provee una explicación de los conceptos básicos de inteligencia artificial y de los aportes que han realizado estas tecnologías al mundo, dentro del artículo se mostrará un ejemplo sencillo de un algoritmo de reconocimiento facial haciendo uso de algunas herramientas utilizadas para la programación de este tipo de algoritmos explicando paso a paso el código fuente utilizado y mostrando cuál será el resultado esperado.

II. Contenido

Dentro de este apartado se definen algunos de los conceptos más importantes a tener en cuenta al momento de hablar sobre inteligencia artificial.

1. Aprendizaje automático

El aprendizaje automático (en inglés *machine learning*) es uno de los conceptos principales de la inteligencia artificial. Ya que este aspecto de la informática se enfoca en que los computadores o máquinas tengan una capacidad de aprendizaje autónomo sin estar programados para realizar la ejecución de alguna tarea en específico. Este aprendizaje es un gran avance en el mundo de la tecnología ya que hacia la década de 1980 los primeros ordenadores fueron explícitamente programados para cumplir una única tarea.

Por el contrario, hoy día gracias al aprendizaje automático muchos dispositivos que utilizamos diariamente cuentan con algoritmos de aprendizaje autónomo sencillos pero muy útiles enfocados a ofrecer una mejor experiencia a los usuarios de forma personalizada. como ejemplo se puede tomar la red social Facebook o los resultados del motor de búsqueda más grande del mundo Google. También se puede observar en nuestro correo electrónico al momento de ver los emails que han sido marcados como spam, este es un algoritmos que a través de la recolección de datos (intereses particulares de cada usuario) ha desarrollado la capacidad de decidir qué información puede ser o no de nuestro interés, adicionalmente también está pre entrenado para detectar qué mensajes son de fuentes seguras o no, este es un ejemplo muy sencillo de cómo los algoritmos de aprendizaje autónomo pueden ser utilizados. Dentro del concepto de aprendizaje automático se encuentran unas subcategorías que se definirán a continuación:

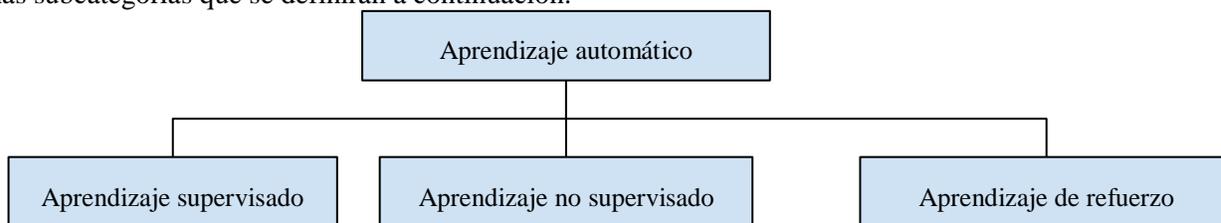


Figura 1: Tipos de aprendizaje profundo. Fuente: Realizada por el autor

a) Aprendizaje supervisado

Estos algoritmos utilizan datos que ya han sido previamente organizados para indicarle cómo tendría que ser categorizada la nueva información, dependiendo siempre de la intervención humana para recibir una retroalimentación, un ejemplo sencillo puede ser enseñar al algoritmo fotografías donde aparece un vehículo en específico, para luego identificar imágenes donde aparezca este vehículo.

b) Aprendizaje no supervisado

Estos algoritmos no necesitan recibir la información previamente organizada o categorizada ya que son capaces de aprender de manera autónoma la forma de realizar esta tarea, de esta forma se evita la intervención humana ya que plantean los posibles problemas y soluciones al desarrollar determinada tarea.

c) Aprendizaje de refuerzo

Estos algoritmos aprenden a través de la experiencia y se les debe dar un “incentivos positivos” por cada acierto que tienen, por ejemplo, cuando se premia a un perro cuando éste aprende a sentarse.

2. Código abierto

El código abierto es un modelo de desarrollo de software colaborativo, el cual permite a toda la comunidad de programadores a nivel mundial compartir sus códigos fuentes y de esta forma generalizar un conocimiento obtenido, y fortaleciendo cada vez más esta comunidad. También se refiere a open source a la adquisición de software gratuito sin necesidad de realizar la compra de licencias o suscripciones, para solventar alguna necesidad.



Figura 2: Logotipo del código abierto. Fuente: <https://opensource.org/>

3. Robótica

La robótica es una rama que está compuesta por varias disciplinas de la ingeniería como lo son la ingeniería mecánica, electrónica, mecánica y ciencias de la computación. Su objetivo es construir una herramienta que

permita el desarrollo de diversas actividades que actualmente realiza un ser humano, ejecutando estas tareas acertada y rápida. La robótica tiene un campo de aplicación muy amplio ya que es utilizada desde la medicina hasta la exploración de nuevos planetas en las incursiones espaciales, y mejorando la calidad de vida de las personas que tienen alguna limitación física.



