XLIV Jornadas de Automática 2023

https://jautomatica.es

Integración en robot social Mini del juego "Veo, veo"

Arrojo, G.[a,*], Borrero, J.[a], Garcia, J.[a], Castillo, J.C.[a], Castro-González, A.[a], Salichs, M.A.[a]

[a]*Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid. Calle de Butarque, 15. Leganés, 29811. Madrid, España.*

To cite this article: Arrojo, G., Borrero, J., Garcia, J., Castillo, J.C., Castro-González, A., Salichs, M.A 2023. Integration in Mini Social Robot of the Game "I spy". XLIV Jornadas de Automática, 501-506. https://doi.org/10.17979/spudc.9788497498609.501

Resumen

Este trabajo presenta el desarrollo de una aplicación de entretenimiento integrada en un robot social para jugar al juego clásico *Veo, veo*. El objetivo es dotar al robot de las capacidades perceptivas necesarias para participar en el juego, pudiendo tomar el rol de adivinar o pensar objetos. El robot utiliza un sistema de visión artificial y modelos de reconocimiento de objetos para identificar los elementos del entorno y obtener características especificas de estos (nombre, posición, tamaño). Además, se ha desarrollado una habilidad de juego que dispone de un sistema de pistas dinámico basado en las características obtenidas en la detección de objetos. Ambos bloques, la detección y la habilidad, han sido integrados en la arquitectura del robot social Mini. Este trabajo tiene como objetivo desarrollar una habilidad de entretenimiento y acompañamiento que fomente las habilidades de deducción y observación del usuario.

Palabras clave: Robótica social, Interacción humano-robot, Visión por computador, Detección de objetos, Entretenimiento, Estimulación cognitiva

Integration in Mini Social Robot of the Game "I spy"

Abstract

This work presents the development of an entertainment application integrated in a social robot to play the classic game I spy. The objective is to provide the robot with the necessary perceptual capabilities to participate in the game, being able to take the role of guessing or thinking objects. The robot uses an artificial vision system and object recognition models to identify the elements of the environment and obtain specific characteristics of these (name, position, size). In addition, a game skill has been developed that has a dynamic hint system based on the features obtained in object detection. Both blocks, the detection and the skill, have been integrated in the architecture of the Mini social robot. This work seeks to develop an entertainment and companion skill that fosters the user's deduction and observation skills.

Keywords: Social robotics, Human-robot interaction, Computer vision, Object detection, Entertainment, Cognitive stimulation

1. Introducción

La creciente necesidad de combatir la soledad y el aburrimiento ha impulsado el desarrollo de soluciones orientadas a satisfacer estas demandas. La robótica social se puede postular como una herramienta asistencial y un nuevo enfoque de entretenimiento, capaz de integrarse con otras áreas en auge como los asistentes de voz o la inteligencia artificial. En la actualidad, existen precedentes de robots sociales con habilidades entretenimiento como Misa[1], Miko[2], Jibo (Rane et al., 2014), Eilik[3].

Una de las principales actividades de entretenimiento son los juegos, donde las interacciones son bastante simples de diseñar ya que las propias reglas de las que se compone un juego sirven como base para diseñar una estructura lógica que puede ser seguida por un robot.

Las relaciones entre los seres humanos están influenciadas

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

[1]`https://www.heymisa.com`

[2]`https://miko.ai/`

[3]`https://energizelab.com/consumerview/eilik`

por el entorno en el que se desarrollan: Moda, decoración del espacio, olores, etc. Por lo tanto, una interacción humano-robot (*HRI*) donde el robot sea capaz de percibir y adaptarse al entorno será percibida de forma más natural por el usuario (Goodrich and Schultz, 2008).

Partiendo de las afirmaciones previas, en este artículo consideramos que el juego Veo, veo es una buena aplicación de entretenimiento a desarrollar en un robot social. Se trata de un juego simple, que utiliza elementos del entorno como base de la interacción (integrando al robot dentro del entorno). El objetivo de este trabajo es el desarrollo e integración de una aplicación de entretenimiento en un robot social, dotándolo de las capacidades de visión necesarias para llevar a cabo el este juego.

