Sistema para la automatización del registro de asistencia en las aulas de clases

System for the automation of the attendance record in the classrooms

Jeisson Paredes[1], Edwar González[2], Jahir Gamar Castillo-Santamaria[3], Lilia Muñoz[4], Vladimir Villarreal[5]

Paredes, J; González, E; Castillo-Santamaria, J.G; Muñoz, L; Villarreal, V. Sistema para la automatización del registro de asistencia en las aulas de clases. *Tecnología en Marcha*. Vol. 36, número especial. Octubre, 2023. V Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software, Salud Electrónica y Móvil. Pág. 83-93.

https://doi.org/10.18845/tm.v36i8.6930

1 Universidad Tecnológica de Panamá. Panamá.
Correo electrónico: jeisson.paredes@utp.ac.pa
https://orcid.org/0000-0002-2818-5815
2 Universidad Tecnológica de Panamá. Panamá.
Correo electrónico: edwar.gonzalez@utp.ac.pa
https://orcid.org/0000-0001-7906-7272
3 Universidad Tecnológica de Panamá, Panamá.
Correo electrónico: jahir5656@gmail.com
https://orcid.org/0000-0002-4693-8673
4 Universidad Tecnológica de Panamá. Panamá.
Correo electrónico: lilia.munoz@utp.ac.pa
https://orcid.org/0000-0002-4011-2715
5 Universidad Tecnológica de Panamá.
Correo electrónico: vladimir.villarreal@utp.ac.pa
https://orcid.org/0000-0003-4678-5977

Palabras claves

Python; reconocimiento facial; asistencia; Excel; aprendizaje profundo.

Resumen

Este proyecto fue realizado con el objetivo de desarrollar un programa que implementara reconocimiento facial, utilizando como lenguaje de programación Python con la capacidad de reconocer a los estudiantes o profesores que ingresan al salón de clases. Sólo colocando su rostro frente a la Webcam, en la pantalla se mostrará un recuadro azul alrededor del rostro del estudiante y mostrará su nombre. El programa almacenará un registro en un documento Excel de aquellas personas que asistieron a clases. En este documento se encontrarán los nombres de los estudiantes y las fechas de la semana. El programa colocará un 1 si asistió y un 0 si no se presentó a la clase, de esta forma se busca evitar el uso de papel y maximizar el tiempo de clases, ya que algunos profesores toman una parte de la clase para crear sus registros de asistencia, este proceso puede tomar más o menos tiempo dependiendo de la cantidad de estudiantes matriculados en el curso.

Keywords

Python; facial recognition; attendance; Excel; deep learning.

Abstract

This project was carried out with the objective of developing a program that would implement facial recognition using Python as a programming language with the ability to recognize students or teachers who enter the classroom. Just by placing your face in front of the Webcam, the screen will display a blue box around the student's face and display their name. The program will store a record in an Excel document of those people who attended classes. In this document you will find the names of the students and the dates of the week. The program will place a 1 if you attended and a 0 if you did not show up to class, in this way it seeks to avoid the use of paper and maximize class time since some teachers take part of the class to create their attendance records. This process may take more or less time depending on the number of students enrolled in the course.

Introducción

Actualmente, no existe una forma establecida para registrar la asistencia en los salones de clases, por lo que cada profesor opta por seguir su método. Algunos realizan formularios en los que los alumnos coloquen sus datos, otros no lo ven como algo importante y simplemente no realizan ningún tipo de registro de asistencia.

Aplicando el reconocimiento de patrones se puede desarrollar un sistema que sea capaz de automatizar el registro de asistencia tanto como de estudiantes como de profesores, con ello, se estaría evitando la utilización innecesaria de papel y optimizando el tiempo dedicado para las clases.

