Jornadas de Automática

Interpretación de gestos en tiempo real empleando GestureNet en un robot social

Jesús García-Martínez*, Juan José Gamboa-Montero, José Carlos Castillo, Álvaro Castro-González, Miguel Ángel Salichs

Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid. Avenida de la Universidad, 30. 28911 Leganés, Madrid. España.

To cite this article: García-Martínez, J., Gamboa-Montero, J.J., Castillo, J.C., Castro-González, Á. and Salichs, M.A. 2024. Real-Time Gesture Interpretation Using GestureNet in a Social Robot
Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10819

Resumen

Este artículo presenta el desarrollo y la integración de un clasificador de gestos manuales en un robot social, con el objetivo de mejorar la comunicación visual durante la interacción humano-robot. Además de las capacidades actuales del robot para escuchar la voz del usuario y recibir comandos táctiles a través de una tableta auxiliar, se ha implementado la capacidad de interpretar gestos visuales. Estos gestos incluyen afirmaciones y negaciones con la mano, así como la mano cerrada y abierta, entre otros. Se ha generado un conjunto de datos para entrenar el modelo de clasificación, y utilizamos una arquitectura diseñada específicamente para este propósito. Como caso de uso del clasificador, se ha desarrollado una aplicación del juego tradicional de piedra, papel o tijera. En dicho juego, durante la interacción con el usuario, el modelo de clasificación se ejecuta en tiempo real. Tanto el módulo de detección como la habilidad de juego se han integrado completamente en la arquitectura del robot, proporcionando una experiencia de usuario fluida y natural a través de este canal de comunicación.

Palabras clave: Aprendizaje profundo, Robótica social, Visión por computador, Interacción humano-robot, Clasificación de imágenes, Tiempo real

Real-Time Gesture Interpretation Using GestureNet in a Social Robot

Abstract

This paper presents the development and integration of a hand gesture classifier in a social robot, aiming to enhance visual communication during human-robot interaction. In addition to the robot's current capabilities to listen to the user's voice and receive touch commands through an auxiliary tablet, the ability to interpret visual gestures has been implemented. These gestures include hand signals for affirmation and negation, as well as open and closed hands. A dataset was generated to train the classification model, and we utilized a specifically designed architecture for this purpose. An application for the traditional game of rock, paper, and scissors was developed as a use case for the classifier. In this game, the classification model runs in real time during user interaction. The detection module and the application have been fully integrated into the robot's architecture, providing a smooth and natural user experience through this communication channel.

Keywords: Deep Learning, Social Robotics, Computer Vision, Human-Robot Interaction, Image Classification, Real Time

*Jesús García-Martínez: jesusgar@ing.uc3m.es

Correos electrónicos: jesusgar@ing.uc3m.es (Jesús García-Martínez), jgamboa@ing.uc3m.es (Juan José Gamboa-Montero), jocastil@ing.uc3m.es (José Carlos Castillo), acgonzal@ing.uc3m.es (Álvaro Castro-González), salichs@ing.uc3m.es (Miguel Ángel Salichs)

1. Introducción

La robótica social es un campo de investigación que se centra en el desarrollo de robots capaces de interactuar de manera efectiva y natural con los humanos, denominado como interacción humano-robot (IHR). Estos robots tienen aplicaciones en diversos ámbitos como la educación, la salud, el entretenimiento o en entornos sociales (Kanda and Ishiguro, 2017). Para que un robot sea capaz de interactuar con eficacia debe ser capaz de percibir su entorno y al usuario integrando sistemas de percepción que le permitan extraer información relevante haciendo uso de micrófonos, sensores de tacto y cámaras. Una de las metas durante la interacción es que el robot pueda adaptarse a las necesidades individuales de cada persona y mejorar la experiencia del usuario.

