XLIV Jornadas de Automática 2023

https://jautomatica.es

# Sistema de interacción humano-robot basado en la mirada para la manipulación de objetos

Menéndez, E.[a,*], Hernandez-Vicen, J.[a], Martínez de la Casa, S.[a], Monje, C.A.[a], Balaguer, C.[a]

[a] *RoboticsLab, Dpto. de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Av. De la Universidad 30, 28911 Leganés, España.*

**To cite this article:** Menéndez, E., Hernandez-Vicen, J., Martínez de la Casa, S., Monje, C.A., Balaguer, C. Gaze-based Human-Robot Interaction System for Object Manipulation. XLIV Jornadas de Automática, 661-666. https://doi.org/10.17979/spudc.9788497498609.661

**Resumen**

En este artículo, se presenta una solución completa de interacción humano-robot basada en la mirada para ayudar al usuario en tareas de manipulación de objetos. El usuario utiliza unas gafas de seguimiento ocular y se detecta la intención en su mirada de coger un objeto entre varios situados sobre una mesa. El robot, una vez identificado el objeto seleccionado, procede a recogerlo y se lo acerca al usuario. Nuestra solución se compone de diferentes componentes, como la estimación de forma y posición de los objetos utilizando supercuádricas en el sistema de referencia del robot, la identificación y selección del objeto coincidente en la imagen del robot mediante redes siamesas, y un proceso adicional que permite al robot recoger el objeto seleccionado por el usuario de manera efectiva. Esta solución proporciona una forma innovadora y efectiva de interacción para realizar tareas asistenciales de manipulación, sin necesidad de marcadores ni posiciones predefinidas de los objetos, mejorando la fluidez en la comunicación y facilitando la interacción natural entre el usuario y el robot.

*Palabras clave:* Aprendizaje automático, Interfaces inteligentes, Robótica Inteligente, Percepción y detección, Robots manipuladores

**Gaze-based Human-Robot Interaction System for Object Manipulation**

**Abstract**

In this article, we present a complete solution for gaze-based human-robot interaction to assist the user in object manipulation tasks. The user wears eye-tracking glasses, and their intention to pick up an object among several placed on a table is detected through their gaze. Once the selected object is identified, the robot proceeds to grasp it and bring it closer to the user. Our solution consists of various components, including the estimation of object shape and position using superquadrics in the robot's reference frame, the identification and selection of the matching object in the robot's image using Siamese networks, and an additional process that enables the robot to pick-up the object selected by the user. This solution offers an innovative and effective way of interaction to perform assistive manipulation tasks, without the need for markers or predefined object positions, enhancing communication fluency and facilitating natural interaction between the user and the robot.

*Keywords:* Machine learning, Intelligent interfaces, Intelligent robotics, Perception and sensing, Robots manipulators

[*] Autor para correspondencia: emenende@pa.uc3m.es

## 1. Introducción

En la actualidad, los robots se han vuelto cada vez más comunes tanto en la vida diaria como en la industria. Los robots ya no se limitan únicamente a realizar tareas simples y repetitivas. Ahora, los robots deben comprender y atender las necesidades de los usuarios de manera más efectiva ((Naneva et al., 2020; Ajoudani et al., 2018)).

Con el objetivo de mejorar la interacción humano-robot, es fundamental que los robots sean capaces de entender la intención del usuario. La mirada es uno de los métodos de comunicación más intuitivos y naturales. La mirada es una forma no verbal de comunicar nuestras intenciones, y su comprensión por parte de los robots permite establecer una interacción más fluida y eficiente (Zhang et al., 2017; Yu et al., 2012).

Este artículo presenta una interacción humano-robot que puede deducir y ejecutar tareas de manipulación de objetos usando solamente la mirada del usuario. El usuario lleva puestas unas gafas de seguimiento ocular (Carter and Luke, 2020) equipadas con dos cámaras: una para detectar las pupilas y otra para capturar el punto de vista del usuario. El seguimiento de la mirada se proyecta en la imagen percibida por el usuario para obtener información sobre su intención visual.