Figura 3 Mano robótica de Shadow Robot Company

Fuente: <https://www.shadowrobot.com/artificial-intelligence-machine-learning/>

4. Redes neuronales

Las redes neuronales son sistemas de procesamiento de la información cuyo diseño y funcionamiento están inspirados en el cerebro humano. Están compuestas por un gran número de elementos simples llamados nodos o neuronas las cuales se están comunicando entre sí de manera constante a través de los enlaces de comunicación, cada una con un peso. El peso representa la información que será usada por la red neuronal para resolver un problema en específico.

De esta forma las redes neuronales son sistemas adaptativos capaces de aprender a resolver problemas a través de la experiencia, esto debido a que aprenden a desarrollar ciertas tareas con ejemplos ilustrativos. Mediante el entrenamiento o aprendizaje, las redes neuronales crean su propia representación de problemas a nivel interno, por esta razón se dice que son auto organizadas. Las redes neuronales se utilizan para resolver una gran cantidad de problemas que desde la programación convencional no sería sencillo resolver, tales como el reconocimiento por voz con el que los teléfonos inteligentes cuentan hoy día, que bajo un comando de activación y una orden por voz se pueden ejecutar tareas como la de leer o escribir mensajes de texto o de WhatsApp, llamar a una persona entre muchas otras funcionalidades.

Actualmente se están llevando a cabo nuevas investigaciones sobre el cerebro humano con la finalidad de ampliar y potenciar las capacidades de una red neuronal, centrándose en crear nuevas conexiones neuronales entre estas redes haciendo que estos enlaces estén mejor estructurados permitiendo un mejor intercambio de información de forma más rápida y segura. [3]

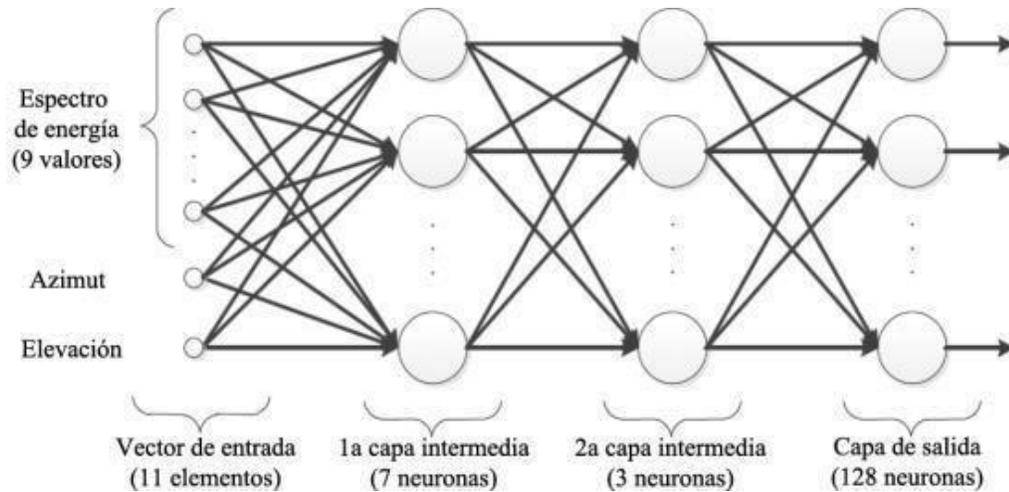


Figura 4 Ejemplo de red neuronal Fuente: https://www.researchgate.net/figure/Arquitectura-basica-utilizada-por-las-redes-neuronales-artificiales-con-dos-capas_fig2_300005515

5. Deep learning

Deep learning conocido en español como aprendizaje profundo son aquellas técnicas de aprendizaje automático que hacen uso de la arquitectura de redes neuronales pre entrenadas. El concepto profundo (Deep) viene de la gran cantidad de capas que pueden ser aplicadas con estas técnicas de trabajo, dado que las redes neuronales convencionales trabajan únicamente con una capa. El Deep learning es un aspecto de la inteligencia artificial que se encarga de simular los métodos de aprendizaje que utilizamos los seres humanos para obtener o generar un tipo de conocimiento. Esto se puede considerar como la automatización de un análisis predictivo. [3]

III. Trabajos realizados con IA

Existen en los últimos años una importante cantidad de trabajos realizados haciendo uso de sistemas con inteligencia artificial, entre ellos se encuentran:

a) Uberflip

Es una plataforma de content marketing que utiliza inteligencia artificial para personalizar la experiencia de contenidos y agilizar el ciclo de venta. Permite entender mejor a cada cliente potencial y predecir qué tipo de contenidos y temas le pueden interesar. Esta herramienta brinda recomendaciones de contenido en el momento correcto, con el formato correcto y dirigidas a la audiencia correcta. Estas recomendaciones se muestran justo en el punto donde es más probable que sean consultadas por el cliente potencial.