El juego del Veo, veo consiste en adivinar objetos. Comienza con un jugador seleccionando mentalmente un objeto de su entorno. A continuación, ese jugador comienza diciendo la frase "Veo, veo ...". El jugador que se encarga de sugerir la palabra a adivinar proporciona pistas descriptivas sobre el objeto seleccionado, generalmente utilizando una característica morfológica o cualidad distintiva como color, tamaño, posición. Los demás jugadores propondrán objetos para tratar de adivinar el elegido. El jugador que lo adivina se convierte en el próximo *buscador*. Juegos tradicionales como el Veo, veo fomenta la observación y la capacidad deductiva, por lo que puede ser empleado como una aplicación de estimulación cognitiva Calvo et al. (2010); **?**. Además, promueve la interacción social y la diversión en grupos de jugadores. Dependiendo del nivel de dificultad deseado, se puede establecer restricciones adicionales en las pistas proporcionadas, como limitar las características descriptivas o incluir categorías específicas de objetos o personas.

El elemento clave en este juego es la capacidad de detectar objetos del entorno. En los últimos años, la detección de objetos ha obtenido grandes avances gracias al aprendizaje profundo y las redes neuronales convolucionales (LeCun et al., 2015). Los procesos de detección de objetos se componen de dos fases: La segmentación de la imagen en busca de las regiones de interés que contienen los objetos y la clasificación de cada una de las regiones. Algunos de los modelos de machine learning más utilizados para la detección de objetos son You Only Look Once (YOLO) (Redmon et al., 2016), Single Shot MultiBox Detector (SSD) (Liu et al., 2016), Faster R-CNN (Girshick, 2015), o Detectron2 (Pham et al., 2020).

Este artículo se estructura de la siguiente manera: En la sección 2 se describen las herramientas empleadas en el desarrollo de la aplicación. En la sección 3 se explica el sistema de percepción desarrollado para la detección de objetos. A continuación, en la sección 4 se describe la implementación del juego dentro del robot social. La sección 5 expone las conclusiones del trabajo realizado.

2. Tecnologías utilizadas

A continuación se abordan algunas de las tecnologías más relevantes para el desarrollo del trabajo, desde las librerías implicadas en la detección de objetos, a la plataforma robótica utilizada.

2.1. Librerías utilizadas

El trabajo ha sido desarrollado en el sistema operativo Linux, específicamente diseñado para funcionar sobre Robot Operating System (ROS) (Quigley et al., 2009). El lenguaje de programación utilizado ha sido Python. A continuación, se detallan algunas de las librerías más importantes utilizadas en el trabajo.

2.1.1. Open CV

Para el procesamiento de imágenes, se ha utilizado Open CV[4], una librería inicialmente desarrollada por Intel para manipular y analizar imágenes. Incluye funciones de procesamiento de imágenes como lectura, escritura, manipulación de píxeles, transformaciones geométricas, filtrado de imágenes, detección y extracción de características, seguimiento de objetos, reconocimiento de patrones, calibración de cámaras o reconstrucción 3D. Se encuentra desarrollado en C++ pero tiene soporte para otros lenguajes como Python o Java. Se trata de una librería multiplataforma compatible con Windows, Linux, macOS, Android e iOS. En este trabajo se ha hecho uso de esta librería para el preprocesamiento de las imágenes (edimensionado y binarización).

2.1.2. YOLOv3

Realizando una comparativa a nivel de rendimiento de los modelos que componen el estado del arte en la detección de objetos: SSD, FRCNN, YOLO; Se ha tomado la decisión de utilizar el modelo YOLOv3, dado que comparado con otros modelos similares como FRCNN y SSD, resulta ser la opción más equilibrada entre tiempo de inferencia y precisión media del modelo. "En un Pascal Titan X, procesa imágenes a 30 FPS y tiene un mAP del 57,9 % en COCO test-dev" (Redmon and Farhadi, 2018). Existen versiones de YOLO más modernas que son capaces de clasificar en un mayor número de categorías, no obstante el gestor de interacción reconoce un numero limitado de palabras, por esto se ha descartado el uso de versiones más complejas.

