



INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE MONTERREY  
ESCUELA DE INGENIERÍA Y CIENCIAS  
MONTERREY, NUEVO LEÓN

INTELIGENCIA ARTIFICIAL AVANZADA PARA LA CIENCIA DE  
DATOS II

**SISTEMA DE PASE DE LISTA Y DETECCIÓN DE PARTICIPACIÓN BASADO EN ML Y  
VISIÓN COMPUTACIONAL**

Profesores: IVÁN MAURICIO AMAYA CONTRERAS, Ph.D.

FÉLIX RICARDO BOTELLO URRUTIA

EDGAR COVANTES OSUNA

FELIPE CASTILLO RENDÓN

HUGO TERASHIMA MARÍN

BLANCA R. RUIZ HERNÁNDEZ

Equipo 6

DIEGO ALBERTO BAÑOS LOPEZ A01275100

CAROL ARRIETA MORENO A01275465

CARLOS DAVID LOZANO SANGUINO A01275316

## Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>  | <b>3</b>  |
| <b>2. Fundamentos teóricos</b>  | <b>4</b>  |
| 2.1. Reconocimiento de rostros . . . . .                                    | 4         |
| 2.2. Detección de pose . . . . .  | 6         |
| 2.3. Algoritmo seleccionado para el reconocimiento de rostro . . . . .      | 6         |
| 2.4. Algoritmo seleccionado para la detección de pose . . . . .             | 7         |
| <b>3. Método propuesto</b>  | <b>8</b>  |
| 3.1. Justificación del método seleccionado . . . . .                        | 8         |
| 3.2. Overview de la solución desarrollada . . . . .                         | 9         |
| 3.2.1. Componentes del Sistema . . . . .                                    | 10        |
| 3.3. Dataset considerado para validar el funcionamiento . . . . .           | 16        |
| 3.4. Métricas consideradas para analizar el desempeño del sistema . . . . . | 17        |
| <b>4. Metodología</b>   | <b>19</b> |
| <b>5. Resultados</b>  | <b>20</b> |
| <b>6. Conclusiones</b>  | <b>21</b> |
| <b>7. Anexos</b>  | <b>23</b> |

## Índice de figuras

|   |    |
|---|----|
| 1. Detección Múltiple de Personas . . . . .                 | 10 |
| 2. Estructura de nuestra base de datos . . . . .            | 11 |
| 3. Detección de Pose y Mano por MediaPipe . . . . .         | 12 |
| 4. Muestra la pagina de inicio de sesión . . . . .          | 13 |
| 5. inicio de sesion realizandose con azure AD B2C . . . . . | 13 |
| 6. Muestra la pagina de Camara . . . . .                    | 13 |
| 7. Muestra la pagina de dashboard . . . . .                 | 14 |
| 8. Vista de todos los registros de participación . . . . .  | 14 |



Abstract. En nuestro proyecto de desarrollo de un sistema de pase de lista, logramos una interfaz innovadora que utiliza eficientemente dos modelos de reconocimiento: facial y de participación. Al integrar FaceRecognition 1.3.0 y la detección de poses con MediaPipe, obtuvimos resultados satisfactorios en la precisión del registro de asistencias. La interfaz fue meticulosamente probada en múltiples ocasiones, asegurando la correcta identificación del alumno mediante su matrícula.

La importancia de este trabajo radica en la mejora de la eficiencia y la seguridad en entornos educativos. Buscábamos crear una solución integral que combinara la precisión del reconocimiento facial con la detección de poses, permitiendo una verificación completa de la asistencia. Las condiciones de prueba analizadas incluyeron variaciones en la iluminación y poses, replicando situaciones del mundo real.

El principal resultado fue el éxito consistente en la identificación y registro de estudiantes, validando la robustez del sistema. En conclusión, nuestra interfaz de pase de lista representa un avance significativo, proporcionando una herramienta confiable y eficiente para la gestión de asistencias en entornos educativos.

## 1. Introducción

En un entorno educativo dinámico y en constante evolución, maximizar la eficiencia de las clases es esencial para el aprendizaje efectivo. En este contexto, nos enfrentamos al desafío de desarrollar un sistema integral de reconocimiento facial y machine learning para optimizar la medición de la asistencia y participación de los estudiantes en las aulas del Tec.

La relevancia del reconocimiento facial para nuestro proyecto radica en su habilidad para ofrecer una solución eficaz y precisa en la medición de la asistencia y participación de los estudiantes. Al basarnos en este avance tecnológico, no solo buscamos optimizar el proceso de pase de lista, sino también ir más allá al identificar y registrar rostros en tiempo real.

Este proyecto tiene como objetivo principal implementar un proceso automatizado de pase de lista mediante el reconocimiento facial de los alumnos. Al aprovechar la ciencia de datos y algoritmos de machine learning, aspiramos a proporcionar a los profesores una herramienta que no solo les permita ahorrar tiempo valioso, sino que también mejore la calidad de la enseñanza al brindar una visión detallada de la asistencia y participación de cada estudiante.

El sistema no se limitará únicamente al registro de rostros, sino que se extenderá a la

detección de la participación del alumno cuando levantan la mano. La implementación de modelos de machine learning específicos para esta tarea brindará a los profesores la capacidad de identificar de manera rápida y precisa quiénes están participando activamente en el aula. El resultado final será la creación de un completo dashboard que exhibirá la participación de cada alumno en tiempo real. Este panel proporcionará información valiosa que puede ser utilizada para evaluar el rendimiento del estudiante y ajustar las estrategias de aprendizaje en consecuencia.

Además, para garantizar la fácil administración de las clases, se desarrollará un portal que permitirá a los docentes gestionar eficientemente la información relacionada con la asistencia y participación. Este enfoque integral no solo simplificará las tareas administrativas, sino que también mejorará la experiencia general de enseñanza y aprendizaje.

En última instancia, al incorporar esta tecnología de vanguardia, no solo estamos simplificando el proceso de asistencia y participación, sino que también estamos allanando el camino hacia un futuro educativo más conectado y adaptativo, donde la tecnología no solo es una herramienta, sino un facilitador integral del aprendizaje.

