Enfoque UTE, V.12-N.4, Oct. 2021, pp. 88-104
http://ingenieria.ute.edu.ec/enfoqueute/
e-ISSN: 1390-6542
CC BY 3.0 EC

Received: 2021/07/30
Accepted: 2021/09/06
Online: 2021/09/10
Published: 2021/10/01
https://doi.org/10.29019/enfoqueute.776

Red neuronal convolucional para la percepción espacial del robot InMoov a través de visión estereoscópica como tecnología de asistencia

(Convolutional Neural Network for Spatial Perception of InMoov Robot Through Stereoscopic Vision as an Assistive Technology)

Juan F. Cortes Zarta[1], *Yesica A. Giraldo Tique*[2], *Carlos F. Vergara Ramírez*[3]

Resumen

En el desarrollo de los robots de asistencia un reto importante consiste en mejorar la percepción espacial de los robots para la identificación de objetos en diversos escenarios. Para ello, es preciso desarrollar herramientas de análisis y procesamiento de datos de visión estereoscópica artificial. Por esta razón, el presente artículo describe un algoritmo de redes neuronales convolucionales (CNN) implementado en una Raspberry Pi 3 ubicada en la cabeza de una réplica del robot humanoide de código abierto InMoov para estimar la posición en X, Y, Z de un objeto dentro de un entorno controlado. Este artículo explica la construcción de la parte superior del robot InMoov, la aplicación de *Transfer Learning* para detectar y segmentar un objeto dentro de un entorno controlado, el desarrollo de la arquitectura CNN y, por último, la asignación y evaluación de parámetros de entrenamiento. Como resultado, se obtuvo un error promedio estimado de 27 mm en la coordenada X, 21 mm en la coordenada Y y 4 mm en la coordenada Z. Estos datos son de gran impacto y necesarios al momento de usar esas coordenadas en un brazo robótico para que alcance el objeto y lo agarre, tema que queda pendiente para un futuro trabajo.

Palabras clave

Robótica humanoide, redes neuronales convolucionales, percepción espacial, aprendizaje de transferencia.

Abstract

In the development of assistive robots, a major challenge is to improve the spatial perception of robots for object identification in various scenarios. For this purpose, it is necessary to develop tools for analysis and processing of artificial stereo vision data. For this reason, this paper describes a convolutional neural network (CNN) algorithm implemented on a Raspberry Pi 3, placed on the head of a replica of the open-source humanoid robot InMoov, to estimate the X, Y, Z position of an object within a controlled environment. This paper explains the construction of the InMoov robot head, the application of Transfer Learning to detect and segment an object within a controlled environment, the development of the CNN architecture, and, finally, the assignment and evaluation of training parameters. As a result, an estimated average error of 27 mm in the X coordinate, 21 mm in the Y coordinate, and 4 mm in the Z coordinate was obtained; data of great impact and necessary when using these coordinates in a robotic arm to reach and grab the object, a topic that remains pending for future work.

Keywords

Humanoid robotic, convolutional neural networks, spatial perception, transfer learning.

1 Escuela Tecnológica Instituto Técnico Central. Bogotá, Colombia. [jfcortesz@itc.edu.co, https://orcid.org/0000-0002-0795-4582]

2 Escuela Tecnológica Instituto Técnico Central. Bogotá, Colombia. [yagiraldot@itc.edu.co, https://orcid.org/0000-0003-1616-1709]

3 Escuela Tecnológica Instituto Técnico Central. Bogotá, Colombia. [cfvergarar@itc.edu.co, https://orcid.org/0000-0003-3843-1255]

1. Introducción

El principal propósito de las tecnologías de asistencia es poder ayudar a mejorar la calidad de vida de personas que sufren alguna discapacidad, disminuyendo de esta manera la dependencia que sienten debido a su incapacidad motriz, la cual los lleva a percibir que son controlados por otras personas (Kerstens et al., 2020).

De acuerdo con estimaciones del Ministerio de Salud y Protección Social de Colombia (2019) para el 2019 existían 1 496 213 personas con alguna discapacidad, de los cuales el 37 % sufre de alteraciones o limitaciones permanentes para usar su cuerpo, manos, brazos o piernas. Además, según los mismos estudios, el 80 % de las personas con discapacidad refirieron pertenecer a los estratos socioeconómicos uno y dos y, precisamente, su discapacidad física es una limitante adicional para tener un empleo formal y estable. Por lo tanto, no solo existe una demanda por tecnologías recientes o específicas para ayudar a personas discapacitadas, sino que el costo de tales soluciones tecnológicas debe garantizar su accesibilidad (Demby's et al., 2019).