No se trata de ofrecer los típicos “artículos recientes” o “artículos populares”, sino más bien de recomendar un contenido que contribuya a que el visitante se convierta en lead o cliente potencial al brindar su información de contacto y avance por sí mismo el proceso de decisión de compra. De esta forma, la conversión será mayor. [16]

b) Articoolo

Es una aplicación de inteligencia artificial aplicada a la generación de contenidos en la cual su algoritmo inteligente crea contenido único y de calidad al simular la manera en que lo haría un cerebro humano. El proceso es simple: primero, se pone el título que se quiere para un post, por ejemplo: “Los mejores libros de superación personal y autoayuda”. A partir del título, la aplicación entenderá el contexto del tema y buscará, en este caso, los mejores recursos de superación personal extrayendo el sentimiento que le ponen y sus palabras clave. Con esto, construirá todo en una pieza coherente de contenido. Finalmente, lo escribirá usando programación neurolingüística para verificar que tenga una semántica correcta y sea legible. Lo más impresionante de todo es que se obtendrá un artículo único y coherente en tan solo un par de minutos. Cabe aclarar que esta herramienta no duplica ni plagia otros contenidos. [17]

c) Cortex

Es una aplicación de inteligencia artificial que se enfoca en mejorar el aspecto visual de las imágenes y vídeos de los posts en redes sociales con el objetivo de generar más interacción y puedan volverse virales. Básicamente, usa datos e insights para automatizar la creación de fotos y vídeos que den mejores resultados. Esta herramienta permite generar hallazgos como qué lenguaje visual utilizar, cómo deben ser las imágenes, qué tipo de fotos publicar, qué hashtags usar, qué tendencias resuenan más con el mercado meta, cómo distinguirse de la competencia, qué colores y qué mensajes utilizar. [18]

d) Adext AI

Es una plataforma dinámica de optimización de audiencias que encuentra las mejores audiencias para anuncios en Google, YouTube, Instagram y Facebook. Asigna de forma autónoma presupuesto a las copias o clones que hace de los anuncios ya existentes para explorar diferentes audiencias y, de acuerdo con los resultados, ajusta el presupuesto canalizando hacia aquellas que tengan la mayor tasa de conversión. Esta plataforma utiliza machine learning para detectar patrones de éxito y aprender con el tiempo qué audiencias son más redituables. También se nutre de big data de otras campañas para predecir tendencias. Aunque *Google* y *Facebook Ads* como tal ya utilizan aprendizaje automático e inteligencia artificial, una herramienta como *Adext AI* puede administrar inteligentemente los presupuestos entre ambas plataformas y combinar los hallazgos entre una y otra. [19]

e) **Node**

Es una aplicación de inteligencia artificial enfocada en la generación de clientes potenciales. Ofrece a los equipos de marketing y ventas una guía para identificar sus mercados y acelerar sus ganancias. Su inteligencia artificial puede sumergirse en montones de datos para encontrar al cliente potencial ideal teniendo en cuenta la información que ha recopilado hasta el momento. Es así como predice y califica cuán perfilado y preparado está el cliente potencial. Esto último es algo que puede ahorrar enormes cantidades de tiempo y energía a las empresas B2B

Los hallazgos accionables que brinda esta aplicación dan respuesta a las incógnitas sobre a quién contactar primero, qué decirle y cuándo decirlo para cerrar más ventas. En este sentido, Node ayuda a priorizar y personalizar la comunicación, aspectos que se traducen en una mayor tasa conversión cliente potencial-cliente. Esto se logra a través de su tecnología avanzada de inteligencia artificial, encuentra la conexión entre gente, compañías, productos y lugares en la web. Y, utilizando machine learning, busca dentro de estas conexiones para encontrar y definir el contexto y unirlos entre sí. Después, usa el procesamiento del lenguaje natural para analizar los principales atributos de un comprador perteneciente al mercado objetivo, y mediante aprendizaje profundo (Deep Learning) entrega proactivamente personas e información relevante para la empresa. Finalmente, los resultados de sus hallazgos son enviados de vuelta a sus algoritmos de inteligencia artificial para facilitar el aprendizaje constante y continuamente mejorar las recomendaciones. [20].

f) **Chatbot Liveperson**

Los chatbots que utilizan inteligencia artificial son una versión más avanzada de chatbots porque, estos hablan realmente como humanos y sus respuestas son muy acertadas y congruentes. Esto hace que la interacción con estos tipos de chat tenga mayor probabilidad de conversación. *Liveperson* es una plataforma conversacional impulsada por inteligencia artificial que hace fácil para los consumidores obtener respuestas sobre los productos que desean comprar dentro de los canales de comunicación que ya usan a diario. Esta aplicación usa inteligencia artificial para examinar las conversaciones con los clientes, identificar intentos de compra en tiempo real y ofrecer hallazgos accionables a las marcas para que optimicen rápidamente su comunicación y operaciones automáticas. [21]

