DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
ISSN: 2007-6363

Publicación Semestral Pädi Vol. X No. 00 (2022) 1–8

Reconocimiento y estimación de posición relativa de objetos en entornos controlados
Recognition and estimation of relative position of objects in controlled environments

Eloy Clemente [a,*], Enrique Luna [a], José Luis Gómez [a], Isaac Villa [a]

[a] *Tecnológico Nacional de México Campus La Paz, Boulevard Forjadores de Baja California Sur No.4720 C.P. 23080*

Resumen

En este proyecto, se presenta un sistema de reconocimiento y clasificación de objetos que se encuentran en su entorno, para un robot de asistencia, así como la estimación de la posición relativa de estos con respecto al robot. Para el reconocimiento y clasificación de los objetos, se aplicaron técnicas de visión artificial basadas en herramientas de segmentación semántica, como son redes neuronales convolucionales. Para la estimación de la posición relativa de los objetos, una vez identificados, se implementó una técnica de visión estereoscopio. Los resultados de los experimentos muestran un 90.6 % de precisión en el reconocimiento y clasificación, y un error medio de 5 cm al estimar la posición relativa de los objetos.

Palabras Clave: Visión artificial, Redes neuronales convolucionales, Segmentación semántica, Visión estereoscopio, Inteligencia artificial.

Abstract

This project presents a system for recognition and classification of objects that are in its environment, for an assistance robot, as well as the estimation of their relative position with respect to the robot. For the recognition and classification of objects, we apply artificial vision techniques based on semantic segmentation tools, such as convolutional neural networks. For the estimation of the relative position of the objects, once identified, a stereoscopic vision technique was implemented. The results of the experiments show a 90.6 % accuracy in recognition and classification, and an average error of 5 cm when estimating the relative position of the objects.

Keywords: Computer vision, Convolutional neural networks, Semantic segmentation, Stereoscopic vision, Artificial intelligence.

1. Introducción

Debido a la escasez y costos de personal de salud y de asistencia en muchos países, los pacientes que están postrados en cama debido a problemas de salud, accidentes o por envejecimiento, en ocasiones no cuentan con el apoyo de personal que se ocupe de sus necesidades. Esto ha conducido a realizar esfuerzos por aplicar la robótica y la automatización para cubrir estas tareas de asistencia.

Para que un robot de asistencia personal realice acciones que satisfagan determinadas necesidades fisiológicas, cuando no se cuenta con apoyo humano, debe de contar con las siguientes funciones: estimar su propia ubicación y la de los objetos, llegar al destino deseado evitando obstáculos, manipular los objetos, identificar las necesidades fisiológicas de los pacientes, y razonar sobre los objetos y operaciones necesarias para satisfacer estas necesidades fisiológicas.

Se estima que en las próximas décadas, en algunos países se duplicará el número de personas mayores de 80 años, y que se tendrá un déficit muy importante de profesionales de la salud. Si pudiéramos desarrollar robots de cuidado personal capaz de realizar tareas de cuidado y asistencia básicas, la necesidad de apoyo de personal humano se reduciría significativamente. Sin embargo, a pesar de que se han desarrollado varios tipos de robots para brindar asistencia y cuidados, aún quedan retos importantes por superar, hasta lograr que los robots sean completamente autónomos y realicen de forma efectiva y segura estas tareas (Yang et al., 2019),(Miseikis et al., 2020).
El objetivo de este proyecto, es aportar en el desarrollo de este tipo de tecnologías, con la intención de que en un futuro estén al alcance de todos los sectores de la población que la requieran. En la organización Olin Robotics se está desarrollando un prototipo de exoesqueleto sobre una plataforma móvil, cuyo objetivo es asistir en el proceso de incorporación (sentado a parado)

*Autor para correspondencia: Eloy Antonio Clemente Rosas L17310793@lapaz.tecnm.mx
Correo electrónico: (jorge.lt@lapaz.tecnm.mx) Enrique Luna, (jose.gt@lapaz.tecnm.mx) José Luis Gómez, (isaac.vm@lapaz.tecnm.mx) Isaac Villa

Fecha de recepción: DD/MM/AAAA Fecha de aceptación: DD/MM/AAAA Fecha de publicación: DD/MM/AAAA
https://orcid.org/0000-0002-6915-5567

del usuario; además se considera que cuando el usuario no tenga el exoesqueleto colocado, este pueda tener funcionamiento autónomo para ir al centro de carga y cubrir algunas tareas de asistencia, como acercar vasos, platos o el móvil del usuario.

2. Antecedentes

En los últimos años, se han propuesto diferentes sistemas para tareas de asistencia y cuidado personal. Por ejemplo, para este fin, en (Miseikis et al., 2020) presentan el desarrollo de una plataforma de robot móvil con un brazo multifuncional, combinando dispositivos de visión, audio, laser, ultrasonido y sensores mecánicos, controlados a través del sistema ROS (Robotic Operating System) y algoritmos de aprendizaje profundo. Durante la pandemia de COVID-19, ajustaron el robot, en corto tiempo, para realizar tareas de desinfección y toma de temperatura.