La Figura 1 muestra el escenario en el que se centra este trabajo, donde el usuario selecciona utilizando su mirada un objeto entre los situados sobre una mesa, tras esto el robot debe coger el objeto seleccionado y acercarlo al usuario. Por ejemplo, en en este caso el usuario se fija en un objeto, y utilizando técnicas de inteligencia artificial se identifica que el usuario quiere el vaso rojo. Sin embargo en nuestro escenario el robot no sabe que objetos tiene delante ni su ubicación, por lo que no podría coger el vaso rojo. Hay varios enfoques para encontrar el objeto seleccionado con la mirada del usuario con respecto al sistema de referencia del robot. En (Shi et al., 2021), el usuario lleva unas gafas de seguimiento ocular para detectar la intención de coger un objeto, pero los objetos están situados en posiciones específicas con respecto al robot, por lo que se conocen los objetos disponibles y donde están situados. En (Weber et al., 2020) resuelven la ubicación de los objetos con respecto al robot situando marcadores para transformar los puntos de fijación al sistema de referencia del robot. En (Huang and Mutlu, 2016) el usuario selecciona el objeto en una matriz con marcadores y cuando se detecta que el usuario fija su mirada en uno de los marcadores, el robot coge el bloque correspondiente.

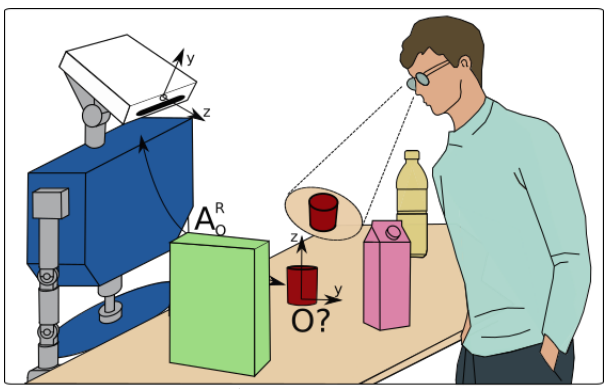



Figura 1: Gracias a unas gafas de seguimiento ocular, el usuario selecciona el objeto que desea. Después, el robot le acerca el objeto seleccionado.

En este artículo se propone una solución integral que permite al robot recoger el objeto seleccionado por la mirada del usuario, sin necesidad de situar marcadores en la mesa ni de que los objetos estén en ubicaciones conocidas previamente. De esta forma, el robot asiste al usuario en tareas de manipulación con una interacción más fluida y natural.

Este artículo se organiza de la siguiente manera: En la Sección 2 se discuten los diferentes componentes del proceso propuesto para que el robot acerque al usuario el objeto en que ha fijado su mirada. La Sección 3 detalla nuestro enfoque para estimar la forma y la posición de los objetos con respecto al sistema de referencia del robot sin asignarles una categoría. En la Sección 4 se presenta la estrategia basada en redes siamesas para determinar qué objeto en el campo de visión del robot coincide con la elección del usuario. Los resultados comparativos de las diferentes redes siamesas utilizadas, se presentan en la Sección 5. Finalmente en la Sección 6, se presentan las conclusiones y las lineas futuras de este trabajo.

## 2. Manipulación del objeto seleccionado con la mirada

Nuestra propuesta presenta una solución completa para identificar el objeto que el usuario ha seleccionado con su mirada en el sistema de referencia del robot, sin necesidad de colocar marcadores o de posicionar los objetos en sitios conocidos. A continuación, el robot recoge el objeto seleccionado. Está solución se basa en ciertas suposiciones. En primer lugar, se considera que los objetos se encuentran sobre una superficie horizontal. En segundo lugar, se permite que los objetos estén parcialmente ocultos por otros objetos, pero no se pueden situar uno encima del otro. Por último, la cámara del robot está orientada hacia abajo, capturando la imagen de los objetos desde un ángulo inclinado. Las etapas principales de esta solución están descritas en la Figura 2.