IV. **Herramientas necesarias para el prototipo planeado**

a) **PYTHON**

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos,

programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Posee una licencia de código abierto, y permite la integración de varias librerías para la implementación de sistemas de inteligencia artificial, las cuales algunas de estas se mencionan a continuación.

b) **Keras**

Keras es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre *TensorFlow*, *Microsoft Cognitive Toolkit* o *Theano*. Está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de Aprendizaje Profundo. Sus fuertes se centran en ser amigable para el usuario, modular y extensible. Keras ha sido concebido para actuar como una interfaz en lugar de ser un *framework de machine learning standalone*. Ofrece un conjunto de abstracciones más intuitivas y de alto nivel haciendo más sencillo el desarrollo de modelos de aprendizaje profundo independientemente del backend computacional utilizado. [10]

c) **TensorFlow**

TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que les permite a los investigadores innovar con el aprendizaje automático y, a los desarrolladores, compilar e implementar con facilidad aplicaciones con tecnología de Aprendizaje Automático.[8]

Compilación sencilla de modelos: *TensorFlow* compila y entrena modelos de Aprendizaje Automático con facilidad mediante API intuitivas y de alto nivel como Keras, con ejecución inmediata, que permite una iteración de modelos inmediata y una depuración fácil.

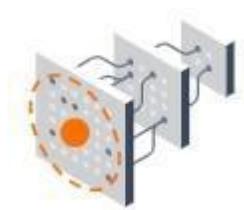


Figura 5 Compilación sencilla de modelos Fuente: <https://www.tensorflow.org/?hl=es-419>.

Producción de aprendizaje automático sólido en cualquier parte:

TensorFlow permite entrenar e implementar con facilidad modelos en la nube, locales, en el navegador o en el dispositivo, sin importar el lenguaje de programación que se esté utilizando.



Figura 6 Producción AA Fuente: <https://www.tensorflow.org/?hl=es-419>.

Importante experimentación para la investigación: *TensorFlow* permite adoptar una arquitectura simple y flexible para llevar las ideas nuevas del concepto al código, a modelos innovadores y a publicaciones con mayor rapidez.



Figura 7 Experimentación para la investigación Fuente: <https://www.tensorflow.org/?hl=es-419>.

d) **Opencv2**

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. *OpenCV* significa Open *Computer Vision* (Visión Artificial Abierta). Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2021 se la sigue mencionando como la biblioteca más popular de visión artificial. Detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV. Su popularidad se debe a:

- Un software libre, publicado bajo licencia BSD, que permite que sea usado libremente para propósitos comerciales y de investigación.
- Es multiplataforma, para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware como x86, x64 (PC), ARM (celulares y Raspberry Pi).
- Posee extensa documentación: la organización tiene una preocupación activa de mantener la documentación de referencia para desarrolladores lo más completa y actualizada posible, ejemplos de uso de sus funciones y

tutoriales accesibles al público no iniciado en visión artificial, además de difundir y fomentar libros y sitios de formación.



Figura 9 Ilustración de detección automática de rostro Fuente: Autor

Las áreas de aplicación de OpenCV incluyen:

- Características 2D y 3D
- Estimación de pose de cámara
- Reconocimiento facial
- Reconocimiento de gestos
- Interacción persona-computadora
- Robótica móvil
- Comprensión de movimientos
- Reconocimiento de objetos
- Segmentación
- Estereoscopia
- Structure from motion (SFM)
- Tracking
- Realidad aumentada

OpenCV está totalmente desarrollado en C++, orientado a objetos y con alta eficiencia computacional. Su API es C++, pero incluye conectores para otros lenguajes como:

- Python
- JAVA

- Matlab • Octave
- Javascript

e) Numpy

NumPy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas. El precursor de *NumPy*, *Numeric*, fue creado originalmente por Jim Hugunin con contribuciones de varios otros desarrolladores. En 2005, Travis Oliphant creó NumPy incorporando características de la competencia Numarray en Numeric, con amplias modificaciones. NumPy es un software de código abierto y cuenta con muchos colaboradores.