## 2. Fundamentos teóricos

### 2.1. Reconocimiento de rostros

El reconocimiento de rostros constituye una disciplina esencial en visión por computadora, con aplicaciones que abarcan desde la seguridad hasta la interacción humano-máquina. En el desarrollo de sistemas de reconocimiento facial, se emplean diversas formulaciones y algoritmos, cada uno diseñado para abordar desafíos específicos. A continuación, se detallan algunos de los principales enfoques utilizados en esta área:

- Algoritmos
  1. Cascada de Haar: Este método se basa en el uso de características tipo Haar para la detección de rostros. Se emplean clasificadores en cascada para filtrar regiones de la imagen que pueden contener rostros, reduciendo así la carga computacional. Se emplean clasificadores en cascada, como el clasificador Adaboost, para evaluar si una región de la imagen contiene o no un rostro. La cascada permite una rápida

eliminación de regiones no faciales, reduciendo la carga computacional al descartar áreas irrelevantes.[9]

2. Local Binary Patterns (LBP): LBP describe texturas locales al examinar la relación de intensidades entre un píxel central y sus vecinos en una ventana. La codificación binaria de estas relaciones forma un patrón que refleja la textura local. En el reconocimiento facial, LBP captura características discriminativas de las texturas de la piel al resaltar patrones repetitivos, como arrugas o texturas específicas del rostro.[8]
3. Modelo de Mezcla Gaussiana (GMM): GMM modela la distribución de los píxeles de una imagen como una combinación de múltiples distribuciones gaussianas. Cada componente representa una fuente potencial de intensidad de píxeles. Es efectivo en entornos donde las condiciones de iluminación pueden variar, ya que permite modelar la variabilidad en la intensidad de los píxeles de manera más flexible.[5]

- Redes Neuronales Convolucionales (CNN)

1. AlexNet, VGG, ResNet: Estas arquitecturas CNN son profundas y complejas, permitiendo la extracción de características jerárquicas en diferentes niveles de abstracción. AlexNet, VGG y ResNet han demostrado eficacia en la clasificación de rostros al aprender representaciones ricas y abstractas.
2. Redes Neuronales Siamesas: Las redes siamesas utilizan dos ramas idénticas para procesar pares de imágenes, extrayendo representaciones discriminativas. Estas redes son eficaces en tareas de verificación y reconocimiento facial, ya que aprenden a comparar y contrastar las similitudes entre las imágenes.

- Aprendizaje Profundo y Detección de Objetos:

1. YOLO (You Only Look Once): YOLO aborda el reconocimiento facial como una tarea de detección de objetos. Divide la imagen en celdas y predice las cajas delimitadoras y las probabilidades de clase para los objetos presentes en cada celda en una sola pasada. YOLO ofrece eficiencia al procesar la imagen de manera integral, permitiendo la detección en tiempo real con una única pasada por la red.

## 2.2. Detección de pose

La detección de pose consiste en la capacidad de un sistema de identificar la posición y orientación de un cuerpo humano, ya sea de cuerpo completo o de partes específicas, así como los brazos, cara, manos, dedos, entre otras. Los principales usos que se le da a la detección de pose son para realidad aumentada y virtual, aplicaciones enfocadas a la salud y ejercicio, entretenimiento y seguimiento de personas para vigilancia. Algunos de los algoritmos para detección de pose son:

1. OpenPose: Desarrollado por CMU Perceptual Computing Lab, este hace uso de redes neuronales (CNN) y es capaz de identificar el cuerpo completo, así como cada parte de este.
2. SPPE (Simple Pose): Este detecta las partes del cuerpo y predice la pose dependiendo de las articulaciones.
3. HRNet (High-Resolution Network): Está diseñada específicamente para la detección de pose y se basa en el uso de diferentes resoluciones para obtener hasta los más mínimos detalles, lo cual ayuda a mejorar la precisión.

De igual manera existen diferentes librerías que nos facilitan la implementación de detección de pose, entre estas están OpenPose; esta se desarrolla en conjunto con el algoritmo explicado anteriormente con el que comparte nombre, MediaPipe; es desarrollada por Google e incluye detección de pose de manos, facial, seguimiento de objetos, entre otras cosas, TF-Pose Estimation; esta se basa en TensorFlow y utiliza modelos pre-entrenados para detectar las poses en tiempo real.

## 2.3. Algoritmo seleccionado para el reconocimiento de rostro

En nuestro proyecto, optamos por implementar el algoritmo de reconocimiento facial proporcionado por la biblioteca "Face Recognition 1.3.0". Esta biblioteca destaca por ser una de las más simples y efectivas para el reconocimiento facial en entornos Python y a través de la línea de comandos. A continuación, se detallan sus características clave:

- Funcionamiento: La biblioteca utiliza la infraestructura de reconocimiento facial de vanguardia de dlib, la cual se basa en técnicas de aprendizaje profundo. Al emplear

este algoritmo, se logra un reconocimiento facial robusto y preciso, con una tasa de precisión del 99.38 por ciento en el reconocido conjunto de datos Labeled Faces in the Wild.

- Parámetros: La precisión del modelo se traduce en su capacidad para reconocer y manipular caras en diversas condiciones. Los parámetros internos están optimizados para extraer características distintivas de las caras y realizar comparaciones precisas entre ellas.
- Fuente y PyPI: La biblioteca Face Recognition 1.3.0 esta construida utilizando la capacidad de reconocimiento facial de dlib y se encuentra disponible en el Python Package Index (PyPI), lo que facilita su instalación y uso en proyectos Python. La documentación detallada proporciona información sobre cómo integrar y utilizar eficientemente las funcionalidades de reconocimiento facial en aplicaciones.[1]
- Face Recognition Command Line Tool: Además de su facilidad de integración en proyectos Python, la biblioteca ofrece una herramienta de línea de comandos que simplifica el proceso de reconocimiento facial en un conjunto de imágenes. Esto permite realizar el reconocimiento de rostros de manera eficiente y práctica, facilitando su implementación en diversos escenarios.[1]
- La elección de "Face Recognition 1.3.0" se fundamenta en su combinación de simplicidad y precisión, lo que la convierte en una herramienta ideal para nuestro proyecto de pase de lista basado en el reconocimiento facial, además de ser open source, lo cual lo vuelve una opción que a grandes rasgos no es restrictiva en su uso.

## 2.4. Algoritmo seleccionado para la detección de pose

La elección para la detección de pose fue últimamente con cual nos pareció mas flexible trabajar, mientras mantenía un majeo relativamente bueno de la pose, en este caso decidimos elegir mediapipe ya que es muy flexible trabajar sobre el, ademas de ser de una herramienta libre de usar.

