

Sistema de reconocimiento facial para control de seguridad en el ingreso a las empresas.**Facial recognition system for security control at company entrances.****Sistema de reconhecimento facial para controle de segurança nas entradas da empresa.**

León León Ryan Abraham¹, Rodríguez Sanchez Melvin Jerson², Villegas Ponce Geisy Camila³, Honorio Escobedo Laly Maribel⁴, Ruiz Honorio Jhan Alcides⁵, Contreras Briceño Jimmy Alexis⁶

Resumen

El presente trabajo de investigación consiste en el desarrollo de un sistema que con base en la detección y reconocimiento facial pueda proporcionar seguridad a las empresas de forma rápida y segura, haciendo uso de una aplicación Python, la cual permite el reconocimiento facial de los trabajadores, y así mantener la seguridad de la empresa. Tras el desarrollo del software, el ordenador procesa las imágenes y videos que están almacenadas en las librerías, las cuales son transferidas a una detección y reconocimiento de rostro, el cual hace que se genere un código de reconocimiento facial desarrollado en el lenguaje de programación Python, el mismo que efectúa la comparación con los rostros almacenados en la base de datos, para finalmente producir un comando de aceptación en donde nos afirma si la usuario detectado se encuentra en la base de datos o no. Los resultados obtenidos, concluimos que se consiguió un diseño funcional, el cual hace un buen uso de la tecnología empleada, dicho diseño permitió identificar eficazmente a cada uno de los usuarios con los que se trabajó en la presente investigación, podemos finalizar diciendo que este sistema puede ser implementado en las empresas y en diversas aplicaciones.

Palabra clave: *Python, Reconocimiento facial, Sistema, Software.*

Abstract

This research work consists of the development of a system based on facial detection and recognition that can provide security to companies quickly and safely, using a Python application, which allows facial recognition of workers, and thus maintain the security of the company. After the development of the software, the computer processes the images and videos that are stored in the libraries, which are transferred to a face detection and recognition, which generates a facial recognition code developed in the Python programming language, which makes the comparison with the faces stored in the database, to finally produce an acceptance command where it states whether the detected user is in the database or not. The results obtained, we conclude that a functional design was achieved, which makes good use of the technology used, this design allowed to effectively identify each of the users with whom we worked in this research, we can conclude by saying that this system can be implemented in companies and in various applications.

Keyword: *Python, Facial recognition, System, Software.*

Resumo

Este trabalho de pesquisa consiste no desenvolvimento de um sistema baseado na detecção e reconhecimento facial que pode fornecer segurança às empresas de forma rápida e segura, fazendo uso de uma aplicação Python, que permite o reconhecimento facial dos trabalhadores, e assim manter a segurança da empresa. Após o desenvolvimento do software, o computador processa as imagens e vídeos armazenados nas bibliotecas, que são transferidos para uma detecção e reconhecimento facial, que gera um código de reconhecimento facial desenvolvido na linguagem de programação Python, que faz a comparação com os rostros armazenados no banco de dados, para finalmente produzir um comando de aceitação onde se declara se o usuário detectado está ou não no banco de dados. Os resultados obtidos, concluímos que foi alcançado um projeto funcional, que faz bom uso

¹ Escuela de Ingeniería Industrial. Magister. Universidad Privada del Norte. Trujillo, Perú. ryan.leon@upn.edu.pe. <https://orcid.org/0000-0002-0599-0141>

² Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo, Perú. N00176003@upn.pe. <https://orcid.org/0000-0002-3171-6613>.

³ Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo, Perú. N00185602@upn.pe. <https://orcid.org/0000-0002-0644-0723>.

⁴ Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo, Perú. N00180253@upn.pe. <https://orcid.org/0000-0002-4100-9138>.

⁵ Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo, Perú. N00171852@upn.pe. <https://orcid.org/0000-0002-4962-3415>.