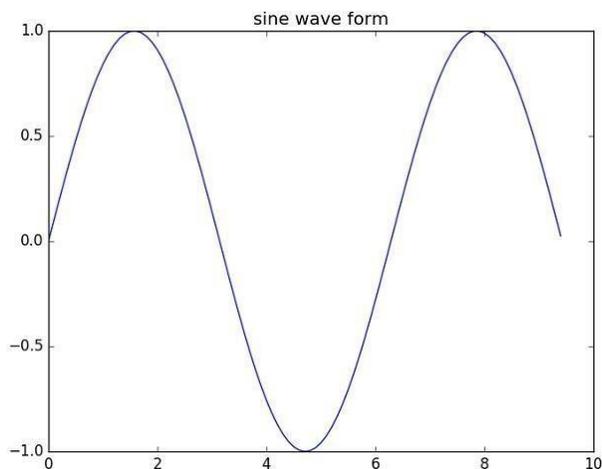


Figura 10 Ejemplo de gráfica generada con NUMPY,

Fuente: https://www.tutorialspoint.com/numpy/numpy_matplotlib.htm

f) Matplotlib

Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy. Proporciona una API, pylab, diseñada para utilizar con MATLAB.

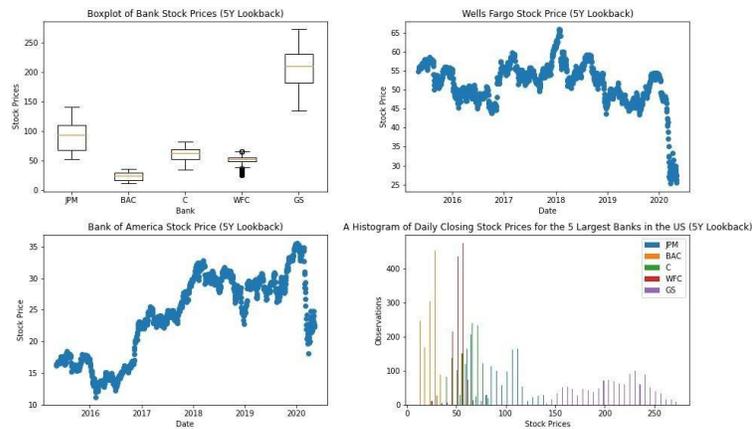


Figura 11 Ejemplo de grafico generado con Matplotlib Fuente: <https://www.freecodecamp.org/news/how-to-embed-interactive-python-visualizations-on-your-website-with-python-and-matplotlib/>

g) Jupyter

Proyecto Jupyter es una organización sin ánimo de lucro, creada para desarrolladores de código abierto o open source, esta plataforma soporta 3 lenguajes de programación los cuales están representados en sus iniciales Ju es Julia, Py es Python y R es lenguaje R

V. Implementación prototipo

a) Detección de rostros

Para la detección de un rostro se debe especificar las áreas de una imagen digital donde se aprecia la cara de una persona de tal forma que esta cree una silueta del rostro identificado. para llevar a cabo esta tarea se puede usar un método llamado plantilla el cual es una técnica empleada para detectar objetos dentro de una escena, donde es representado por formas, en el caso de los rostros se toma como objeto los ojos y la nariz y se utiliza como una plantilla parametrizada lo cual al interactuar con la imagen de un rostro buscando alguna similitud con los objetos creados (nariz y ojos), para determinar si es o no la persona correcta.

b) Reconocimiento facial

El reconocimiento facial es una técnica que está basada en la capacidad de un ser humano para identificar rostros tomando algunas características faciales de una persona como el color de piel, tamaño de los ojos y la nariz, y sus rasgos únicos. Luego estos datos son comparados en una base datos buscando una similitud de un rostros en específico dentro de muchos datos.

c) Aplicación práctica

En este numeral se hará la implementación de un sistema de reconocimiento facial partiendo de la detección de rostros usando python como lenguaje de programación y librerías como tensorflow y keras.

La librería *tensorflow* permite utilizar la red convolucional *mobilenet* herramienta que nos permitirá identificar rostros. En la librería *keras* se puede utilizar *facenet* que es el sistema que hace el reconocimiento facial. Se utilizará la librería *OPENCV* para leer / escribir imágenes del disco duro y almacenarlas. Se utilizará la librería *Numpy* para hacer cálculos y comparaciones entre los rostros que se han identificado y determinar a qué persona corresponde, se utilizará *matplotlib* para dibujar las imágenes que se vayan obteniendo y finalmente se utilizará la librería *Os* la cual permitirá leer los datos del disco duro. A continuación, se muestra en detalle el código donde se cargan las librerías:

```
IMPORT TENSORFLOW AS TF
FROM TENSORFLOW.KERAS.MODELS IMPORT LOAD_MODEL
IMPORT CV2
IMPORT NUMPY AS NP
IMPORT MATPLOTLIB.PYLOT AS PLT
IMPORT OS
```

Se aclara que el entorno utilizado para el procesamiento del código que se utiliza para esta demostración será JUPYTER el cual proporciona un entorno pensado para satisfacer necesidades concretas y ajustarse al flujo de trabajo de la ciencia de datos y la simulación numérica, permitiendo desde una misma interfaz escribir, documentar y ejecutar código, visualizar datos, realizar cálculos y ver los resultados. Inicialmente se tendrán una serie de rostros conocidos (rostros de referencia), almacenados dentro del proyecto en una carpeta que se nombra “knowns” como se muestra a continuación:

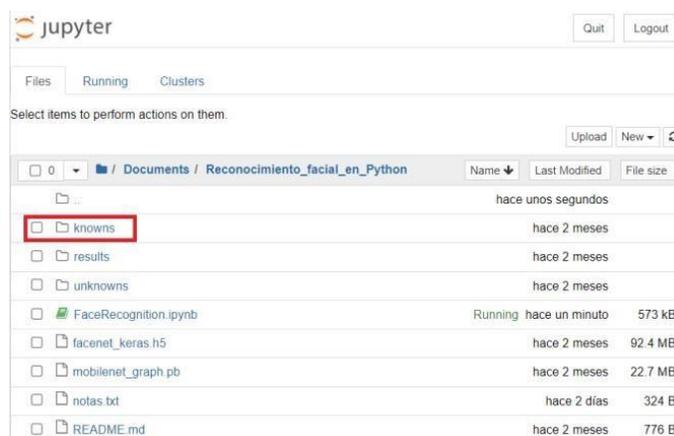


Figura 12: Archivos del proyecto señalando carpeta knowns Fuente: Autor



Figura 13 Contenido de la carpeta knowns, Fuente: Autor

La idea principal del desarrollo de este prototipo es solicitarle al sistema que detecte las características esenciales de estos rostros de referencia ubicados en la carpeta “knowns”. Ahora bien, una vez cargadas las librerías como se indicó en la imagen 11, se van a crear los directorios de referencia donde se tienen los rostros conocidos, desconocidos y donde vamos a almacenar los resultados del proceso de detección de rostros.

```
DIR_KNOWNS = 'knowns'
DIR_UNKNOWNNS = 'unknowns'
DIR_RESULTS = 'results'
```

Ahora se va a leer la información del archivo mobilenet_graph.pb, el cual corresponde a una red convolucionada que ya está entrenada previamente para detectar rostros en imágenes, esta red neuronal se leerá con ayuda de la librería TensorFlow. A continuación, se muestra en detalle el código que permite la lectura de la red convolucional:

```
# Leer mobilenet_graph.pb with
tf.io.gfile.GFile('mobilenet_graph.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=")
```

Luego de tener cargada la red neuronal se creará una función básica para cargar imágenes donde se utilizará para tal fin la librería de OpenCV, esta función recibirá como parámetros el directorio de la imagen y el nombre de la imagen, como se muestra a continuación:

```
# Cargar imagen def load_image(DIR,
NAME):
    return cv2.cvtColor(cv2.imread(f'{DIR}/{name}'), cv2.COLOR_BGR2RGB)
```

Ahora se creará la función para detectar rostros la cual recibe como parámetro una imagen y un umbral (Probabilidad de certeza de que el rostro corresponde al sujeto definido en la carpeta “knowns”). El resultado de esta función será una lista de imágenes con rostros que se asemejan al rostro del sujeto conocido.

```
def detect_faces(image, score_threshold=0.7):
    global boxes, scores
    (imh, imw) = image.shape[:-1]    img
    = np.expand_dims(image,axis=0)
    # Inicializar mobilenet
    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')    scores =
mobilenet.get_tensor_by_name('detection_scores:0')    # Predicción
(detECCIÓN)
    (boxes, scores) = sess.run([boxes, scores], feed_dict={image_tensor:img})
    # Reajustar tamaños boxes, scores
boxes = np.squeeze(boxes,axis=0)
scores = np.squeeze(scores,axis=0)
# Depurar bounding boxes
    idx = np.where(scores>=score_threshold)[0]
    # Crear bounding boxes
    bboxes = []    for index in idx:        ymin,
xmin, ymax, xmax = boxes[index,:]
        (left, right, top, bottom) = (xmin*imw, xmax*imw, ymin*imh, ymax*imh)
left, right, top, bottom = int(left), int(right), int(top), int(bottom)
    bboxes.append([left,right,top,bottom])    return bboxes
```

Ahora se creará una función para dibujar la(s) imagen(es) que retorna la función “detect_faces”, esta función recibe los siguientes parámetros:

1. Imagen (image)
2. Espacio donde se pintará la imagen (box)
3. Color de la línea que bordeara el rostro identificado (color)
4. Ancho de la línea que bordeará el rostro identificado (line_width), por defecto se colocará en 6

Ahora bien, una vez definidas las funciones para leer las imágenes, detectar rostros y dibujar los rostros que detecta la red neuronal se procede a ejecutar toda esta definición de código, donde para la ejecución se tomará como ejemplo la imagen dispuesta en el directorio “knowns”, como se muestra a continuación:

```
name = 'juan_camilo.jpg'
image = load_image(DIR_UNKNOWNNS,name)
bboxes = detect_faces(image) for box in bboxes:
detected_faces = draw_box(image,box,(0,255,0))
fig = plt.figure(figsize=(10,10))
plt.imshow(detected_faces)
```