YOLO es de un detector de objetos que no está basado en clasificadores, al contrario que otros detectores como Faster R-CNN. YOLO realiza predicciones con una sola evaluación, analizando todo el contexto de la imagen a diferencia de otros detectores que necesitan cientos de evaluaciones para cada imagen, lo que permite reducir en gran medida el tiempo necesario para realizar las detecciones[5]. El modelo preentrenado utilizado, YOLOv3, ha sido entrenado con la base de datos COCO que contiene más de 200,000 imágenes etiquetadas que representan 80 categorías. Es posible obtener el modelo entrenado con esta base de datos en el repositorio darknet (Wang et al., 2022)[6]. YOLOv3 ha servido como herramienta para detectar objetos y obtener características preliminares de estos (nombre, centroide y región de interés).

[4] `https://opencv.org/`
[5] `https://pjreddie.com/darknet/yolo/`
[6] `https://github.com/pjreddie/darknet`

2.2. *Mini*

El robot social Mini (Salichs et al., 2020) (ver figura 1) ha sido la plataforma utilizada para el desarrollo de esta aplicación. El objetivo inicial de este robot era realizar tareas de estimulación cognitiva con personas mayores. A raíz de su desarrollo, se le han incorporado funcionalidades y aplicaciones nuevas relacionadas, entre otras, con el entretenimiento. Respecto a las características físicas de Mini, se trata de un robot de sobremesa de forma humanoide. Su tamaño es de aproximadamente unos 50 centímetros, está dotado de motores para simular el movimiento de brazos torso y cabeza, aunque carece de capacidad de desplazamiento. Dispone de luces LED RGB repartidas por toda su estructura, además de dos ojos animados capaces de mostrar diferentes expresiones o incluso seguir al usuario. La forma de interacción principal de Mini es a través de comunicación oral aunque dispone de una tableta que sirve como sistema auxiliar para algunas interacciones específicas.

Figura 1: Robot social Mini.

El robot Mini dispone de dos cámaras: una cámara RGB situada en la barbilla del robot, y una cámara de profundidad RGB-D (Intel RealSense depth-camera-d435i"[7]) utilizada en este trabajo para tomar las imágenes del entorno del robot. El *software* se ejecuta en un ordenador embarcado que permite la ejecución en tiempo real de toda la arquitectura sin necesidad de utilizar servidores externos. Además, Mini dispone de dos aceleradores gráficos que permiten ejecutar en tiempo real modelos de inteligencia artificial: Google Coral TPU[8] y el Intel Movidius NCS2[9] y no sobrecargar la CPU.

La arquitectura *software* del robot Mini está implementada sobre ROS (Quigley et al., 2009) y se divide en 5 módulos principales: el *sistema de percepción* (Alonso Martín et al., 2017) obtiene y procesa la información del entorno del robot y del usuario con el que interactúa a través del conjunto de sensores cámaras, micrófonos, etc; el *gestor de interacción* (Rodicio, 2021) es el encargado de gestionar la información del sistema de percepción y a su vez generar los datos necesarios para enriquecer la interacción con el usuario; el *sistema de toma de decisiones* (Maroto-Gómez et al., 2018)ejecuta y controla las habilidades del robot; y el *gestor de expresiones* (Maroto-Gómez et al., 2022) gestiona el control de los actuadores robot, movimiento, sonido, luces.

3. Detección de los objetos en el entorno del robot

La acción principal que constituye la base de toda la interacción en el juego Veo, veo consiste en buscar y adivinar objetos presentes en el entorno. Por consiguiente, resulta imperativo proveer al robot de una capacidad que le permita observar su entorno y detectar los objetos (ver figura 2). Para ello, se ha desarrollado un sistema de percepción encargado de procesar las imágenes de la cámara de la base del robot y obtener el listado de objetos de su entorno. En la figura 3 se puede observar las etapas por las que pasa el módulo de percepción con el objetivo de obtener una lista de objetos junto a sus respectivas características: Nombre, posición, tamaño y distancia.