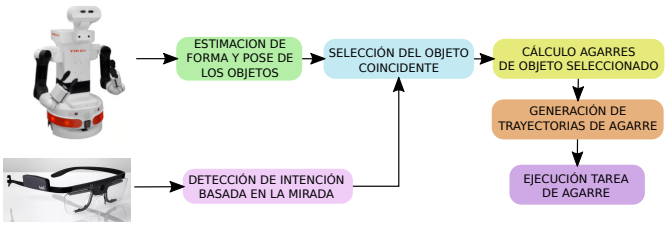



Figura 2: Arquitectura general del proceso.

En la primera etapa, el robot se encarga de estimar las formas básicas y las posiciones de los objetos situados sobre la superficie horizontal. Este proceso se lleva a cabo sin determinar la categoría a la que pertenecen los objetos. Tras esto, se obtienen cuadros delimitadores (bounding boxes en inglés) para cada forma en la imagen RGB capturada por la cámara del robot.

Simultáneamente, el usuario emplea unas gafas de seguimiento ocular que captan su mirada. Utilizando técnicas de inteligencia artificial, como las descritas en (Gonzalez-Diaz et al., 2019), se interpreta la intención del usuario de coger un objeto. Además, se genera un cuadro delimitador que encuadra el objeto deseado desde el punto de vista del usuario, identificando la categoría del objeto.

Una vez que el usuario ha indicado su intención de recoger un objeto utilizando la mirada, se procede a seleccionar el objeto desde la perspectiva del robot. Para ello, se emplean redes siamesas que comparan las características visuales del objeto deseado en la imagen captada con las gafas de seguimiento ocular, con las características visuales de los objetos capturados por la cámara del robot. Esta comparación permite identificar el objeto más parecido en la imagen del robot, su forma y su posición con respecto al sistema de referencia del robot.

Una vez identificado el objeto correspondiente, se procede a calcular las poses de agarre adecuadas. Este cálculo considera varios factores: la forma y posición estimada del objeto, su tamaño y peso, la orientación deseada después del agarre, el entorno circundante para evitar colisiones, y las capacidades y limitaciones del propio robot. Utilizando la cinemática inversa, se determinan las configuraciones articulares realizables que permitirán al robot sujetar el objeto de manera segura.

A continuación, se planifica una trayectoria que sea viable y libre de colisiones hasta la pose de agarre para uno de los brazos del robot. Esta planificación tiene en cuenta las restricciones cinemáticas y dinámicas del brazo robótico, los límites de las articulaciones, las restricciones del espacio de trabajo y la necesidad de evitar obstáculos en el entorno. Después, el robot ejecuta la trayectoria hasta la pose de pre-agarre del objeto.

Finalmente, se planifican y ejecutan las trayectorias para aproximar el objeto hacia el usuario. En este paso, se consideran adicionalmente las restricciones de evitación de colisiones con el entorno teniendo en cuenta que el objeto está sujeto por el efector final del brazo del robot.

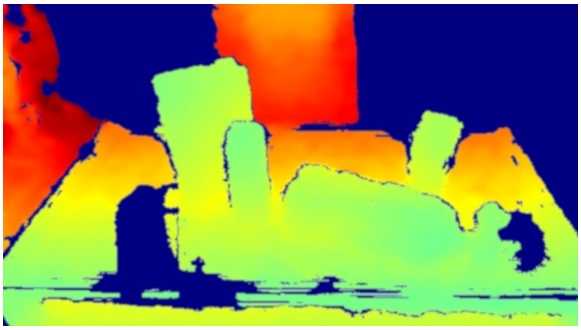
En este trabajo, se describe dos etapas críticas del proceso mostrado en esta sección. La primera es la etapa en la que el robot estima la posición y la forma de los objetos en relación con su propio sistema de referencia. La segunda etapa descrita es la selección del objeto en el sistema de referencia del robot que coincide con la fijación de la mirada del usuario utilizando redes siamesas.