Una vez ejecutada la función definida en el script 5 se obtiene el siguiente resultado:



Figura 14 Resultado de la detección de rostros Fuente: Autor

VI. Trabajos futuros

a) Guardias de seguridad

Actualmente se está experimentando con guardias de seguridad robotizados en varios lugares del mundo, entre los que destaca el guardia de seguridad robotizado K5 de Knightscope, que ha sido utilizado por múltiples empresas e instituciones, incluyendo la Sociedad para la Prevención de la Crueldad contra los Animales (SPCA), ubicada en San Francisco, donde existe un robot encargado de disuadir el vandalismo.

El uso de la inteligencia artificial y el aprendizaje de máquinas, cuando se combina con guardias de seguridad robóticos, podría proporcionar una variedad de beneficios a las autoridades locales, incluyendo imágenes de video vigilancia de 360 grados que utilizan reconocimiento facial y análisis de comportamiento predictivo, así como imágenes térmicas, grabación y reconocimiento de voz y RFID de largo alcance.

b) Operativo de call center

Los avances en el procesamiento del lenguaje natural y los sistemas de aprendizaje automático han permitido recibir una llamada de un robot, algo con lo que todos probablemente ya se ha experimentado y con lo que se ha familiarizado, y esta tendencia parece que va a continuar en el futuro.

En la actualidad hay muchas empresas en todo el mundo que utilizan asistentes virtuales que funcionan con máquinas de aprendizaje en sus centros de llamadas y esta tendencia se está adoptando cada vez más.

En algunos casos, investigaciones realizadas por compañías que han introducido operativos de centros de llamadas con tecnología de inteligencia artificial han demostrado que los clientes prefieren interactuar con robots de procesamiento de lenguaje natural en lugar de abrirse camino a través de menús y preguntas frecuentes.

c) Taxis

Con el advenimiento de los vehículos autónomos y los coches sin conductor, sin duda vendrá un cambio en la forma en que utilizamos los servicios de transporte. Es probable que los impulsores humanos sean sustituidos por sistemas automatizados alimentados por IA que probablemente integren una serie de tecnologías de vanguardia, como la informática de vanguardia y las redes de comunicaciones inalámbricas 5G. La IA y los taxis autónomos impulsados por aprendizaje automático no sólo no requerirían un conductor humano, sino que también podrían utilizar la información de tráfico en vivo y de viaje que se le suministra desde otros vehículos e infraestructura de transporte, lo que le permitiría determinar la mejor ruta para su cliente.

d) Chef

Una de las sorpresas de la inteligencia artificial y las tecnologías de aprendizaje automático fue la creación del Chef Watson, que utilizó las tecnologías Watson de IBM. El chef Watson es capaz de generar ideas y recetas de comidas totalmente nuevas utilizando su amplio conocimiento de los maridajes de alimentos y sabores. [30]. Por otro lado, encontramos a Flippy, el robot de preparación de comidas de Miso Electronics. Flippy es capaz de preparar y servir comidas y bocadillos a velocidades mucho más rápidas que los chefs humanos [31]. Con sólo estos dos ejemplos queda claro que la robótica impulsada por la IA podría hacerse cargo también de nuestras cocinas.

VII. Conclusiones

La aplicación de la inteligencia artificial es un área de la ingeniería que crea muchas expectativas cuando se habla de ella, no obstante, la realización de este trabajo nos ha mostrado que este es un tema que se viene desarrollando desde mucho tiempo atrás y que con el pasar de los años se ha ido mejorando cada vez más. Aplicado en muchos ámbitos profesionales e incluso de nuestra vida diaria, ya que hoy día en el mercado se

encuentran teléfonos inteligentes que cuentan con un sistema de reconocimiento facial para bloquear entre muchas otras funcionalidades.

Por otro lado, está el surgimiento de las redes neuronales y la robótica que se han combinado para permitir que un robot tenga comportamientos más parecidos a los humanos y esté en la capacidad de aprender por su cuenta de acuerdo con las experiencias que tenga, ya que una red neuronal le permite auto entrenarse en muchas experiencias, y le da la capacidad de generar una solución a cualquier tipo de problema.

