UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

Implementación de un Sistema de Reconocimiento Visual para Mejorar la Interacción en el Robot Social UDAbot

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autor:

CHRISTIAN MAX PEÑA VILLALBA

Director:

DANIEL ESTEBAN ITURRALDE PIEDRA

CUENCA, ECUADOR

2026

DEDICATORIA

Dedico esta tesis a mi querida madre, Ruth Villalba, quien ha sido un pilar fundamental a lo largo de mi vida. Su apoyo incondicional durante mi formación académica y en cada etapa personal ha sido clave para alcanzar este logro. Gracias a sus enseñanzas, valores y ejemplo de perseverancia, he aprendido a enfrentar los desafíos con responsabilidad, humildad y determinación.

A mi hermana, Angélica, quien ha sido una fuente constante de inspiración en mi desarrollo personal y profesional. Su esfuerzo, dedicación y visión de superación me han motivado a seguir creciendo y a plantearme metas cada vez más altas. Su influencia ha sido esencial en mi camino hacia este logro.

Al amor de mi vida, Fabiola Boni, por su compañía incondicional a lo largo de este proceso. Su apoyo, paciencia y confianza han sido fundamentales en los momentos más exigentes. Gracias por estar siempre presente, por creer en mí, brindándome la fortaleza necesaria para culminar esta etapa.

Christian Max Peña Villalba

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a la universidad por haber contribuido a mi formación profesional, brindándome no solo los conocimientos técnicos necesarios, sino también un entorno que fomentó el pensamiento crítico y el desarrollo integral. De igual manera, agradezco el apoyo del personal académico y administrativo, quienes, constantemente, contribuyen a elevar el nivel estudiantil.

Extiendo mi profundo agradecimiento al Ing. Daniel Iturralde, PhD, tutor de este trabajo, por su guía permanente a lo largo de todo el proceso. Su experiencia, conocimientos y observaciones fueron fundamentales para superar dificultades e impulsar el desarrollo de esta investigación.

Finalmente, agradezco a todos los docentes que formaron parte de mi proceso académico, quienes con sus enseñanzas, consejos y exigencia contribuyeron a forjar mis aptitudes profesionales. Gracias a ellos, no solo adquirí conocimientos, sino también valores y criterios que serán esenciales en mi desempeño como futuro profesional.

IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO VISUAL PARA MEJORAR LA INTERACCIÓN EN EL ROBOT SOCIAL UDABOT

En el ámbito de la robótica social, la interacción con el usuario constituye un área en constante evolución. Los sistemas de reconocimiento facial permiten mejorar dicha interacción; sin embargo, su implementación en sistemas embebidos representa un desafío computacional significativo, lo que hace necesario el uso de hardware especializado. El sistema desarrollado emplea el modelo SSD_MobileNet, dedicado para el sensor IMX500, para la detección y clasificación de objetos, filtrando la clase persona. Posteriormente, se utilizan modelos de MediaPipe para la detección y corrección de la alineación facial, en conjunto con ArcFace para la comparación de identidades. El sistema alcanza una exactitud del 72% y una precisión del 94% en tiempo real, con un consumo reducido de recursos. Estos resultados demuestran la viabilidad de implementar soluciones de inteligencia artificial en plataformas embebidas orientadas a la robótica social.

Palabras clave: Robot social, Reconocimiento facial, Inteligencia artificial, Raspberry Pi.

IMPLEMENTATION OF A VISUAL RECOGNITION SYSTEM FOR ENHANCED INTERACTION IN THE UDABOT SOCIAL ROBOT

In the field of social robotics, user interaction is a continuously evolving area. Facial recognition systems enable improvements in such interaction; however, their implementation on embedded systems poses a significant computational challenge, requiring the use of specialized hardware. The proposed system employs the SSD MobileNet model, optimized for the IMX500 sensor, to perform object detection and classification, specifically filtering the person class. Subsequently, MediaPipe models are utilized for facial landmark detection and alignment, in combination with ArcFace for identity verification. The system achieves 72% accuracy and 94% real-time precision while maintaining low resource consumption. These results demonstrate the feasibility of implementing artificial intelligence solutions on embedded platforms for social robotics applications.

Keywords: Social robot, Facial recognition, Artificial intelligence, Raspberry Pi

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Implementación de un Sistema de Reconocimiento Visual para Mejorar la Interacción en el Robot Social UDAbot

Christian Peña
Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
max_gford@es.uazuay.edu.ec

***Resumen*—En el ámbito de la robótica social, la interacción con el usuario constituye un área en constante evolución. Los sistemas de reconocimiento facial permiten mejorar dicha interacción; sin embargo, su implementación en sistemas embebidos representa un desafío computacional significativo, lo que hace necesario el uso de hardware especializado. El sistema desarrollado emplea el modelo SSD_MobileNet, dedicado para el sensor IMX500, para la detección y clasificación de objetos, filtrando la clase persona. Posteriormente, se utilizan modelos de MediaPipe para la detección y corrección de la alineación facial, en conjunto con ArcFace para la comparación de identidades. El sistema alcanza una exactitud del 72% y una precisión del 94% en tiempo real, con un consumo reducido de recursos. Estos resultados demuestran la viabilidad de implementar soluciones de inteligencia artificial en plataformas embebidas orientadas a la robótica social.**

***Palabras clave*—Robot social, Reconocimiento facial, Inteligencia artificial, Raspberry Pi.**

I. Introducción

En la actualidad, la aplicación de robots abarca una amplia variedad en áreas industriales, sociales y educativas. El uso más común se da en la automatización industrial, donde se alcanzó un promedio mundial de 162 unidades robóticas por cada 10.000 empleados en 2023. El país líder es la República de Corea, con un promedio de 1.012 unidades, gracias a su fuerte presencia en la industria automotriz. Este crecimiento refleja un incremento anual entorno al 6%, evidenciando la rápida adopción de la robótica en el ámbito industrial, que se expande hacia otras áreas de interés, impulsada por la reducción de costos, la digitalización y la creciente demanda de automatización inteligente [1].