⁶ Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo, Perú. N00069318@upn.pe. <https://orcid.org/0000-0001-5658-3637>.

da tecnologia utilizada, este projeto permitiu identificar efetivamente cada um dos usuários com os quais trabalhamos nesta pesquisa, podemos concluir dizendo que este sistema pode ser implementado em empresas e em várias aplicações.

Palavras-chave: *Python, Reconhecimento Facial, Sistema, Software.*

Introducción

Hoy en día en el Perú la tecnología va en desarrollo ascendente, esto debido a que las empresas con procesos mecánicos avanzan para automatizar cada aspecto de ellas, desde el proceso de elaboración de algún producto, el uso eficiente de energía, hasta la seguridad de la misma empresa. Debido a esto las empresas buscan invertir en tecnología para así desarrollar procesos más modernos. Es por eso que en la actualidad la seguridad va de la mano de la tecnología y que mejor que las empresas que hoy en día van creado distintos tipos de reconocimientos para sus trabajadores, ya sea como el reconocimiento facial, dactilar, etc. El sistema de detección y reconocimiento facial es utilizado por varios países europeos y asiáticos como China, Canadá, Estados Unidos entre otros países en vías de desarrollo. De esta manera se pretende disminuir la inseguridad ciudadana en todos sus aspectos. Permitiendo determinar con rapidez la identidad de una persona luego de capturar la imagen de su rostro y verificar características de su fisonomía comparándolas con las que están registradas en una base de datos.

Esto permite que las técnicas de identificación basadas en el reconocimiento facial ofrecen una solución de identificación robusta, dado que utiliza los rasgos de las personas. Los rasgos físicos pueden ser extraídos del ojo (iris, retina), el contorno de la cara, los labios, etc. De acuerdo con lo expuesto, en este trabajo de investigación se pretende desarrollar un sistema de reconocimiento de personas basado en el reconocimiento facial el cual use como sistema de seguridad y de identificación de los trabajadores para las empresas. Debido a la coyuntura provocada por el COVID-19, y pensando en la importancia que tiene actualmente la desinfección personal, la cámara usada para el reconocimiento facial debe ser instalada en una cabina de esterilización, para que el proceso sea a la par con la desinfección.

La biometría es la ciencia que analiza las características físicas (rostro, huellas digitales, iris de los ojos, etc.) o comportamientos (la voz, la manera de firmar, etc.) que cada individuo posee, los cuales pueden ser utilizados para identificarla o validar restricciones de acceso. (Marín, 2009). Según (Serratosa, 2008) un sistema biométrico debe tener una precisión y velocidad aceptable con un número de recursos razonable asimismo no puede ser nocivo para los usuarios y ser suficientemente seguro ante los métodos fraudulentos. El Reconocimiento facial te ayuda a la verificación, autenticación e identificación comparando la imagen de una cara desconocida con todas las imágenes de caras conocidas que se encuentran en la base de datos para determinar su identidad. (Moreano, 2017). Las principales técnicas de reconocimiento facial son la de métodos basados en imágenes fijas, métodos basados en imágenes 3d, métodos basados en vídeo y sistemas de reconocimiento en tiempo real implementaciones en hardware, cada uno de estos métodos trabajan de diferente manera, pero para la obtención de buenos resultados la iluminación, alineación, escalado que se use para realizar la selección de patrones debe ser buena. (Fuentes A., 2011).

Python fue creado a finales de los 80 y principios de los 90 por Guido van Rossum, quien lo desarrollo cuando se encontraba trabajando en el sistema operativo Amoeba, primariamente se concibe para manejar excepciones y tener interfaces con Amoeba como sucesor del lenguaje ABC, pero guido decidió crear uno nuevo y utilizarlo para poder superar los problemas con los que había encontrado en otros proyectos con ABC (Challenger-Pérez, 2014). Actualmente Python tiene un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo, dos de las características principales de Python son la de interpretado y multiplataforma, asimismo nos ofrece más facilidades para la programación orientada a objetivos. (Montoro, 2013)

