



INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERIA EN SISTEMAS COMPUTACIONALES

Nombre del Proyecto:
Reconocimiento Facial

Presenta:

Gillot de Leon Diana Guadalupe

161080124 25%

Hernandez Diaz Mitzi Mabel

161080181 25%

Martinez Rodriguez Gerardo Emmanuel

161080185 25%

Ramirez Peña Carlos Ivan

161080215 25%

Asesor Interno:

Parrá Hernández Abiel Tomás



Índice

Resumen	1
ABSTRACT	2
Introducción	3
Descripción del problema.	3
Objetivo General	4
Objetivos Específicos.....	4
Supuesto.	4
Justificación.....	5
Marco Teorico	6
Marco Historico del Reconocimiento Facial.	6
Marco Conceptual	8
Sistemas Informáticos.....	8
Python	8
[LIBRO] La referencia del lenguaje Python	12
Metodologia de Trabajo.	13
Desarrollo e implementación.	14
Resultados.....	20
Conclusiones.....	21
Gillot de Leon Diana Guadalupe.....	21
Hernandez Diaz Mitzi Mabel	21
Martinez Rodriguez Gerardo Emmanuel.....	21
Ramirez Peña Carlos Ivan	21
Fuentes de Informacion.	23

Resumen

Aunque muchas veces se usan los nombres indistintamente, Detección y Reconocimiento facial no significan exactamente lo mismo.

Se conoce como detección facial el proceso de encontrar rostros humanos en una imagen, sin llegar a identificar a las personas.

En cambio, el reconocimiento facial es ir un paso más allá: no solo buscamos rostros, sino que intentamos verificar la identidad de las personas que aparecen en la imagen.

En el siguiente proyecto veremos cómo aplicar ambos. En la primera parte utilizaremos OpenCV para hacer detección facial, y en la segunda parte utilizaremos Face_Recognition para hacer reconocimiento facial.

OpenCV utiliza el algoritmo de Viola & Jones para detectar rostros, que desde su publicación en 2001 es probablemente el algoritmo por excelencia para hacer este tipo de tareas. Este algoritmo está pensado para detectar objetos concretos, ya sean rostros, coches, gatos, plátanos...

Para poder detectar objetos se necesita un clasificador. Podemos imaginar los clasificadores como una especie de “plantilla” para detectar un determinado tipo de objeto. Este clasificador es entrenado proporcionándole cientos de imágenes de ejemplo del objeto que se quiere detectar.

Cuando se quieren detectar objetos, el algoritmo aplica una serie de clasificadores (una “cascada de clasificadores”) sobre una región de interés en la que considera que podría hallarse el objeto en cuestión. Si todos los clasificadores acaban dando positivo, el algoritmo considera que hay una coincidencia y se ha encontrado el objeto deseado.

Palabras Clave: Reconocimiento facial, deteccion facial, open CV, algoritmo, objeto.

ABSTRACT

Although the names are often used interchangeably, Face Detection and Recognition do not mean exactly the same thing.

Face detection is the process of finding human faces in an image, without actually identifying people.

Instead, facial recognition is going one step further: we not only search for faces, but we also try to verify the identity of the people who appear in the image.

In the next project we will see how to apply both. In the first part we will use OpenCV to do face detection, and in the second part we will use Face_Recognition to do facial recognition.

OpenCV uses the Viola & Jones algorithm to detect faces, which since its publication in 2001 is probably the algorithm par excellence to do this type of task. This algorithm is designed to detect specific objects, whether they are faces, cars, cats, bananas
...

In order to detect objects you need a classifier. We can imagine classifiers as a kind of "template" to detect a certain type of object. This classifier is trained by providing it with hundreds of example images of the object to be detected.

When you want to detect objects, the algorithm applies a series of classifiers (a "cascade of classifiers") on a region of interest in which it considers that the object in question could be found. If all the classifiers are positive, the algorithm considers that there is a match and the desired object has been found.

Keywords: Face recognition, face detection, open CV, algorithm, object.

Introducción

El presente proyecto se refiere al diseño y desarrollo de una aplicación de reconocimiento facial.

Se implementará con un gestor de base de datos. Mediante las base de datos se va a permitir comprobar las pruebas y error.

La característica principal es gestionar mediante python, un reconocimiento facial ya que será reconocer cada uno del personal que va a ingresar al Instituto Tecnológico de Iztapalapa, o para otra institución.

Para analizar la problemática es necesario mencionar sus causas. Una de ellas es, el robo al instituto. Las personas que no son del Instituto no las reconocerá así también lograr una lata transparencia de agilidad administrativa reduciendo tiempos.

Descripción del problema.

En la actualidad institutos a nivel Universitaria, aun no se sabe que tengan una plataforma de reconocimiento facial que da un facilidad de reconocer personas del instituto y sin perder mucho tiempo, desafortunadamente no hay este proyecto puesto por falta de interés, recursos, etc.

Este proyecto surge a partir de la necesidad del Instituto Tecnológico de Iztapalapa, que no cuentan con esta plataforma.