### 3. Método propuesto

#### 3.1. Justificación del método seleccionado

En la evaluación de modelos preentrenados para nuestro proyecto de reconocimiento facial, consideramos varias alternativas, entre ellas, YOLOv5 v8, OpenFace, DeepFace y ArcFace. Cada opción fue evaluada en función de criterios clave, y la elección de Face Recognition 1.3.0 se fundamenta en su rendimiento, simplicidad y capacidad de integración con dlib.

- Alternativas consideradas

1. OpenCV: una librería integral en el ámbito de visión por computadora, ofrecía una opción atractiva debido a su robustez y diversidad de funciones. Su capacidad para realizar tareas de detección de rostros y extracción de características constituye una ventaja, especialmente en aplicaciones donde se busca una solución integral.[7]
2. FaceNet: FaceNet, desarrollado por Google, destaca por su enfoque en la generación de embeddings faciales altamente discriminativos. Este modelo preentrenado tiene la capacidad de aprender representaciones faciales robustas, favoreciendo el reconocimiento preciso en diversas condiciones. Basada en tripletas, permite una representación tridimensional eficaz de las caras, contribuyendo a la capacidad del modelo para diferenciar entre individuos con alta precisión.[3]
3. OpenPose: Se destaca por su capacidad para detectar y rastrear poses humanas, lo cual puede complementar la información facial al considerar la postura corporal en contextos de reconocimiento facial. La capacidad de OpenPose para trabajar con información detallada de poses agrega una dimensión adicional al análisis de imágenes, lo que puede mejorar la robustez del sistema en situaciones donde la postura del cuerpo es un factor relevante.[2]

- Elección de Face Recognition 1.3.0

1. Simplicidad de Uso: Face Recognition 1.3.0 destaca por su simplicidad de uso, lo que facilita su implementación en nuestro sistema de pase de lista. La biblioteca proporciona una interfaz amigable y una documentación clara, lo que acelera el proceso de desarrollo y redunda en una eficiencia operativa.[4]

2. Integración con dlib: La biblioteca se construye sobre dlib, una potente librería de procesamiento de imágenes y aprendizaje automático. Esta integración permite aprovechar las capacidades avanzadas de dlib en reconocimiento facial basado en aprendizaje profundo, lo que contribuye a la robustez del modelo.[4]
3. Rendimiento en el Benchmark LFW: Face Recognition 1.3.0 ha demostrado un rendimiento impresionante en el reconocido conjunto de datos Labeled Faces in the Wild (LFW), alcanzando una precisión del 99.38 por ciento. Esta alta tasa de precisión fue un factor determinante en la elección, ya que refleja la capacidad del modelo para generalizar y reconocer rostros en condiciones realistas y diversas.[4]
4. Facilidad de Implementación: La disponibilidad de la biblioteca en el Python Package Index (PyPI) simplifica la implementación y mantenimiento del modelo preentrenado en nuestro entorno de desarrollo. La presencia de una herramienta de línea de comandos adicional para realizar reconocimiento facial también agrega versatilidad a la solución.[4]

La elección de Face Recognition 1.3.0 se basa en su combinación única de simplicidad, rendimiento excepcional en benchmarks relevantes, y una integración eficaz con dlib. Estas características hicieron que este modelo preentrenado destacara como la opción ganadora, proporcionando una base sólida para el éxito de nuestro sistema de pase de lista basado en reconocimiento facial.

### **3.2. Overview de la solución desarrollada**

El modelo de reconocimiento facial propuesto se fundamenta en la biblioteca Face Recognition 1.3.0, la cual utiliza la capacidad de reconocimiento facial de dlib. Esta biblioteca, disponible en el Python Package Index (PyPI), facilita su integración en proyectos Python. La propuesta se centra en el reconocimiento facial de rostros conocidos mediante la codificación de los mismos. Se ha implementado un sistema que, a través de una cámara, captura y analiza frames de video en tiempo real. La detección facial se realiza inicialmente mediante el empleo de clasificadores Haar cascades, y posteriormente, se utiliza la capacidad de reconocimiento facial de Face Recognition 1.3.0 para identificar rostros previamente codificados. La herramienta de línea de comandos proporcionada por la biblioteca simplifica este proceso.

### 3.2.1. Componentes del Sistema

- Modelo Face Recognition 1.3.0: EL componente principal del sistema es la biblioteca Face Recognition 1.3.0, que aprovecha la potencia de dlib para el reconocimiento facial. Este componente permite identificar y comparar rostros conocidos en tiempo real, brindando una solución robusta y precisa para la autenticación de identidad. Face Recognition 1.3.0 utiliza un modelo preentrenado para codificar y representar los rasgos únicos de los rostros conocidos. La codificación resultante permite una comparación eficiente con los rostros en tiempo real, logrando una alta precisión en la identificación. Se complementa el reconocimiento facial mediante el uso de clasificadores Haar cascades de OpenCV como se menciono anteriormente para la detección inicial de rostros en la imagen. Esta combinación de enfoques mejora la eficiencia del sistema al filtrar regiones que probablemente contengan rostros.



Figura 1: Detección Múltiple de Personas

- Modelo de Asistencia de Mano con MediaPipe

El modelo de asistencia de mano se basa en la librería Mediapipe para la detección de hombros, brazos y cara. Su funcionamiento detallado es el siguiente:

1. Inicialización: Se inicializa la captura de video y se configuran los modelos de MediaPipe para la detección de hombros, brazos y cara.
2. Detección Facial Inicial: Se utiliza un clasificador Haar cascades para la detección facial. Las regiones faciales identificadas se envían al modelo de reconocimiento facial para la verificación de rostros conocidos.
3. Procesamiento con MediaPipe: Se procesa el frame de video utilizando el modelo de MediaPipe para la detección de hombros, brazos y cara.

4. Visualización de Resultados: Los puntos del cuerpo detectados se dibujan en el frame, proporcionando una representación visual de la postura corporal.
5. Almacenamiento en Base de Datos SQL: La información de reconocimiento facial se guarda en una base de datos SQL, asociando las identidades reconocidas con un identificador único. La información sobre la postura corporal también puede ser almacenada para análisis posterior.