Rodríguez Cárave (2020), en su tesis titulada “Creación de una aplicación de reconocimiento emocional en Python para Neuromarketing y Neuromanagement” para la obtención del título de grado de Ingeniería de las Tecnologías Industriales publicado por la Universidad de Sevilla, expone que se elaboró un modelo para detectar las emociones faciales de las personas realizando un experimento para comprobar su utilidad, mediante el entrenamiento de una red neuronal de programación Python. Además, se usó el software Open Source Ogama para el experimento con tres candidatos a tres puestos de trabajos, a quienes se les detectó las puntuaciones de sus emociones. Los resultados obtenidos mostraron la

predisposición de los sujetos para cada uno de los puestos de trabajo, facilitando la asignación a los puestos existentes en la empresa. En conclusión, el reconocimiento emocional puede brindar una variable esencial para tener en cuenta en la optimización de la asignación de los recursos dentro del entorno de la organización industrial y de otros entornos de las empresas.

Costa Mari (2020), en su tesis titulada “Análisis de un sistema de reconocimiento facial a partir de una base de datos realizado mediante python” para la obtención del título profesional de grado en Ingeniería Electrónica Industrial y Automática, publicado por la Universidad Politécnica de Catalunya, expone que este proyecto está basado en el análisis, tanto teórico como experimental, de un método de reconocimiento facial a raíz de una base de datos. Se realizó mediante el lenguaje de programación Python junto a la librería de machine learning y visión por computador OpenCV. El proceso de identificación de una cara se divide en dos subprocesos sucesivos. Respecto al primer subproceso que es la detección facial, el método usado se enfoca en el algoritmo de Viola-Jones basado en Haar Cascades. Con el segundo subproceso, el reconocimiento facial, los métodos utilizados en este proyecto son Fisherfaces, Eigenfaces y LBPH (Local Binary Pattern Histogram), de los cuales se ejecutó su código y un estudio detallado. De esta forma, se ha realizado una comparación de los tres métodos de reconocimiento facial para establecer su comportamiento bajo distintas condiciones.

Artola Moreno (2019) en su tesis titulada “Clasificación de imágenes usando redes neuronales convolucionales en Python” para la obtención del título profesional Grado en Ingeniería de las Tecnologías de Telecomunicación, publicado por la Universidad de Sevilla, expone que el objetivo del presente proyecto es el de analizar redes neuronales convolucionales aptas para aprender sin supervisión a partir de datos sin estructurar y el valor que tienen en el análisis de imágenes. Centrándonos en el objetivo de entrenar la red para alcanzar un alto porcentaje de precisión, sensibilidad y especificidad, se procesará una investigación de los lenguajes de programación más usados en el campo de la Inteligencia Artificial, bases de datos para explotar la información y el funcionamiento propio de las Redes Neuronales Convolucionales.

Esta investigación se realiza porque existe la necesidad de mejorar el nivel de eficiencia en el proceso de reconocimiento facial dentro de las empresas o alguna entidad. El tema propuesto en el proyecto permitirá encontrar soluciones de optimizaciones, generará una nueva técnica, se aportará en la solución de problemas de seguridad dentro de las empresas, permitirá disminuir el número de incidencias delincuenciales, etc. Esta técnica eficiente permitirá obtener un sistema de reconocimiento facial de apoyo en el monitoreo de reconocimiento de individuos ante una situación que quieran delinquir, etc.

Materiales y Métodos

Software

Tabla 1.

Especificaciones del software.

| Software | Descripción |
|----------|---|
| Python | Lenguaje de programación multiparadigma |

Hardware, referente al hardware solamente se ha utilizado un ordenador portátil con WebCam para la realización de todo el proyecto.

Tabla 2.