Se inicio este trabajo desarrollando una revisión de artículos que guardan relación con la solución que se plantea; uno de los proyectos similares fue el desarrollado por estudiantes de la Universidad Regional Autónoma de los Andes, Ecuador quienes desarrollaron un trabajo que consistía en la utilización del lenguaje de programación *Python* para el desarrollo del programa implementado en una *Raspberry Pi 4*, con un módulo cámara de 5 megapíxeles, que permitía

adquirir imágenes en alta definición. Se utilizó *MySQL* como motor de la base de datos y se desarrolló un sitio web implementando *HTML, CSS y JavaScript* en el cual se podía acceder para la revisión del registro de asistencia [1]. En la Universidad del Norte (Colombia), División de Ingenierías, Departamento de Ingeniería Eléctrica y Electrónica se diseñó un sistema constituido por una interfaz Web y una red neuronal mediante el uso de *Django y Python*, la cual es capaz de permitir a un usuario poder ingresar fotos de un grupo cualquiera y poder generar, mediante reconocimiento facial, una lista de las personas que asistieron y no asistieron a dicha reunión del grupo [2].

En la Universidad Peruana de Ciencias Aplicadas, Facultad de Ingeniería en el Programa Académico de Ingeniería de Sistemas de Información, se presentó el proyecto para el desarrollo de un modelo tecnológico que tiene como objetivo la identificación de pacientes mediante un servicio cognitivo de reconocimiento facial en *Cloud Computing* para la necesidad que tienen los sectores de salud de prevenir la suplantación de identidad [3].

Por otro lado, en la Universidad Católica de Colombia, Facultad de Ingeniería Electrónica y Telecomunicaciones en Bogotá 2019, se desarrolló un proyecto utilizado *Raspberry Pi 3 modelo B+,* una cámara *full HD Ecam* 8000 y una pantalla táctil de 3.5 pulgadas adherida a la placa de desarrollo. En cuanto al software se usó Linux distribución debían (Raspbian) para Raspberry pi, Python como lenguaje de programación junto con *OpenCV* para la manipulación de las imágenes. Con estas herramientas se desarrolló el Sistema de reconocimiento facial, el cual permitía el control de acceso a una casa [4].

Nuestra propuesta consiste en desarrollar un sistema autónomo que sea capaz registrar la asistencia de un salón de clase mediante el reconocimiento de patrones para analizar los rostros de los estudiantes o profesores que asisten al salón de clases, desarrollando un programa con el lenguaje de programación *Python*, utilizaremos *librerías para la* visión artificial y las implementaremos para comparar los patrones de los rostros capturados por la cámara en el momento en el que la persona ingresa al salón, con los patrones almacenados, identificando así si el alumno asistió a la clase, generando al final un registro de los alumnos o profesores que asistieron.

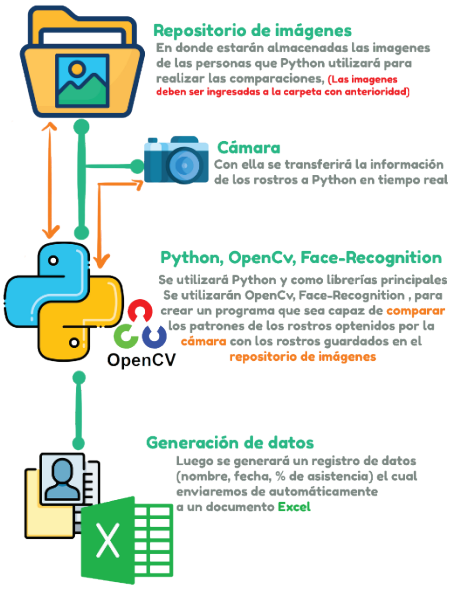


Figura 1. Diagrama de funcionalidad.

En la figura 1 se observa el diagrama de funcionalidad del proyecto el cual cuenta con 4 fases, a continuación, las explicamos.

La primera fase del diagrama consiste en llenar el repositorio de imágenes. Para el proyecto se utilizaron fotos de pruebas de tres compañeros, estas imágenes son utilizadas por *Python* para hacer comparaciones de patrones. En la segunda fase se utiliza una cámara web, la cual funciona como los ojos del programa, ella estará registrando quien entra al salón.