El diseño que se planteo mediante gestor de programación en python, que permitió realizar la gestión rápida y eficaz, permitiendo satisfactoriamente el diseño de la plataforma.

Objetivo General

Diseñar y desarrollar una aplicación para la identificación de personas del Instituto Tecnológico de Iztapalapa mediante el programa python.

Objetivos Específicos

- Diseñar una aplicación vía escritorio que permita el control de identificación del personal y alumnos Instituto Tecnológico de Iztapalapa.
- Implementar bajo los estándares del Tecnológico Nacional de México, la administración que permita el fácil manejo de la plataforma, para llevar a cabo la identificación referente al personal administrativo y alumnado.
- Comprobar la transparencia de información administrativa y la eficiencia del programa a través de la reducción de tiempos de accesos a la Institución.

Supuesto.

- Al diseñar una aplicación para el reconocimiento facial para el Instituto Tecnológico de Iztapalapa ayudará a tener una mejor administración, reduciendo tiempos y robos al instituto.

Justificación

Derivado de la experiencia en la gestión de los tiempos de acceso del Instituto Tecnológico de Iztapalapa. Ya que tenemos credenciales pero eso no es seguro, puesto que se las pueden prestar y los guardias no se dan cuenta quien realmente está entrando. El no tener la identificación de cada alumno y cada docente provoca que no se garantice la seguridad del ITIZ, ya que se ha provocado ciertas tragedias en el Instituto. Además, los horarios mixtos del personal no se sabe que persona está entrando. Así también ocasionan que al momento de la entrada se detengan y ocasione molestia de la comunidad estudiantil o al personal por los tiempos de acceso.

Utilizaremos el programa Python, se llevara la programación ya que se está utilizando más en la Inteligencia Artificial

Los Guardas del ITZI, se pueden reducir en este apoyo humano y se puede asignar a otras áreas de oportunidad dentro del ITIZ, debido a que una plataforma de impacto tecnológico posiciona las capacidades y especialización en mejorar los procesos, no solo ayuda la parte administrativa si no reduce tiempos y daría una mejor imagen para el ITIZ.

El impacto tecnológico que se requiere dentro de una institución donde se imparten carreras de sistemas computacionales, es ser partícipe de esta transformación para contribuir a la comunidad del TECNM, posicionando con la plataforma una mejora en los procesos administrativos, siendo esta plataforma útil no únicamente para el ITIZ, además puede llevarse acabo la implementación de una plataforma global para otras Instituciones o empresas y así mismo ayudar a los demás planteles a disminuir la misma problemática.

Marco Teorico

En este capítulo hablaremos sobre el antecedente de la Inteligencia Artificial acerca del Reconocimiento Facial.

Marco Historico del Reconocimiento Facial.

Se dice que, Woodrow Wilson Bledsoe fue como el pionero de esta tecnología, ya que en 1960 trabajó en un sistema para clasificar los rasgos del rostro humano a través de la tabla RAND. Este sistema utilizaba un lápiz óptico y unas coordenadas para situar los ojos, la nariz o la boca de las personas de forma precisa, pero era un procedimiento todavía muy manual.

Así el reconocimiento facial se fue haciendo un hueco en las aplicaciones de seguridad, en particular aquellas concernientes al Estado. En 2001, nace el Viola-Jones Object Detection Framework, que propone algoritmos para detectar objetos dentro de imágenes, y que enseguida fue utilizado para la detección de rostros de forma exitosa. En la década actual, aparecen los Convolutional Neural Networks, que hasta hoy suponen el mejor modo de detectar rostros, haciendo uso de sistemas muy potentes capaces de almacenar información en la nube.

El reconocimiento facial (RF) automatizado por computadora es un área de investigación en pleno desarrollo. Identificar a una persona por medio de una imagen de su rostro implica emular el proceso cognitivo que realiza un ser humano al reconocer a sus pares. Aunque existen trabajos en el campo de la psicofísica y la neurociencia, aun no se sabe con certeza como funcionan estos procesos internamente en el cerebro humano. (Scarel, Germán Matías pag.1)

La detección implica encontrar las aéreas dentro de una imagen que contienen un rostro. Básicamente se trata de descartar todo lo que sea fondo, y así obtener la ubicación y tamaño exacto de la cara.

El reconocimiento consiste en clasificar las características extraídas de cada rostro. Esta clasificación puede ser realizada de manera supervisada, en la cual un patrón de entrada es identificado como miembro de una clase predefinida, o de manera no supervisada, donde el patrón es asignado a una clase desconocida. Aquí, cada clase es un sujeto, por lo tanto, al clasificar las características se está indicando a que sujeto pertenecen.

El uso del color en el procesamiento de imágenes es de importancia ya que suele simplificar la identificación y extracción de objetos en una escena. Un modelo de color especifica un sistema de coordenadas y un subespacio en el cual un color es definido por un punto. Muchos modelos están orientados al hardware como lo son el RGB (rojo, verde y azul), CMY (cian, magenta y amarillo) y el CMYK (cian, magenta, amarillo y negro); otros están orientados hacia aplicaciones donde se pretende manipular el color, tales como HSI (tono, saturación e intensidad), HSV (tono, saturación y valor) y el YCbCr (luminancia, crominancia azul y crominancia roja). (Germán Matias,pag.9).