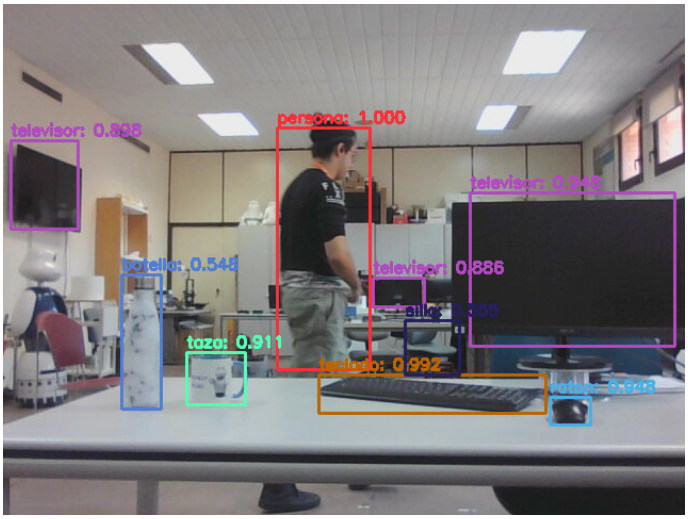

Figura 2: Detección de objetos en un fotograma tomado por Mini durante el juego.

El primer paso de la detección de objetos consiste en *calcular el número de imágenes* a tomar, depende directamente de la relación entre el ángulo de visión de la cámara y el rango de rotación abarcado por el robot. A continuación, se realiza la *toma de imágenes* con la cámara de la base obteniendo un listado de imágenes RGB que se transformarán a binarias y se redimensionarán a 416 x 416 píxeles para coincidir con la capa de entrada del modelo utilizado. Para cada una de las imágenes de color, se obtiene la imagen de profundidad que se utilizará mas adelante para ampliar la información característica de cada objeto. En el tercer paso, se realiza la *detección de objetos*(figura 3, color rosa) utilizando el modelo *YOLO* para cada una de las imágenes. Con la información obtenida del detector (nombre, centroide

[7] https://www.intelrealsense.com/depth-camera-d435i/
[8] https://coral.ai/
[9] https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html

y región de interés), se genera un listado de objetos para cada imagen. El último paso consiste en utilizar los datos obtenidos por el detector de objetos y la información proporcionada por la imagen de profundidad para *Extraer información relevante*, obteniendo la *distancia*, *posición* relativa al robot o el *tamaño* del objeto. Para calcular la distancia entre el robot y los objetos, se normaliza la imagen de profundidad adaptándola al rango de medida de la cámara: [0.3-3] metros. A partir de las coordenadas X,Y del centroide de cada objeto se obtiene la distancia correspondiente en la imagen normalizada. Para calcular la posición relativa, utilizando los mismos parámetros (coordenadas X,Y, profundidad) con las funciones propias de la cámara utilizada, se calculan las transformaciones geométricas obteniendo la posición en los tres ejes (X,Y,Z) respecto al robot medidos en centímetros. Finalmente, para el cálculo del tamaño del objeto se calcula el área de la región de interés medido en píxeles. Esta parte de desarrollo ha sido integrada dentro del Sistema de percepción del robot Mini.

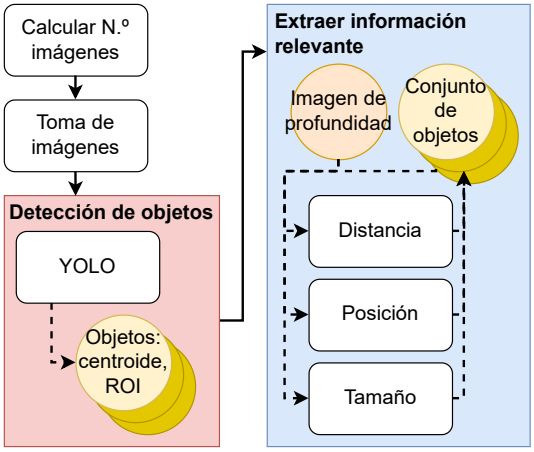

Figura 3: Esquema de la detección de objetos.