En (Jishnu et al., 2020) proponen el diseño de un robot de asistencia, controlado a través de comandos de voz, para transportar objetos a cortas y largas distancias. La comunicación humano-robot se establece a través de dispositivos móviles vía Bluetooth. El robot cuenta con una cámara para la detección de los objetos, así como para calcular la distancia aproximada de estos, aplicando el algoritmo YOLO (You Only Look Once), basado en Redes Neuronales Convolucionales.

Así mismo, en (Diddeniya et al., 2018) presentan el diseño de un robot que opera en ambientes de oficina, utilizando sensores para visión 3D, controlados también a través del sistema ROS. Los comandos se transmiten a través de un dispositivo móvil, que se conecta con una estación de trabajo central y el robot, vía Wi-Fi.

Independientemente de la aplicación, se han desarrollado una variedad de sistemas para localización de objetos y navegación autónoma. Por ejemplo, en (Ferreira, 2013) muestran el diseño de un vehículo para tareas de navegación autónoma, a partir de un mapa que crea de su entorno el propio sistema. El vehículo recopila la información a través de un sistema de visión, para identificar la trayectoria a seguir y detectar obstáculos. Para completar todas sus tareas, el vehículo es controlador de forma remota.

En (Purwanto et al., 2017) presentan el desarrollo de un sistema de navegación para interiores, a través de un sistema de visión multipunto a nivel del piso, para detectar obstáculos y áreas libres. Con esta información, aplican un sistema de inferencia difusa, para establecer la navegación del vehículo de forma autónoma.

3. Desarrollo

En este proyecto se propone un sistema de reconocimiento y clasificación de objetos, así como la estimación de la posición relativa de estos, con respecto a una cámara montada sobre un robot móvil. En la Figura 1 se muestra el diagrama general del sistema propuesto. El sistema tiene dos etapas de operación; la primera etapa abarca el preprocesamiento de las imágenes para entrenar el modelo y el entrenamiento del mismo. Estas dos tareas se realizan *offline*, en una computadora de escritorio. La segunda etapa corresponde al reconocimiento y clasificación de los objetos, así como la estimación de la posición relativa de estos. Estas tareas se realizan *online*, en un sistema Raspberry Pi 4.

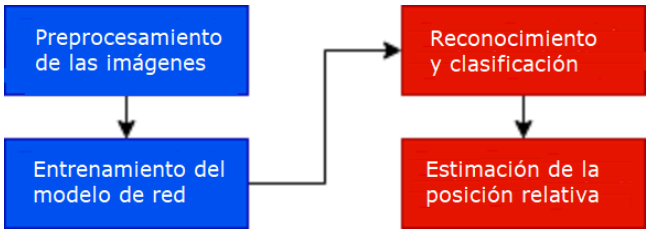

Figura 1: Diagrama general del funcionamiento del sistema.

3.1. *Preprocesamiento de las imágenes*

La preparación de las imágenes para el entrenamiento del modelo, consistió en la siguientes tareas:

- Se definió un conjunto de objetos de interés, a los cuales se les asignó un índice de clase y un color (ver Tabla 1).

Tabla 1: Definición de los objetos de interés.

Objeto	Color RGB	Índice
Fondo	(0, 0, 0)	0
Conector	(0, 128,128)	1
Celular	(128, 0, 0)	2
Plato	(0, 0, 128)	3
Vaso	(128, 128, 0)	4
Mesa	(0, 128, 0)	5
Personas	(192, 0, 0)	6
Herramienta	(128, 0, 128)	7
Garrafón de agua	(0,192,0)	8
Obstáculos	(0,0,192)	9

- Se capturaron un total de 397 imágenes, que incluyen a los objetos de interés, a no más de 2.5 metros de distancia. Las imágenes se guardaron originalmente en formato JPG, con una resolución de 640x480 pixeles. En la Tabla 2 se muestran las características básicas de la cámara utilizada.

Tabla 2: Especificaciones de la cámara.

Tipo de obturador	Rodante
Velocidad de fotogramas	21fps 8mp
	30fps 1080p
	120fps 720p
Ángulo de visión	75 grados (horizontal)

- Se delimitó manualmente el contorno de los objetos de interés presentes en cada imagen, utilizando el software *VIA VGG Image Anotattor, Versión 2.0.11* [1] (ver Figura 2).

[1] https://www.robots.ox.ac.uk/ vgg/software/via/

- Se generó un archivo JSON con la definición de los polígonos que representan el contorno correspondiente de cada objeto.

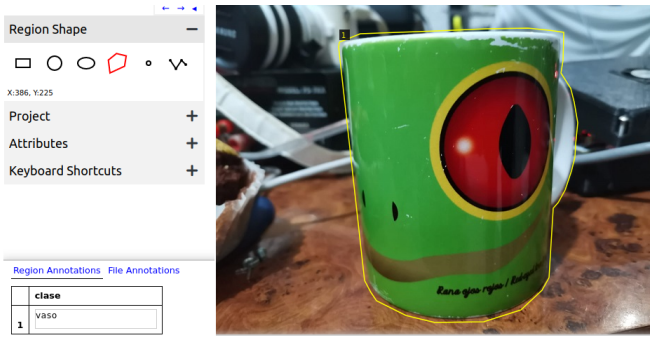

Figura 2: Delimitación del contorno de los objetos de interés.

