VALIDACIÓN DE IDENTIDAD BIOMÉTRICA PARA EL REGISTRO DE ENTRADA DEL PERSONAL IIDAI

Alex Cristian Ramos Colque
cramoscolque@gmail.com
Ingeniería Informática
Universidad Nacional "Siglo XX"
Llallagua – Bolivia

Resumen - *Se ha desarrollado un sistema de autenticación basado en el reconocimiento facial, utilizando diferentes tecnologías como Python, flash, etc. El sistema permite a los usuarios registrarse mediante la captura de una imagen facial y posteriormente autenticar su identidad comparando una nueva imagen con las imágenes ya registradas en la base de datos. La implementación se llevó a cabo utilizando diversas bibliotecas de reconocimiento facial y procesamiento de imágenes, incluyendo OpenCV para la visión por computadora. Estas herramientas permiten detectar y analizar características faciales con alta precisión. El sistema garantiza una autenticación eficiente y segura, evitando la necesidad de contraseñas tradicionales y reduciendo el riesgo de acceso no autorizado. Además, el uso de Flask facilita la creación de una interfaz web interactiva y amigable para el usuario. En conjunto, esta solución ofrece una mejora significativa en la seguridad y la usabilidad para la autenticación de usuarios.*

Palabras clave - *Autenticación, Flask, Procesamiento de imágenes, Reconocimiento facial.*

Abstract - *An authentication system based on facial recognition has been developed using different technologies such as Python, flask, etc. The system allows users to register by capturing a facial image and subsequently authenticate their identity by comparing a new image with images already registered in the database. The implementation was carried out using various facial recognition and image processing libraries, including OpenCV for computer vision. These tools allow high precision detection and analysis of facial features. The system guarantees efficient and secure authentication, avoiding the need for traditional passwords and reducing the risk of unauthorized access. In addition, the use of Flask facilitates the creation of an interactive and user-friendly web interface. Taken together, this solution offers a significant improvement in security and usability for user authentication.*

Keywords - *Authentication, Facial recognition, Flask, Image processing,*

1.- INTRODUCCIÓN

La autenticación biométrica se ha convertido en una tecnología prominente debido a su capacidad para proporcionar una seguridad mejorada en comparación con los métodos tradicionales basados en contraseñas. El reconocimiento facial, en particular, ha ganado popularidad debido a su naturaleza no intrusiva y a la disponibilidad de hardware compatible. Este proyecto desarrolla un sistema de login basado en reconocimiento facial utilizando Python, Flask y la biblioteca face recognition.

2.- MATERIALES Y MÉTODOS

2.1 Entorno de Desarrollo

Se utilizó el IDE PyCharm Community Edition 2024.1.1 debido a su capacidad para virtualizar un servidor web y facilitar la codificación en distintos lenguajes de programación. La plataforma de despliegue que se utilizó fue pythonanywhere.

2.1.1 Lenguaje de Programación

a) Python: Lenguaje principal del proyecto.

b) Html, Css, JavaScript: Para la creación de interfaz de usuario.

2.1.2 Framework Web: Flask escrito en Python, ideal para crear aplicaciones web de forma rápida y sencilla. Así también permite construir una amplia gama de aplicaciones web, simples hasta APIs complejas.

```python
import os
import time
import face_recognition
from flask import Flask, request, Response, send_from_directory

PATH_IMAGES_DIR = './enrrolados'
TEMP_DIR = './temp'
STATIC_DIR = './static'
enrrolados_faces = []
res = os.listdir(PATH_IMAGES_DIR)
app = Flask(__name__)

1 usage (1 dynamic)
@app.route('/')
def index():
    return Response(open('index.html', encoding='utf-8').read(), mimetype="text/html")
@app.route('/enrrolar.html')
def enrrolarhtml():
    return Response(open('enrrolar.html',encoding='utf-8').read(), mimetype="text/html")
@app.route('/identificar.html')
def identificarhtml():
    return Response(open('identificar.html').read(), mimetype="text/html")
@app.route('/final.html')
def finalhtml():
    return Response(open('final.html', encoding='utf-8').read(), mimetype="text/html")
@app.route('/index.html')
def indexhtml():
    return Response(open('index.html', encoding='utf-8').read(), mimetype="text/html")
@app.route('/utils.js')
def utilsjs():
    return Response(open('utils.js').read(), mimetype="text/javascript")
@app.route('/opencv.js')
```

Fig. 1: Creación de aplicación web con Flask

Fuente: Elaboración Propia

2.2 Procesamiento de imágenes

Se utilizó diferentes bibliotecas de python para este procesamiento de imágenes. OpenCv haarcascade_frontalface_default.xml biblioteca para la visión por computadora Face Recognition biblioteca para el reconocimiento facial, Dlib biblioteca de herramientas de aprendizaje automático.