Es decir, para el caso de imágenes faciales, se pretende extraer las características más relevantes en una imagen facial, las cuales pueden o no estar relacionadas a las características faciales como boca, nariz, ojos, etc.. En términos matemáticos, se quiere encontrar las componentes principales de la distribución de caras, o lo que es lo mismo, la matriz de covarianza del conjunto de imágenes caras [TP91]. Estos eigenvectores pueden ser vistos como el conjunto de vectores que caracterizan la variación entre las imágenes faciales. A partir de esto, cada cara puede ser representada exactamente como una combinación lineal.

Marco Conceptual

En el capítulo III, hablaremos sobre los sistemas informáticos, los cuales utilizaremos para hacer el diseño y desarrollo de una aplicación de reconocimiento facial, los cuales serán: PYTHON.

Sistemas Informáticos.

Python

Python se remonta a finales de la década de 1980, y su implementación fue iniciada en diciembre de 1989 por Guido van Rossum en CWI en los Países Bajos como sucesor del lenguaje de programación ABC capaz de manejar excepciones y interactuando con el sistema operativo Amoeba. Van Rossum es el autor principal de Python

Python es un lenguaje de programación de alto nivel y propósito general muy utilizado. Su diseño la filosofía enfatiza la legibilidad del código, y su sintaxis permite a los programadores expresar conceptos en menos líneas de código de lo que sería posible en lenguajes como C ++ o Java.

El lenguaje proporciona construcciones destinado a permitir programas claros tanto a pequeña como a gran escala. Python admite múltiples paradigmas de programación, incluidos los orientados a objetos, imperativos y programación funcional o estilos procedimentales. Cuenta con un sistema de tipo dinámico y memoria automática. Gestión y tiene una biblioteca estándar amplia y completa.

Los intérpretes de Python están disponibles para instalación en muchos sistemas operativos, lo que permite la ejecución de código Python en una amplia variedad de sistemas. La programación es totalmente compatible y hay una serie de características de lenguaje que admiten funciones programación y programación orientada a aspectos.

Se admiten muchos otros paradigmas mediante extensiones, incluyendo diseño por contrato y programación lógica. Python usa escritura dinámica y una combinación de recuento de referencias y un recolector de basura con detección de ciclos para la gestión de la memoria.

Características importantes:

- Python es una resolución dinámica de nombres (enlace tardío), que enlaza los nombres de métodos y variables durante el programa. Ejecución.
- El diseño de Python ofrece cierto soporte para la programación funcional en la tradición Lisp.

- El lenguaje tiene funciones de mapa, reducción y filtrado; comprensiones para listas, diccionarios y conjuntos; y generador expresiones.

Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited to many tasks. This is deliberate and has been described as a "batteries included" Python philosophy. For Internet-facing applications, a large number of standard formats and protocols (such as MIME and HTTP) are supported.

Además, Python tiene ventajas significativas sobre los distintos lenguajes de programación, lo cual una de ellas es:

- Sintaxis limpia, esto es; para la asignación de bloques para utilizar excepciones.
- Programas de tolerancia, quiere decir que es característico de la mayoría de los idiomas interpretados.
- La distribución normal tiene muchos módulos útiles ya que tiene incluido el módulo para desarrollar GUI.

El uso de Python en modo interactivo (muy útil para experimentar y resolver problemas simples, la distribución normal es un entorno de desarrollo simple pero al mismo tiempo bastante potente, que es llamado IDLE y lo que está escrito en Python.

Python tiene una gran biblioteca estándar, comúnmente citada como una de las mayores fortalezas de Python, proporcionando herramientas adecuadas para muchas tareas. Esto es deliberado y se ha descrito como "pilas incluidas".

Filosofía Python. Para aplicaciones orientadas a Internet, una gran cantidad de formatos y protocolos estándar (como MIME y HTTP) son compatibles. Módulos para crear interfaces gráficas de usuario, conectarse a relacionales de bases de datos, generadores de números pseudoaleatorios, aritmética con decimales de precisión arbitraria, También se incluyen expresiones regulares y pruebas unitarias. Algunas partes de la biblioteca estándar están cubiertas por especificaciones, pero la mayoría de los módulos no lo son. Están especificados por su código, interno documentación y conjunto de pruebas (si se proporciona). Sin embargo, debido a que la mayor parte de la biblioteca estándar es multiplataforma Código Python, solo hay unos pocos módulos que deben ser alterados o reescritos por completo por alternativas implementaciones.

El mecanismo principal para proponer nuevas características importantes, para recopilar información de la comunidad sobre un problema y para documentar las decisiones de diseño que se han aplicado a Python. La mejora del lenguaje va de la mano con el desarrollo de la referencia CPython implementación.