Luego entra en juego el programa en Python que tomará la información generada por la cámara en la segunda fase, para esto utiliza las librerías *OpenCV* y *Face-Recognition* para hacer comparaciones con las imágenes guardadas en el repositorio.

Por último, el programa generará datos que luego se enviarán a un documento de Excel para crear el registro de asistencia.

Materiales y metodología

Materiales por utilizar

Para el desarrollo de este proyecto se necesitan los siguientes materiales:

- Una cámara *Web* modelo L*ogitech c920,* también utilizamos la webcam integrada de la computadora, para las primeras pruebas.
- Computadora: plataforma (equipo) en el que se desarrollará y ejecutará el programa.
- *Python 3.7.0:* lenguaje de programación [5].
- Librerías *OpenCV [6] y face-recognition [10]:* para el realizar el reconocimiento de patrones, además se usaran otras como: *numpy [7], CMake [8], dlib [9], OpenPyXL [14].*
- *Visual Studio Code Como editor de Código [11].*
- *Microsoft Excel*: hoja de cálculo donde se almacenará el registro de asistencia.

Metodología

La primera fase para el desarrollo del proyecto consiste en la instalación de los requerimientos de este.

Se procede a instalar el desarrollo para escritorio con *C++* de *Visual Studio [12]* con el objetivo de utilizar el compilador de *C++, esto para poder implementar CMake y dlib.* Luego se instalan las librerías: *CMake, dlib, face- recognition, numpy y OpenCV.*

Una vez se adquiere todo lo que necesita comienza la fase codificación del prototipo, se utiliza como *IDE PyCharm y Python* en su versión 3.7.0

Primera versión del prototipo: en esta versión se trabajó con las imágenes sin una entrada externa, es decir se compararon imágenes de una carpeta sin recibir información de una cámara.

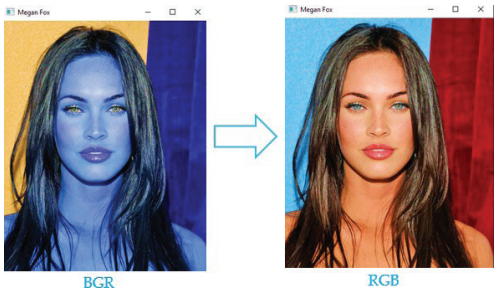


Figura 2. BGR RGB [13].

Se importan las imágenes del repositorio utilizando la librería *face-recognition*, se cambiaron sus formatos de *BGR* a *RGB* utilizando la función *cvtColor* de la librería *OpenCV.* En la figura 2, se puede observar como una imagen en formato BGR pasa ser RGB.

Una vez las imágenes están disponibles se empieza por localizar los rostros con la función *face_location,* la cual retorna cuatro valores que representan las coordenadas del rostro en la imagen.

Se codificaron los rostros con la función *face_encode,* la cual retorna 128 valores que luego serán comparados entre ellos para determinar si los rostros son iguales.

Se calcula el índice de parentesco con la función *face_distance,* que se utilizó para elegir el rostro con índice de parentesco más alto, ya que al momento de utilizar un gran volumen de imágenes cabe la posibilidad de que existan rostros parecidos.

Para este parte del proyecto se obtienen los primeros resultados a nivel de terminal.

La figura 3 muestra la ejecución del código con dos imágenes de una sola persona, donde se obtiene un resultado correcto ya que reconoció que es la misma persona con un índice de parentesco de 0.42861203.

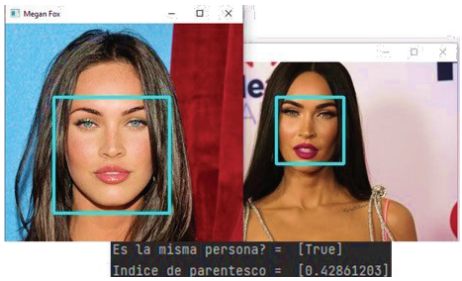


Figura 3. Primera Ejecución Caso Verdadero [15].