Dado que la selección del objeto, ya sea por parte del usuario o del robot, siempre se lleva a cabo al comienzo del juego, se considera que el funcionamiento bajo demanda barca las necesidades de la habilidad, sin necesidad de una ejecución en tiempo real. Como las modificaciones en el entorno no afectarán al objeto previamente seleccionado, el robot tomará una serie de fotografías que abarquen la mayor área de visión posible. De esta forma se evitará un sobrecoste innecesario de procesamiento.

4. Desarrollo de la habilidad Veo, veo para el robot social Mini

El trabajo realizado tiene como objetivo el desarrollo e integración de una aplicación de entretenimiento inspirada en el clásico juego infantil Veo, veo. Esta aplicación requiere de información proveniente del entorno y se ha diseñado para que los usuarios puedan interactuar por voz con el robot de manera natural y fluida. No requiere de accesorios complementarios como tabletas.

4.1. Estructura de la habilidad

El funcionamiento de la habilidad es la siguiente (ver figura 4): Al inicio de la habilidad, el robot realizará una breve presentación del juego en el que va a participar el usuario. Primero le preguntará si conoce las reglas del juego, en caso negativo, le *dará las instrucciones*. Después, se pedirá al usuario que*establezca un modo de juego* (Usuario o robot piensa objeto). Tras este proceso, comienza la operación general del juego: En primer lugar, la habilidad activará momentáneamente el módulo de detección para que realice una toma de imágenes y realice el*reconocimiento de objetos* en la escena (figura 2). En paralelo, se da un breve periodo de tiempo al usuario para hacer lo mismo. Una vez el detector finalice, le devolverá a la habilidad el listado con los objetos detectados, donde cada objeto contendrá toda la información obtenida por el detector (nombre, tamaño, posición, entre otros). Tras la detección de objetos objetos existen dos flujos de funcionamiento posibles dependiendo de quién decide el objeto a adivinar: usuario o robot.

En el caso de que el *usuario piense un objeto*(ver figura 4, bloque de color morado): El robot intentará *adivinar el objeto* para esto es posible configurar la aplicación para aplicar diferentes estrategias de selección de entre las desarrolladas: como seleccionar por el número de veces que se repite el objeto, por tamaño o de forma aleatoria. Si acierta se terminará la partida. En caso de que no acierte existe la posibilidad de que el robot pida una pista al usuario, la probabilidad de que esto suceda decrece a medida que el robot pide más pistas hasta que ya no queden pistas posibles por dar (ninguna pistas pedidas, 100 %. Todas las pistas perdidas, 0 %.) . Si el robot *pide pista* selecciona una de las pistas disponibles que no haya sido planteada previamente y *filtra* el conjunto de objetos, eliminando aquellos que no cumplan los requisitos de la pista obtenida. En cualquier caso, el robot volverá a *decidir un objeto* hasta dar con el adecuado o quedarse sin opciones que cumplan las condiciones.

En el caso de que el *robot piense un objeto*(ver figura 4, color azul), el robot accederá al listado de objetos detectados previamente y *seleccionará un objeto* siguiendo las mismas directrices que *adivinar el objeto* 4.1 en *usuario piense un objeto*. Una vez seleccionado, procederá a *preguntar al usuario* "Veo, veo una cosita, ¿qué crees que es?" si el usuario acierta se terminará la partida. En caso contrario es posible que el robot proponga una pista con una probabilidad variable como la de *pide pista* 4.1 en *usuario piense un objeto*. Si el robot decide *dar pista*, escogerá aleatoriamente una que no haya dado previamente, aplicando una función al objeto seleccionado (primera letra, número de letra, tamaño, entre otros). En ambos casos el robot volverá a *preguntar al usuario* hasta que éste acierte o se rinda.

Una vez finalizada la partida el robot propondrá al usuario la posibilidad de volver a jugar. En este caso volverá al inicio de la *estructura general del juego*.