La lista de correo python-dev es el foro principal de discusión sobre el lenguaje desarrollo; Los problemas específicos se discuten en el rastreador de errores Roundup que se encuentra en python.org.

El desarrollo tiene lugar en un repositorio de código fuente autohospedado que ejecuta Mercurial. Un número de alfa, beta, y los candidatos a lanzamiento también se publican como vistas previas y para probar antes de que se realice el lanzamiento final. Aunque hay un cronograma aproximado para cada versión, esto a menudo se retrasa si el código no está listo. El equipo de desarrollo supervisa el estado del código ejecutando el gran conjunto de pruebas unitarias durante el desarrollo, y utilizando el sistema de integración continua BuildBot. La comunidad de desarrolladores de Python también ha contribuido más de 72.000 módulos de software (a partir de enero de 2016) al índice de paquetes de Python (llamado PyPI), el repositorio de bibliotecas de terceros para Python.

<http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf?sequence=3>

Python es un lenguaje de programación de propósito general de alto nivel interpretado . La filosofía de diseño de Python enfatiza la legibilidad del código con su notable uso de sangría significativa . Sus construcciones de lenguaje , así como su enfoque orientado a objetos , tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala.

Python usa tipado dinámico y una combinación de recuento de referencias y un recolector de basura con detección de ciclos para la gestión de la memoria . También cuenta con resolución dinámica de nombres como; enlace tardío, que enlaza los nombres de métodos y variables durante la ejecución del programa.

En lugar de tener todas sus funciones integradas en su núcleo, Python fue diseñado para ser altamente extensible con módulos. Esta modularidad compacta lo ha hecho particularmente popular como un medio para agregar interfaces programables a aplicaciones existentes.

La visión de Van Rossum de un lenguaje básico pequeño con una gran biblioteca estándar y un intérprete fácilmente extensible surgió de sus frustraciones con ABC , que abrazó el enfoque opuesto.

Python se esfuerza por lograr una sintaxis y una gramática más simples y menos desordenadas, al tiempo que ofrece a los desarrolladores una opción en su metodología de codificación. Python adopta la filosofía de diseño de "debe haber una, y preferiblemente sólo una, forma obvia de hacerlo".

Alex martelli , miembro de la Python Software Foundation y autor del libro de Python, escribe que "describir algo como 'inteligente' no se considera un cumplido en la cultura Python".

La gran biblioteca estándar de Python, comúnmente citada como una de sus mayores fortalezas, proporciona herramientas adecuadas para muchas tareas. Para las aplicaciones orientadas a Internet, se admiten muchos formatos y protocolos estándar, como MIME y HTTP . Incluye módulos para crear interfaces gráficas de usuario, conectarse a bases de datos relacionales, generar números pseudoaleatorios, aritmética con decimales de precisión arbitraria, manipular expresiones regulares y pruebas unitarias .

El desarrollo de Python se lleva a cabo en gran parte a través del proceso de Propuesta de mejora de Python (PEP), el mecanismo principal para proponer nuevas características importantes, recopilar la opinión de la comunidad sobre problemas y documentar las decisiones de diseño de Python. El estilo de codificación de Python se trata en PEP. La comunidad de Python y el consejo directivo revisan y comentan los PEP sobresalientes.

La mejora del lenguaje corresponde al desarrollo de la implementación de referencia CPython. La lista de correo python-dev es el foro principal para el desarrollo del lenguaje. Los problemas específicos se discuten en el rastreador de errores de Roundup alojado en bugs.python.org . Desarrollo tuvo lugar

originalmente en un auto-organizado repositorio de código de origen que ejecuta Mercurial , hasta que se trasladó a Python GitHub en enero de 2017.

Los lanzamientos públicos de CPython vienen en tres tipos de versiones, a continuación les vamos a decir cada uno de ellos.

1.- Versiones incompatibles con versiones anteriores, donde se espera que el código se rompa y deba ser transferido manualmente. Se incrementa la primera parte del número de versión. Estos lanzamientos ocurren con poca frecuencia: la versión 3.0 se lanzó 8 años después de la 2.0.

2.- Lanzamientos importantes o de "características", ocurrieron aproximadamente cada 18 meses, pero con la adopción de una cadencia de lanzamiento anual que comienza con Python 3.9, se espera que sucedan una vez al año, Son en gran parte compatibles pero introducen nuevas características. La segunda parte del número de versión se incrementa. Cada versión principal es compatible con correcciones de errores durante varios años después de su lanzamiento.

3.-Las versiones de corrección de errores, que no introducen nuevas características, ocurren aproximadamente cada 3 meses y se realizan cuando se ha corregido una cantidad suficiente de errores desde la última versión. Las vulnerabilidades de seguridad también se reparan en estas versiones. Se incrementa la tercera y última parte del número de versión

file:///C:/Users/user/Downloads/reference%20(1).pdf

[LIBRO] [La referencia del lenguaje Python](#)

Metodología de Trabajo.

Lean: está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando así a un segundo plano el tiempo y los costes. El aprendizaje, las reacciones rápidas y potenciar el equipo son fundamentales.