## 3. Estimación de forma y posición de los objetos

El proceso de la estimación de los objetos sin conocer su categoría necesita una imagen de profundidad obtenida con cámara la RGBD del robot. La Figura 7(a) muestra la imagen de profundidad de una escena usual dónde algunas partes de los objetos están en el lado oculto de la cámara o están parcialmente ocultas por otros objetos. Combinando los parámetros intrínsecos de la cámara y la configuración del robot mediante la cinemática directa, la imagen de profundidad adquirida con la cámara RGBD del robot se convierte en una nube de puntos en 3D en el sistema de referencia del robot.

La nube de puntos 3D generada contiene puntos tanto de los objetos como de la mesa donde se encuentran. Para determinar qué puntos corresponden a la superficie de la mesa, se emplea el algoritmo RANSAC (Fischler and Bolles, 1981). Una vez identificados, estos puntos se eliminan de la nube, dejando solo los puntos correspondientes a los objetos.

Posteriormente, los puntos restantes se segmentan en clusters distintos correspondientes a cada objeto. Esta segmentación se realiza mediante la técnica de clustering utilizando la distancia euclídea. Seguidamente, se filtran los clusters que son más pequeños que un umbral mínimo, ya que es probable que correspondan a ruido en la nube de puntos.



(a) Imagen de profundidad

(b) Imagen RGB

Figura 3: Imágenes capturadas por la cámara del robot, mostrando una escena con varios objetos situados sobre una mesa.

La Figura 4 muestra los clusters restantes que corresponden a los objetos sobre la mesa. Para poder distinguir entre los diferentes objetos, se asigna a cada cluster un identificador único.

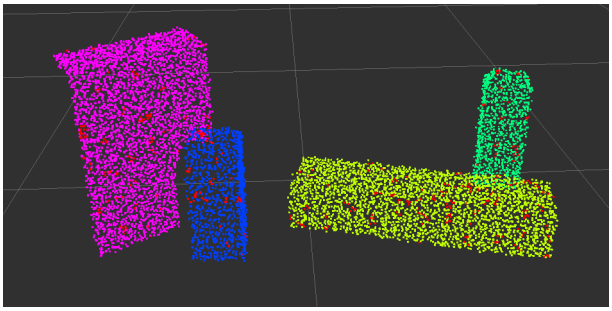


Figura 4: Nube de puntos 3D tras la segmentación. Cada cluster coloreado representa un objeto distinto sobre la mesa.

A partir de los clusters de los objetos obtenidos, se puede encontrar la supercuádrica que mejor representa cada objeto. Las supercuádricas son una representación compacta de formas simples, basada en un conjunto de parámetros (Vezzani et al., 2017). Una supercuádrica se representa por la función de dentro-fuera mostrada en (1), considerando un sistema de referencia centrado en el objeto:

$$F(x,y,z) = \left( \left( \frac{x}{\lambda_1} \right)^{\frac{2}{\lambda_5}} + \left( \frac{y}{\lambda_2} \right)^{\frac{2}{\lambda_5}} \right)^{\frac{\lambda_5}{\lambda_4}} + \left( \frac{z}{\lambda_3} \right)^{\frac{2}{\lambda_4}} \quad (1)$$

donde $\lambda_1$, $\lambda_2$ y $\lambda_3$ son las longitudes de los semiejes, $\lambda_4$ y $\lambda_5$ son los parámetros de forma, y x, y, z son las coordenadas de un punto en el espacio 3D.

La función dentro-fuera determina la posición de un punto $(x, y, z)$ en relación con la supercuádrica: dentro ($F < 1$), en la superficie ($F = 1$), o fuera ($F > 1$). Para situar la supercuádrica en el sistema de referencia del robot, se incorporan variables de traslación $(p_x, p_y, p_z)$ y ángulos RPY $(\phi, \theta, \gamma)$ para la orientación.