Los robots sociales no poseen una definición bien establecida, debido a la falta de consenso acerca de sus funciones y sobre qué los hace realmente sociales. Diversos autores plantean parámetros destacables como el aspecto físico, las habilidades sociales, el grado de autonomía, el nivel de inteligencia, la distancia y duración de la interacción, así como el entorno de uso. Dependiendo del objetivo de la aplicación, existen robots destinados a entornos educativos, terapéuticos, laborales o de acompañamiento. No obstante, todavía persiste una brecha entre la teoría y la implementación práctica, especialmente en lo referente a la interacción humano-robot, lo que genera dificultades en el diseño general, la apariencia, la comunicación y la viabilidad técnica. Por ello, diversos proyectos actuales se enfocan en optimizar diseños y recursos computacionales para lograr una mayor aceptación social [2].

El desarrollo de distintos sistemas basados en inteligencia artificial para la interacción humano-robot es un tema de gran interés para los investigadores, y los resultados han mejorado desde que se cambió el enfoque hacia métodos de aprendizaje profundo, en comparación con los antiguos métodos tradicionales. En busca de implementar capacidades esenciales para los robots de asistencia social (SAR), las investigaciones se centran principalmente en tareas como la detección de rasgos faciales, la postura del cuerpo o de la cabeza, el seguimiento de la mirada, el procesamiento del habla y el desempeño de diversas funciones. Sin embargo, la efectividad de estos modelos varía según el entorno de aplicación, ya que los robots deben adaptarse a distintos contextos sociales, ya sean educativos, clínicos u otros. Este énfasis contextual podría garantizar una mayor confiabilidad y efectividad de los SAR para lograr interacciones más naturales [3].

Los sistemas embebidos como la Raspberry Pi presentan ciertas dificultades para integrar modelos de inteligencia artificial de manera eficiente debido al peso computacional y su construcción basada en redes neuronales, esto se refleja como errores en la precisión o latencias altas si no se adaptan correctamente. Hay diversas soluciones como dispositivos para optimizar y acelerar el procesamiento, o cámaras con circuitos integrados para la ejecución directa de modelos de IA y reducir el consumo de recursos del resto del sistema [4].

El considerable desarrollo de la inteligencia artificial y de las técnicas de procesamiento de imágenes basadas en redes neuronales convolucionales han impulsado significativamente las capacidades de los robots sociales, ampliando su campo de aplicación en búsqueda de establecer una interacción humano-robot cada vez más natural, precisa y eficiente. En este contexto, resulta fundamental realizar una revisión de los avances recientes para identificar limitaciones y áreas de mejora en el diseño de robots sociales.

En el artículo [5] se desarrolló un sistema de control parental

para un prototipo de robot de asistencia social (SAR), construido por Robotis Bioloid, cuyo hardware incluía un procesador Intel NUC Core i3-8109U, cámara web, micrófonos, conectividad Wi-Fi y redes móviles. Se entrenaron modelos basados en redes neuronales convolucionales para la detección de objetos y rostros, mientras que para el asistente de voz se utilizó la API de Google Assistant. El robot tenía la capacidad de identificar cuatro objetos, tres rostros e interpretar nueve comandos de voz mediante palabras clave. Cada interacción con el SAR se registraba y enviaba a un Bot de Telegram a través de la conexión Wi-Fi. Los resultados reportaron una precisión del 85.41%.

En la publicación [6] se realizó una comparación entre diferentes modelos para la detección de rostros: varios basados en redes neuronales convolucionales (CNN, R-CNN, Fast R-CNN, Faster R-CNN) y los modelos YOLO (V1, V2, V3). Los modelos de la familia R-CNN resultaron ser demasiado complejos y lentos en comparación con los de YOLO, lo cual representa un aspecto crítico para sistemas de reconocimiento facial en tiempo real. En los experimentos realizados, los modelos R-CNN alcanzaron una velocidad de 7 FPS, mientras que YOLOv3 logró ejecutarse a 45 FPS.

En [7] se implementó un sistema de reconocimiento facial en el robot social OhBot, utilizando modelos Tiny-ML optimizados para microprocesadores como Raspberry Pi. Con el fin de mejorar el rendimiento se integró un Intel Neural Stick 2 para optimizar y acelerar el procesamiento de modelos de IA en robots sociales. Se evaluaron diferentes combinaciones de modelos Tiny-ML (YOLOv4, YOLOv5s) junto con el acelerador neural, obteniendo los mejores resultados con YOLOv5s y el Neural Stick, alcanzando una velocidad de procesamiento de 17 FPS. Finalmente, se implementó un sistema de interacción con usuarios para conversar, sugerir o corregir el uso de la mascarilla.

En [8] se presentó un proyecto con el robot antropomórfico de servicio doméstico y de atención médica CHARMIE, en el cual se implementó un sistema de reconocimiento facial 3D mediante una cámara Intel RealSense D455. Este dispositivo permite capturar de forma simultánea imágenes RGB y mapas de profundidad. Para la etapa de clasificación se entrenaron dos redes neuronales convolucionales, una para cada tipo de imagen, utilizando los entornos TensorFlow y Keras. La incorporación del mapa de profundidad tuvo como propósito incrementar la robustez del sistema, evitando intentos de suplantación a través de fotografías impresas. Las pruebas reportaron un rendimiento promedio del 94% con imágenes RGB y del 72% con mapas de profundidad.

En [9] se presentó la construcción de un robot humanoide social denominado Okao Vision, diseñado con capacidades de interacción social, asistencia, procesamiento de imágenes y expresión emocional. Para el reconocimiento facial y análisis de gestos se incorporó una cámara OMRON HVC-P, la cual incluye el procesamiento embebido que permite identificar rostros, localizar el cuerpo humano y analizar expresiones faciales en el mismo módulo del sensor. El control del robot se implementó mediante una placa Arduino, integrando seis servomotores que permitieron la gesticulación de las cejas, ojos, parpados y un panel led para la boca, además contaba con la posibilidad de entablar conversaciones con los usuarios y reaccionar a las emociones.

En [10] se desarrolló un sistema multimodal de interacción afectiva bidireccional empleando al robot social Pepper junto con el entorno de diálogo abierto Rasa. El sistema fue diseñado para reconocer y responder a emociones humanas mediante la fusión de tres canales de entrada: voz, expresiones faciales y dirección de la mirada. Haciendo uso de su pantalla táctil el robot generaba repuestas emocionales combinando voz, gestos y emojis. El proyecto planteó como una solución accesible, de código abierto y fácil de implementar. Los resultados de la evaluación reflejaron una precisión cercana al 89%.

El resto del artículo se organiza de la siguiente manera: la Sección II describe el sistema implementado y la metodología utilizada; la Sección III presenta los resultados obtenidos así como la evaluación del rendimiento del sistema propuesto; finalmente, en la Sección IV se exponen las conclusiones del trabajo.