Especificaciones del hardware.

| Especificación | Propiedades |
|-------------------|---------------|
| Procesador | Intel Core i7 |
| Memoria RAM | 16 GB |
| Tipo de sistema | 64 bits |
| Sistema operativo | Windows |
| Pantalla | 15.6" |

El sistema de reconocimiento facial propuesto consta de las siguientes etapas:

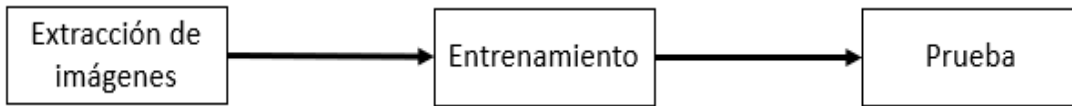


Figura 1. Etapas del desarrollo del software de reconocimiento facial.

Extracción de imágenes. En esta primera etapa se realizará la extracción de imágenes a partir de videos considerando la importación de cuatro librerías necesarias para la identificación de procesamiento a nivel de visión artificial, así mismo se iniciará con la especificación de la ruta donde se almacenaran las fotos de cada usuario que posteriormente serán utilizadas para la etapa de entrenamiento, todo esto con el fin de obtener las características visuales de las imágenes (iluminación, filtros, textura, colores, sombras, etc.) para así acelerar el desarrollo de la siguiente fase.

Entrenamiento. En esta etapa se ejecutará el entrenamiento al sistema utilizando las imágenes extraídas en la fase anterior, las cuales fueron pasadas a una escala de grises y a una propiedad de color para generar una uniformidad y mejoras en las imágenes, resaltando características que facilitan el análisis en la última etapa. Dentro del entrenamiento se cuenta con una lista de usuarios que serán analizados e identificados por el Software el cual recolectará las características suficientes de cada persona con el fin de almacenarlos.

Prueba. En esta etapa final se realizará las pruebas de testeo, seleccionando videos de cada una de las personas estudiadas con el fin de identificar de quien es el rostro de acuerdo a los parámetros y características guardadas en la detección y entrenamiento. A través del video es importante capturar el análisis para poder hacer una clasificación a través de la ponderación. Finalmente, si el sistema logra identificar al usuario, se muestra su nombre caso contrario mostrará la palabra “desconocido”.

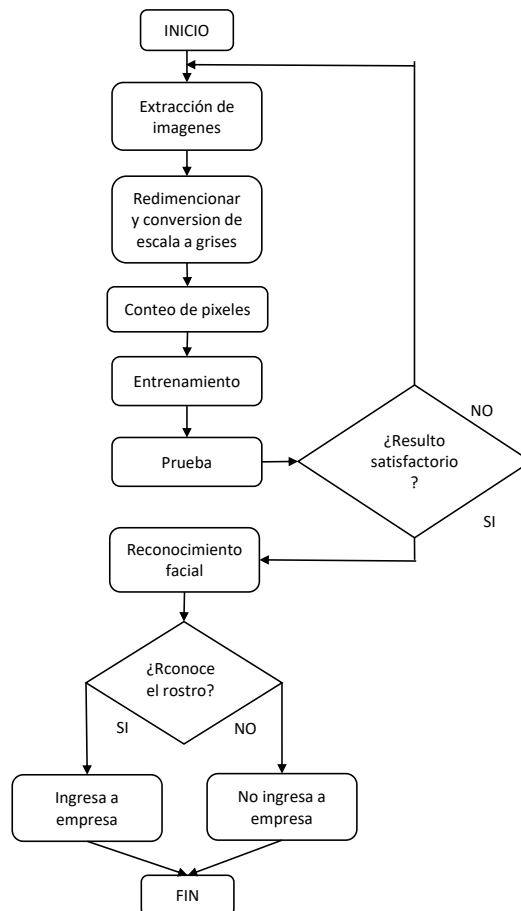


Figura 2. Flujograma del proceso de reconocimiento facial.

```
#extraer imagenes
dataPath = 'C:/Users/AQUINO/AppData/Local/Programs/Python/Python39/Rostros/Fotos'
capt = cv2.VideoCapture('Video.mp4')
count = 0

while True:
    ret, frame = capt.read()
    if ret == False: break
    cv2.imwrite(dataPath+'IMG_{}.jpg'.format(count),frame)
    count = count+1
    if count >= 1000:
        break
    capt.release()
"""
```