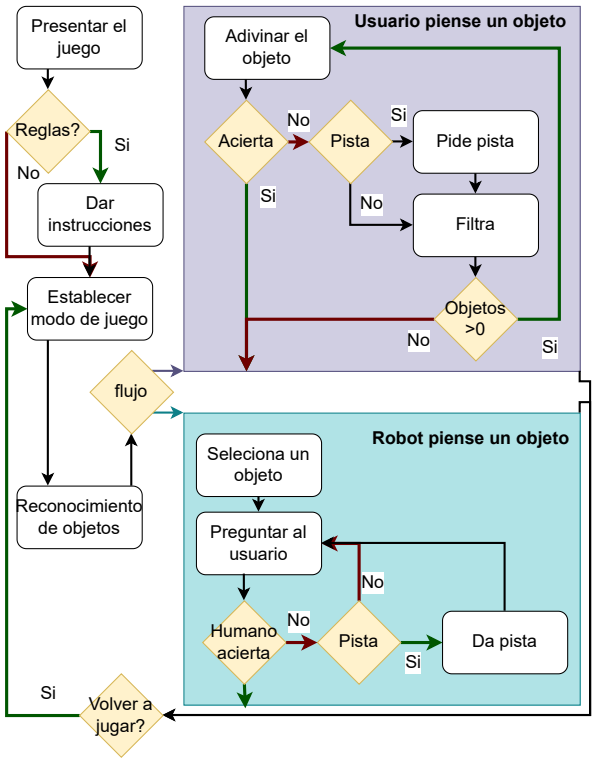

Figura 4: Diagrama de flujo de la habilidad Veo, veo

.

4.2. Niveles de acciones

Uno de los objetivos de este trabajo es obtener una habilidad de percepción modular, que pueda adecuarse a distintos experimentos y sirva como precedente para futuros proyectos de visión y percepción del entorno. Atendiendo al objetivo de las acciones implicadas en la habilidad podemos diferenciar dos niveles: Nivel de interacción, aquellas acciones enfocadas en la experiencia de usuario. Y un nivel estratégico, que son acciones enfocadas en dotar al juego de dinamismo.

- Nivel de interacción: En este nivel se encuentran todas las acciones que afectan directamente a la experiencia de usuario, las frases y expresiones usadas por el robot o la estructura general del juego. Acciones como presentar el juego, dar instrucciones, preguntar al usuario.
- Nivel de estrategia: Agrupa todos aquellos elementos específicos del juego que, aunque son independientes de la interacción, pueden afectar la experiencia de usuario. En este nivel se incluyen acciones como los métodos de selección de objetos o los sistemas de pistas.

5. Conclusiones

En el presente trabajo se ha desarrollado e integrado una aplicación de entretenimiento en un robot social, cuya evaluación esta sujeta a futuras pruebas con usuarios. Dicha aplicación consta de dos módulos: (i) un sistema de percepción encargado de gestionar la información del entorno del robot y listar los objetos junto a sus propiedades más relevantes y (ii) una habilidad que guía al usuario durante la interacción con naturalidad. Además, se ha desarrollado un software que permite su adaptación y reestructuración a diversos niveles, permitiendo su reutilización en otro tipo de interacciones que impliquen reconocimiento de objetos en el entorno.

No obstante, cabe destacar que el rendimiento de los algoritmos de detección de objetos en una imagen están sujetos a un coste computacional relevante. Consideramos que las soluciones bajo demanda pueden ser una alternativa, que satisfaga las necesidades de visión en aquellos casos donde los datos no varían significativamente “como en la habilidad del veo veo”.

Creemos que la integración de habilidades de percepción dentro de interacciones sociales puede abrir el campo de desarrollo a nuevas habilidades, no solo enfocadas en el entretenimiento, sino que abarquen tareas como la estimulación cognitiva (reconocer objetos junto con el usuario, señalarlos o brindar características pueden ser consideradas como algunas de estas capacidades).