II. Metodología

El presente proyecto como se indica en la Fig.1 responde a la necesidad de integrar capacidades de procesamiento visual en el robot social UDAbot, mediante la implementación de la Raspberry Pi AI Camera. Esta incorporación permitirá dotarlo de funciones avanzadas de reconocimiento del entorno, detección de objetos y personas, así como de identificación del personal docente, optimizando su interacción con los usuarios y su entorno. El uso de la AI Camera con procesamiento basado en inteligencia artificial posibilita la ejecución de diversos modelos de detección de manera local, reduciendo la dependencia de servicios en la nube y mejorando la eficiencia temporal del sistema. Se propone el desarrollo e integración de un módulo de reconocimiento visual en operación continua que utilice modelos de librerías de Pi AI Camera y MediaPipe para diversas tareas de detección de elementos en el entorno e identificación de personas, en caso de coincidencia iniciará la interacción social. Este desarrollo busca consolidar la integración del UDAbot en el entorno universitario, fomentar el interés de los estudiantes por la investigación y aplicación de tecnologías de visión artificial y robótica social.

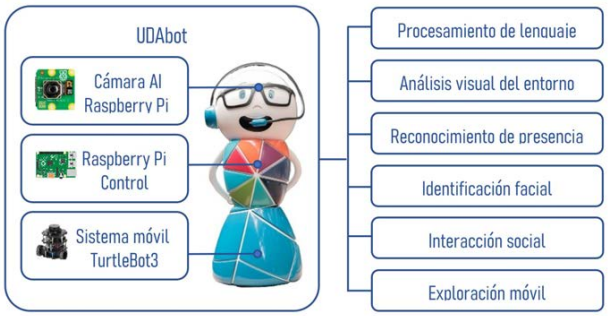

Fig. 1. Partes y funciones del UDAbot

En la Fig. 2 se presenta el diagrama de flujo general del sistema, en el cual se describe la estructura y la secuencia lógica de las etapas necesarias para su correcto funcionamiento. Dicho diagrama permite visualizar el proceso de adquisición y procesamiento de imágenes capturadas por la Raspberry Pi AI Camera, así como la ejecución de los modelos de visión artificial encargados de la detección de objetos, personas y reconocimiento facial.

A. Raspberry AI Camera

En la primera etapa se hace uso de la Raspberry Pi AI Camera, la cual está basada en el sensor IMX500 que integra un procesador de señal de imagen y un acelerador de inteligencia artificial en el propio módulo de la cámara, tal como se observa en el diagrama de la Fig. 3. El procesador de señal se encarga de transformar la imagen capturada en un vector de entrada, aplicando procesos de recorte y escalado según los requerimientos del modelo cargado. Posteriormente, la inferencia se ejecuta directamente en la cámara, generando vectores de salida que contienen los resultados y que son enviados a la Raspberry Pi [11].

B. Detección de objetos

En esta etapa, el módulo de la cámara hace uso del chip IMX500 fabricado por Sony, el cual dispone de distintos modelos preentrenados y optimizados para el postprocesamiento de imágenes [12]. Estos modelos permiten la ejecución de diversas tareas de visión artificial, tales como:

- Clasificación de imágenes.
- Detección de objetos.
- Segmentación semántica.
- Estimación de pose.

El proyecto se construye sobre la detección de objetos, para la cual existen varios modelos Tiny-ML optimizados para este hardware, como SSD MobileNet y YOLO, entre otros. Dichos modelos han sido entrenados utilizando el conjunto de datos COCO (Common Objects in Context), que contiene más de 118,000 imágenes y 80 clases etiquetadas. En la Fig. 4 se describe el funcionamiento de la etapa de detección de objetos, cuyo objetivo es filtrar la clase persona, la cual será procesada en etapas posteriores, considerando una capacidad ajustable de hasta cinco personas para su identificación.

C. Detección de rostros

Como se aprecia en la Fig. 5, una vez que la etapa de detección de objetos ha identificado y recortado la región correspondiente a una persona, esta subimagen se envía al módulo de detección de rostros para su análisis detallado. Para este fin se emplea el modelo MediaPipe Face Detector, un modelo de aprendizaje automático diseñado para localizar regiones faciales y generar cuadros delimitadores de uno o varios rostros dentro de una imagen [13]. Se contempla el escenario en el que se detecte una persona cuyo rostro no sea visible; asimismo, en caso de que se identifiquen más de cinco personas, se seleccionan los rostros más cercanos de acuerdo con la proximidad medida a partir de las dimensiones