Varios estudios se centran en el desarrollo IHR multimodal, como en el trabajo de (Shrestha et al., 2024), donde los autores presentan un conjunto de datos que fusiona comandos de voz y gestos naturales basados en la posición del esqueleto del usuario, sincronizados con demostraciones de comportamiento del robot. Del mismo modo, (Chen et al., 2022) estudia el efecto que tiene un robot social que es capaz de percibir las emociones del usuario y ajustar su comportamiento durante la interacción frente a un robot que carece de dicha capacidad. Los resultados de su estudio demostraron que los usuarios valoraban de manera más positiva al robot afectivo respecto al que no lo era. Existen otros estudios que se centran en la percepción del tacto durante la interacción, como en el estudio de Zhou et al. (2021), donde los autores evaluaban la percepción del usuario cuando tocaban a un robot o viceversa, las impresiones de la interacción y las actitudes a los robots humanoides. O estudios que se centran en la interacción por voz y dispositivos auxiliares. En el trabajo de Andronas et al. (2021), los autores presentan arquitectura de interacción multimodal entre humanos y robots para facilitar la comunicación y colaboración en entornos industriales, utilizando dispositivos como relojes inteligentes capaces de reconocer comandos de voz y gestos y tabletas para comandar a los robots durante el ensamblado.

El sistema que se presenta en este trabajo se ha integrado en un robot social que actualmente interactúa con los usuarios a través de múltiples modalidades. El robot Mini (Salichs et al., 2020) dispone de un micrófono que capta comandos de voz, permitiéndole reconocer y responder a las instrucciones verbales del usuario. Además, cuenta con una tableta auxiliar donde los usuarios pueden introducir información de manera táctil, así como sensores de tacto que permiten iniciar y detener actividades según las interacciones físicas del usuario. Nuestro objetivo es ampliar la capacidad de interacción de este robot incorporando un sistema de interpretación de gestos manuales mediante visión por computador. Esto mejorará las capacidades de comunicación de la plataforma robótica, de forma que el usuario podrá realizar gestos tales como mano abierta o cerrada para indicar negación, y símbolo de "ok" y pulgar arriba para indicar afirmación. Esta modalidad, añadida a las que ya posee el robot contribuirá a que la interacción sea más versátil y natural.

En trabajos anteriores (Borrero et al., 2023) se hizo una primera aproximación al problema, implementando un detector de puntos de referencia de la mano para clasificar gestos según la posición de los dedos para una aplicación de entretenimiento. Una de las carencias que se observó utilizando este tipo de modelos es que en caso de oclusión parcial de uno o varios dedos, el modelo estima la posición de donde debería encontrarse el dedo, generando falsos positivos sobre si el dedo se encuentra extendido o no, lo que resultaba en clasificaciones incorrectas comprometiendo a la precisión del detector. Con la motivación de corregir estas limitaciones, en este estudio se presenta un modelo de clasificación de gestos basado en el reconocimiento de imágenes completas de la mano, en lugar de utilizar modelos de detección de puntos de referencia. Este enfoque permite identificar de manera precisa los gestos, incluso con oclusiones parciales de los dedos. Como caso de uso de este nuevo clasificador, se ha implementado y probado el juego tradicional de piedra, papel o tijera como una aplicación de la plataforma robótica, ya que este juego requiere el reconocimiento preciso de varios gestos manuales.

El presente trabajo tiene la siguiente estructura: En la sección 2 se explican las tecnologías utilizadas, incluyendo el robot social donde se realiza la integración y la arquitectura software empleada para el entrenamiento del modelo de clasificación y la aplicación. En la sección 3 se describe como se ha generado la base de datos de las imágenes de los gestos, el diseño y entrenamiento del modelo entrenado y como se realizan las clasificaciones. El funcionamiento de la aplicación integrada en el robot se detalla en la sección 4. Las conclusiones obtenidas a partir del trabajo realizado son recogidas en la sección 5.

2. Materiales

En esta sección, por un lado, se profundiza la plataforma robótica en la que el sistema se integra. Por otro lado, se listan las librerías que se han utilizado durante el desarrollo del trabajo, destacando las implicadas en la visión por computador y para el entrenamiento de modelos de clasificación basadas en aprendizaje profundo.