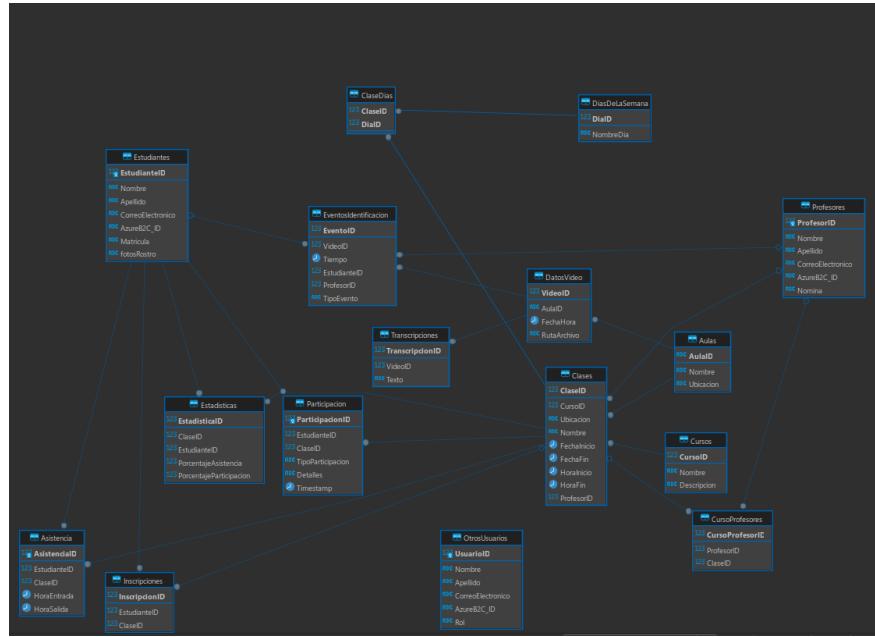


Figura 2: Estructura de nuestra base de datos

6. Visualización en Azure: La información almacenada en la base de datos se puede visualizar en Azure, proporcionando a los usuarios autorizados un acceso fácil y seguro a los registros de asistencia y datos de reconocimiento facial.



Figura 3: Detección de Pose y Mano por MediaPipe

- Interfaz de usuario: La interfaz de usuario consiste en un login en el cual se podrá acceder usando como método de acceso Azure AD B2c, el cual permite identificar al usuario, así como permite identificar el rol con el que esta, dependiendo del rol que tenga se vera limitado el uso o en general a que puede acceder de la pagina web, siendo administrador el rango mas alto, con la capacidad de acceder a todos los cursos y a todo lo que ofrece la interfaz, seguido de profesores que tienen la misma vista solo que tiene la limitación respecto a los cursos que tengan acceso y que solo puede registrar alumnos y al final Alumno el cual lo único relevante que puede hacer es registrar su cara en el menú de inicio, la interfaz cuenta con 3 vistas importantes Inicio, en donde se puede hacer tanto el registro de nuevos usuarios como el registro de caras de alumnos ya existentes, dashboard que muestra las estadísticas en general, permite descargar datos acerca de las asistencias y las participaciones que se tenga y hay una pequeña gráficas relacionadas con los datos recolectados, de igual forma se puede ver los alumnos que están registrados dentro de la clase seleccionada, y por ultimo esta el botón de cámara que nos permite accionar el sistema de registro de asistencias y participaciones