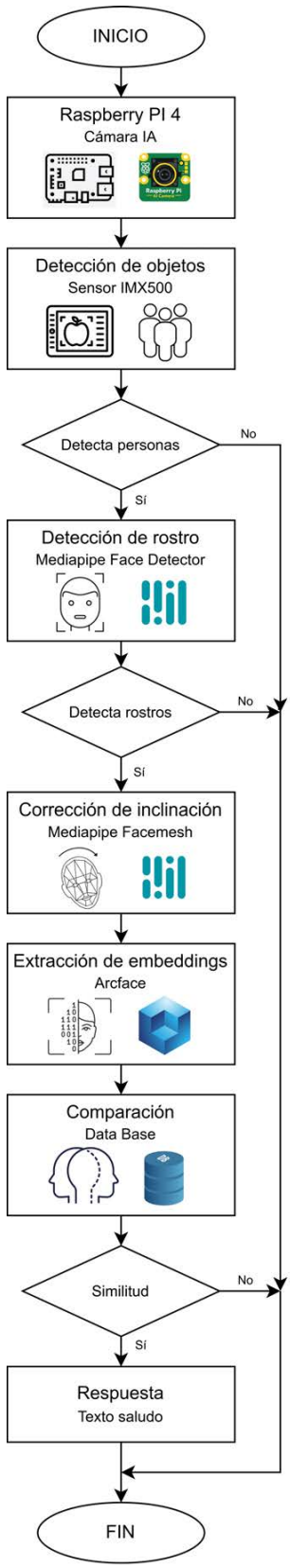

Fig. 2. Diagrama de flujo del funcionamiento del sistema

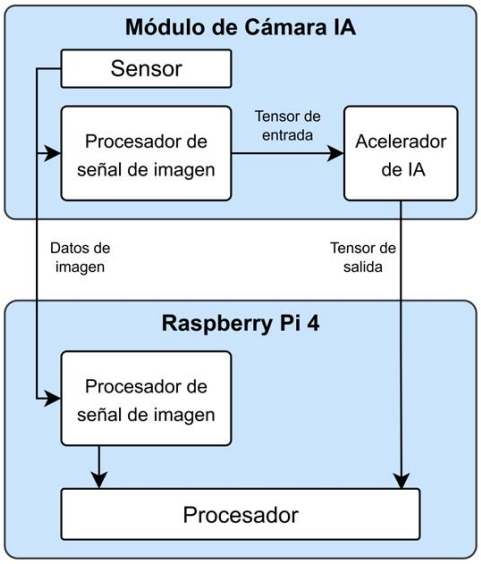

Fig. 3. Diagrama de funcionamiento de Raspberry IA Camera

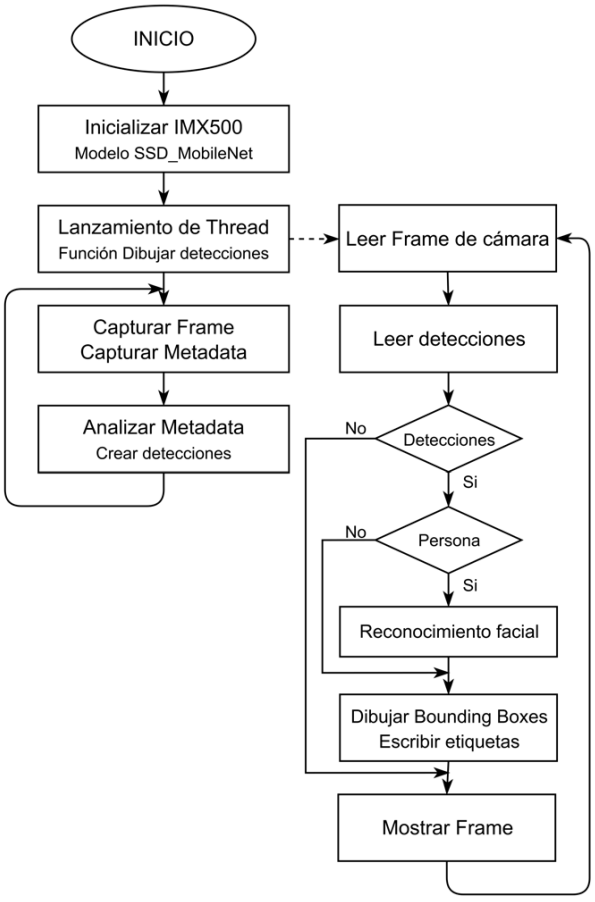

Fig. 4. Diagrama de flujo de la detección de objetos

del recuadro facial. La salida de esta etapa proporciona las coordenadas de los rostros detectados, las cuales son necesarias para las fases posteriores de comparación facial.

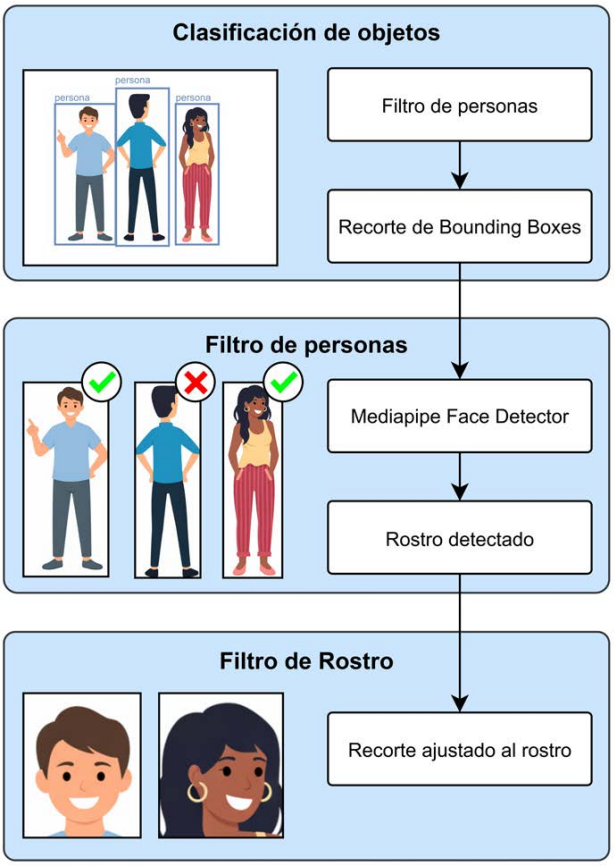

Fig. 5. Diagrama de detección de rostro

D. Corrección de inclinación de rostro

En esta etapa se corrige la orientación del rostro para normalizar su inclinación antes de continuar con la extracción de características faciales. A partir de los rostros detectados, se utiliza MediaPipe Face Landmarker, el cual es un modelo que genera un conjunto de 478 puntos de referencia tridimensionales distribuidos en la superficie facial, como se puede apreciar en la Fig. 6 [14].

Como indica la Fig 7. Para la corrección de la inclinación del rostro se emplea un proceso de alineación facial basado en cinco puntos de referencia, correspondiente al estándar utilizado por ArcFace. A partir de los landmarks detectados (ojos, nariz y comisuras de la boca), se calcula una transformación geométrica que ajusta el rostro a una plantilla normalizada de tamaño 112×112 píxeles [15]. Este procedimiento permite corregir rotaciones e inclinaciones del rostro, garantizando una orientación frontal consistente. La normalización obtenida mejora la estabilidad y precisión de la posterior extracción de embeddings faciales y su comparación.