- Mediante rutinas escritas en Python, a partir del archivo JSON, se generaron imágenes en formato PNG, con los objetos segmentados con su color correspondiente (ver Figura 3).

Figura 3: Segmentación de los objetos de interés.

- Se escalaron, tanto las imágenes originales, como las imágenes segmentadas, a una dimensión de 320x240 pixeles.
- Se convirtieron las componentes de color RGB, de cada pixel de las imágenes segmentadas, a un valor único que corresponde a la clase del objeto al que pertenece el pixel, y se almacenó la matriz resultante en un archivo con formato numpy (NPY).

3.2. *Entrenamiento del modelo*

Con el objetivo de reconocer y clasificar los objetos en etapas posteriores, se implementó una estrategia de *Segmentación Semántica*, la cual consiste en clasificar las imágenes a nivel de píxel. Para esto, se diseñó y entrenó un modelo de red neuronal, que combina capas convolucionales y convolucionales transpuestas.

La arquitectura de la red utilizada es del tipo U-Net, basada en el modelo ResNet18, al cual se le eliminaron las dos últimas capas, y se le agregaron cuatro capas convolucionales transpuestas, combinadas con capas convolucionales comunes (ver Tabla 3).

Tabla 3: Arquitectura de la red neuronal utilizada.

Capas	Canales	Dimensión	K	Stride	Padding
Imagen	3	320x240			
Conv	64	160x120	7x7	2x2	3x3
Conv	64	160x120	3x3	1x1	1x1
Conv	64	160x120	3x3	1x1	1x1
Conv	64	160x120	3x3	1x1	1x1
Conv	64	160x120	3x3	1x1	1x1
Conv	128	80x60	3x3	2x2	1x1
Conv	128	80x60	3x3	1x1	1x1
Conv	128	80x60	3x3	1x1	1x1
Conv	128	80x60	3x3	1x1	1x1
Conv	256	40x30	3x3	2x2	1x1
Conv	256	40x30	3x3	1x1	1x1
Conv	256	40x30	3x3	1x1	1x1
Conv	256	40x30	3x3	1x1	1x1
Conv	512	20x15	3x3	2x2	1x1
Conv	512	20x15	3x3	1x1	1x1
Conv	512	20x15	3x3	1x1	1x1
Conv	512	20x15	3x3	1x1	1x1
Conv Trans.	256	40x30	2x2	1x1	2x2
Conv	256	40x30	3x3	1x1	1x1
Conv Trans.	128	80x60	2x2	1x1	2x2
Conv	128	80x60	3x3	1x1	1x1
Conv Trans.	64	160x120	2x2	1x1	2x2
Conv	64	160x120	3x3	1x1	1x1
Conv Trans.	64	320x240	2x2	1x1	2x2
Conv	10	320x240	3x3	1x1	1x1

El entrenamiento de la red se implementó en Python versión 3.9.7, utilizando las librerías Pytorch versión 1.9.0 y Torchvision versión 0.10.0. Se utilizaron 350 imágenes para entrenamiento y 47 para pruebas. En la Tabla 4 se muestran los parámetros del entrenamiento, y en la Tabla 5 se muestran las características del equipo de cómputo donde se ejecutó el entrenamiento.

Tabla 4: Parámetros del entrenamiento.

Tamaño de lote	10
Número de épocas	100
Tasa de aprendizaje	0.0001
Factor de decaimiento	0.001
Función de pérdida	Entropía Cruzada
Función de optimización	Adam

Tabla 5: Características del equipo de cómputo para el entrenamiento.

Procesador	Intel Core i7-4790 3.60GHz x 8
Tarjeta gráfica	NVIDIA GeForce GTX TITAN X
Memoria RAM	16 GB
Disco duro	SSD 240 GB
Sistema operativo	Ubuntu 20.04 LTS

En la Figura 4 se muestra la estadística del entrenamiento. El valor final de la función de pérdida, a nivel de pixel, sobre

los datos de entrenamiento, es de 0.00641 (color azul) y el factor IoU de 0.87202 (color naranja). Sobre los datos de prueba, la función de pérdida concluyó con un valor de 0.07024 (color verde) y el factor IoU de 0.76395 (color rojo).

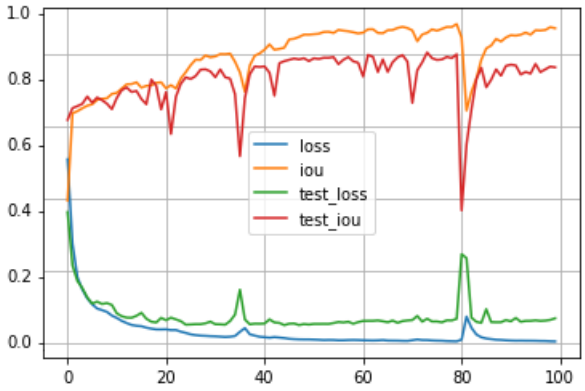

Figura 4: Estadística del entrenamiento.