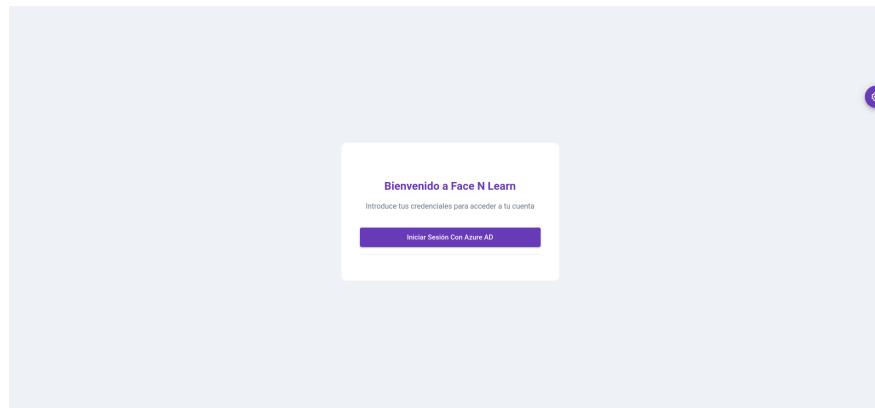


Figura 4: Muestra la pagina de inicio de sesión

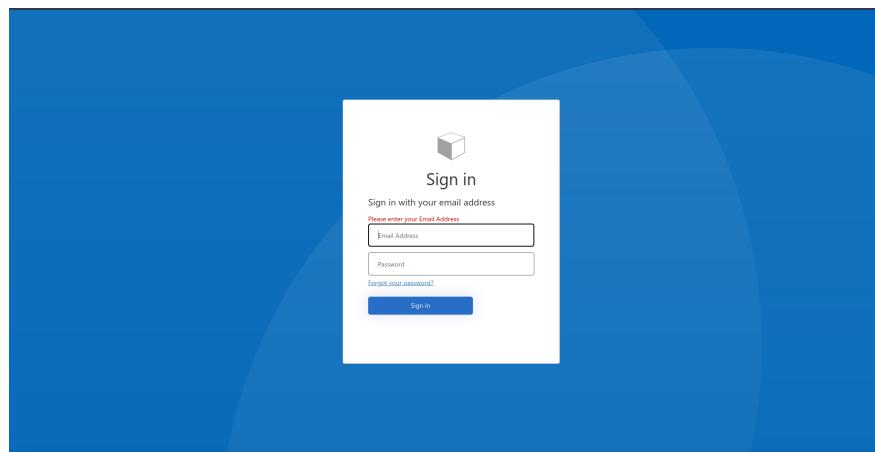


Figura 5: inicio de sesion realizandose con azure AD B2C

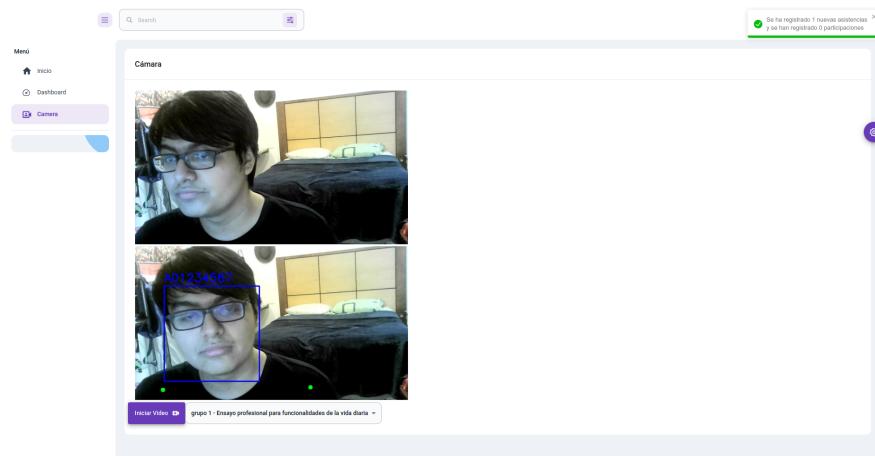


Figura 6: Muestra la pagina de Camara

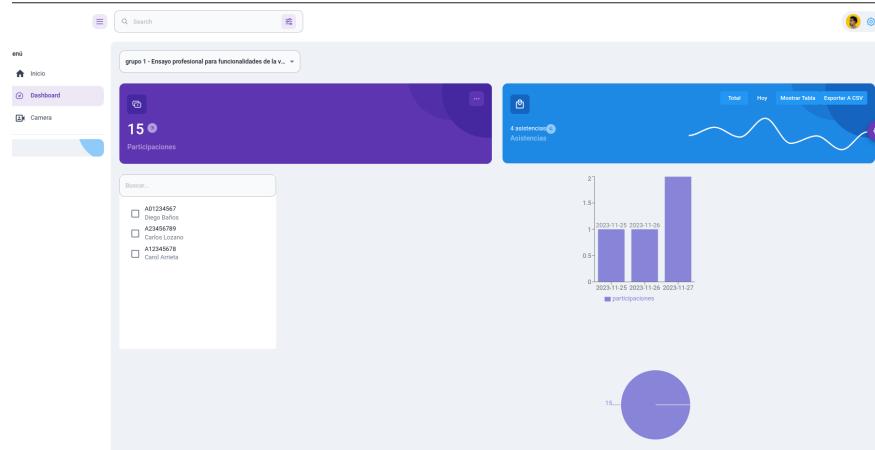


Figura 7: Muestra la pagina de dashboard

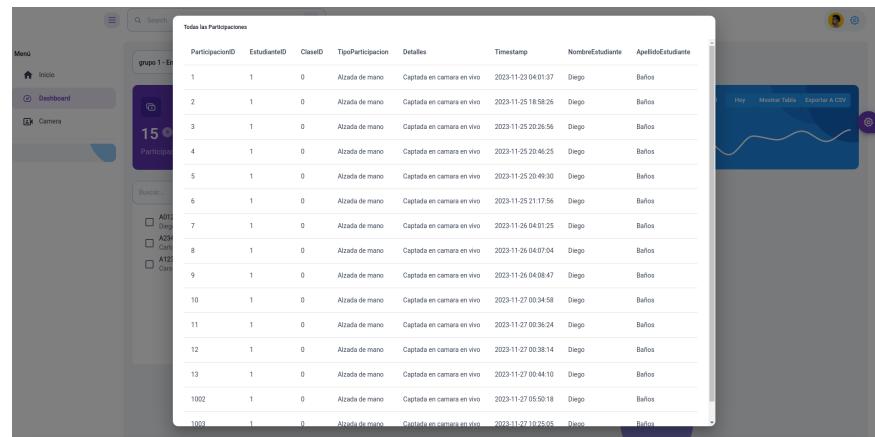


Figura 8: Vista de todos los registros de participación

The homepage features a central registration form titled 'Registra a tu usuario'. It includes fields for Nombre, Apellido, Dirección de correo electrónico / Nombre de usuario, Contraseña, and a Registrar button. Below the form are links for 'Ocultar Registro' and 'Registrar Caja de Usuario'.

Figura 9: Muestra la pagina de inicio con el registro de usuarios abierto

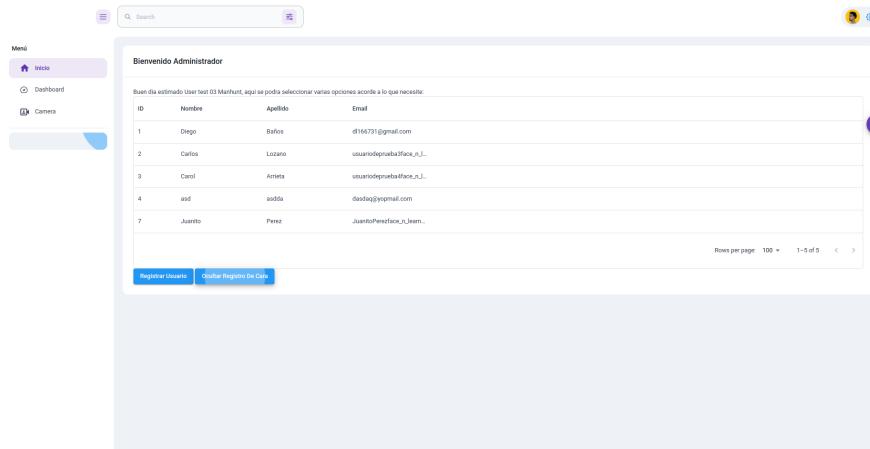


Figura 10: Muestra la pagina de inicio con el registro de caras de alumnos abierta