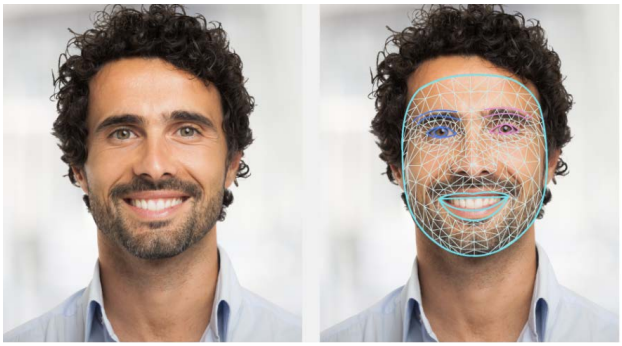

Fig. 6. MediaPipe Face Landmarks - Puntos de referencia

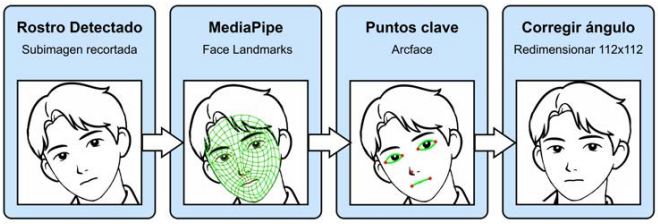

Fig. 7. Diagrama de corrección de inclinación de rostro

E. Extracción de embeddings - Arcface

En esta etapa como se ilustra en la Fig 8. Se realiza la extracción de embeddings faciales a partir de los rostros previamente alineados. Para ello se emplea un modelo de reconocimiento facial basado en ArcFace, el cual corresponde a una red neuronal profunda que genera un vector de características de alta dimensión, capaz de representar de forma compacta la identidad facial [15]. Este proceso genera un archivo ligero de aproximadamente 2 Kb tipo *.npy* el cual es el formato nativo de NumPy para guardar arreglos de float32, es más compacto, preciso y mucho más fácil de cargar en comparación con otros tipos como *.txt* o *.json*.

El modelo ArcFace introduce una función de pérdida basada en distintos parámetros de clasificación, similaridad, separación y compactación, como se describe en la Ecuación (1),

$$L_{\text{ArcFace}} = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1,\, j\neq y_i}^{N} e^{s\cos(\theta_j)}} \tag{1}$$

donde y_i es la clase correcta asociada, N el número total de identidades, y θ_j el ángulo entre el embedding y el vector de la clase j. El parámetro s es un factor de escala que amplifica el valor del coseno, mientras que m es el margen angular aditivo aplicado al ángulo de la clase correcta θ_{y_i}. ArcFace normaliza tanto los vectores de características como los pesos de clasificación, de modo que la decisión dependa únicamente del ángulo entre ellos, además de incorporar un margen angular que fuerza una mayor compactación intra-clase y una mayor separación inter-clase sobre una hiperesfera normalizada. Este enfoque permite obtener embeddings faciales más robustos frente a variaciones de posiciones, iluminación y expresión, mejorando significativamente el desempeño en tareas de reconocimiento facial.

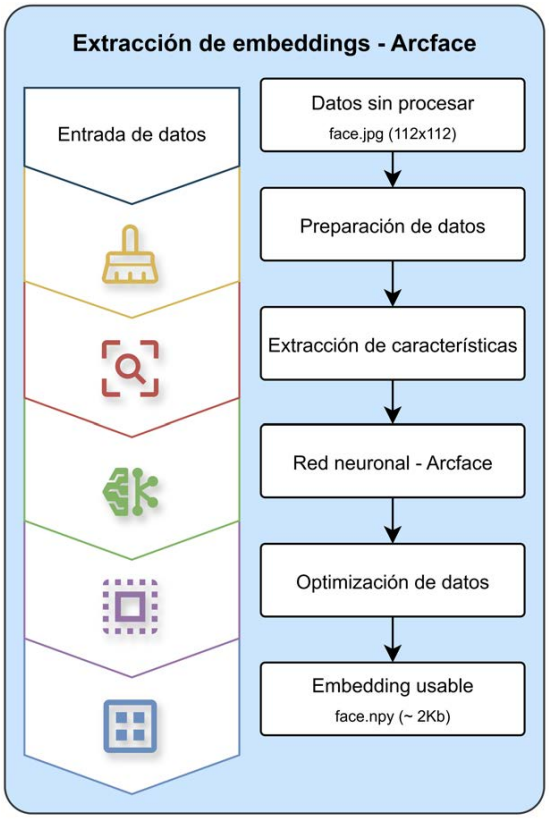

Fig. 8. Diagrama de extracción de embeddings

F. Comparación facial

Una vez obtenidos los embeddings faciales, y dependiendo de la cantidad de personas detectadas en el instante de captura del frame, se procede a la etapa de comparación individual con la base de datos previamente registrada y cargada en memoria RAM al inicio del programa, con el fin de optimizar los tiempos de respuesta. Cada embedding extraído es contrastado con los vectores almacenados mediante una métrica de similitud, empleando la similitud coseno como criterio principal de comparación. Dado que los embeddings generados por ArcFace se encuentran normalizados, pueden considerarse vectores unitarios distribuidos sobre una hiperesfera como se aprecia en la Fig 9.

La similitud coseno entre dos embeddings es equivalente a su producto punto [15]. Además, para vectores normalizados, la distancia euclidiana está directamente relacionada con el ángulo entre ellos, como se aprecia en la Ecuación (2)

$$\|A - B\|^2 = 2 - 2\cos\theta \tag{2}$$

donde θ es el ángulo entre los vectores A y B. Por lo tanto, maximizar la similitud coseno equivale a minimizar la distancia euclidiana, lo que justifica el uso de la métrica angular

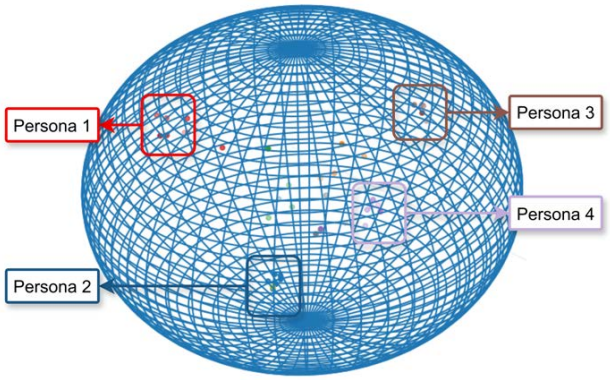

Fig. 9. Hiperesfera - Distribución de embeddings faciales

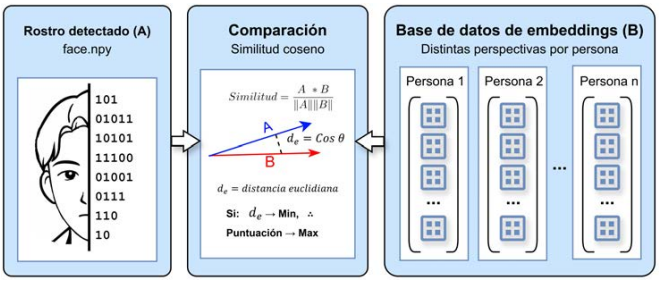

Fig. 10. Diagrama de comparación de embeddings

como criterio de comparación en el proceso de identificación facial como indica la Fig 10.