Se escogió dicha metodología ya que a diferencia de las metodologías tradicionales contamos con poco tiempo, además, el diseño de la interfaz será un diseño sencillo por el tiempo antes mencionado.

Además, que en las metodologías ágiles se ubican 4 etapas que son:

- **Planteamiento**
- **Requerimientos**
- **Iteración(es)**
- **Puesta en marcha**



a diferencia de una metodología tradicional la cual consta de 6 etapas, las cuales son:

- **Planteamiento**
- **Análisis**
- **Diseño**
- **Programación**
- **Pruebas**
- **Puesta en marcha**



por consiguiente, fue que se escogió **Lean** como metodología para el proyecto.

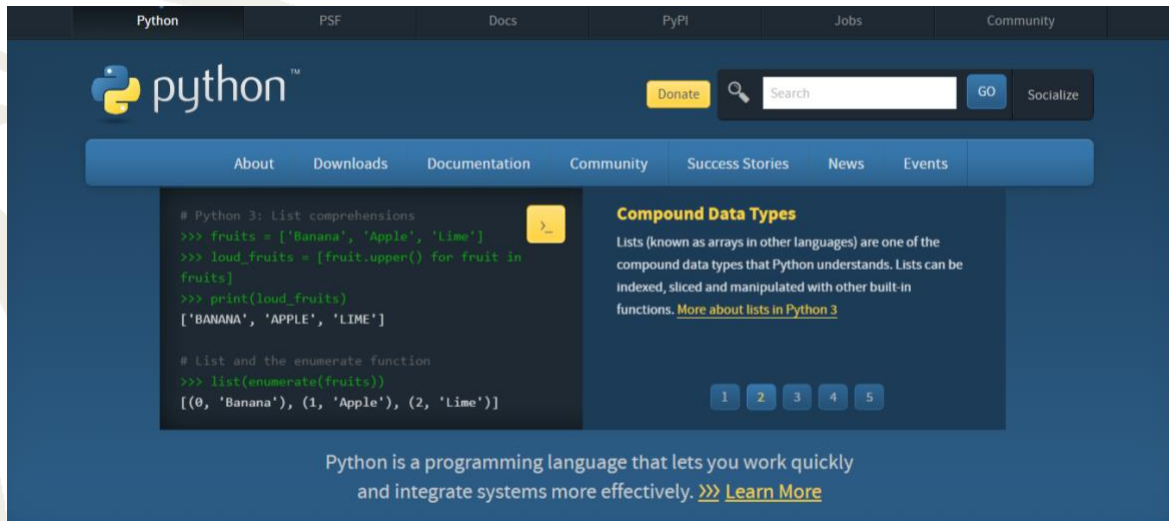
Fuente: <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html>

Desarrollo e implementación.

1.- Primero debemos de instalar Python en la página oficial después damos en el apartado download y descargamos la versión mas actual

Liga para descargar python

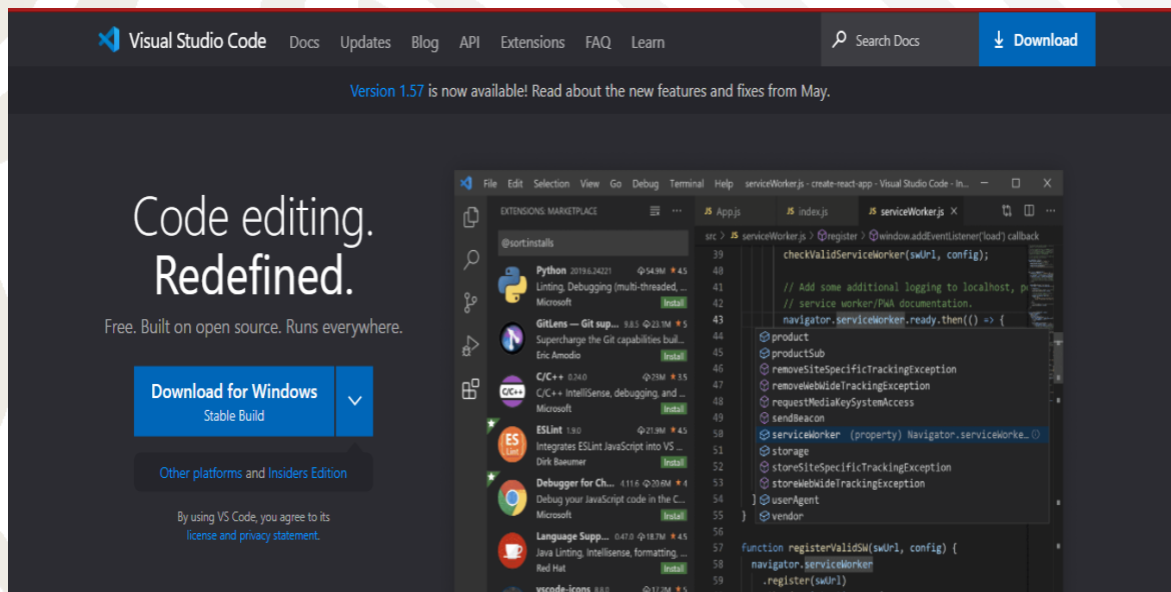
<https://www.python.org/>



2.- Instalamos Python con la variable de entorno y poder instalar las librerías que necesitamos como CV2 desde la terminal de Windows