3.3. *Reconocimiento y clasificación*

El proceso de reconocimiento y clasificación de los objetos se implementó en lenguaje Python, sobre un sistema Raspberry Pi 4, equipado con un módulo de cámara. Para esto, se almacenó previamente el modelo de la red entrenada en formato Pytorch (.pt), y se cargó sobre la Raspberry.

El proceso inicia con la captura de una imagen a través de la cámara, y se pasa a la red para su evaluación. La red devuelve una matriz tridimensional con la predicción de la clase a la que pertenece cada pixel.

Las dimensiones de la matriz son 320 x 240 x 10. Las primeras dos dimensiones corresponden al ancho y alto de la imagen, en pixeles, y la tercera dimensión corresponde a la cantidad de clases de los objetos de interés. El valor de cada celda indica la probabilidad estimada por la red, con valor entre 0 y 1, de que el pixel que se encuentra en esa fila y columna, sea de la clase que corresponde al subíndice de la tercera dimensión.

En la Figura 5 se muestra un ejemplo de predicción de la red, para el pixel que se encuentra en la esquina superior derecha (fila=0, columna=319). De acuerdo con este diagrama, el pixel tiene una probabilidad de 0.1 de pertenecer a la clase 1 o a la clase 8, y una probabilidad de 0.8 de pertenecer a la clase 2. Por lo tanto, la clase asignada a este pixel es la número 2.

Con base en este criterio, a continuación se genera una matriz bidimensional, con las dimensiones de la imagen de entrada, donde el valor de cada celda indica la clase inferida por la red para el pixel que se encuentra en esa posición.

En la Figura 6 se muestra un ejemplo de una imagen utilizada como entrada a la red. Y en la Figura 7 se muestra la imagen segmentada a partir de la inferencia realizada por la red.

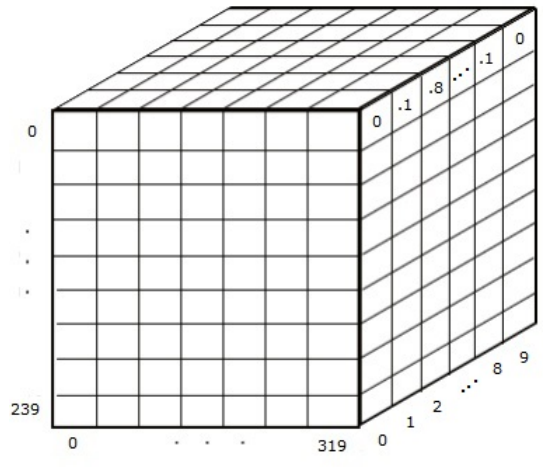

Figura 5: Ejemplo de matriz de salida de la red neuronal.

Figura 6: Imagen original de entrada a la red.

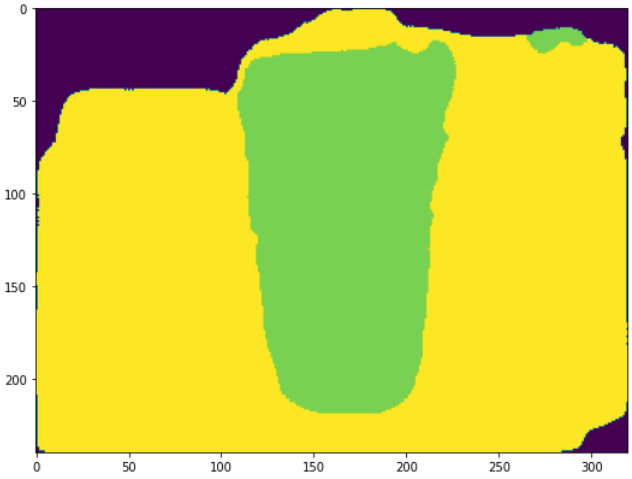

Figura 7: Segmentación inferida por la red neuronal.

3.4. *Estimación de la posición relativa*

Una vez identificados y segmentados los objetos en la imagen, el siguiente paso es estimar su posición relativa con res-

pecto al robot. Para esto, se aplicó una técnica de visión estereoscópica, para calcular la distancia a la que se encuentran los objetos del robot, a partir de dos imágenes tomadas en posiciones diferentes de la cámara.

Esta estrategia, asume que, a mayor distancia de los objetos con respecto a la cámara, menor será su desplazamiento, al comparar su posición en ambas imágenes. Y de forma correspondiente, a menor distancia de los objetos, mayor será su desplazamiento.

Para obtener la relación, entre el desplazamiento de los objetos en las imágenes y la distancia a la que se encuentran de la cámara, se capturaron ocho imágenes de tres puntos de control, posicionando la cámara a 50, 80, 100 y 150 cm de estos. Para cada una de estas distancias, se capturaron dos imágenes, colocando la cámara a 6 cm de distancia entre una y otra toma. (ver Figura 8).