La puntuación obtenida corresponde a un valor entre el rango de 0 a 1, teniendo como criterio de interpretación los rangos definidos en la Tabla I. El umbral para considerar un reconocimiento facial confiable en tiempo real se estima a partir del 0.6.

TABLA I
CRITERIO DE INTERPRETACIÓN DE LA SIMILITUD COSENO

Rango de puntuación	Interpretación
0.00 - 0.20	Coincidencia nula
0.21 - 0.40	Similitud baja
0.41 - 0.60	Similitud media
0.61 - 0.80	Similitud alta
0.81 - 1.00	Similitud muy alta

III. RESULTADOS

En esta sección se presenta la implementación del sistema de registro de usuarios, así como los resultados experimentales obtenidos durante la validación del reconocimiento facial.

A. Sistema de registro y base de datos

Para complementar el correcto funcionamiento del sistema de identificación facial, se implementó un módulo de registro encargado de capturar, procesar y almacenar las características faciales de nuevos usuarios en una base de datos estructurada.

1) Registro de usuarios: El módulo desarrollado consiste en una interfaz visual, como se muestra en la Fig. 11, cuyo propósito es facilitar el proceso de registro de usuarios. Cada imagen capturada atraviesa las etapas previamente descritas: detección de rostro, corrección de inclinación y extracción de embeddings.

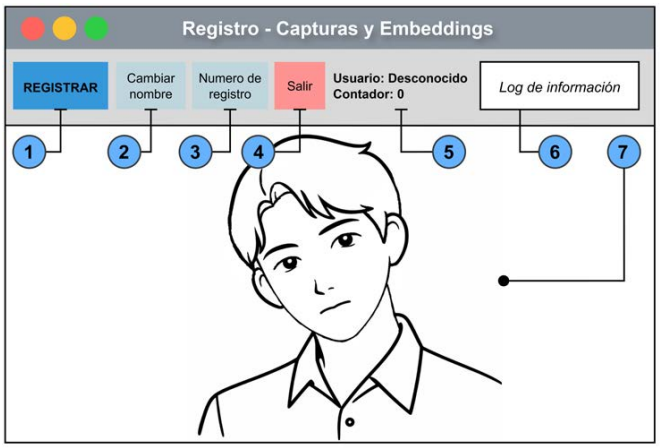

Fig. 11. Interfaz visual de registro de usuario.

Los elementos que conforman la interfaz son:

1) Detecta y almacena el embedding extraído; también permite registrar mediante la tecla “Enter”.
2) Permite asignar o modificar el nombre del usuario a registrar.
3) Permite cambiar el índice del embedding actual.
4) Cierra el programa; también puede utilizarse la tecla “ESC”.
5) Muestra el usuario actual y el contador de muestras registradas; al cambiar de usuario el contador se reinicia.
6) Registra los eventos generados durante el proceso.
7) Visualización de la Cámara IA.

2) Base de datos de embeddings: Para la construcción de la base de datos se definió un esquema en el cual se almacenan seis embeddings por cada usuario. Estos corresponden a distintas perspectivas o ángulos del rostro como se indica en la Tabla. II, con el objetivo de capturar una mayor variabilidad de características por persona y mejorar la robustez del sistema ante cambios de posiciones, evaluaciones específicas con ArcFace muestran que un conjunto de posiciones cercanas a - 45°, 0° y 45° es suficiente para lograr un reconocimiento robusto sin necesidad de incluir perspectivas de perfiles extremos, ademas que interferiría con la corrección del ángulo en los procesos de MediaPipe.

Para cada archivo se le corresponderá un nombre de usuario y un indice de la siguiente manera *usuario_indice.npy*, permitiendo mantener un orden estructurado de las muestras. Los ángulos propuestos para el registro, como se ilustra en la Fig 12, son:

TABLA II
PERSPECTIVAS DEFINIDAS PARA EL REGISTRO DE EMBEDDINGS

Índice	Perspectiva del rostro
1	Vista frontal
2	Giro derecha 30°
3	Giro derecha 45°
4	Vista frontal superior 20°
5	Giro izquierda 30°
6	Giro izquierda 45°

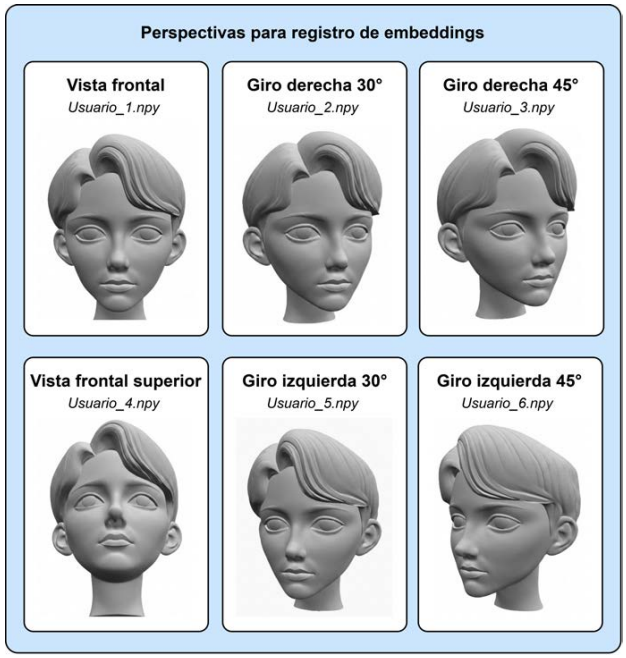

Fig. 12. Perspectivas para registro de embeddings.

B. Funcionamiento del módulo de reconocimiento visual

El robot social UDAbot previamente cuenta con un sistema de movilidad e interacción. Sobre esta infraestructura se integra el sistema de reconocimiento facial, permitiendo que ambos trabajen de manera coordinada mediante eventos de activación, como por ejemplo el uso de *wakewords* para iniciar el proceso de identificación. Con este propósito, el sistema fue desarrollado siguiendo una arquitectura modular, con el fin de minimizar el consumo computacional y evitar interferencias con las funciones previamente implementadas en el robot.

En la Fig. 13 se ilustra el diagrama de flujo correspondiente a la integración del sistema de reconocimiento visual dentro del funcionamiento general del UDAbot. Durante el arranque del sistema se ejecuta la inicialización de los modelos de inteligencia artificial, como MediaPipe FaceMesh, Face Landmarker y ArcFace, así como las librerías de Picamera y la configuración del modelo SSD_MobileNet para la Raspberry Pi AI Camera.