Para encontrar la supercuádrica que mejor representa a cada objeto, se busca el conjunto de valores de los parámetros que permita que la mayoría de los puntos ($p_i = [x_i, y_i, z_i]$) de cada cluster se ajusten a la superficie de la supercuádrica. Este ajuste busca minimizar la distancia entre los puntos del cluster y la superficie de la supercuádrica, como se indica en (2).

$$\min_{\mathbf{v}} \sum_{i=1}^{N} (\sqrt{\lambda_1\lambda_2\lambda_3}(F(\mathbf{p_i}, \mathbf{v}) - 1))^2, \tag{2}$$

donde $N$ es el número de puntos en cada cluster. En la Figura 5, se muestra el resultado de aplicar (2) a cada cluster de puntos de la Figura 4. Las formas coloreadas corresponden a las supercuádricas ajustadas, que representan la forma estimada de los objetos en la mesa y su posición con respecto al sistema de referencia del robot.

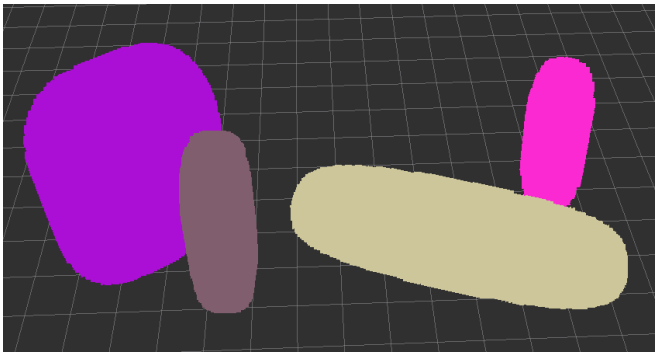


Figura 5: Supercuádricas ajustadas para los objetos situados en la mesa.

## 4. Selección del objeto coincidente

Esta sección se enfoca en la identificación del objeto que el usuario está mirando en el sistema de coordenadas del robot. Este módulo empareja el recorte del objeto observado en la imagen de las gafas con el recorte del objeto más similar en la imagen del robot. Este método, basado en una red siamesa, elimina la necesidad de marcadores y de la detección previa de todos los objetos en ambas imágenes.

El módulo consta de tres pasos: Primero, genera los cuadros delimitadores de los objetos en la imagen RGB del robot utilizando las supercuádricas obtenidas en la sección anterior. Para ello transforma los puntos de cada supercuádrica a las coordenadas de la imagen y, posteriormente, se calculan los valores máximos y mínimos en las coordenadas de la imagen. Después calcula los vectores de características de los objetos recortados mediante una red neuronal entrenada para aprender similitud. También, se obtiene el vector de características para el recorte en la imagen de las gafas con el objeto seleccionado. Finalmente, este vector se compara con los vectores correspondientes a los recortes de cada objeto, obteniendo el objeto más similar en el sistema de referencia del robot.

El componente principal de este módulo es la red neuronal siamesa (Melekhov et al., 2016) que ha sido entrenada para detectar similitud entre las imágenes de objetos capturadas con las gafas de seguimiento ocular y imágenes obtenidas con la cámara del robot.

### *4.1. Diseño y entrenamiento de la red siamesa*

Las redes neuronales siamesas aprenden métricas de similitud y contienen varias redes idénticas (usualmente dos o más) que comparten configuración, parámetros y pesos. Al procesar las imágenes con una red idéntica (o ramas), la red aprende a generar vectores de características similares para imágenes similares (en nuestro escenario, imágenes del mismo objeto) y distintos para las que no lo son. Las redes tripletas están diseñadas para aprender a partir de un trío de muestras: una imagen ancla, una positiva y una negativa. Las imágenes ancla y positiva son similares, mientras que las imágenes ancla y negativa son disímiles.