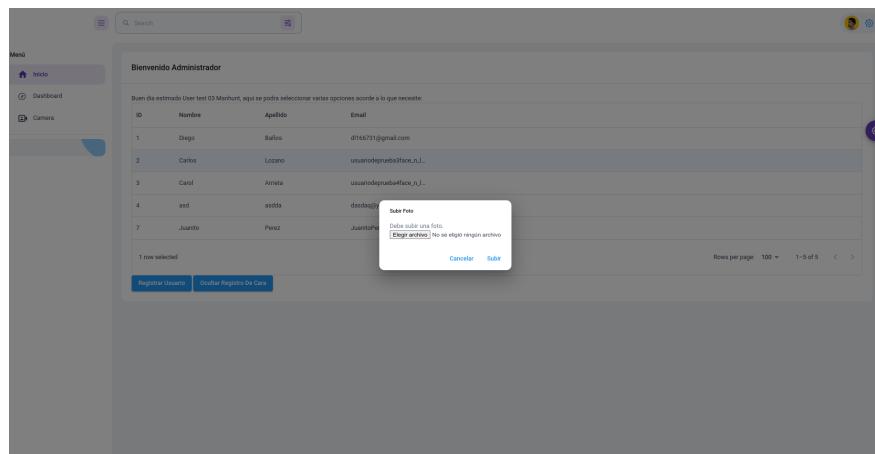


Figura 11: Un alumno seleccionado por un profesor para actualizar su cara

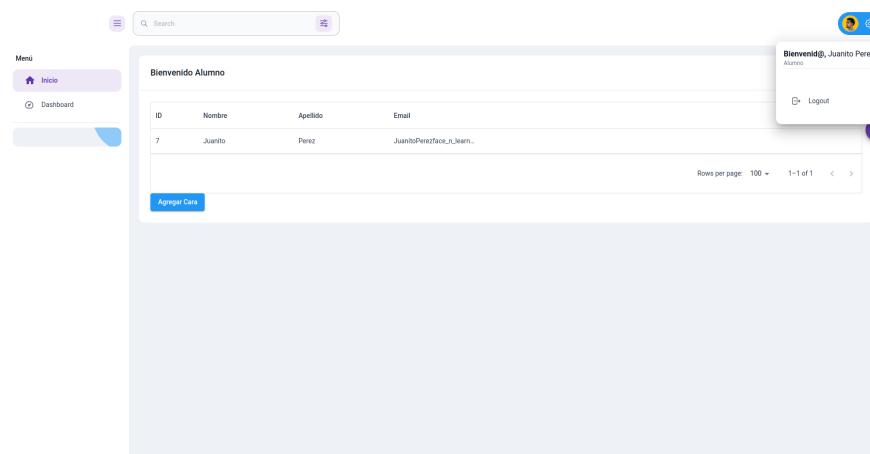


Figura 12: Vista de la pagina de inicio de un alumno, solo puede registrar a si mismo

### 3.3. Dataset considerado para validar el funcionamiento

La elección del dataset "Faces Labeled Wild" para entrenar nuestro modelo de reconocimiento facial preentrenado se basa en su amplitud y diversidad, proporcionando un conjunto de datos robusto que abarca una variabilidad significativa en términos de condiciones de iluminación, poses y expresiones faciales. Este dataset se ha convertido en un estándar en la investigación de reconocimiento facial y ofrece la capacidad de evaluar el rendimiento del modelo en escenarios del mundo real.

El dataset consta de más de 13,000 imágenes de rostros recopiladas de la web, abarcando retratos individuales, grupos sociales y situaciones cotidianas. Cada rostro ha sido etiquetado con el nombre de la persona retratada. La diversidad del conjunto refleja la complejidad del mundo real, incluyendo variaciones en la edad, género, etnia y entorno. La selección de este dataset permite al modelo aprender características distintivas y adaptarse a una amplia gama de contextos.[6]



Figura 13: Muestra de Rostros del Dataset FLW

Cada imagen en el dataset viene acompañada de etiquetas que indican la presencia de rostros y, posiblemente, coordenadas de regiones faciales. Estas etiquetas son fundamen-

tales para el entrenamiento supervisado del modelo, ya que proporcionan información sobre la ubicación y características de los rostros en las imágenes. Al entrenar el modelo con estas etiquetas, se busca que el sistema adquiera la capacidad de reconocer rostros y asociar identidades. El enfoque en la utilización de este conjunto de datos se centra en la accesibilidad y disponibilidad de las imágenes de manera libre y gratuita, evitando problemas potenciales de privacidad de datos, ya que las imágenes están recopiladas de fuentes públicas en la web

### **3.4. Métricas consideradas para analizar el desempeño del sistema**

En general para poder medir los resultados del sistema de reconocimiento facial y el sistema de detección de poses para la asistencia y para la participación de un aula se realizaron dentro del programa de python funciones que ayudan a detectar en si como funciona dicho programa, en general verifica las caras y posiciones, se realizan comparaciones dentro de los datos que se acumulan, en este caso serian los rostros y las posiciones de la persona, contar cuando se detecta correctamente una cara o una pose, cuando llegara a suceder por ejemplo un falso positivo, y al final se realiza un calculo para determinar por ejemplo la precisión del modelo. Para activar las metricas es necesario cambiar offline\_mode a true.

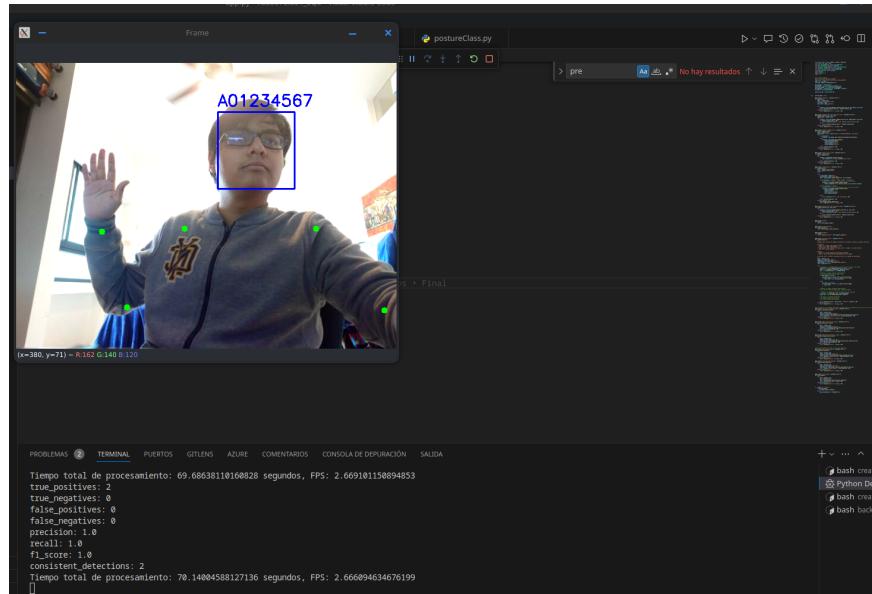


Figura 14: Muestra del programa corriendo mientras se observa las métricas en consola