2.1. El robot social Mini

Mini (Salichs et al., 2020) es un robot social de sobremesa diseñado y fabricado en la Universidad Carlos III de Madrid. El propósito original con el que se diseñó este robot era la estimulación para mayores con deterioro cognitivo moderado, por lo que las aplicaciones estaban desarrolladas con ese enfoque. Actualmente el rango de aplicaciones abarca a todo tipo de usuarios y edades incluyendo aplicaciones de estimulación física, cognitiva y entretenimiento.

2.1.1. Hardware

En lo referido a su nivel de hardware, se trata de un robot impreso en 3D recubierto con una malla de peluche personalizable (ver figura 1 parte inferior derecha). Dispone de cinco grados de libertad en cabeza, cuello, brazos y tronco para realizar diferentes expresiones no verbales. Tiene iluminación led en las mejillas, cabeza y corazón utilizados para simular los latidos del corazón y emociones. Dispone de micrófonos integrados en la cabeza y en el torso para captar el sonido ambiente y la voz del usuario, y de un altavoz en la cabeza para comunicarse verbalmente. Los ojos del robot son dos pantallas

uOLED configurables que permiten mostrar diferentes expresiones así como simular el parpadeo con naturalidad. Además, cuenta con sensores capacitivos en torso, brazos y cabeza para detectar cuando el usuario toca al robot. Para captar información visual del entorno y del usuario dispone de una cámara RGBD (RealSense D435i[1]) situada en el tronco del robot. Para reducir el consumo de la CPU, situada en la base del robot, Mini integra dos aceleradores gráficos: Google Coral TPU[2] e Intel Neural Compute Stick 2[3] para ejecutar modelos de inteligencia artificial.

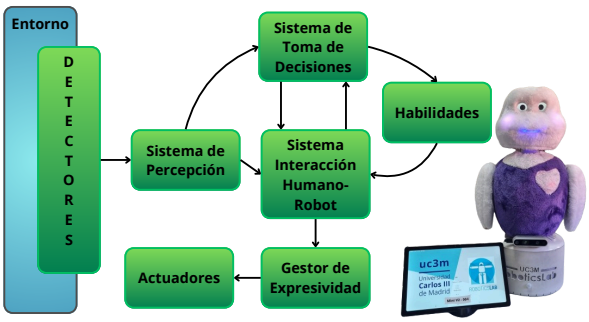

Figura 1: Arquitectura software del robot Mini.

2.1.2. *Software*

Respecto a su arquitectura software está desarrollada en ROS y consta de seis módulos principales (ver figura 1): Comenzando desde la adquisición de información hasta la actuación del robot, siguiendo el flujo del diagrama, los detectores captan la información del entorno y del usuario, esta información incluye los canales visuales, táctil y auditivos. El Sistema de Percepción se encarga de estandarizar, procesar y agrupar la información. Esta información agrupada es utilizada por el Sistema de Toma de Decisiones, encargado de seleccionar la actividad que debe realizar el robot en cada momento (Habilidades), y por el Sistema de Interacción Humano-Robot, encargado de gestionar la comunicación con el usuario por la vía oral y por la tableta auxiliar. El Gestor de expresividad se encarga de seleccionar en cada momento el gesto que debe realizar el robot, comandando finalmente el módulo de Actuación que controla los motores, luces y ojos de Mini.

En este trabajo las principales contribuciones son, en primer lugar, un detector que permita interpretar los gestos manuales que hace el usuario, por lo que se integra en la arquitectura de percepción. En segundo lugar, se ha desarrollado una nueva aplicación de entretenimiento, integrada en el módulo de habilidades del robot, que hace uso del clasificador de gestos.

2.2. *Librerías utilizadas*

El sistema operativo sobre el que está construida la arquitectura del robot es Ubuntu, concretamente diseñado empleando *Robotic Operating System* (ROS) Quigley et al. (2009). Por ello, los desarrollos presentados en este trabajo se han llevado a cabo en el lenguaje de programación Python. Se destacan a continuación algunas de las librerías que se han utilizado.