La red tripleta propuesta consta de tres ramas, cada una de las cuales está compuesta por una red neuronal convolucional (CNN por sus siglas en inglés) basada en la arquitectura ResNet-50 (He et al., 2016), pero sin sus capas completamente conectadas (FC por sus siglas en inglés) originales. En cada rama, se inserta una capa de agrupación de pirámides espaciales (SPP por sus siglas en inglés) (He et al., 2015) tras la última capa convolucional para manejar imágenes de distintos tamaños. SPP divide los mapas de características en una serie de cuadrículas de tamaño fijo, utilizando ventanas de agrupación de tamaños [1×1, 2×2, y 4×4]. Finalmente, tras la capa SPP, se añaden tres capas FC para generar los vectores de características finales.

Durante el entrenamiento, como se ilustra en la Figura 6, la red tripleta aprende a minimizar una función triplet loss (Schroff et al., 2015). Las imágenes ancla, en este caso, son imágenes de objetos tomadas con las gafas de seguimiento ocular, mientras que las imágenes positivas y negativas se obtienen con la cámara del robot. La función triplet loss busca minimizar la distancia entre los vectores de características de la imagen ancla y las imágenes positivas, mientras maximiza la distancia entre los vectores de la imagen ancla y las imágenes negativas.

Si $f(x)$ es el vector de características obtenido para la entrada $x$, y dado un conjunto de N tríos $x_a^i$, $x_p^i$, $x_n^i$, (donde $x_a^i$ es la imagen ancla, $x_p^i$ es la imagen positiva, y $x_n^i$ es la imagen negativa), la función triplet loss se define en (3):

$$L_{tri} = \sum_{i=1}^{N} max(0, d(f(x_a^i), f(x_p^i)) - d(f(x_a^i), f(x_n^i)) + \alpha) \tag{3}$$

Donde $d(f(x_a^i), f(x_p^i))$ denota el cuadrado de la distancia euclídea entre las vectores de características de $x_a^i$ y $x_p^i$, $d(f(x_a^i), f(x_n^i))$ denota el cuadrado de la distancia euclídea entre los vectores de características de $x_a^i$ y $x_n^i$, y $\alpha >= 0$ es el margen máximo entre los pares $(x_a^i, x_p^i)$ y $(x_a^i, x_n^i)$.

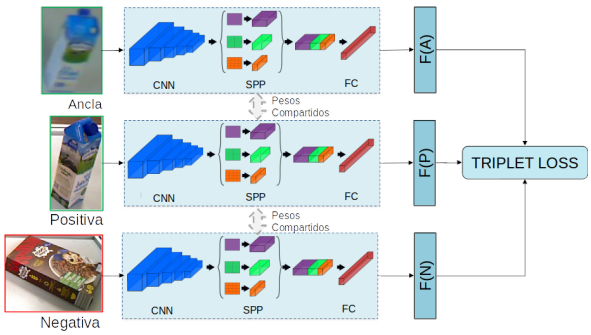



Figura 6: Ilustración de la red tripleta en el proceso de entrenamiento. La red neuronal toma tres entradas: la imagen ancla, la imagen positiva y la imagen negativa. Durante el entrenamiento, se calcula y minimiza la función triplet loss.

Para el entrenamiento se aplica una estrategia de minería de tríos en línea (online triplet mining en inglés) dentro de cada mini-batch de datos. Esta estrategia selecciona intencionadamente los tríos más difíciles de clasificar correctamente, basándose en la distancia entre los vectores de características. Al entrenar con los tríos más desafiantes, se mejora la capacidad del modelo para diferenciar entre imágenes similares y disímiles.

#### *4.1.1. Uso de la red en la implementación*

Una vez entrenada nuestra red para identificar la similitud entre imágenes de objetos captadas con ambas cámaras, se utiliza una sola rama de la red tripleta. Se empieza con los recortes de los objetos en la imagen RGB del robot, derivados de los cuadros delimitadores calculados con supercuádricas. Cada uno de estos recortes tiene asignado un número identificador asociado a su supercuádrica correspondiente. Se utiliza la rama entrenada para generar vectores de características para estos recortes.