En la figura 4 se observa el caso contrario a lo anterior dicho, son dos personas diferentes con un índice de parentesco de 0.8315924

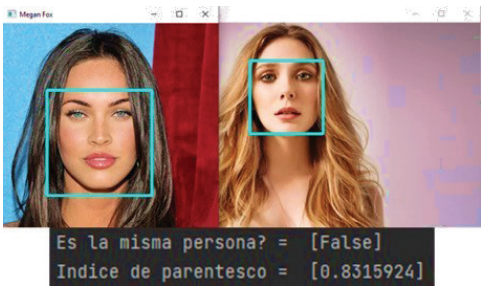


Figura 4. Primera Ejecución Caso Falso [16].

En la versión final se implementa la entrada de información por una Webcam y se realiza la comparación de patrones en tiempo real. Este proceso se logró importando las imágenes, pero esta vez estableciendo un *path,* usando la librería *os* que permite usar funcionalidades dependientes del sistema operativo. Establecer un path permite analizar toda imagen que se encuentre en esa dirección, esto facilita el trabajo de modo que no se deba definir una variable para cada imagen, además de que no limita el número de imágenes. Luego de establecer el *path* se creó un ciclo para obtener los nombres e ingresarlos en un vector, al igual que las imágenes. Después se crea una función llamada codificador, que codifica los rostros de cada imagen y los ingresa a un vector, para luego compararlos con los patrones que se generen en tiempo real a través de la *Webcam.*

Por último, se utilizan la librería *OpenPyXL* y algunas de funciones (*Workbook, Font, PatternFill, numbers, Alignment*), estas funciones se utilizan para el manejo de Excel desde *Python*, con ellas se automatizó la creación del archivo.xlsx y se diseñó una plantilla beta que es capaz de registrar solamente una semana de clases, a través de decisiones se controla la inserción de los datos, 1 para indicar que estuvo presente y 0 para indicar ausencia.

Resultados

En la figura 5 se puede observar la función codificador la cual se encarga de rellenar una lista con todos los valores que se generan al codificar los rostros, la función básicamente recibe un vector con todas las imágenes ubicadas en el repositorio de imágenes y luego en un ciclo for se codifica cada rostro, por último se van agregando todos los valores de cada rostro en una lista, con la que luego se harán las comparaciones con los valores que se generen con los patrones obtenidos por la Webcam en tiempo real.

```
# FUNCIÓN PARA CODIFICAR LAS IMÁGENES
# mandamos las imagenes al la función
def codificador(imagenes):
    # vector para guardar los códigos
    listaCodificada = []

    for img in imagenes:

        # CONVERTIMOS EL FORMATO DE LAS IMGs DE BGR A RGB
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # Codifica el rostro
        codigo = face_recognition.face_encodings(img)[0]  # Codifica el rostro
        listaCodificada.append(codigo)

    # retorno un vector con todos los patrones de todas las imágenes
    return listaCodificada
```

Figura 5. Función Codificador.