2.2.1. *OpenCV*

La librería OpenCV[4] (*Open Source Computer Vision Library*) se ha utilizado para el procesamiento de las imágenes capturadas con la cámara del robot. Se trata de una biblioteca de software de código abierto que proporciona una infraestructura común para aplicaciones de visión por computador y aprendizaje automático. La librería contiene algoritmos que permiten realizar transformaciones geométricas y morfológicas sobre la imagen, como la detección de bordes, contornos cerrados u áreas, o la segmentación a través de la extracción de características. Además de esto, dispone de funciones para realizar el calibrado de cámaras, o aplicaciones con modelos preentrenados para la detección de caras, seguimiento de objetos, entre otras. La librería es multiplataforma y tiene soporte en Python. En este trabajo se ha hecho uso principalmente para el guardado de las imágenes a la hora de generar la base de datos, durante el preprocesado de las imágenes a la hora de entrenar el modelo de clasificación y durante la inferencia en tiempo real.

2.2.2. *TensorFlow*

Para el entrenamiento del clasificador de gestos se ha utilizado la librería TensorFlow[5]. Es una biblioteca de software de código abierto desarrollada por *Google Brain* para el aprendizaje automático y el aprendizaje profundo. Proporciona una infraestructura para la construcción y el entrenamiento de modelos de inteligencia artificial. La biblioteca dispone de funciones que permiten el procesamiento de grandes conjuntos de datos, realizar el entrenamiento de modelos y la posterior evaluación e inferencia. En el trabajo se ha utilizado principalmente durante las fases de procesado de la base de datos de imágenes, para entrenar el modelo presentado en este trabajo y para realizar las clasificaciones durante la interacción con el usuario.

3. Diseño e implementación del clasificador de gestos

Para realizar el entrenamiento del modelo de clasificación de imágenes se han considerado varias técnicas: Transferencia de aprendizaje (Torrey and Shavlik, 2010), definido como el entrenamiento de un modelo que parte de unos pesos precargados, y que se lleva a cabo congelando los pesos de todas las capas intermedias y modificando únicamente durante el entrenamiento los pesos de las capas de salida del modelo. Ajuste fino (Vrbančič and Podgorelec, 2020), descrito como una extensión del método de transferencia de aprendizaje, en la que se aplica una etapa final donde se descongelan los pesos de las capas intermedias para realizar pequeños ajustes sobre los pesos del modelo al completo. Por último, planteamos el entrenamiento del modelo completo desde cero (Boyd et al.,

[1] https://www.intelrealsense.com/depth-camera-d435i/
[2] https://coral.ai/products/accelerator
[3] https://www.intel.la/content/www/xl/es/products/sku/140109/intel-neural-compute-stick-2/specifications.htmlr
[4] https://opencv.org/
[5] https://www.tensorflow.org/?hl=es-419

2019), que consiste en entrenar el modelo al completo asignando unos pesos iniciales aleatorios y entrenar toda la arquitectura.

En nuestro diseño de aplicación consideramos interpretar cinco gestos, que se corresponden con el número de clases que el modelo deberá aprender. Por ello, tanto nuestro conjunto de datos como número de clases es reducido. Dado que el objetivo es que el clasificador de gestos pueda ejecutarse en tiempo real, consideramos que no es necesario utilizar modelos preentrenados con una alta densidad de capas que aumenten el tiempo de inferencia. Por todo ello, de las tres técnicas descritas, hemos decidido entrenar desde cero el modelo *GestureNet* (Rosebrock et al., 2019). La arquitectura de este modelo tiene características comunes tanto de *AlexNet* como de *VGGNet*, muy utilizados en la literatura para la clasificación de imágenes, pero reduce la complejidad de capas de estos modelos mejorando el tiempo de inferencia.