Figura 3. Código para extraer imágenes.

```
#entrenamiento
# camila jerson jhan jimy laly
dataPath = 'C:/Users/AQUINO/AppData/Local/Programs/Python/Python39/Rostros/Fotos_Train'

peoplelist = os.listdir(dataPath)
print('Usuarios: ',peoplelist)

labels = []
facesData = []
label = 0

for nameDir in peoplelist:
    personPath = dataPath + '/' + nameDir

    for file in os.listdir(personPath):
        # print('Rostros: ', nameDir + '/' + file)
        labels.append(label)
        #facesData.append(cv2.imread(personPath+'/'+file,0))
        #frame = cv2.imread(personPath+'/'+file,0)
        #frame = imutils.resize(frame, width=32)
        #facesData.append(frame)

        image = cv2.imread(personPath+'/'+file)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        facesData.append(image)

    label = label +1
```

Figura 4. Código para entrenamiento del software.

```
# prueba
videoTest = 'Video11.mp4'

peoplelist = os.listdir(dataPath)
print('Usuarios: ',peoplelist)

face_recognizer = cv2.face.LBPHFaceRecognizer_create()
face_recognizer.read('modeloLBPHFace.xml')

cap = cv2.VideoCapture(videoTest)

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

Figura 5. Código de prueba para el testeo del reconocimiento facial.

Resultados

Al finalizar la implementación del sistema propuesto, se ha obtenido los siguientes resultados de acuerdo con los objetivos planteados.

Tabla 3.

Especificaciones del software.

| Etapas | Respuesta esperada | Resultado |
|------------------------|-------------------------------------|-----------|
| Extracción de imágenes | Captura y visualización de imágenes | Ok |
| Entrenamiento | Entrenamiento del sistema | Ok |
| Prueba | Identificación de rostro | Ok |

Pruebas. Se realizó en base a 688 imágenes mediante la base de datos Python, a cada imagen se le aplico el método de detección de rostros cuyos los resultados para la persona 2 se muestran a continuación:



Figura 6. Detección de Rostro para la Persona 2.

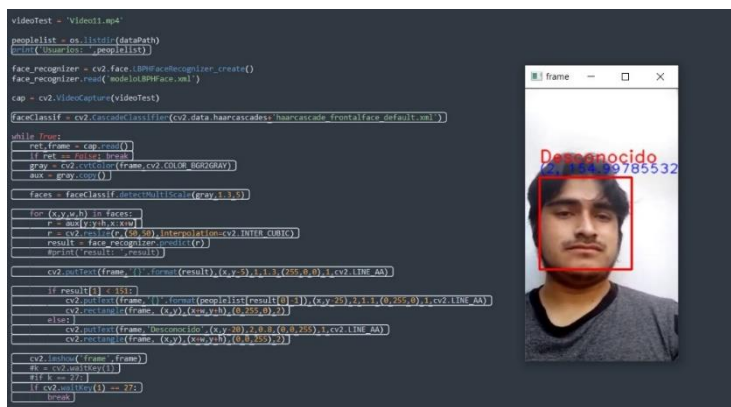


Figura 7. Fallo en la Detección de Rostro para la Persona 2.

Tabla 4.
Resultados de prueba.

| Prueba | Usuario | Nº de fotos | Video (s) | Reconoce (s) | No reconoce (s) | % error | % nivel de confianza | % Meta |