Adicionalmente, el programa hace uso de *threads* (hilos) para evitar la interrupción de procesos en el núcleo principal del robot. De esta manera, se actualizan continuamente los datos de la lista de personas detectadas con las similitudes más aproximadas y sus respectivos puntajes. En caso de no detectarse ningún rostro, los datos se eliminan y la lista permanece vacía. Esta información puede consultarse desde el proceso principal para su debida gestión.

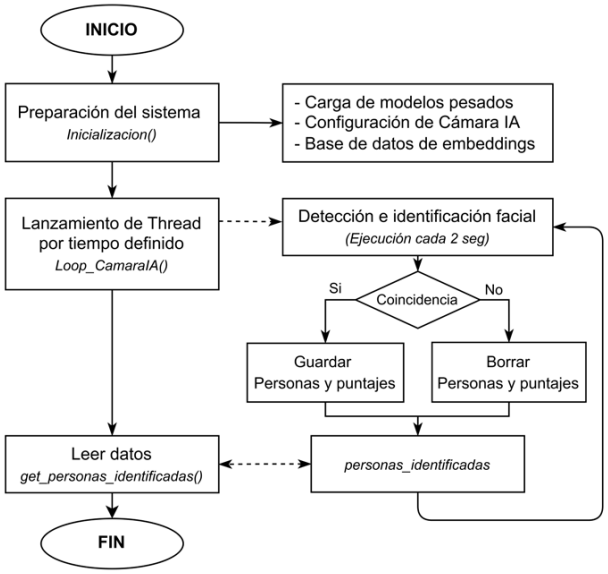

Fig. 13. Diagrama de integración del módulo de reconocimiento facial.

C. Pruebas de funcionamiento

Para la etapa de pruebas se tomaron en cuenta distintos parámetros, como el consumo de recursos de CPU, GPU y RAM, en función de los elementos de dibujado en pantalla, así como la precisión y exactitud del sistema, la cantidad de personas presentes en escena, las condiciones de iluminación y los tiempos de respuesta.

1) Rendimiento de CPU, GPU y RAM: Para optimizar el consumo de recursos del programa se establecieron dos versiones durante el desarrollo del sistema. En la primera se hizo uso de una versión inicial que incluía el dibujado de recuadros con sus respectivas etiquetas, con el objetivo de comprobar el funcionamiento del modelo SSD_MobileNet en la Raspberry Pi AI Camera.

Posteriormente, en la versión 2 se eliminó el procesamiento de dichos recuadros y de la interfaz visual, manteniendo únicamente las funciones esenciales para la ejecución del sistema. Dentro del funcionamiento general se distinguen dos fases para el reconocimiento facial: en la primera se detecta la presencia de rostros, y en la segunda se procede con la comparación facial. Se realizó una contraste del rendimiento entre ambas versiones, como se indica en la Fig. 14, para la etapa de detección de rostros y en la Fig. 15, para la etapa de comparación facial, obteniendo resultados favorables en el consumo de recursos del sistema.

2) Precisión y exactitud del sistema: En la Fig. 16, se ilustra una captura durante las pruebas de funcionamiento

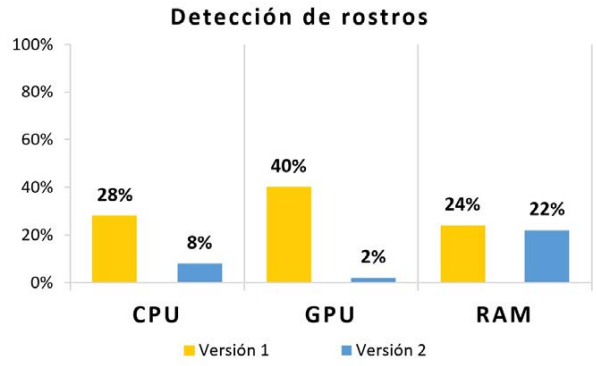

Fig. 14. Comparación del rendimiento durante la detección de rostros.

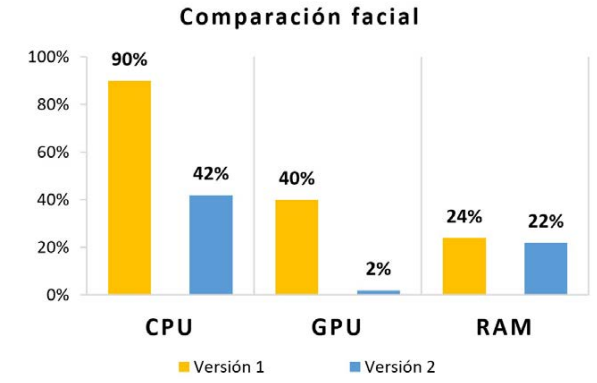

Fig. 15. Comparación del rendimiento durante la identificación facial.

del sistema con los participantes y los resultados obtenidos en ese momento. Dentro de los parámetros de configuración del modelo SSD_MobileNet se estableció un umbral bajo con el fin de mejorar la detección de personas, en procesos subsecuentes del sistema los posibles falsos positivos se filtran. Como se indica en la Fig. 17, bajo condiciones de iluminación regular y con la presencia de cinco participantes se obtuvo una precisión del 94% y una exactitud del 72%. Estos valores fueron obtenidos a partir del promedio general de los aciertos y de las puntuaciones registradas para todos los participantes durante las pruebas realizadas bajo el mismo escenario.

De la misma manera, se realizaron pruebas bajo condiciones de iluminación limitada, obteniendo una precisión del 86% y una exactitud del 64%. El valor de exactitud obtenido se mantiene dentro del rango considerado como similitud alta según el criterio establecido en la Tabla 1.

3) Tiempos de respuesta: Durante la recolección de datos también se registraron los tiempos transcurridos en cada proceso de identificación facial. Tras realizar un promedio general de todas las pruebas, se obtuvieron dos tiempos de respuesta principales. El primero corresponde a la etapa de detección y filtrado de rostros, con un tiempo aproximado de 52 ms, debido a que gran parte de este proceso se realiza dentro de la Cámara IA. El segundo tiempo corresponde al proceso de comparación

Fig. 16. Pruebas de funcionamiento.