3.1. *Entrenamiento del modelo*

Hemos recopilado un total de 2500 imágenes pertenecientes a cinco clases: Pulgar arriba, pulgar abajo, mano abierta, mano cerrada y símbolo "ok". Para generar el conjunto de datos hemos utilizando ambas manos realizando los diferentes gestos que queremos que el modelo aprenda capturando un total de 500 imágenes por clase. Dado que utilizamos información de profundidad se ha variado la posición, cercanía y orientación de la mano durante la adquisición de imágenes para reducir la repetitividad y permitir que el modelo extraiga características mejor y que pueda generalizar. Con el mismo propósito, se ha realizado la captura de las imágenes cada 0,3 segundos. Además, se han aplicado técnicas de eliminación de duplicados para garantizar que no haya imágenes similares (*Image hashing*). Estas técnicas consisten en calcular el valor *hash* resultante de cada imagen comparando las diferencias de brillo entre píxeles adyacentes en la imagen. Y garantizar que no existen dos imágenes iguales en cada clase del conjunto de datos (Fitas et al., 2021).

Tabla 1: Hiperparámetros aplicados durante el aumento de datos.

Hiperparámetro	**Valor**
Rango de Rotación	20
Rango de Zoom	0.15
Desplazamiento Horizontal	0.2
Desplazamiento Vertical	0.2
Cizallamiento	0.15
Volteo Horizontal	True
Modo de Relleno	cercano

En lo referido al procesamiento del conjunto de datos, lo hemos dividido en tres conjuntos de manera aleatoria: Entrenamiento, validación y testeo siguiendo una distribución de 70 %-20 %-10 % respectivamente. Estos valores de reparto de conjuntos de imágenes se han utilizado en diferentes estudios recientes de entrenamiento de clasificadores (Himami et al. (2021), Mudduluru et al. (2023)). Aplicamos técnicas de aumento de datos, con una configuración de hiperparámetros mostrada en la tabla 1. Con esta técnica junto a los métodos anteriormente aplicados tenemos como objetivo que el modelo sea capaz de generalizar y reducir el sobreajuste.

Figura 2: Gráfico de entrenamiento del modelo de clasificación de gestos.

Durante la fase de entrenamiento hemos fijado los siguientes hiperparámetros del modelo: Utilizamos el optimizador *Adam*, dado que es en el que se utilizaba en el estudio que presentaba el modelo. La tasa de aprendizaje utilizada es de de 0,001. Y el *batch size* es de 8. Aplicamos una función de decaimiento que aplica sobre la tasa de aprendizaje reduciendo el ajuste de los pesos del clasificador con el objetivo de que, a medida que pasan las épocas, el modelo sólo realice pequeños ajustes sobre los pesos dado que ya ha aprendido las clases. Entrenamos sobre un total de 75 épocas. Valor arbitrario dado que aplicamos el método de *EarlyStoping* para detener el entrenamiento. Este método evalúa la evolución de la precisión en la validación durante el entrenamiento, en caso de que la validación no mejore durante 8 épocas, se detiene el entrenamiento dado que el modelo comenzaría a sobreajustar. En la figura 2 se observa como el error de entrenamiento y validación (*train_loss*, *val_loss*) disminuyen hasta estabilizarse en valores cercanos a cero, activando el método que detiene el entrenamiento (época 34), mientras que las curvas de precisión del entrenamiento y validación (*train_acc*, *val_acc*) incrementan.

Tabla 2: Evaluación del modelo de clasificación sobre el conjunto de test.

Clase	**Precisión**	**Recall**	**F1-Score**	**Soporte**
Mano cerrada	0.96	1.00	0.98	50
Símbolo ok	1.00	1.00	1.00	50
Mano abierta	1.00	0.98	0.99	50
Pulgar abajo	1.00	0.98	0.99	49
Pulgar arriba	0.98	0.98	0.98	50
Accuracy		0.99 (249)		
Macro avg	0.99	0.99	0.99	249
Weighted avg	0.99	0.99	0.99	249

3.2. *Evaluación del modelo*

Siguiendo los pasos previamente descritos, hemos evaluado nuestro modelo midiendo el *accuracy* y el *F1-Score* obteniendo 0,99 en el conjunto de datos de test. Desglosado por clases el modelo ha sido capaz de aprender las cinco clases obteniendo las métricas mostradas en la tabla 2.