|--------|---------|-------------|-----------|--------------|-----------------|---------|----------------------|--------|
| 1 | Camila | 773 | 16 | 15.12 | 0.88 | 5.50% | 94.50 | 100 |
| | Jerson | 688 | 10 | 9.37 | 0.63 | 6.30% | 93.70 | 100 |
| 2 | Camila | 773 | 16 | 15.26 | 0.74 | 4.63% | 95.38 | 100 |
| | Jerson | 688 | 10 | 9.31 | 0.69 | 6.90% | 93.10 | 100 |
| 3 | Camila | 773 | 16 | 15.26 | 0.74 | 4.63% | 95.38 | 100 |
| | Jerson | 688 | 10 | 9.4 | 0.6 | 6.00% | 94.00 | 100 |
| 4 | Camila | 773 | 16 | 15.12 | 0.88 | 5.50% | 94.50 | 100 |
| | Jerson | 688 | 10 | 9.3 | 0.7 | 7.00% | 93.00 | 100 |
| 5 | Camila | 773 | 16 | 15.21 | 0.79 | 4.94% | 95.06 | 100 |
| | Jerson | 688 | 10 | 9.42 | 0.58 | 5.80% | 94.20 | 100 |
| 6 | Camila | 773 | 16 | 15.12 | 0.88 | 5.50% | 94.50 | 100 |
| | Jerson | 688 | 10 | 9.31 | 0.69 | 6.90% | 93.10 | 100 |
| 7 | Camila | 773 | 16 | 15.33 | 0.67 | 4.19% | 95.81 | 100 |
| | Jerson | 688 | 10 | 9.04 | 0.96 | 9.60% | 90.40 | 100 |
| 8 | Camila | 773 | 16 | 14.98 | 1.02 | 6.38% | 93.63 | 100 |
| | Jerson | 688 | 10 | 9.37 | 0.63 | 6.30% | 93.70 | 100 |
| 9 | Camila | 773 | 16 | 15.03 | 0.97 | 6.06% | 93.94 | 100 |
| | Jerson | 688 | 10 | 9.08 | 0.92 | 9.20% | 90.80 | 100 |
| 10 | Camila | 773 | 16 | 15.02 | 0.98 | 6.13% | 93.88 | 100 |
| | Jerson | 688 | 10 | 9.2 | 0.8 | 8.00% | 92.00 | 100 |

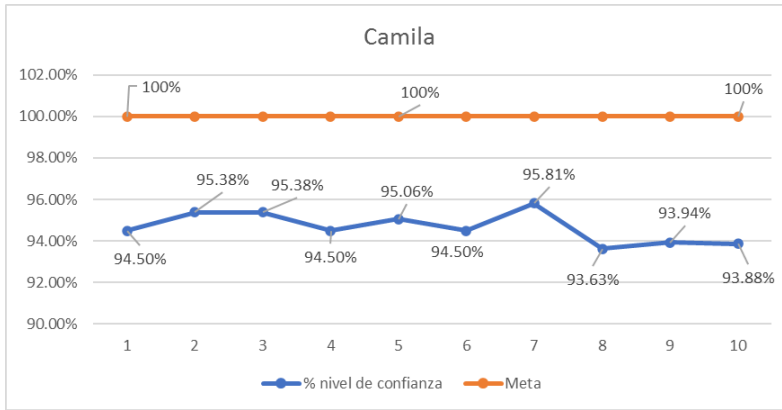


Figura 8. Efectividad del reconocimiento del usuario 1.

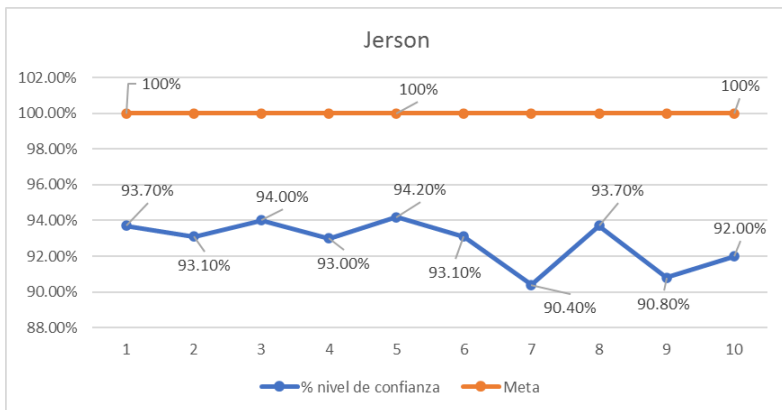


Figura 9. Efectividad del reconocimiento del usuario 2.

A continuación, podemos observar el porcentaje de aciertos al realizar la búsqueda para cada persona, utilizando imágenes frontales.

Tabla 5.

Porcentaje de error vs Efectividad

| | Usuario | Reconoce (s) | No reconoce (s) | % error | % nivel de confianza |
|----------|---------|--------------|-----------------|---------|----------------------|
| Promedio | Camila | 15.145 | 0.855 | 5.34 | 94.66 |
| | Jerson | 9.280 | 0.720 | 7.20 | 92.80 |

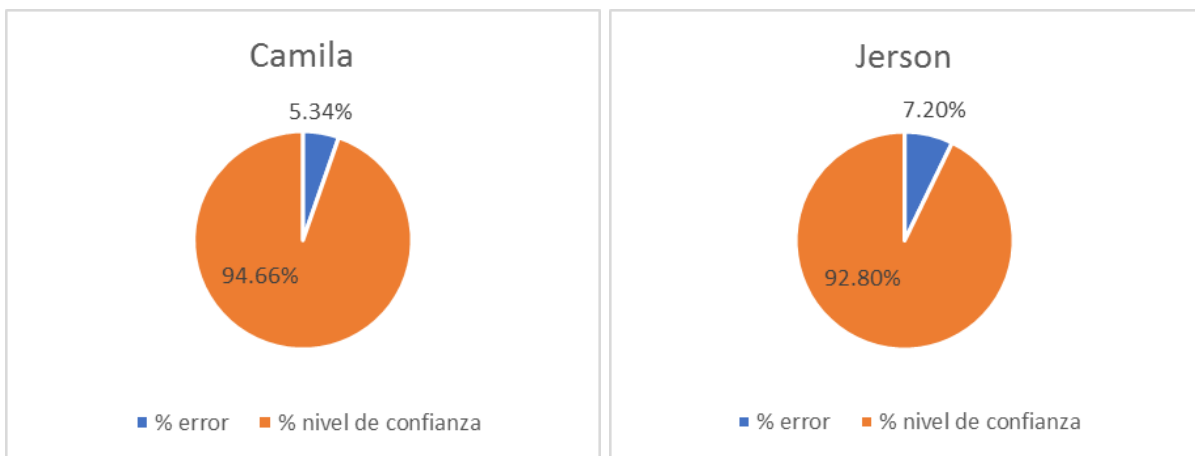


Figura 10. Erros vs Efectividad para ambos usuarios.

Podemos observar en la Figura 10, que para el primer usuario el porcentaje de error con respecto al segundo es menor, esto se debe a la cantidad de imágenes utilizadas, ya que entre más imágenes el nivel de confianza disminuye esto se debe a que el sistema puede obtener mayores características faciales.