Se muestra una parte de la función de control para la inserción de asistencia en la figura 6, concretamente hay una parte verdadera (cuando la fecha almacenada es igual a la fecha actual del sistema) y otra falsa (cuando la fecha actual es diferente a la última almacenada).

```
# SI LA FECHA EN EXCEL ES LA MISMA QUE LA ACTUAL
if str(fecha) == str( hoja.cell( hojaControl['B2'].value, hojaControl['C2'].value).value ):

    for fila in range(4, int(y) ):

        if  hoja.cell(fila, 2).value == nombre and hoja.cell(fila, hojaControl['C2'].value ).value != 1:
            hoja.cell(fila, hojaControl['C2'].value ).value = 1
            #centrado
            hoja.cell(fila, hojaControl['C2'].value ).alignment = Alignment(horizontal='center')
            lector.save('REGISTRO ASISTENCIA IA.xlsx')

else:
    #genero nueva ubicación para la fecha
    if hojaControl['B3'].value != 5:
        fecha = date.today()
        hoja.cell( hojaControl['B2'].value, int(hojaControl['C2'].value) + 1 ).value = str(fecha)
        hoja.cell( hojaControl['B2'].value, int(hojaControl['C2'].value) + 1 ).alignment = Alignment(horizontal='center')
        hoja.cell( hojaControl['B2'].value, int(hojaControl['C2'].value) + 1 ).font = Font( color = '008cff', bold=True )
        lector.save('REGISTRO ASISTENCIA IA.xlsx')

        #verifico ausentes
        ausentes(y , int(hojaControl['C2'].value) )
        hojaControl['C2'] = int(hojaControl['C2'].value) + 1
        lectorControl.save('Proyecto Final IA\data.xlsx')

    else:
        #verifico ausentes
        ausentes(y , int(hojaControl['C2'].value) )
```

Figura 6. Función de Inserción.

La parte verdadera se encargar de ingresar el 1 en la dirección (fila, columna), fila del nombre correspondiente y la columna de la fecha actual.

La falsa ingresa una nueva fecha en la siguiente columna siempre y cuando no hayan pasado los 5 días límites y coloca un 0 a cada estudiante que no asistió la fecha anterior.

Pruebas

Se realizaron distintas pruebas, utilizando tres estudiantes, a cada uno de ellos se les tomó una fotografía frontal del rostro, y se agregaron al repositorio de imágenes. Al ejecutar el programa se desplegará una ventana flotante *(cv2.imshow)* en la cual se mostrará la cámara, de esta manera comienza a reconocer a los estudiantes, remarcando con un cuadro azul los rostros e indicando sus nombres *(cv2.putText)*

En la figura 7 se muestra como el programa logra reconocer a uno de los estudiantes de los cuales se analizaron sus patrones faciales.

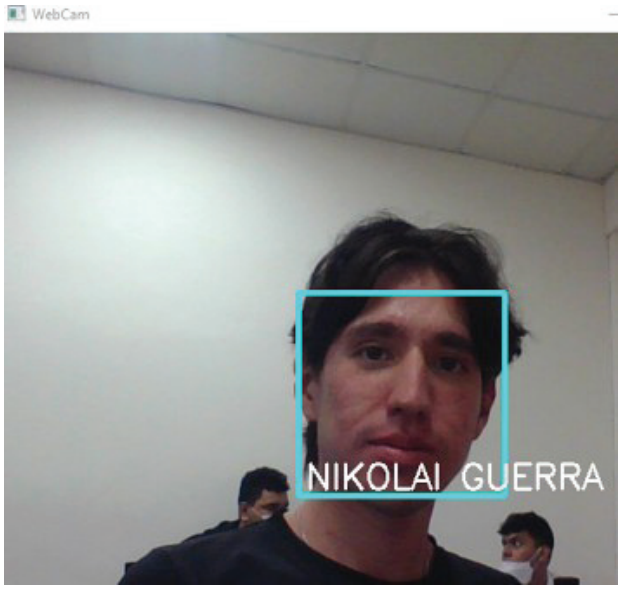


Figura 7. Reconocimiento facial de un estudiante.

En el cuadro.1 se puede observar las pruebas que se hicieron con los tres sujetos (estudiantes), en la columna de frontal se puede observar que el programa reconoció a todas las personas, esto era lo esperado, ya que las fotos eran en perspectiva frontal, por otra parte, se puede observar que en las perspectivas laterales el programa nunca fue capaz de reconocer a las personas.

Cuadro 1. Resultados de Ejecución.

		Frontal		Laterial Derecha		Lateral Izquierdo	
	Estudiante	Aciertos	Fallos	Aciertos	Fallos	Aciertos	Fallos
1	Nikolai Guerra	5	0	0	5	0	5
2	Jeisson Paredes	5	0	0	5	0	5
3	Edwar Gonzalez	5	0	0	5	0	5

Figura 8. Reconocimiento facial con complementos.