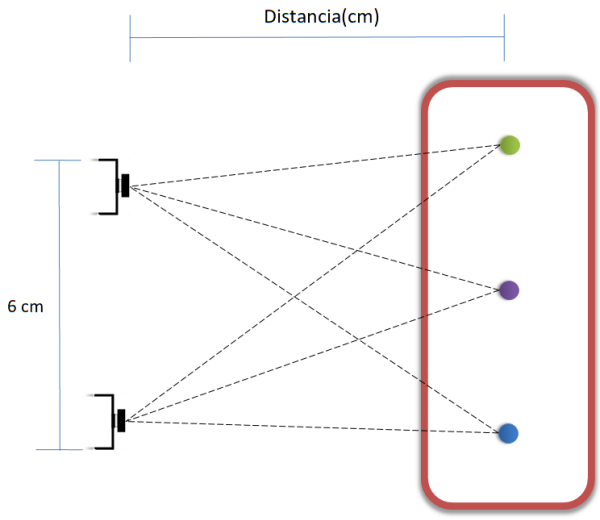

Figura 8: Captura de puntos de control desde dos posiciones distintas.

En la Tabla 6 se muestran la posiciones de los puntos de control en cada una de las imágenes capturadas. Las posiciones están dadas en pixeles sobre el eje X. Así mismo, se muestra el desplazamiento de los puntos entre las dos imágenes. Como desplazamiento asociado a cada una de estas distancias, se tomó el promedio de los tres puntos de control.

Tabla 6: Desplazamiento de los puntos de control.

		Posición en el eje X (pixeles)					
		Control uno		Control dos		Control tres	
		Pos.	Desp.	Pos.	Desp.	Pos.	Desp.
50 cm	Origen	320	73	197	74	76	72
	A 6 cm	247		123		4	
80 cm	Origen	320	47	239	48	160	48
	A 6 cm	273		191		112	
100 cm	Origen	320	38	254	39	190	39
	A 6 cm	282		215		151	
150 cm	Origen	320	26	276	27	232	26
	A 6 cm	294		249		206	

Una vez obtenidos los desplazamientos asociados a estas cuatro distancias, se aplicó análisis de regresión con la librería *scipy.optimize* de Python, para obtener la función que mejor describa esta relación. Se probaron diferentes funciones, como la lineal (ver Figura 9), exponencial (ver Figura 10), y la hiperbólica (ver Figura 11). Finalmente, la función hiperbólica dada en (1), resultó ajustarse bien a los datos.

$$distancia(cm) = 3836{,}15/desplazamiento(px) \qquad (1)$$

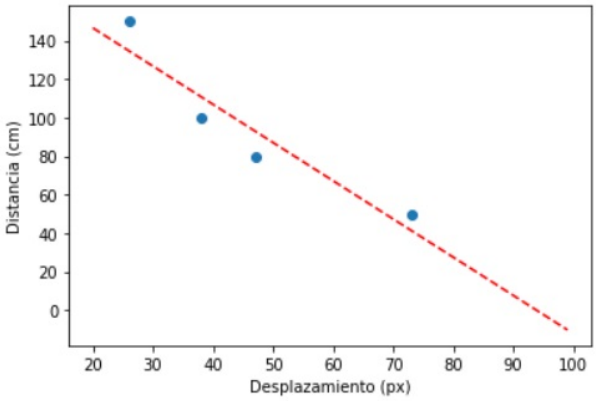

Figura 9: Regresión lineal.

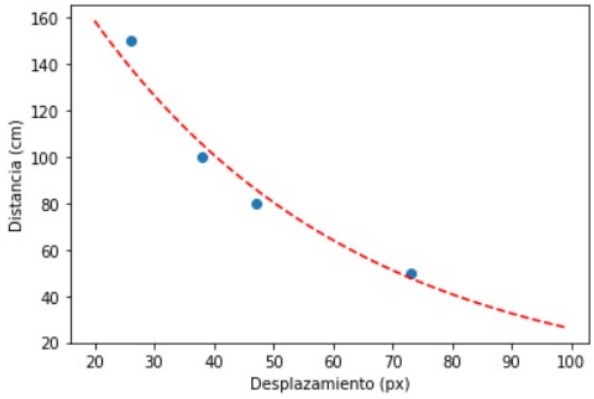

Figura 10: Regresión exponencial.

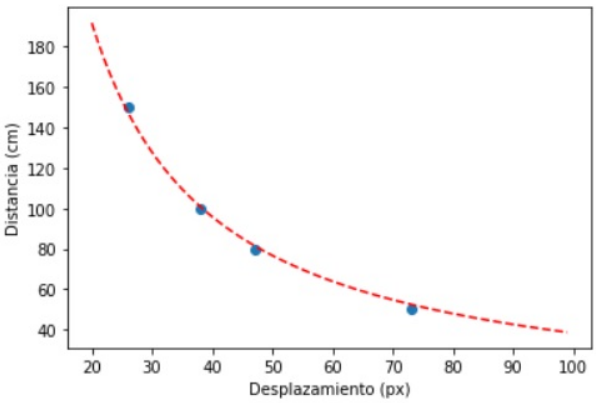

Figura 11: Regresión hiperbólica.

Con la función (1), se puede estimar la distancia relativa de los objetos con respecto al robot, a partir del desplazamiento de estos en las imágenes tomadas desde dos puntos diferentes.

Para esto, una vez segmentadas las imágenes, se obtiene el contorno y el centroide de los objetos en ambas imágenes, utilizando la librería *OpenCV* de Python. En la Figura 12 se muestra un ejemplo de la generación del contorno y centroide para un mismo objeto, en las dos imágenes capturadas.