Discusión

En el presente artículo de investigación se desarrolló un sistema de reconocimiento facial para control de seguridad en el ingreso de empresas, por lo que se reafirma lo dicho por (Ibarra y Paredes, 2018) en su trabajo de investigación “Redes neuronales artificiales para el control de acceso basado en reconocimiento facial” que afirma que con el desarrollo del control basado en reconocimiento facial se tiene un mayor control en el acceso a una empresa, asimismo nos menciona que este sistema es poco invasivo al usuario ya que solo tiene que acercar el rostro a la cámara y el sistema se encarga del reconocimiento.

El margen de error en el desarrollo del reconocimiento fácil de nuestro proyecto fue leve debido a las características e incremento de imágenes siendo esto un factor favorable para la disminución de fallas tal como se presenta en (Molina, J. C. 2011) en su trabajo de investigación “Reconocimiento de facial basado en FPGA” en el que concluyeron que, al incrementar el número de interacciones, el error de clasificación es de 2% y tiende a cero, conforme se incrementa el número de interacciones.

El sistema de reconocimiento facial presentado en esta investigación no cuenta con una efectividad del 100% esto se debe a diversos factores, tal como lo menciona (Castro y Lucintuña, 2019) en su trabajo de grado titulado “Sistema de reconocimiento facial por medio del lenguaje de programación Python, para el control del acceso al laboratorio de la escuela de sistemas de la universidad estatal de bolívar”, en donde concluye que el reconocimiento facial mediante el lenguaje de programación Python les permitió un reconocimiento facial del 97 % de efectividad, este porcentaje tiene variaciones ya que depende de la iluminación así como la posición del rostro al momento de capturar el rostro del usuario previo el análisis de características faciales del individuo que pasa por el sistema de reconocimiento.

Conclusiones

El objetivo principal de este proyecto fue desarrollar un sistema de reconocimiento facial para control de seguridad en el ingreso de empresas, el cual fue alcanzado con éxito. Se comprobó que el sistema desarrollado permitió identificar a cada uno de los usuarios con los que se trabajó en esta investigación.