En cuanto a las posibles mejoras del sistema, es necesario considerar que el sistema depende en gran medida de técnicas de detección de objetos. Sería posible introducir otros detectores que permitan extraer mayor información de la imagen, como el color de los objetos, u otras propiedades o modelos preentrenados con otras clases con el fin de obtener experiencias de usuario más personalizadas y completas. Además, sería posible reestructurar el bloque de interacción para orientarse a aplicaciones asistenciales como ayuda en la búsqueda y localización de objetos en el hogar del usuario.

Agradecimientos

La investigación que ha conducido a estos resultados ha recibido financiación de los proyectos: «RoboCity2030-DIH-CM», Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, financiado por «Programas de Actividades I+D en la Comunidad de Madrid» y cofinanciado por el Fondo Social Europeo (FSE) de la EU; «Robots sociales para mitigar la soledad y el aislamiento en mayores» (SoRoLi), PID2021-123941OA-I00, financiado por la Agencia Estatal de Investigación (AEI), Ministerio Español de Ciencia e Innovación; «Robots sociales para reducir la brecha digital de las personas mayores» (SoRoGap), TED2021-132079B-I00, financiado por la Agencia Estatal de Investigación (AEI); «Mejora del nivel de madurez tecnológica del robot Mini» (MeNiR), PDC2022-133518-I00, financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea NextGenerationEU/PRTR.

Referencias

Alonso Martín, F., Castillo, J. C., Castro González, Á., Gamboa, J. J., Maroto, M., Marqués Villaroya, S., Pérez Vidal, A. J., Salichs, M. Á., et al., 2017. Percibiendo el entorno en los robots sociales del roboticslab. Actas de las XXXVIII Jornadas de Automática.

Calvo, B. F., Castillo, I. C., Serna, A., de Lucena, V. M., Campos, F. R., 2010. El efecto del formato de intervención individual o grupal en la estimulación cognitivade pacientes con enfermedad de alzheimer. Revista de psicopatología y psicología clínica 15 (2), 115–123.

Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448.
URL: https://github.com/rbgirshick/

Goodrich, M. A., Schultz, A. C., 2008. Human–robot interaction: A survey. Foundations and Trends® in Human–Computer Interaction 1, 203–275.
URL: http://dx.doi.org/10.1561/1100000005
DOI: 10.1561/1100000005

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521 (7553), 436–444.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C., 2016. SSD: Single shot multibox detector. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 9905 LNCS. Springer Verlag, pp. 21–37.
URL: https://link.springer.com/chapter/10.1007/978-3-319-46448-0{_}2
DOI: 10.1007/978-3-319-46448-0_2

Maroto-Gómez, M., Castro-González, Á., Castillo, J. C., Malfaz, M., Salichs, M. A., 2018. A bio-inspired motivational decision making system for social robots based on the perception of the user. Sensors 18 (8), 2691.

Maroto-Gómez, M., Villarroya, S. M., Malfaz, M., Castro-González, Á., Castillo, J. C., Salichs, M. Á., 2022. A preference learning system for the autonomous selection and personalization of entertainment activities during human-robot interaction. In: 2022 IEEE International Conference on Development and Learning (ICDL). IEEE, pp. 343–348.

Pham, V., Pham, C., Dang, T., 2020. Road damage detection and classification with detectron2 and faster r-cnn. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, pp. 5592–5601.

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.

Rane, P., Mhatre, V., Kurup, L., 2014. Study of a home robot: Jibo. International journal of engineering research and technology,.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2016-Decem. pp. 779–788.
URL: http://pjreddie.com/yolo/
DOI: 10.1109/CVPR.2016.91

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Rodicio, E. F., 2021. Human-robot interaction architecture for interactive and lively social robots. Ph.D. thesis, Universidad Carlos III de Madrid.

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., dec 2020. Mini: A New Social Robot for the Elderly. International Journal of Social Robotics 12 (6), 1231–1249.
URL: https://link.springer.com/article/10.1007/s12369-020-00687-0
DOI: 10.1007/s12369-020-00687-0

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M., 2022. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.
URL: https://arxiv.org/abs/2207.02696
DOI: 10.48550/ARXIV.2207.02696