La biblioteca face recognition se ha utilizado en proyectos de reconocimiento facial debido a su precisión en la detección y análisis de características faciales (Steve Wooding 2018).

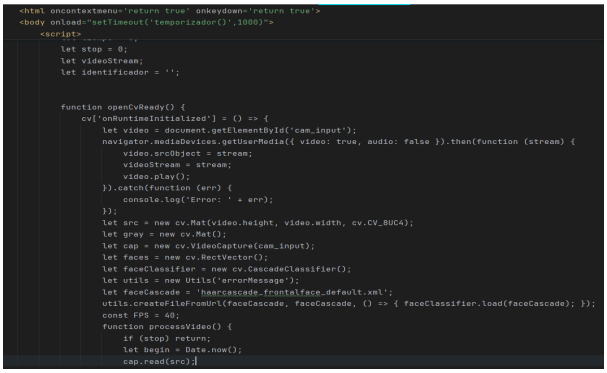

Fig. 2: Visión por computadora
Fuente: Elaboración Propia

2.3 Modelos entrenados para el reconocimiento, detección y predicción de rostros.

a) mmod_human_face_detector.dat: Este modelo está entrenado con el conjunto de datos:http://dlib.net/files/data/dlib_face_detection_dataset-2016-09-30.tar.gz. Se creó el conjunto de datos encontrando imágenes de rostros en muchos conjuntos de datos de imágenes disponibles públicamente (excluyendo el conjunto de datos FDDB).

b) shape_predictor_5_face_landmarks.dat: Este es un modelo de referencia de 5 puntos que identifica las esquinas de los ojos y la parte inferior de la nariz. Se entrenó con el conjunto de datos que se encuentra en: http://dlib.net/files/data/dlib_faces_5points.tar, que consta de 7198 rostros. Se creó este conjunto de datos descargando imágenes de Internet y anotándolas con la herramienta imglab de dlib. Este modelo está diseñado para funcionar bien con el detector de rostros HOG de dlib y el detector de rostros CNN (mmod_human_face_detector.dat).

c)shape_predictor_68_face_landmarks.dat

Este modelo está entrenado en el conjunto de datosbug300W(https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/).

d)dlib_face_recognition_resnet_model_v1.dat: La red se entrenó desde cero con un conjunto de datos de aproximadamente 3 millones de rostros. Este conjunto de datos se deriva de dos conjuntos de datos.

El conjunto de datos de face scrub (http://vintage.winklerbros.net/facescrub.html) y el conjunto de datos de VGG (http://www.robots.ox.ac.uk/~vgg/data/vgg_face/). Se hizo este modelo entrenando una CNN de reconocimiento facial y luego usando métodos de agrupación de gráficos y mucha revisión manual para limpiar el conjunto de datos. Al final, aproximadamente la mitad de las imágenes son de VGG y fase scrub. Además, el número total de identidades individuales en el conjunto de datos es 7485. Se evitaron superposiciones de identidades en LFW.

3.- RESULTADOS

El sistema desarrollado permite el registro y la autenticación de usuarios utilizando imágenes faciales. Durante las pruebas, el sistema mostró una alta precisión en la identificación de rostros previamente registrados. Se tiene los archivos de index.html, enrrolar.html, identificar.html y final.html como interfaz para el registro y la autenticación.

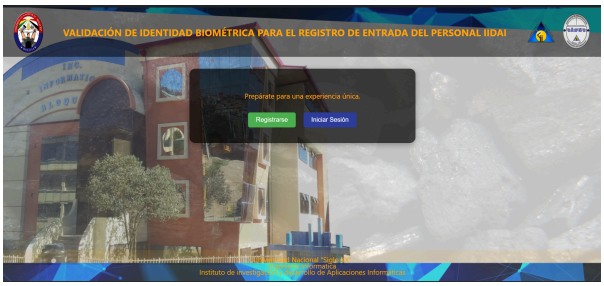

Fig. 1: Interfaz de la página web.
Fuente: Elaboración propia

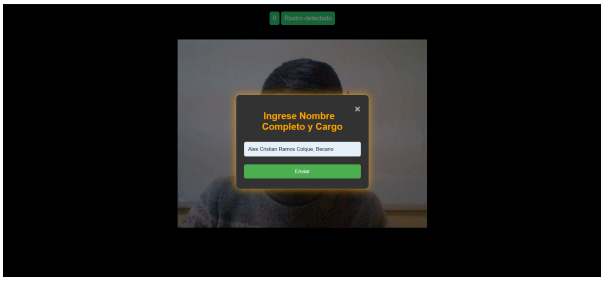

Fig. 2: Interfaz de registro único mediante contraseña y la captura de fotografía.
Fuente: Elaboración Propia