El clasificador de gestos desarrollado realiza las siguientes etapas de preprocesado antes de realizar las inferencias: A partir de la imagen RGBD se aplica un filtro de distancia,

fijado en nuestro caso entre 20 y 60 cm. Lo suficiente para detectar el gesto de la mano realizado por el usuario frente a la cámara del robot con el brazo estirado. Para ello, realizamos una máscara que contiene los píxeles de la imagen de profundidad que se encuentran en dicho rango (ver figura 3(a)). Posteriormente, se redimensiona la imagen para coincidir con el tamaño de entrada del modelo. Este procedimiento es el que se ha aplicado durante la generación del conjunto de datos para coincidir con el entrenamiento. Los diferentes gestos correctamente clasificados por el modelo se muestran en la figura 3.

(a) Ejemplo de máscara sobre la que se realiza la inferencia en el caso del símbolo ok.

(b) Clasificación del símbolo ok.

(c) Clasificación del pulgar arriba.

(d) Clasificación del pulgar abajo.

(e) Clasificación de la mano abierta.

(f) Clasificación de la mano cerrada.

Figura 3: Clasificación de los signos de la mano para interacción visual.

4. Desarrollo de una aplicación para el robot social

El juego de piedra, papel o tijera es un juego de manos en el que dos jugadores eligen simultáneamente uno de los tres gestos: piedra (representado por el puño cerrado), papel (representado por la mano abierta) y tijera (representado por los dedos índice y corazón extendidos y el resto contraídos). En base a las normas del juego, la piedra aplasta a la tijera, la tijera corta el papel y el papel envuelve a la piedra, estableciendo una normativa de qué gesto predomina entre los disponibles. El objetivo es predecir y contrarrestar la elección del oponente para ganar la partida. Se trata de un juego de azar y estrategia en el que los jugadores esconden las manos, y cuando se realiza una cuenta regresiva se muestra el gesto seleccionado en búsqueda de un ganador.

La aplicación que se ha diseñado para este trabajo ha sido integrada como parte del módulo de "Habilidades" del robot (mencionado en la sección 2.1.2), por lo tanto de ahora en adelante nos referiremos a ella como "habilidad". En la fase de diseño de la habilidad del robot se ha emulado la estructura del juego real. Se han añadido dos etapas adicionales previas al comienzo del juego: La primera está enfocada a que el usuario aprenda a mostrarle las manos al robot, mientras que en la segunda fase el robot le comentará al usuario las normas del juego en caso de que no las conozca. Como la aplicación está planteada para que puedan utilizarla usuarios de todas las edades, el robot en todo momento guiará al usuario a lo largo de la interacción indicándole lo que debe hacer tanto por comunicación verbal como visual utilizando la tableta auxiliar del mismo. En el diseño de la aplicación la cuenta regresiva para indicar cuando mostrar las manos siempre la realiza el robot en voz alta dado que es el que está guiando al usuario.

En la figura 4 se observa el diagrama de flujo por el que pasa la habilidad del robot durante el transcurso de la partida: al inicio del juego, se realiza la etapa de calibración donde el robot le pide al usuario que muestre los diferentes gestos con el objetivo de que el usuario aprenda a como enseñarle las manos al robot. En caso de que el robot no sea capaz de reconocer el gesto que hace el usuario, se repite esta etapa hasta que el reconocimiento sea satisfactorio. Posteriormente el robot le pregunta al usuario si conoce las reglas o por el contrario si quiere recordar las instrucciones.

El juego está configurado al mejor de cinco rondas, con el objetivo de que no se pueda empatar. En cada ronda el robot realizará una cuenta regresiva y cuando dé la señal, tanto el usuario como el robot deben sacar sus gestos a la vez. En base al diseño, el usuario solo puede utilizar una mano de manera simultánea pero el robot es capaz de clasificar debidamente los gestos con ambas manos. Mini no tiene dedos físicos en los brazos, por lo que mostrará en la tableta el gesto escogido a la vez que lo comunica verbalmente. En el momento de la cuenta regresiva, se leen los valores del clasificador de gestos explicado en la sección 3. Para garantizar que el gesto mostrado por el usuario sea el correcto y darle tiempo al usuario para enseñar las manos, se almacenan todas las clasificaciones de los gestos durante dos segundos, descartando aquellas con una baja confianza.