El valor del desplazamiento se obtiene con el valor absoluto de la diferencia de la componente X de ambos centroides. En el ejemplo de la Figura 12, la componente X del centroide del objeto en la imagen superior es 227, y en la imagen inferior es de 176, resultando un desplazamiento de 51 pixeles.

Aplicando la función (1), la distancia inferida, entre el robot y el objeto, es de 71 cm, lo cual es cercano a la distancia real, que es de 70 cm.

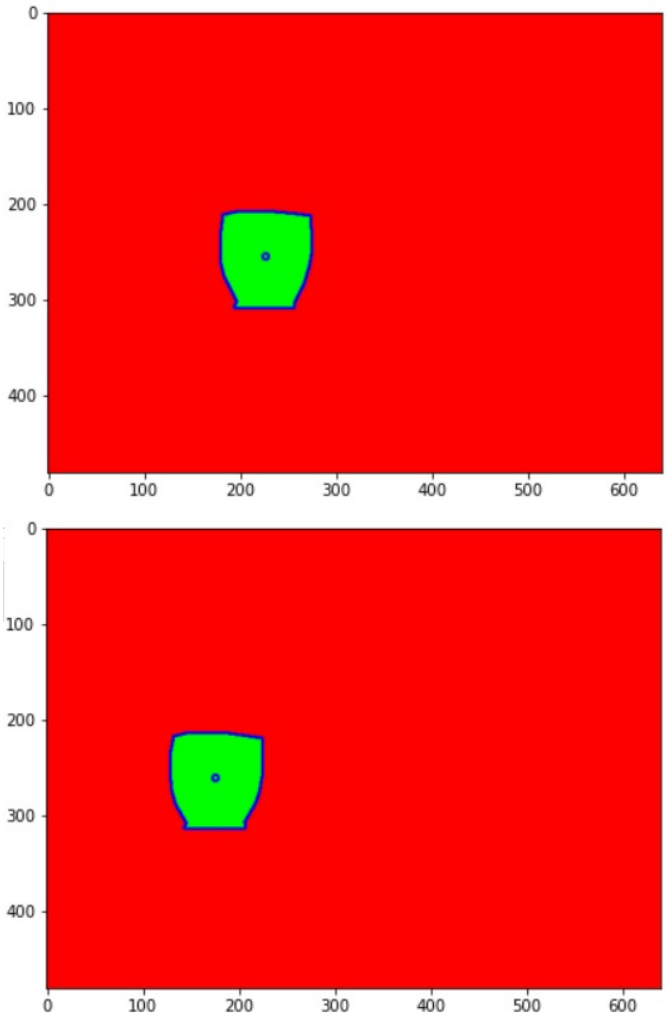

Figura 12: Ejemplo de la generación de contornos y obtención de centroides.

Un problema detectado al aplicar esta estrategia de estimación de distancias, es que si los objetos se encuentran en un extremo de la imagen, parcialmente fuera del campo de visión de la cámara, el cálculo del centroide no es del todo exacto. Por lo tanto, el dato del desplazamiento tampoco lo es.

La propuesta para estos casos, es utilizar las coordenadas del borde del objeto, que se encuentra del lado contrario al extremo por donde sale parcialmente de la imagen. Esto es, si el objeto sale parcialmente por el lado izquierdo de la imagen, se toma la componente X máxima (contorno derecho) del objeto (círculo amarillo en la Figura 13). Si el objeto sale parcialmente por el extremo derecho, se toma la componente X mínima. Los resultados de aplicar esta estrategia, se presentan en la siguiente sección.

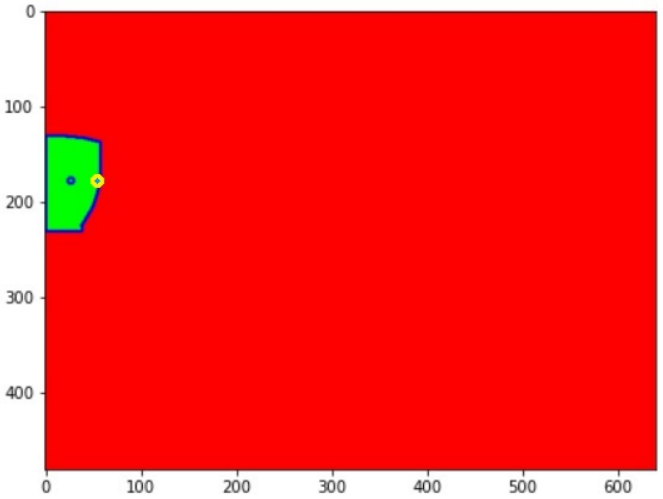

Figura 13: Objeto parcialmente fuera del campo de visión.

4. Experimentos y resultados

4.1. Pruebas de reconocimiento y clasificación

Las pruebas consistieron en aplicar el sistema de reconocimiento y clasificación, sobre 47 imágenes que contienen un total de 127 ocurrencias de los objetos de interés.