VIII. Referencias

- [1] L. Rouhiainen, «Google Scholar,» 2018. [En línea]. Available: https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf.
- [2] J. J. M. MORENO, «Google Scholar,» 2002. [En línea]. Available: <https://www.tesisenred.net/bitstream/handle/10803/9441/tjjmm1de1.pdf?sequence=1&isAllowed=y>.
- [3] D. J. Matich, «Google Scholar,» 03 2001. [En línea]. Available: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matichredesneuronales.pdf.
- [4] D. L. CASTAÑO SAAVEDRA y J. D. ALONSO SIERRA, «Google Scholar,» 2019. [En línea]. Available: <https://repository.ucatolica.edu.co/bitstream/10983/24032/1/Final%20Trabajo%20de%20grado.pdf>.
- [5] N. Maynez, «inboundcycle,» 16 01 2020. [En línea]. Available: <https://www.inboundcycle.com/blogdeinbound-marketing/aplicaciones-inteligencia-artificial-2020>.
- [6] Lanner, «Lanner,» 03 10 2018. [En línea]. Available: <https://www.lanner-america.com/es/bloges/10trabajos-que-pronto-podrian-ser-realizados-por-robots-maquinas-y-vehiculos-impulsados-por-la-ia/>.
- [7] Semana, «Cinco conceptos básicos para entender la inteligencia artificial, según Google,» 03 11 2020.
- [8] Tensorflow, «Tensorflow,» [En línea]. Available: <https://www.tensorflow.org/?hl=es-419>.
- [9] Google Scholar, «Google Scholar,» [En línea]. Available: <https://www.redalyc.org/pdf/1815/181531232001.pdf>.
- [10] Torres, J. (2018). Deep Learning Introducción Practica Con Kera. [11] Google Scholar, «Google Scholar,» [En línea]. Available: <http://mapir.isa.uma.es/varevalo/drafts/arevalo2004lva1.pdf> [12] Google Scholar, «Google Scholar,» [En línea]. Available: <http://adsabs.harvard.edu/pdf/2005ASPC..347...91B>.
- [13] Google Scholar, «Google Scholar,» [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/46984/Documento_completo.pdf?sequence=1&isAllowed=y.
- [14] Euskadinnova, «Euskadinnova,» [En línea]. Available: [https://www.spri.eus/euskadinnova/es/enpresadigitala/agenda/sistemas-prediccionmachinelearning/8580.aspx#:~:text=Los%20sistemas%20de%20prediccion%20son,aprendizaje%20automático%20\(machine%20learning\)](https://www.spri.eus/euskadinnova/es/enpresadigitala/agenda/sistemas-prediccionmachinelearning/8580.aspx#:~:text=Los%20sistemas%20de%20prediccion%20son,aprendizaje%20automático%20(machine%20learning)).
- [15] X. B. Olabe, «Google Scholar,» [En línea]. Available:

https://ocw.ehu.es/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/librodelcurso.pdf.

- [16] Página web oficial de Uberflip [En línea]. Available: <https://www.uberflip.com/>
- [17] Página web oficial de Articoolo [En línea]. Available: <https://articoolo.com/?lang=es>
- [18] Página web oficial de Cortex [En línea]. Available: <https://www.capterra.co/software/172547/cortex>.
- [19] Página web oficial de Adext [En línea]. Available: <https://www.adext.ai/>
- [20] Página web oficial de Node [En línea]. Available: <https://www.sugarcrm.com/es/>
- [21] Página web oficial de Liveperson [En línea]. Available: <https://www.liveperson.com/> [22] Página web oficial de IBM [En línea]. Available: <https://www.ibm.com/blogs/think/eses/2018/03/20/chef-watsoninspira-nuevas-recetas-lanzarote-2/>
- [23] Página web oficial de Miso Electronics [En línea]. Available: Flippy - Miso Robotics
- [24] Jordi, TORRES. 2018. DEEP LEARNING Introducción práctica con Keras. 2018. Kaggle. Kaggle. [En línea] Kaggle. [Citado el: 16 de 02 de 2020.] https://www.kaggle.com/keras/resnet50#imagenet_class_index.json. [25] Clever Data Big Data Prediction, Que es el Machine learning?, rescatado en Septiembre del 2016 desde <http://cleverdata.io/que-es-machine-learningbig-data/>
- [26] Alberto, and Jerónimo Ríos. 2017. “Reconocimiento Facial Por El Método De Eigenfaces.” *Pistas Educativas* 127 (04): 66–81. <http://itcelaya.edu.mx/ojs/index.php/pistas>.
- [27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529--533, 2015.
- [28] J. Roos, I. Lane, E. Botha, and G. P. Hancke. Using neural networks for non-intrusive monitoring of industrial electrical loads. In *Instrumentation and Measurement Technology Conference, 1994. IMTC/94. Conference Proceedings. 10th Anniversary. Advanced Technologies in I & M., 1994 IEEE*, pages 1115--1118. IEEE, 1994. <http://dx.doi.org/10.1109/IMTC.1994.351862> [pathdoi:10.1109/IMTC.1994.351862](https://doi.org/10.1109/IMTC.1994.351862)
- [29] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193--202, 1980.
- [30] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012.

Publicación Facultad de Ingeniería y Red de Investigaciones de Tecnología Avanzada _ RITA

REVISTA

TIA