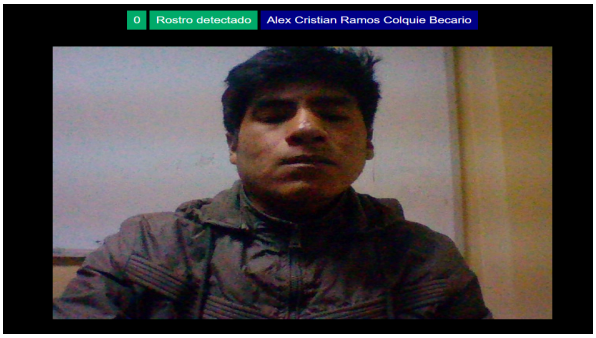

Fig. 3: Interfaz de inicio de sesión
Fuente: Elaboración Propia

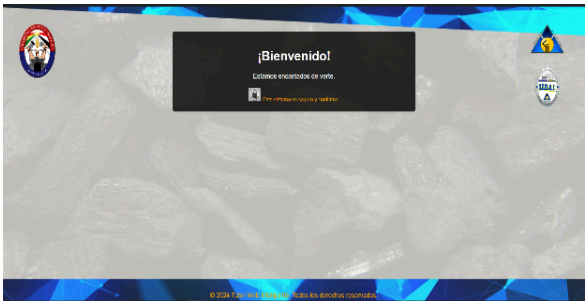

Fig. 4: Interfaz de Bienvenida una vez reconocido con éxito al usuario
Fuente: Elaboración propia

```
face_image = face_recognition.load_image_file(temp_path)
face_encodings = face_recognition.face_encodings(face_image)

if not face_encodings:
    raise ValueError("No se encontró un rostro en la imagen.")

unk_face_encoding = face_encodings[0]
results = face_recognition.compare_faces(enrrolados_faces, unk_face_encoding, tolerance=0.6)
```

Fig. 5: Proceso de reconocimiento facial
Fuente: Elaboración propia

Se procesa la imagen para extraer y codificar las características faciales:

- **face_recognition.load_image_file()**: carga la imagen.
- **face_recognition.face_encodings():** utiliza modelos pre entrenados para extraer las características faciales del usuario
- **face_recognition.compare_faces()**: utiliza las codificaciones faciales y las compara con las caras registradas, con un umbral de tolerancia.

4.- CONCLUSIONES

El desarrollo de un sistema de login con reconocimiento facial utilizando Python y Flask demostró ser una solución viable y eficiente para la autenticación de usuarios. La detección y comparación precisa del usuario y su respectivo dato personal se puede lograr mediante el uso de herramientas tecnológicas como detección de rostros, reconocimiento facial, representación facial, además del uso de bibliotecas de visión artificial y framework web.

La implementación presentada en este artículo puede ser utilizada como base para futuros desarrollos en el campo de la biometría y la seguridad informática.

Destacar que, si bien el sistema ha demostrado ser efectivo, aún existen aspectos que pueden ser mejorados en futuras investigaciones. Entre ellos, se encuentran la optimización del rendimiento para procesar imágenes y vídeos en tiempo real, la mejora de la precisión del reconocimiento facial en condiciones adversas de iluminación y la implementación de mecanismos de seguridad aún más robustos para proteger los datos de los usuarios.

5. REFERENCIAS

Jose F. Velez Serrano, Ana B. Moreno Diaz, Angel Sanchez C, Jose Luis E. Sanchez- Marin Visión por computador:http://www.visionporcomputador.es/libroVision/VisionPorComputador.pdf

Face Recognition with OpenCV (2018). Steve Wooding

Visión artificial: Fundamentos y aplicaciones (2017). De la visión artificial en la actualidad https://bcnvision.es/blog-vision-artificial/los-fundamentos-y-aplicaciones-de-la-vision-artificial-en-la-actualidad/

Gonzalo Pajares Martinsanz (2010). Visión por Computador Imágenes Digitales y Aplicaciones

Stuart Russell, Peter Norvig. Inteligencia Artificial Fundamentos, Práctica y Aplicaciones

Handbook of Face Recognition (2011). Stan Zisserman, Michael Black, y David J. Teller

Face Recognition: Adam Geitgey https://github.com/ageitgey/face_recognition

Desarrollo Web con Flask Miguel Grinberg

Davisking / dlib-models https://github.com/davisking/dlib-models