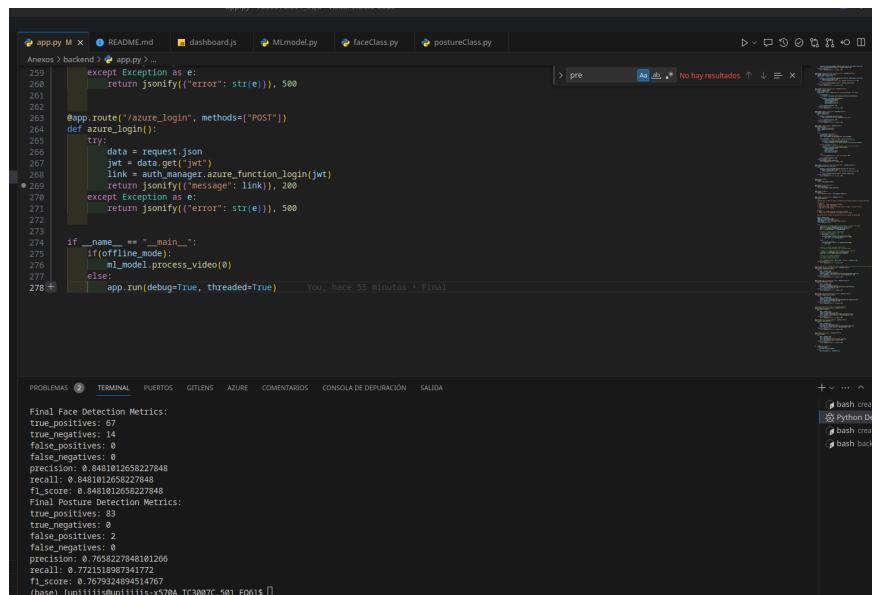


Figura 15: Muestra del programa cuando termina muestra un total de las métricas en consola

```
Final Face Detection Metrics:  
true_positives: 67  
true_negatives: 14  
false_positives: 0  
false_negatives: 0  
precision: 0.8481012658227848  
recall: 0.8481012658227848  
f1_score: 0.8481012658227848  
Final Posture Detection Metrics:  
true_positives: 83  
true_negatives: 0  
false_positives: 2  
false_negatives: 0  
precision: 0.7658227848101266  
recall: 0.7721518987341772  
f1_score: 0.7679324894514767  
(base) [upiiiiis@upiiiiis-x570A TC3007C.501 E061$ ]
```

Figura 16: Un mejor vistazo a lo que arroja consola al final

#### 4. Metodología

El plan de trabajo establecido para alcanzar el objetivo del proyecto se presenta en la figura 1. El sistema propuesto está dividido en tres fases. La primera fase es fundamental, centrándose en la recolección de datos a través de cámaras instaladas en las aulas. Estas cámaras estarán ubicadas en ángulos y alturas adecuadas para garantizar la grabación y monitoreo integral del salón. Durante la entrada de los estudiantes, se recopilará información para que la cámara registre los rostros de quienes ingresaron a la clase.

La segunda parte del sistema se subdivide en dos secciones. Por un lado, se encuentra el modelo de Detección y Reconocimiento Facial, que constituye la base esencial para almacenar los datos faciales, identificar a los alumnos presentes, y activarse durante la entrada en vivo para reconocer a los participantes. Por otro lado, se encuentra la Detección de Poses, un modelo independiente que identificará a aquellos estudiantes que levanten la mano para participar o plantear preguntas al profesor. La información recopilada en esta fase se utilizará en las siguientes etapas del proyecto.

Los modelos empleados son preentrenados, con una precisión superior al 90 por ciento, capacitados con miles de rostros para obtener resultados óptimos. Aunque se entrenarán con los nuevos rostros de los alumnos, el preentrenamiento contribuirá a obtener mejores resultados, especialmente dada la limitada disponibilidad actual de conjuntos de datos faciales.

La última fase implica registrar los datos recopilados en una base de datos SQL,

donde se almacenarán todos los registros de asistencias y participaciones de los alumnos. Posteriormente, estos datos se transferirán o utilizarán para visualizarlos en una interfaz creada por nosotros en la plataforma Azure. Esta interfaz permitirá a los profesores acceder y analizar de manera rápida y sencilla los registros de asistencia y participación de los alumnos, proporcionando retroalimentación valiosa.

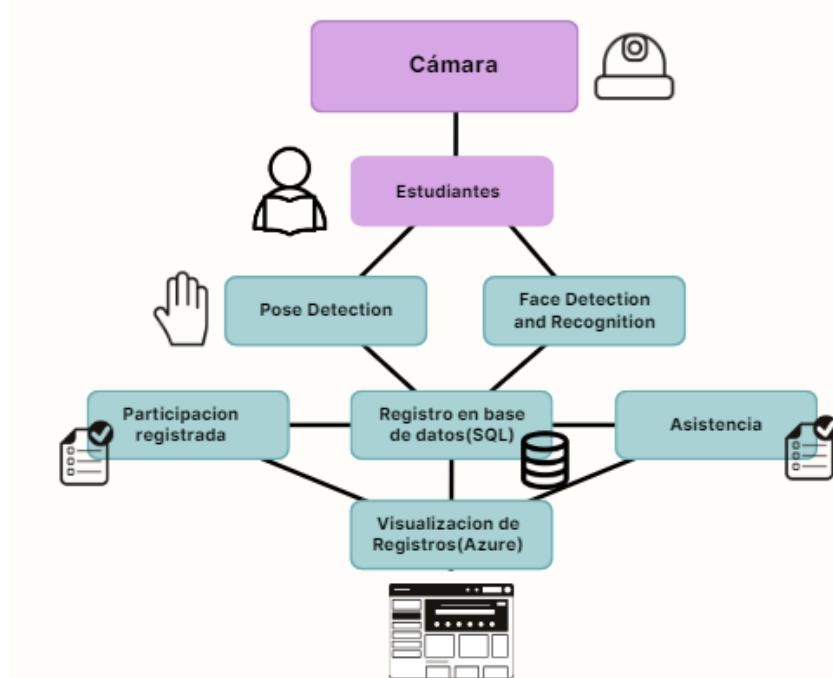


Figura 17: Marco general del sistema propuesto.

## 5. Resultados

Recordando lo anterior respecto a las métricas y como estas son calculadas, realizamos pruebas en ambientes en tiempo real en donde mediamos el rendimiento del sistema a diferentes circunstancias, en general la precisión del modelo se ve afectada a lo que seria la luz, el ambiente, la calidad de la imagen que se usa de referencia para identificar al alumnado, entre otros factores, en general se ha observado que dentro de un ambiente óptimo la integración del modelo puede llegar a precisiones de punto mas alto de 91 % mientras que en momentos puede llegar a precisiones de 60 %, así que en general dependerá ampliamente del entorno en donde se ejecute pero considerando que un aula de clases es bastante sistematizado y controlado creemos que estos resultados son positivos para poder previsualizar la asistencia y la participación que se requiere

en este tipo de proyectos, de igual forma se genero una interfaz para poder activar el modelo en tiempo real de tal forma que cuando un usuario con credenciales autorizadas accede dentro de la plataforma, pueda seleccionar la opción de cámara y ahí el usuario indica para que clase esta tomando asistencia y al final llama al modelo programado para retro alimentar al modelo definido con la cámara que haya seleccionado el usuario. En general se observa que bajo el marco de lo planteado se logro integrar los componentes y los sistemas de una manera funcional el cual ayuda a completar los objetivos previamente planteados en este proyecto.