El módulo que detecta la intención de la mirada del usuario nos proporciona no sólo el cuadro delimitador del objeto que el usuario ha seleccionado en la imagen de las gafas, y su categoría, sino que también distingue cuándo el usuario tiene la intención de recoger dicho objeto. Para este objeto seleccionado, se genera un vector de características utilizando la rama entrenada y se obtiene el cuadrado de la distancia euclídea con respecto a los vectores correspondientes a los recortes en la imagen del robot.

El recorte en la imagen del robot con la menor distancia se identifica como el objeto que el usuario ha seleccionado con su mirada e intenta coger. Por tanto, no solo se adquiere información sobre la categoría del objeto seleccionado, sino también sobre su forma y su ubicación en relación con el sistema de referencia del robot. Con esta información esencial, el robot puede calcular las zonas de agarre más adecuadas y las trayectorias necesarias para coger de forma efectiva el objeto seleccionado.

## 5. Resultados

En esta sección, se presentan los resultados del entrenamiento de la red siamesa utilizando nuestra base de datos. Esta consta de 10,000 imágenes tomadas con las gafas y 10,000 imágenes capturadas con el robot. La base de datos, dividida en conjuntos de entrenamiento, validación y prueba, incluía imágenes de 16 objetos comúnes de cocina para los dos primeros conjuntos, añadiendo imágenes 2 nuevos objetos en el conjunto de pruebas.

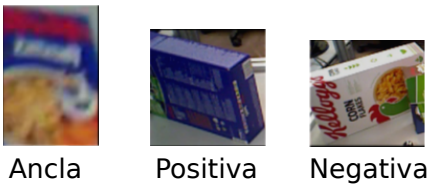
Todas las redes evaluadas comparten una estructura común: una red neuronal convolucional (CNN), una capa de agrupación de pirámides espaciales (SPP) y capas totalmente conectadas. Las diferencias radican en las CNNs utilizadas, siendo MOBILENET-v2, VGG-19 y ResNet-50 las que se evaluaron. La Tabla 1 muestra los resultados medidos en términos de precisión y F1-score en el conjunto de prueba.

Tabla 1: Resultados Comparativos de las Redes Tripletas

| | Precisión | F1 |
|---|---|---|
| S-MOBILENETV2-SPP-FC | 0.9080 | 0.9054 |
| S-VGG19-SPP-FC | 0.8612 | 0.8631 |
| S-RESNET50-SPP-FC | 0.9546 | 0.9540 |

Según los resultados, la red basada en ResNet-50 tuvo el mejor rendimiento en nuestro conjunto de prueba, tanto en términos de F1-score como de precisión, por lo que fue la red tripleta seleccionada para nuestra aplicación. La Figura 5 ilustra ejemplos de emparejamiento correcto e incorrecto utilizando la red seleccionada en la base de datos de prueba. En el emparejamiento correcto, la red identifica acertadamente la imagen positiva como la más similar a la ancla. Sin embargo, en el caso de emparejamiento incorrecto, la red selecciona erróneamente la imagen negativa como la más parecida a la ancla.


(a) Emparejamiento correcto.




(b) Emparejamiento incorrecto

Figura 7: Ejemplos de emparejamiento correcto e incorrecto utilizando la red tripleta S-RESNET50-SPP-FC. La imagen ancla es capturada con las gafas de seguimiento ocular, mientras que las imágenes positiva y negativa son capturadas con la cámara del robot. En el emparejamiento correcto, la red identifica la imagen positiva como la más similar a la ancla. Por otro lado, en el emparejamiento incorrecto, la red incorrectamente asocia la imagen negativa como la más similar.