Posteriormente, se busca el gesto más repetido en dicho listado y es el que el robot considera como válido. En el caso de uso, utilizamos el gesto de la mano abierta como "papel", mano cerrada como "piedra", y el símbolo "ok" como "tijeras". El robot compara el gesto del usuario con su propio gesto, que selecciona de manera aleatoria, y determina el ganador de la ronda. Las reglas son las mismas que en el juego original. Si ambos muestran el mismo gesto, se considera un empate y se repite la ronda. En caso de que el usuario se olvide de mostrar las manos en el tiempo indicado (no hay gestos detectados), o que el robot detecte otro gesto que no se corresponde con ninguno de los utilizados en el juego, le avisará por voz y volverá a realizar la cuenta atrás repitiendo la ronda.

[6]https://youtu.be/s3vg1vSbTFg

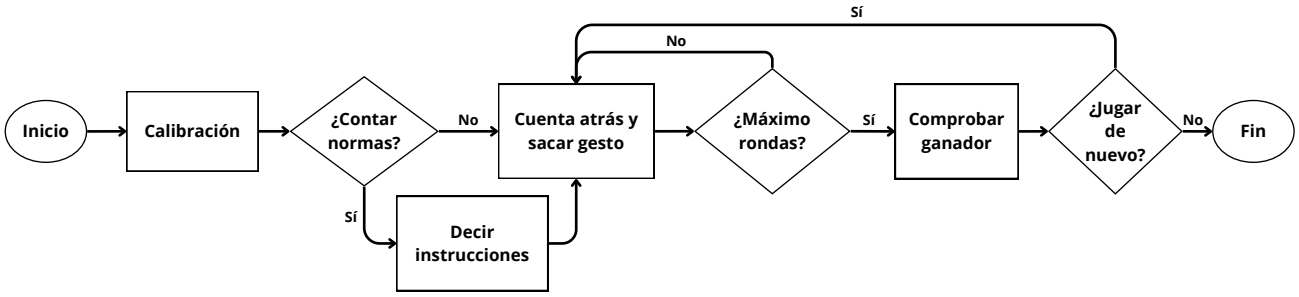

Figura 4: Diagrama de flujo de la aplicación.

Este proceso se repite hasta que haya un jugador victorioso. Tras finalizar el juego, el robot preguntará al usuario si quiere volver a jugar. Un ejemplo de la interacción completa donde se ve el funcionamiento de la aplicación y del clasificador ha sido recogido en vídeo[6].

5. Conclusiones

En este trabajo se ha desarrollado e integrado un sistema de interpretación de gestos manuales en un robot social y se ha implementado una aplicación como caso de uso para el detector. El sistema consta de dos módulos principales: un clasificador de gestos capaz de reconocer cinco gestos diferentes, integrado en el sistema de percepción del robot, y la aplicación del juego piedra, papel o tijera, que utiliza este clasificador y está integrado en el sistema de habilidades del robot.

La capacidad de interpretar gestos manuales permite al robot añadir un canal comunicativo adicional a los ya existentes, mejorando así la naturalidad de la interacción con el usuario. Aunque ya existen algoritmos para detectar puntos característicos de las manos, estos suelen fallar con oclusiones parciales. En cambio, el uso de modelos de clasificación ofrece una mayor precisión en la interpretación de gestos al incluir una variedad de muestras de cada gesto durante el entrenamiento y descartando clasificaciones con baja confianza. Los siguientes objetivos de esta línea de trabajo consistirán, en primer lugar, en evaluar a través de la aplicación desarrollada si la integración de este canal de comunicación afecta significativamente a la experiencia del usuario durante la interacción humano-robot. En segundo lugar, este mismo estudio nos permitiría evaluar el rendimiento del modelo en tiempo real durante la ejecución de la aplicación. Por último, se puede realizar una comparativa con otros modelos orientados al reconocimiento de gestos presentes en la literatura, tales como *ResNet*, *AlexNet* o *MobileNet*, entre otros.