El editor de código que utilizaremos en Visual Studio Code

<https://code.visualstudio.com/>



3.-Checamos la version de python instalada y procedes a instalar la paqueteria de open CV.

```
C:\Users\1043998>python
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pip install cv2
```

4.-Generaremos un nuevo archivo con la extension .py en el cual capturaremos pimero todos los rostros. Una vez instaladas las librerías la importamos en nuestro editor de codigo

```
import imutils
import os
import cv2
```

5.- empezaremos poniendo el nombre del data(en este caso sera el nombre de la persona de la cual vamos a capturar las imágenes para que posteriormente las reconozca), tambien vamos a introducir la ruta donde se guardaran dichos archivos.

```
personName = 'ivan 161080215'
# ruta donde se almaceno el Data
dataPath = 'C:/Users/1043998/Desktop/Reconocimiento/rostros'
personPath = dataPath + '/' + personName
```

6.-en este punto crearemos la carpeta con el nombre de la persona que habiamos puesto anteriormente,en la cual se van a guargar todos los archivos .jpg, asi como tambien podremos de donde va a agarrar esas imágenes.

```
if not os.path.exists(personPath):
    print('Carpeta creada: ', personPath)
    os.makedirs(personPath)

#cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)else:
cap = cv2.VideoCapture('ivan.mp4')

faceClassif = cv2.CascadeClassifier(
    cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
count = 0
```


7.-con las siguientes líneas de código lo que haremos, es que se ejecutara el video y hara el proceso de identificación de rostros y los ira guardando uno por uno, dependiendo de cuantas imágenes querramos guardar sera el numero que pondremos en el contador.

```
while True:

    ret, frame = cap.read()
    if ret == False:
        break
    frame = imutils.resize(frame, width=640)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = frame.copy()

    faces = faceClassif.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        rostro = auxFrame[y:y+h, x:x+w]
        rostro = cv2.resize(rostro, (150, 150),
            interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count), rostro)
        count = count + 1
    cv2.imshow('frame', frame)

    k = cv2.waitKey(1)
    if k == 27 or count >= 200:
        break

cap.release()
cv2.destroyAllWindows()
```

Entrenando

8.-Después creamos un nuevo archivo llamado entrenando con la extensión .py, e igualmente llamamos las librerías.

La librería numpy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales.

```
import cv2
import os
import numpy as np
```


9.- ponemos la extension donde se encuentra el data y posteriormente listamos las carpetas dentro de esta, crearemos un array vacio llamado labels, y otro llamado facesData, y ponemos un contador llamado label=0.

```
dataPath = 'C:/Users/1043998/Desktop/Reconocimiento/rostros'
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0
```

10.- creamos el metodoeigenfaces, el cual es un entrenamiento de prediccion, el metodo asume que todas las imágenes deben estar en escala de grises y hace el test para ver que todas las imágenes esten del mismo tamaño.

```
for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName,0))

    label = label + 1
```

11.- los siguientes comandos crearan un archivo.xml, el cual almacenara el reconocedor de rostros.

```
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

print("Entrenando ... ")
face_recognizer.train(facesData, np.array(labels))

face_recognizer.write('modeloLBPHFace.xml')
print("Modelo almacenado ... ")
```

12.-Finalmente creamos un tercer archivo con la extensión .py, y de igual manera lo vamos a llamar las librerías de openCV.

```
import cv2
import os
```

13.vamos a especificar la ruta donde se encuentran las carpetas de la(s) persona(s) que vamos a reconocer y las listaremos para obtener únicamente sus nombres.

```
dataPath = 'C:/Users/1043998/Desktop/Reconocimiento/rostros'
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)
```

14.-vamos a llamar al método eigenfaces_recognizer como en el anterior archivo, posteriormente leeremos el modelo que recientemente almacenamos (el archivo.xml que se creó), luego proseguiremos con el detector de rostros.

```
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Leyendo el modelo

face_recognizer.read('modeloLBPHFace.xml')

cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
cap = cv2.VideoCapture('ivan.mp4')

faceClassif =
cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
```