En la Figura 14 se muestra un ejemplo de una imagen de prueba, en la Figura 15 se muestra la imagen segmentada manualmente, bajo los mismos criterios con los que se entrenó la red, y en la Figura 16 se muestra la segmentación inferida por el sistema. Se puede observar que la segmentación inferida es similar a la segmentación manual.

En ambas imágenes segmentadas, se muestran en color amarillo los pixeles que se etiquetaron como parte de la mesa, o como equipo electrónico sobre esta. Los colores mostrados en las pruebas no corresponden necesariamente al color utilizado en la preparación de las muestras (la etiqueta asignada sí). En ambas imágenes, se etiquetaron los pixeles del celular en color azul turquesa, y en color violeta los pixeles del fondo, que incluye piso, pared, cortinas y otros objetos no considerados de interés.

En este ejemplo, se detectó como falso positivo la segmentación del brillo en el borde del celular, que se etiquetó como parte de un garrafón de agua. Para corregir este tipo de errores, se propone reentrenar la red con más ejemplos de objetos con brillo.

Figura 14: Imagen de prueba original.

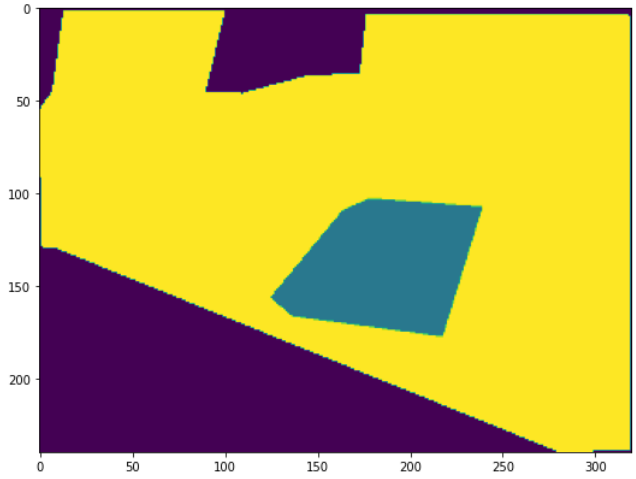

Figura 15: Imagen segmentada manualmente.

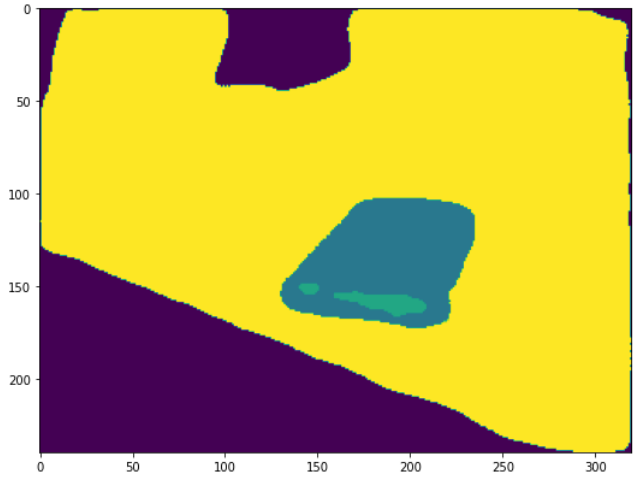

Figura 16: Imagen con segmentación inferida por el sistema.

En la Tabla 7 se muestran los resultados de las pruebas. El número de objetos reconocidos y clasificados con la clase correcta es de 116 (positivos correctos), la cantidad de objetos identificados con una clase no correcta es de 12 (falsos positivos), y el número de objetos presentes en las imágenes, no identificados por el sistema, es de 11 (falsos negativos).

Tabla 7: Pruebas de reconocimiento y clasificación.

Positivos correctos	116
Falsos positivos	12
Falsos negativos	11
Precisión	90.6 %
Sensibilidad	91.3 %

4.2. *Pruebas de estimación de posiciones relativas*

Para estas pruebas, se colocaron objetos en diferentes posiciones con distancias conocidas, y se aplicó la estrategia propuesta. Por cada posición de los objetos, se capturaron dos imágenes, la primera imagen se tomó con la cámara colocada en una posición considerada como base. La segunda imagen se tomó con la cámara colocada a 6 cm de la posición base.

Una vez obtenidas las imágenes, se generó el contorno y centroide de los objetos de prueba, se calculo el desplazamiento de los centroides, se estimó la distancia de los objetos respecto a la cámara y se obtuvo el error de la estimación (ver Tabla 8).

En la quinta columna de la Tabla 8, se muestra el error de estimación, al considerar únicamente los centroides para el cálculo del desplazamiento. Se puede observar que el error para las pruebas 8 y 10 es bastante elevado. Esto es debido a que en estos casos, el objeto se sale parcialmente de la imagen, y el centroide estimado no es correcto. En la columna ocho se muestra el error para estas dos pruebas, con el cálculo del desplazamiento ajustado con base al borde izquierdo del objeto. Se puede observar que el error disminuye sensiblemente.

Tabla 8: Pruebas de estimación de la posición relativa de los objetos.