Agradecimientos

Estos resultados han sido financiados por los proyectos PID2021-123941OA-I00, financiado por MCIN/AEI/10.13039/501100011033 y por ERDF A way of making Europe; TED2021-132079B-I00 financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea NextGenerationEU/PRTR; Mejora del nivel de madurez tecnológica del robot Mini (MeNiR) financiado por MCIN/AEI/10. 13039/501100011033. 13039/501100011033 y por la Unión Europea NextGenerationEU/PRTR; Robot social portable con alto grado de vinculación (PoSoRo) PID2022-140345OB-I00 financiado por MCIN/AEI/10.13039/501100011033 y ERDF A way of making Europe.

Referencias

Andronas, D., Apostolopoulos, G., Fourtakas, N., Makris, S., 2021. Multimodal interfaces for natural human-robot interaction. Procedia Manufacturing 54, 197–202.

Borrero, J., Arrojo Fuentes, G. A., García, J., Castillo, J. C., Castro-González, Á., Salichs, M. Á., 2023. Implementación del juego pares o nones en un robot social. In: XLIV Jornadas de Automática. Universidade da Coruña. Servizo de Publicacións, pp. 539–544.

Boyd, A., Czajka, A., Bowyer, K., 2019. Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch? In: 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, pp. 1–9.

Chen, L., Wang, K., Li, M., Wu, M., Pedrycz, W., Hirota, K., 2022. K-means clustering-based kernel canonical correlation analysis for multimodal emotion recognition in human–robot interaction. IEEE Transactions on Industrial Electronics 70 (1), 1016–1024.

Fitas, R., Rocha, B., Costa, V., Sousa, A., 2021. Design and comparison of image hashing methods: A case study on cork stopper unique identification. Journal of Imaging 7 (3), 48.

Himami, Z. R., Bustamam, A., Anki, P., 2021. Deep learning in image classification using dense networks and residual networks for pathologic myopia detection. In: 2021 International Conference on Artificial Intelligence and Big Data Analytics. IEEE, pp. 1–6.

Kanda, T., Ishiguro, H., 2017. Human-robot interaction in social robotics. CRC Press.

Mudduluru, S., Maryada, S. K. R., Booker, W. L., Hougen, D. F., Zheng, B., 2023. Improving medical image segmentation and classification using a novel joint deep learning model. In: Medical Imaging 2023: Computer-Aided Diagnosis. Vol. 12465. SPIE, pp. 599–608.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al., 2009. Ros: an open-source robot operating system. In: ICRA workshop on open source software. Vol. 3. Kobe, Japan, p. 5.

Rosebrock, A., PhD, D. H., MSc, D. M., Thanki, A., Paul, S., 2019. Raspberry pi for computer vision: Hobbyist bundle-v1. 0.1. Baltimore, MD: PyImageSearch. com.

Salichs, M. A., Castro-González, Á., Salichs, E., Fernández-Rodicio, E., Maroto-Gómez, M., Gamboa-Montero, J. J., Marques-Villarroya, S., Castillo, J. C., Alonso-Martín, F., Malfaz, M., 2020. Mini: a new social robot for the elderly. International Journal of Social Robotics 12, 1231–1249.

Shrestha, S., Zha, Y., Banagiri, S., Gao, G., Aloimonos, Y., Fermuller, C., 2024. Natsgd: A dataset with speech, gestures, and demonstrations for robot learning in natural human-robot interaction. arXiv preprint arXiv:2403.02274.

Torrey, L., Shavlik, J., 2010. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, pp. 242–264.

Vrbančič, G., Podgorelec, V., 2020. Transfer learning with adaptive finetuning. IEEE Access 8, 196197–196211.

Zhou, Y., Kornher, T., Mohnke, J., Fischer, M. H., 2021. Tactile interaction with a humanoid robot: Effects on physiology and subjective impressions. International Journal of Social Robotics 13, 1657–1677.