El sistema presenta un leve margen de error para cada uno de los usuarios, esto es atribuido principalmente a los videos que se ha utilizado para el procesamiento difieren en tamaño, luminosidad, entre otras características. El nivel de confianza para el usuario 1 y 2 es de 94.66% y 92.80 respectivamente.

El código de programación utilizado para cada una de las etapas es correcto ya que fue capaz de hacer captura y visualización de imágenes, entrenamiento del sistema y reconocimiento facial.

Referencias

- Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python.
- Castro Serrano, L. C., & Lucintuña Hurtado, W. G. (2019). Sistema de reconocimiento facial por medio del lenguaje de programación Python, para el control del acceso al laboratorio de la Escuela de Sistemas de la Universidad Estatal de Bolívar, año 2019 (Bachelor's thesis, Universidad Estatal de Bolívar).

Facultad de Ciencias Administrativas, Gestión Empresarial e Informática. Carrera de Ingeniería en Sistemas Computacionales).

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20(2), 1-13.

Costa Mari, D. (2020). Análisis de un sistema de reconocimiento facial a partir de una base de datos realizado mediante Python (Bachelor's thesis, Universitat Politècnica de Catalunya).

Fuentes, H. A. (2011). Sistemas de reconocimiento basados en la imagen facial. *Avances en Sistemas e Informática*, 8(3), 7-16.

Ibarra-Estévez, J., & Paredes, K. (2018). Redes neuronales artificiales para el control de acceso basado en reconocimiento facial. *revistapuce*.

Marín, M. R., Uribe, J. C. R., & Morales, J. C. O. (2009). Una mirada a la biometría. *Avances en Sistemas e Informática*, 6(2), 29-38.

Molina, J. C. (2011). Reconocimiento de Facial basado en FPGA. *Revista ECIPerú*, 8(1), 6-6.

Montoro, A. F. (2012). Python 3 al descubierto. *Rc Libros*.

Moreano, J. A. C., Pulloquina, R. H. M., Lagla, G. A. F., Chisag, J. C. C., & Pico, O. A. G. (2017). Reconocimiento facial con base en imágenes. *Revista Boletín Redipe*, 6(5), 143-151.

Rodríguez Cárave, J. (2020). Creación de una aplicación de reconocimiento emocional en Python para Neuromarketing y Neuromanagement.

Serratos, F. (2008). La biometría para la identificación de las personas. *Universitat Oberta de Catalunya*, 8-20.