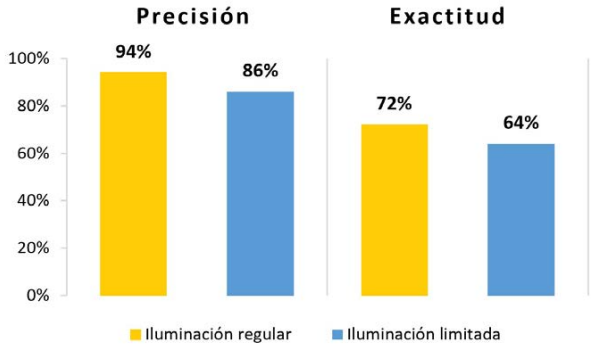

Fig. 17. Resultados de exactitud y precisión del sistema.

facial por cada usuario presente en escena, con un promedio de 675 ms. En consecuencia, a medida que aumenta el número de personas presentes en escena, el tiempo total requerido para obtener los resultados también se incrementa.

IV. Conclusiones

En el presente proyecto se implementó un módulo para el reconocimiento del entorno e identificación facial, con el fin de mejorar la interacción del robot social UDAbot en el entorno universitario. A partir del desarrollo realizado, se establecen las siguientes conclusiones.

La recolección de información permitió identificar las opciones más viables para el desarrollo del sistema, destacando el uso de modelos específicos como ArcFace y MediaPipe, adaptados para su ejecución en Raspberry Pi. Librerías completas como DeepFace resultan demasiado demandantes

para sistemas embebidos. Asimismo, se analizaron modelos compatibles con la Raspberry Pi AI Camera, basada en el sensor IMX500 desarrollado por Sony. En caso de requerir modelos para distintas aplicaciones, se podrían entrenar haciendo uso de su plataforma propietaria AITRIOS. Dentro de las alternativas para detección de objetos se evaluaron los modelos SSD_MobileNet y YOLO; sin embargo, debido a su configuración más sencilla y a un rendimiento ligeramente superior, se optó por el primero.

Durante la etapa de pruebas se establecieron diferentes estrategias para mejorar el proceso de identificación facial. Con el fin de evitar el solapamiento entre personas, se ajustó el umbral de detección de objetos a 0.45; si bien esto genera falsos positivos en etapas iniciales, estos son correctamente filtrados en fases posteriores del sistema, logrando un rendimiento general adecuado. En el proceso de registro de usuarios se observó una mejora en los puntajes de comparación cuando las muestras eran capturadas con la misma cámara del sistema. Asimismo, se evidenció una variabilidad en los resultados cuando los usuarios utilizaban lentes, atribuida a imprecisiones del modelo MediaPipe FaceMesh en la localización de los puntos claves de ojos y cejas, reduciendo aproximadamente en un 10% en el puntaje de similitud. También se determinaron los ángulos de perspectiva desde - 45° a 45°, rango en los cuales la identificación se mantiene confiable, observándose que vistas de perfil generan mayores errores debido a dificultades en la correcta alineación del rostro.

Debido a las limitaciones de hardware, se implementaron diversas optimizaciones, como la carga de la base de datos de embeddings en memoria RAM. Dado su reducido tamaño, esta estrategia no representa una carga computacional significativa, evitando errores de lectura de archivos y permitiendo obtener tiempos de respuesta satisfactorios. El sistema alcanzó un promedio de exactitud del 72% y una precisión del 94%. Los tiempos de comparación podrían mejorarse mediante el uso de hardware más potente o plataformas especializadas para la ejecución de modelos de inteligencia artificial en sistemas embebidos.

Finalmente, la implementación de este módulo representa un avance significativo en las capacidades del robot social UDAbot, integrándose de manera efectiva con el sistema de interacción previamente desarrollado.

Referencias

[1] I. F. of Robotics, "Global robot density in factories doubled in seven years," 2024, accessed: 2025-11-22. [Online]. Available: https://ifr.org/ifr-press-releases/news/global-robot-density-in-factories-doubled-in-seven-years

[2] A. Henschel, G. Laban, and E. Cross, "What makes a robot social? a review of social robots from science fiction to a home or hospital near you," *Current Robotics Reports*, vol. 2, pp. 9–19, 2021.

[3] B. Salamat Ravandi, I. Khan, P. Gander, and R. Lowe, "Deep learning approaches for user engagement detection in human-robot interaction: A scoping review," *International Journal of Human–Computer Interaction*, vol. 41, no. 20, pp. 13 074–13 092, 2025.

[4] S. Bhoopalan, P. Ragunath, K. Sanjay, and M. Subash, "Adaptive autonomous assistance using raspberry pi," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 3, no. 3, pp. 734–739, Mar 2025.

[5] D. H. Fuadi, D. Novita, and M. Taufik, "Socially assistive robot interaction by objects detection and face recognition on convolutional neural network for parental monitoring," in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, 2021, pp. 1–6.

[6] S. M. M, A. Geroge, A. N, and J. James, "Custom face recognition using yolo.v3," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 2021, pp. 454–458.

[7] E. Probierz, N. Bartosiak, M. Wojnar, K. Skowroński, A. Gałuszka, T. Grzejszczak, and O. Kedziora, "Application of tiny-ml methods for face recognition in social robotics using ohbot robots," in *2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2022, pp. 146–151.

[8] S. Baixo, T. Ribeiro, G. Lopes, and A. F. Ribeiro, "3d face recognition using inception networks for service robots," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2022, pp. 47–52.

[9] Y. Miura, H. Yihao, S. Kuchii, and G. C. Sern, "Research and development of the social robot using speech recognition and image sensing technology," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2015, pp. 66–69.

[10] T. Applewhite, V. J. Zhong, and R. Dornberger, "Novel bidirectional multimodal system for affective human-robot engagement," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–7.

[11] Raspberry Pi Ltd., "Ai camera," 2024, accessed: 2026-2-5. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/ai-camera.html

[12] Raspberry pi Ltd., "Raspberry pi ai camera (imx500) model zoo," 2026, accessed: 2026-2-5. [Online]. Available: https://github.com/raspberrypi/imx500-models

[13] Google AI Edge Google Developers-MediaPipe, "Guía de detección de rostro," 2025, accessed: 2026-2-6. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector?hl=es-419

[14] Google Ai Edge Google Developers-MediaPipe, "Guía de detección de puntos faciales," 2024, accessed: 2026-2-6. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker?hl=es-419

[15] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 5962–5979, 2022.