En la figura 8 se puede observar otros aspectos, como que el programa era capaz de reconocer a personas que contaban con la mascarilla un poco por debajo de la nariz, gorras y lentes.

Como trabajo a futuro se podría actualizar el programa de manera de que sea capaz de crear un Excel más sofisticado, es decir que el programa pueda crear un documento completo con cada semana del semestre en un *Sheet* diferente y que se muestre un resumen de datos con respectivos gráficos en cada una de ellas.

Por otra parte, se podría implementar el programa en Arduino o Raspberry para de esta manera lograr un prototipo más compacto, sin necesidad de utilizar una computadora para realizar la función. Además, de agregar fotos en las diferentes vistas de cada persona para que el programa no tenga limitaciones al momento de reconocer las personas, de cualquier ángulo.

En figura 9 se puede observar cómo durante la ejecución el programa no es capaz de reconocer a una persona por su perfil, este error se puede solucionar insertando fotografías de los laterales de los rostros al repositorio imágenes.

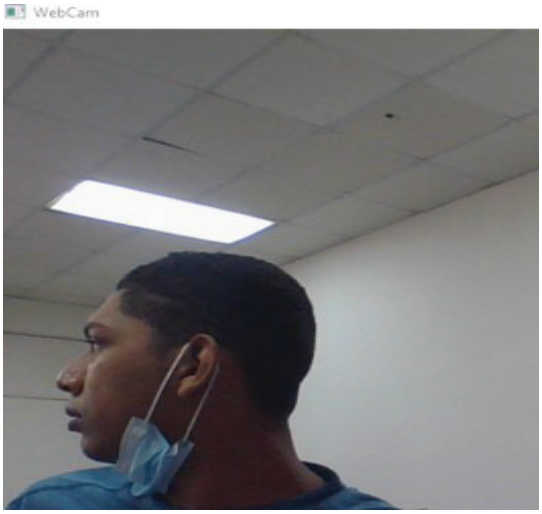


Figura 9. Reconocimiento Nulo de Perfil.

PLANTILLA DEL REGISTRO DE ASISTENCIA_BETA							
	NOMBRES	2022-06-6	2022-06-7	2022-06-8	2022-06-9	2022-06-10	%
1	Edwar Gonzalez	1	1	1	1	1	100%
2	Jeisson Paredes	1	0	1	0	0	40%
3	Nikolai Guerra	0	0	1	0	1	40%

Figura 10. Tabla de Registro de Asistencia.

En la figura 10 se puede observar el documento Excel automatizado luego de ejecutar el programa durante una semana de clases.

Conclusiones

En este proyecto se realizó un sistema automatizado para crear un registro de asistencia en las aulas de clases, los estudiantes solo necesitaran colocar su rostro frente a la cámara y el sistema generará un cuadro en el cual mostrará su nombre, confirmando así que ha sido identificado y siendo agregado a la lista de asistencia, para ello se necesita una cámara web y

una computadora con la cual se ejecutara el programa, disminuyendo el tiempo que se emplea en tomar la asistencia, esto en comparación con otros métodos utilizados como lo son: firmar una lista de asistencia o llamar por su nombre a cada estudiante.

Agradecimiento

A la Secretaria Nacional de Ciencia, Tecnología e Innovación de Panamá, por el financiamiento parcial para el desarrollo del proyecto. La Dra. Lilia Muñoz y el Dr. Vladimir Villarreal son miembros del Sistema Nacional de Investigación de la SENACYT.