Prueba	Dist. Real	Desplaz. (centr.)	Dist. Estim.	Error (cm)	Desplaz. (borde)	Dist. Estim.	Error (cm)
1	150	25	153	3			
2	150	23	166	16			
3	120	31	123	3			
4	120	31	123	3			
5	100	36	106	6			
6	100	38	101	1			
7	70	51	75	5			
8	70	31	124	54	53	68	2
9	50	72	53	3			
10	30	69	56	25	111	26	4
			media =	12.3			5.06
			desv. =	15.6			4.16

En la Figura 17 se muestra gráficamente la distancia de los objetos inferida por el sistema (componente Y de los puntos naranjas), a partir del desplazamiento de los centroides. Se muestra también la distancia real de los objetos (componente Y de los puntos azules). La componente X de cualquier punto (azul o naranja), corresponde al desplazamiento.

En la Figura 18 se muestran las distancias, una vez que se ajustó el desplazamiento de los objetos que salen parcialmente

de la imagen (pruebas 8 y 10). Para estos objetos, se utilizó el desplazamiento de sus bordes izquierdos, en lugar de sus centroides. Se puede observar que las distancias estimadas se ajustan mejor a las distancias reales.

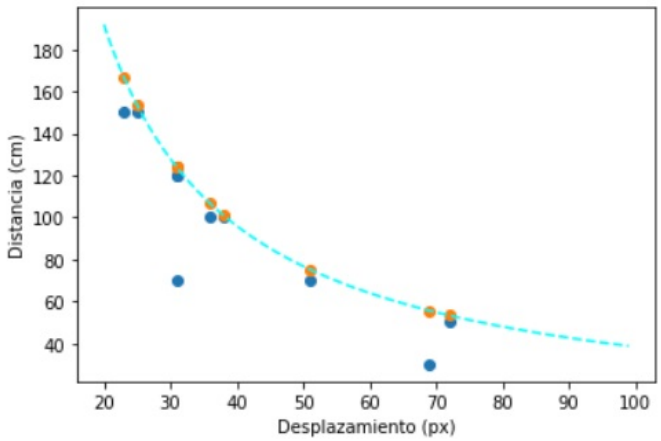

Figura 17: Estimación de las distancias con base a centroides.

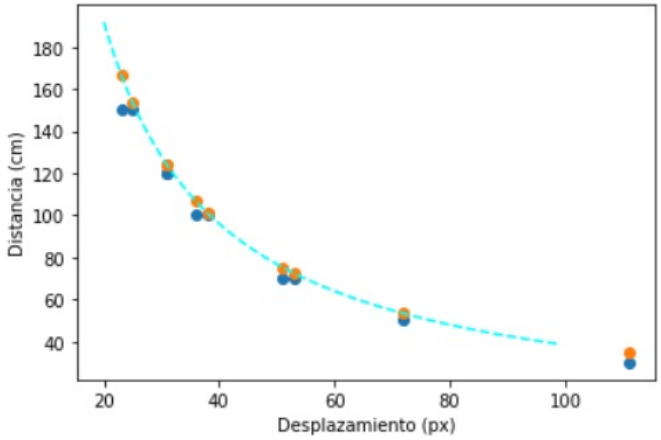

Figura 18: Estimación de las distancias con base a centroides y bordes.

5. Conclusiones

En este trabajo se propone un sistema para el reconocimiento y clasificación de objetos, basado en redes neuronales convolucionales, así como la estimación de la posición relativa de estos con respecto a una cámara montada sobre un robot móvil, aplicando una técnica de visión estereoscópica.

El objetivo de contar con este sistema, es integrarlo, en una siguiente etapa del proyecto, sobre un robot de cuidados y asistencia básica.

Los experimentos realizados arrojaron resultados alentadores, sin embargo, consideramos que es posible mejorar la precisión del reconocimiento de los objetos, a través de un mayor y más variado conjunto de muestras de entrenamientos.

Agradecimientos

Agregue sus agradecimiento hasta la aceptación del manuscrito.

Referencias

Diddeniya, S. I. A. P., Adikari, A. M. S. B., Gunasinghe, H. N., Silva, P. R. S. D., Ganegoda, N. C., and Wanniarachchi, W. K. I. L. (2018). Vision based office assistant robot system for indoor office environment. *3rd International Conference on Information Technology Research (ICITR)*.

Ferreira, L. (2013). Localization and navigation in an autonomous vehicle. *Universidade de Aveiro Departamento de Electrónica, Telecomunicaciones e Informática*.

Jishnu, K., Indu, V., Ananthakrishnan, K., Amith, K., Reddy, P., and Pramod, S. (2020). Voice controlled personal assistant robot for elderly people. *Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020)*, pages 269–274.

Miseikis, J., Caroni, P., Duchamp, P., Gasser, A., R. Marko, N. M., Zwilling, F., Castelbajac, C., Eicher, L., Fruh, M., and Fruh, H. (2020). Lio-a personal robot assistant for human-robot interaction and care applications. *in IEEE Robotics and Automation Letters*, 5:5339–5346.

Purwanto, D., Rivai, M., and Soebhakti, H. (2017). Vision-based multi-point sensing for corridor navigation of autonomous indoor vehicle. *International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 67–70.

Yang, G., Wang, S., and Yang, J. (2019). Desire-driven reasoning for personal care robots. *in IEEE Access*, 7:75203–75212.