A pesar de que la redes tripletas fueron entrenadas con 16 objetos, demostraron una excelente capacidad para generalizar y detectar similitudes en los nuevos objetos introducidos en la fase de pruebas. Esto hace que la redes tripletas sean una elección adecuada para la tarea de identificar el objeto que el usuario está mirando en el sistema de coordenadas del robot, incluso cuando se introducen nuevos objetos en el escenario.

## 6. Conclusiones

Se ha desarrollado una solución completa que permite al robot acercar al usuario el objeto seleccionado mediante la mirada, sin necesidad de utilizar marcadores ni conocer los objetos presentes. Nuestro sistema utiliza supercuádricas como una representación precisa y compacta de la forma y posición de los objetos. Además, las redes siamesas implementadas han aprendido a identificar similitudes entre las imágenes de objetos capturadas con las gafas de seguimiento ocular y las imágenes de objetos obtenidas con la cámara del robot. Estas redes han demostrado ser efectivas y han logrado generalizar a nuevos objetos introducidos en el entorno. Esto permite que el robot pueda seleccionar de forma precisa el objeto coincidente con el que el usuario ha mirado. Como trabajos futuros, se propone mejorar la precisión tanto de las supercuádricas como de las redes siamesas en escenarios donde los objetos están parcialmente ocluidos.

## Agradecimientos

La investigación que ha conducido a estos resultados ha recibido financiación del proyecto COMPANION-CM: Inteligencia artificial y modelos cognitivos para la interacción simétrica humano-robot en el ámbito de la robótica asistencial, con referencia Y2020/NMT-6660, financiado por Proyectos Sinérgicos de I+D la Comunidad de Madrid, y del proyecto SOFIA: Articulación blanda inteligente con capacidades de reconfiguración y modularidad para plataformas robóticas, con referencia PID2020-13194GBI00, financiado por el Ministerio de Economía, Industria y Competitividad.

## Referencias

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., Khatib, O., 2018. Progress and prospects of the human–robot collaboration. Autonomous Robots 42, 957–975.

Carter, B. T., Luke, S. G., 2020. Best practices in eye tracking research. International Journal of Psychophysiology 155, 49–62.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24 (6), 381–395.

Gonzalez-Diaz, I., Benois-Pineau, J., Domenger, J.-P., Cattaert, D., de Rugy, A., 2019. Perceptually-guided deep neural networks for ego-action prediction: Object grasping. Pattern Recognition 88, 223–235.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37 (9), 1904–1916.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.

Huang, C.-M., Mutlu, B., 2016. Anticipatory robot control for efficient human-robot collaboration. In: 2016 11th ACM/IEEE international conference on human-robot interaction (HRI). IEEE, pp. 83–90.

Melekhov, I., Kannala, J., Rahtu, E., 2016. Siamese network features for image matching. In: 2016 23rd international conference on pattern recognition (ICPR). IEEE, pp. 378–383.

Naneva, S., Sarda Gou, M., Webb, T. L., Prescott, T. J., 2020. A systematic review of attitudes, anxiety, acceptance, and trust towards social robots. International Journal of Social Robotics 12 (6), 1179–1201.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823.

Shi, L., Copot, C., Vanlanduit, S., 2021. Gazeemd: Detecting visual intention in gaze-based human-robot interaction. Robotics 10 (2), 68.

Vezzani, G., Pattacini, U., Natale, L., 2017. A grasping approach based on superquadric models. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 1579–1586.

Weber, D., Santini, T., Zell, A., Kasneci, E., 2020. Distilling location proposals of unknown objects through gaze information for human-robot interaction. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 11086–11093.

Yu, C., Schermerhorn, P., Scheutz, M., 2012. Adaptive eye gaze patterns in interactions with human and artificial agents. ACM Transactions on Interactive Intelligent Systems (TiiS) 1 (2), 1–25.

Zhang, Y., Beskow, J., Kjellström, H., 2017. Look but don't stare: Mutual gaze interaction in social robots. In: Social Robotics: 9th International Conference, ICSR 2017, Tsukuba, Japan, November 22-24, 2017, Proceedings 9. Springer, pp. 556–566.