Referencias

[1] Lara-Jacho Steven Luis Orlando Albarracín-Zambrano & Dionisio Vitalio Ponce-Ruiz, “Prototipo de reconocimiento facial para mejorar el control de asistencia de estudiantes en UNIANDES, Quevedo”, Revista Arbitrada Interdisciplinaria Koinonía, agosto de 2020. Accedido el 14 de mayo de 2022. [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=7608931

[2] D. A. Caro Barranco y A. C. López Roncancio, “Sistema inteligente para el registro de asistencia basado en procesamiento digital de imágenes y redes neuronales convolucionales”, UniNorte, octubre de 2018. Accedido el 14 de mayo de 2022. [En línea]. Disponible: https://manglar.uninorte.edu.co/handle/10584/8485#page=1

[3] L. M. Arroyo D. Alonso B. Rivera and M. Humberto, “Modelo tecnológico de reconocimiento facial para la identificación de pacientes en el sector salud”, Repositorio Académico UPC, 2019. Accedido el 14 de mayo de 2022. [En línea]. Disponible: https://repositorioacademico.upc.edu.pe/handle/10757/648832

[4] Alonso-Sierra J. D. & Castaño-Saavedra D. L., “Sistema de reconocimiento facial para control de acceso a viviendas”, Repositorio Institucional Universidad Católica de Colombia, 2019. Accedido el 14 de mayo de 2022. [En línea]. Disponible: https://hdl.handle.net/10983/24032

[5] Python Software Foundation. “Download python”. Python.org. https://www.python.org/downloads/ (accedido el 1 de junio de 2022).

[6] OpenCV team. “Releases - OpenCV”. OpenCV. https://opencv.org/releases/ (accedido el 1 de junio de 2022).

[7] NumPy. “NumPy”. NumPy. https://numpy.org (accedido el 1 de junio de 2022).

[8] CMake. “Download | CMake”. CMake. https://cmake.org/download/ (accedido el 1 de junio de 2022).

[9] “Dlib C++ library”. dlib C++ Library. http://dlib.net (accedido el 1 de junio de 2022).

[10] Python Software Foundation. “Face-recognition”. PyPI. https://pypi.org/project/face-recognition/ (accedido el 1 de junio de 2022).

[11] Microsoft. “Visual studio code - code editing”. Visual Studio Code. https://code.visualstudio.com (accedido el 11 de junio de 2022).

[12] Microsoft. “Descargar visual studio tools: Instalación gratuita para windows, mac, linux”. Visual Studio. https://visualstudio.microsoft.com/es/downloads/ (accedido el 1 de junio de 2022).

[13] “Megan Fox ya es mamá”. Atlántico. https://www.atlantico.net/articulo/gente/megan-fox-mama/20121020143109198639.html (accedido el 7 de junio de 2022).

[14] Openpyxl. “Openpyxl”. openpyxl. https://openpyxl.readthedocs.io/en/stable/ (accedido el 9 de junio de 2022).Megan fox”. Vía País. https://viapais.com.ar/via-libre/megan-fox- poso-entre-petalos-de-rosas-y-transparencias-siempre-quise-parecer- una-baraja-de-tarot/ (accedido el 10 de junio de 2022).

[15] Elizabeth olsen”. Pinterest. https://www.pinterest.com/pin/elizabeth- olsen--132011832816623967/ (accedido el 10 de junio de 2022).

[16] A. Geitgey. “Machine learning is fun! Part 4: Modern face recognition with deep learning”. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning- c3cffc121d78 (accedido el 10 de junio de 2022).

[17] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Sistema para la Automatización del Registro de Asistencia en las aulas de clases

Jeisson Paredes, Dariend Del Cid, Edwar Gonzalez
Licenciatura en Ingeniería de Sistemas y Computación, Facultad de Ingeniería de Sistemas Computacionales

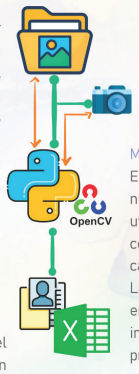
Resumen

Este proyecto fue realizado con el objetivo de desarrollar un programa que implementara reconocimiento facial utilizando como lenguaje de programación Python con la capacidad de reconocer a los estudiantes o profesores que ingresan al salón de clases. Sólo colocando su rostro frente a la Webcam, en la pantalla se mostrará un recuadro azul alrededor del rostro del estudiante y mostrará su nombre

Introduccion

Actualmente, no existe una forma establecida para registrar la asistencia en los salones de clases, por lo que cada profesor opta por seguir su método. Aplicando el reconocimiento de patrones se puede desarrollar un sistema que sea capaz de automatizar el registro de asistencia de los estudiantes.