15.- en este punto ya estaremos corriendo el reconocedor de rostros, debemos tomar en cuenta que debemos redimensionar a los pixeles que pusimos a escala de girses en el primer archivo, ya que fue mismo que utilizamos en el reconocedor de rostros.

```
while True:
    ret, frame = cap.read()
    if ret == False: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = gray.copy()

    faces = faceClassif.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        rostro = auxFrame[y:y+h, x:x+w]
        rostro = cv2.resize(rostro, (150, 150), interpolation=
cv2.INTER_CUBIC)
        result = face_recognizer.predict(rostro)

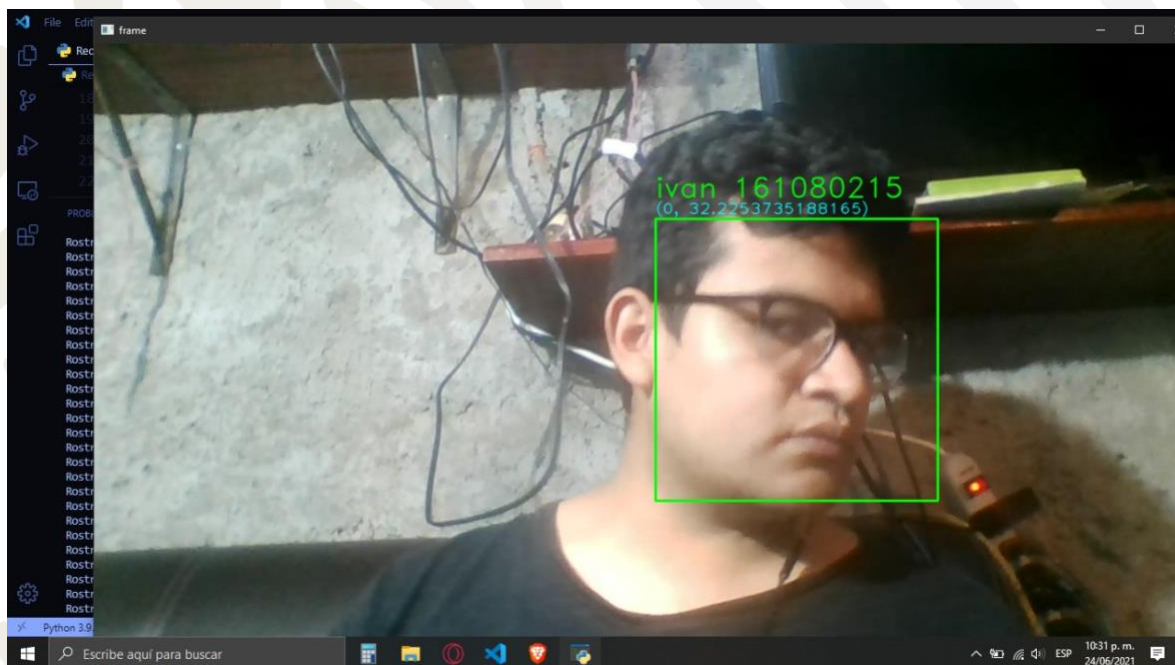
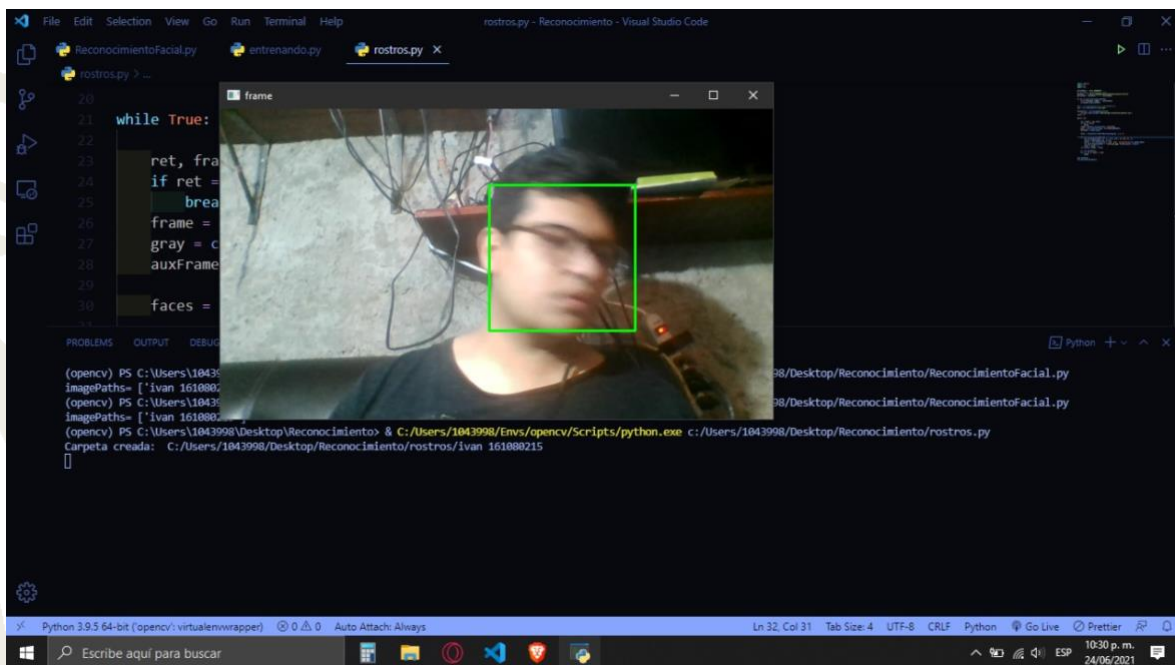
        cv2.putText(frame, '{}'.format(result), (x, y-5), 1, 1.3,
(255, 255, 0), 1, cv2.LINE_AA)
        ...
```

```
if result[1] < 70:
    cv2.putText(frame, '{}'.format(imagePaths[result[0]]), (x, y-
25), 2, 1.1, (0, 255, 0), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
else:
    cv2.putText(frame, 'Desconocido', (x, y-20), 2, 0.8,
(0, 0, 255), 1, cv2.LINE_AA)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)

cv2.imshow('frame', frame)
k = cv2.waitKey(1)
if k == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

Resultados



Conclusiones.

Gillot de Leon Diana Guadalupe

Esta tecnología ha existido durante décadas, pero su uso se ha vuelto más notable y accesible en los últimos años, ya que ahora se utiliza con soluciones innovadoras, como aplicaciones de fotos personales y autenticación secundaria en dispositivos móviles. Para comprender estas capacidades novedosas.

El reconocimiento facial resulta útil en numerosas aplicaciones e industrias. Hoy en día, esta tecnología ayuda a identificar a cualquier tipo de personas en diferentes áreas de la industria.

Hernandez Diaz Mitzi Mabel

La realización del proyecto, me ha servido para darme cuenta de que la Inteligencia Artificial no es algo nuevo, lleva décadas de estudio y está en constante evolución. Mediante la realización del trabajo del reconocimiento facial, ah sido para darnos cuenta que puede ver mucha seguridad, no tanto para el ITIZ. Si no simplemente para otras Instituciones o empresas que puede llegar hacer un proyecto capaz de tener una mejor imagen y sobre todo a reducir el tiempo, como evitar cualquier robo. Sab

Sabemos que la mayoría de la gente, al hablar de inteligencia artificial tiende a relacionarlo con el mundo de la robótica y, más concretamente a los robots con formas humanas, capaces de relacionarse. Pues en este caso al trabajar con PYTHON nos abre muchas puertas a muchas dependencias con este proyecto, porque sabemos que Pyhton se está manejando con la inteligencia Artificial