## 6. Conclusiones

El sistema de reconocimiento facial implementado representa una solución integral y efectiva para la gestión de la asistencia y participación de estudiantes en entornos educativos. La combinación de una interfaz desarrollada en React, un modelo creado en Python y el almacenamiento en la nube mediante Azure ha proporcionado una plataforma completa y accesible. Sin embargo, es fundamental analizar tanto los puntos fuertes como las áreas de mejora del sistema.

### Puntos Fuertes:

**Alto Nivel de Precisión y Eficiencia:** El modelo desarrollado exhibe una precisión significativa al compararse con métodos tradicionales, asegurando una identificación confiable de los estudiantes. La eficiencia del sistema contribuye al ahorro de tiempo, simplificando la toma de asistencia y seguimiento de participación.

**Automatización y Recopilación de Datos:** La automatización del proceso de asistencia y participación proporciona una solución eficaz para la recopilación de datos, permitiendo un análisis detallado del desempeño de los alumnos. Las gráficas generadas ofrecen una visión visual y comprensible de las tendencias y participación.

### Áreas de Mejora:

Privacidad de Datos: La consideración principal radica en la privacidad de los datos, especialmente al tratarse de un sistema de reconocimiento facial. Es crucial implementar medidas de seguridad robustas para garantizar la protección de la información personal de los estudiantes y evitar posibles violaciones de privacidad.

Experiencia del Usuario: Se busca mejorar la experiencia del usuario para agilizar aún más el proceso de toma de asistencia y participación. Esto podría incluir la implementación de funcionalidades intuitivas en la interfaz, así como la optimización de la velocidad y la eficiencia del sistema.

Dashboard Interactivo: Para futuras mejoras, se propone trabajar en la creación de un dashboard interactivo más completo. Esto proporcionaría a los usuarios una visión detallada y personalizable de la información sobre asistencia y participación, facilitando la toma de decisiones y el análisis de datos.

En conclusión, el sistema de reconocimiento facial junto con su interfaz asociada ha demostrado ser una herramienta eficiente y efectiva para la gestión de asistencia y participación de estudiantes en entornos educativos. La combinación de tecnologías, desde React para la interfaz hasta Python para el modelo, respaldado por la capacidad de almacenamiento en la nube de Azure, ha brindado resultados sólidos en términos de funcionalidad y rendimiento.

A pesar de los éxitos obtenidos, es imperativo reconocer que cualquier sistema siempre puede beneficiarse de mejoras continuas. La optimización de la privacidad de los datos, la experiencia del usuario y la exploración de nuevas funcionalidades son áreas que pueden ser objeto de futuras mejoras.

En este momento, el sistema proporciona una solución automatizada y precisa para el seguimiento de la asistencia, aprovechando la potencia de la tecnología de reconocimiento facial. Sin embargo, la búsqueda constante de la excelencia nos lleva a considerar posibles refinamientos y evoluciones para garantizar que el sistema siga cumpliendo con

las expectativas y demandas cambiantes en el ámbito educativo y más allá.

## 7. Anexos

En esta sección se hará referencia a los recursos propios generados y utilizados en el reto.

1: Código del backend - Carpeta raíz: [https://github.com/5100-chap/TC3007C.501\\_EQ6/tree/main/Anexos/backend](https://github.com/5100-chap/TC3007C.501_EQ6/tree/main/Anexos/backend)

2: Código del backend - Carpeta donde se encuentra la clase con los modelos: [https://github.com/5100-chap/TC3007C.501\\_EQ6/tree/main/Anexos/backend/models](https://github.com/5100-chap/TC3007C.501_EQ6/tree/main/Anexos/backend/models)

3: Código del frontend - Carpeta raíz: [https://github.com/5100-chap/TC3007C.501\\_EQ6/blob/main/Anexos/create-react-app/README.md](https://github.com/5100-chap/TC3007C.501_EQ6/blob/main/Anexos/create-react-app/README.md)  
4: Anexos del GitHub - Script de la base de datos: [https://github.com/5100-chap/TC3007C.501\\_EQ6/blob/main/Anexos/dbsqlscript.sql](https://github.com/5100-chap/TC3007C.501_EQ6/blob/main/Anexos/dbsqlscript.sql)

5: Anexos del GitHub - Aquí se podrá encontrar archivos que históricamente se evaluaron: [https://github.com/5100-chap/TC3007C.501\\_EQ6/tree/main/Anexos](https://github.com/5100-chap/TC3007C.501_EQ6/tree/main/Anexos)

6: Repositorio del proyecto: [https://github.com/5100-chap/TC3007C.501\\_EQ6/](https://github.com/5100-chap/TC3007C.501_EQ6/)

## Referencias

- [1] Ageitgey. *The world's simplest facial recognition api for Python and the command line*. URL: [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition).
- [2] Cmu-Perceptual-Computing-Lab. *OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation*. URL: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.
- [3] Davidsandberg. *Face recognition using Tensorflow*. URL: <https://github.com/davidsandberg/facenet>.
- [4] face-recognition. *PyPI*. URL: <https://pypi.org/project/face-recognition/>.
- [5] N. Kasa y H. Watanabe. *A mechanical sensorless control system for salient-pole brushless DC motor with autocalibration of estimated position angles*. URL: <https://ieeexplore.ieee.org/document/836354>.
- [6] University of Massachusetts. *LFW Face Database*. URL: <https://vis-www.cs.umass.edu/lfw/>.
- [7] OpenCV. *OpenCV Face Recognition*. URL: <https://opencv.org/opencv-face-recognition/>.
- [8] A. Rosebrock. *Local Binary Patterns with Python OpenCV - PyImageSearch*. URL: <https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>.
- [9] . Viola y M. Jone. *Rapid object detection using a boosted cascade of simple feature*. URL: <https://ieeexplore.ieee.org/document/990517>.