Nuestra propuesta consiste en desarrollar un sistema autónomo que sea capaz registrar la asistencia de un salón, mediante el reconocimiento de rostros de los estudiantes.



Materiales y Método

Materiales

-Una cámara Web
-Python 3.7.0
-(Librerías) OpenCV, Face-recognition, entre otras
-Un IDE, para este caso Visual Studio Code
-Microsoft Excel

Método

En resumen se podría indicar que el desarrollo de nuestro sistema tomó dos fases, en la primera, utilizando OpenCv y face-recognition se desarrolló un código capaz de comparar, ubicar y reconocer rostros captados por una WebCam.

La segunda fase consistió en el manipulación de datos, en esta se crearon funciones que daban formato e insertaban datos en un documeto Excel en donde principalmente se podría ver el registro de asistencia semanal en cuestión.

Resultados

El prototipo fue capaz de reconocer a los estudiantes en el 100% de los casos cuando los estudiantes se colocaban frontalmente a la cámara, en los casos en que los estudiantes se colocaban de perfil el reconocimiento fue nulo ya que en el repositorio de imágenes sólo hay imágenes en perspectiva frontal.

Durante la ejecución se observaron otros aspectos, el programa era capaz de reconocer a personas que contaban con la mascarilla un poco por debajo de la nariz, gorras y lentes.

Resultados de Pruebas

		frontal		Laterial Derecha		Lateral Izquierdo	
	Estudiante	Aciertos	Fallos	Aciertos	Fallos	Aciertos	Fallos
1	Nikolai Guerra	5	0	0	5	0	5
2	Jeisson Paredes	5	0	0	5	0	5
3	Edwar Gonzalez	5	0	0	5	0	5

Resultados del registro

REGISTRO DE ASISTENCIA DE UNA SEMANA

#	NOMBRES	LU 2022-6-21	MA 2022-6-22	MI 2022-6-23	JU 2022-6-24	VI 2022-6-25	SA 2022-6-26	DO 2022-6-27	%
1	Edwar Gonzalez		1		1				5%
2	Jeisson Paredes		1						3%
3	Nikolai Guerra								0%

SEMANA #1

Luego de ejecutar el programa durante 2 días de la semana, se insertaron los unos que indican asistencia en las celdas correspondientes en el documento que se generó en la primera ejecución.

Conclusión

En este proyecto se realizó un sistema automatizado para crear un registro de asistencia en las aulas de clases, los estudiantes solo necesitaran colocar su rostro frente a la cámara y el sistema generara un cuadro en el cual mostrara su nombre, confirmando así que ha sido identificado y siendo agregado a la lista de asistencia, para ello se necesita una cámara web y una computadora con la cual se ejecutara el programa, disminuyendo el tiempo que se emplea en tomar la asistencia, esto en comparación con otros métodos utilizados como lo son: firmar una lista de asistencia o llamar por su nombre a cada estudiante.

Agradecimiento

Le agradecemos a Adam Geitgey por su artículo "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning" el cual fue una base para la generación de este proyecto por otra parte le damos las gracias al profesor Vladimir Villarreal por guiarnos en el desarrollo del artículo, y a Nikolai Guerra por brindarse como sujeto de prueba para las ejecuciones.

Referencias

-A. Geitgey. "Machine learning is fun! Part 4: Modern face recognition with deep learning". https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78 (accedido el 10 de junio de 2022).
-Python Software Foundation. "Face-recognition". PyPI. https://pypi.org/project/face-recognition/ (accedido el 1 de junio de 2022).
-OpenCV team. "Releases - OpenCV". OpenCV. https://opencv.org/releases/ (accedido el 1 de junio de 2022).