Martinez Rodriguez Gerardo Emmanuel

Sin duda alguna el realizar este proyecto cabio mi forma de ver las cosas, ahora sabemos que la Inteligencia Artificial esta dia a dia en nuestras vidas, desde las aplicaciones que usamos dia a dia como lo es Facebook, Instagram, etc., hasta el desarrollo de una super infraestructura como lo son las fotomultas, tambien me ayudo a enter los procesos del Reconocimiento Facial, ya que muchos pensamos que es igual la Deteccion Facial y el Reconocimiento, a su vez me ayudo a experimentar con librerias que no conocia, tambien me ayudo a experimentar y adentrarme mas en mi caso en mi SO, ya que uso OSX, y eso implica mayores desafios para mi como persona, porque algunas librerias no se ejecutan ni se intalan igual, en concusion me agrado trabajar con este proyecto porque en base a este ahora puedo crear una BD y configurar este proyecto a una camara en mi casa.

Ramirez Peña Carlos Ivan



La gran problemática que debemos enfrentarnos hoy en día, es la seguridad ya que vivimos en un país peligroso la cual no respeta situación económica y la universidad del itiz no queda exenta de la delincuencia ya se ha dado accidentes. La cual se optado en dale la oportunidad a la tecnología en especifico la inteligencia artificial con reconocimiento facial el cual permitió detectar el a los estudiantes y personal del itiz y evita que personas ajenas al itiz no entren a la institución gracias a base de conocimiento que tiene el software.

Esto beneficiara al itiz en cuanto a la seguridad y apoyara a comunidad a mejorar la seguridad, mejora la vigilancia de una manera más eficiente, tiene un mejor control de las personas que acceden al itiz,

Fuentes de Informacion.

<https://www.delftstack.com/es/howto/python/how-to-access-environment-variables-in-python/#:~:text=Las%20variables%20de%20entorno%20son,un%20programa%20Python%20desde%20dentro.>

<https://opencv.org/>

<https://es.wikipedia.org/wiki/OpenCV#:~:text=La%20misi%C3%B3n%20de%20OpenCV%20es,en%20la%20facilidad%20de%20uso.>

<https://www.esic.edu/rethink/tecnologia/para-que-sirve-python>

<https://unpocodejava.com/2013/10/09/que-es-opencv/>

http://soporte.tvc.mx/Ingenieria/DAHUA/ARCHIVOS_COMUNES/Diferencia%20entre%20Face%20Detecti%C3%B3n%20y%20Face%20Recogniti%C3%B3n%20Dahua.pdf

<http://aprendiendoingenieria.es/instalar-opencv-con-visual-studio-2017/>

<https://www.andreasjkl.com/basics-of-ar-anchors-keypoints-feature-detection/pip-install-opencv-python/>

<https://geekytheory.com/configurar-visual-studio-code-programar-python>

<https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>

https://es.wikipedia.org/wiki/Inteligencia_artificial