La robótica ha estado involucrada en diferentes aspectos de la tecnología de asistencia como la movilidad personal, funciones sensoriales y de monitoreo (Hassan, Abou-Loukh y Ibraheem, 2020), asistencia dentro de entornos controlados (González et al., 2008), entre otros. Tradicionalmente, los robots se han enfocado en funciones de automatización preprogramadas y en áreas de trabajo controladas (Xu y Wang, 2021). Sin embargo, en los últimos años la robótica ha avanzado constantemente, creando de esta manera robots más robustos con la capacidad de realizar acciones o tareas con una complejidad progresivamente mayor. También se ha estado implementado algoritmos de inteligencia artificial (IA) para otorgarle a los robots autonomía y adaptabilidad, las cuales son características fundamentales para su eventual implementación como tecnología de asistencia (Li, et al., 2021).

Uno de los principales problemas de la autonomía en los robots es lograr identificar correctamente su entorno y los objetos que lo rodean, ya sea para evitar colisiones, trazar trayectorias o simplemente para detectar objetos (Jardón, et al., 2008; Huang, et al., 2020; Miseikis et al., 2018). La técnica de visión computacional por triangulación matemática ha sido bastante utilizada para solventar el problema de la autonomía. Sin embargo, esta técnica es susceptible a cambios de los parámetros intrínsecos y extrínsecos de las cámaras, por lo que no se adapta muy bien a entornos dinámicos y, a la vez, esta técnica necesita un modelo matemático a base de reglas para cumplir su objetivo que, en este caso, es la estimación de las coordenadas de un objeto (O'Mahony et al., 2020). Por otra parte, una red neuronal convolucional (CNN) es un tipo de red artificial que intenta emular el ojo humano y la interpretación de imágenes que realiza el cerebro (Ghosh, et al., 2020). Por tal motivo, esta técnica soluciona problemas relacionados a imágenes de forma natural, utilizando principios de aprendizaje de neuronas orgánicas (O'Mahony et al., 2020; Smola y Vishwanathan, 2008), por lo que su uso, aplicación y resultados tienden a ser más orientados a la interpretación nativa de las imágenes que realizan los seres humanos.

Estos campos están permitiendo a los robots adoptar la capacidad con la que los humanos y gran parte de los animales interactúan con su entorno, mejorando y potencializando la teoría tradicional del control. Tal que se replican mecanismos complejos de la biología como, por ejemplo, la visión estereoscópica, la cual es una forma biológica utilizada para identificar profundidad y relieve en una imagen (Valencia et al., 2016).

El aporte de esta investigación es la utilización de CNN para visión estereoscópica en un robot como InMoov (Langevin, 2012), el cual tiene cualidades físicas y funcionales equivalentes

al del ser humano y que puede ser utilizado para discernir la profundidad y localización de un objeto dentro de una escena. De manera que esta funcionalidad es el cimiento de fases posteriores en esta investigación que le permitirán al robot InMoov localizar y alcanzar un objeto que necesite una persona con discapacidad motriz que se encuentre dentro de un entorno doméstico.

2. Metodología

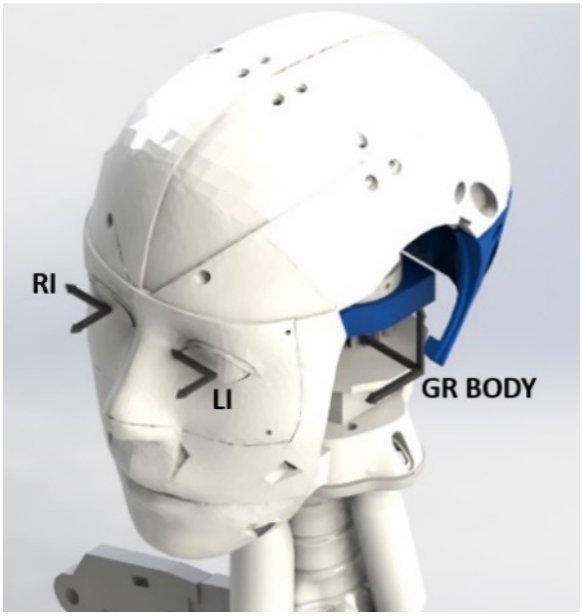
Con el objetivo de permitirle a la réplica del robot humanoide InMoov aprender la percepción espacial de un entorno doméstico controlado, se desarrolló una arquitectura de redes neuronales convolucionales (CNN) para encontrar el objeto, segmentarlo y estimar sus coordenadas cartesianas relativas (X, Y, Z) con respecto al robot, utilizando como punto de partida las metodologías planteadas por Leitner, et al. (2013) y Demby's, et al. (2019).

2.1. Robot humanoide InMoov

El algoritmo de percepción espacial se aplicó para darle esta funcionalidad a una réplica del proyecto del robot humanoide de código abierto (*open source*) InMoov. Las piezas se fabricaron a través de impresión 3D por FDM (modelado por deposición fundida) a partir de los modelos 3D que se encuentran disponibles en la página oficial del proyecto InMoov (Langevin, 2012).

En la Figura 1 se esquematizan los sistemas de referencia del robot, de los cuales dos corresponden a las cámaras del robot identificadas como LI '*Left Image*' y RI '*Right Image*'. Para la cabeza se determina también el sistema coordenado GR Body, el cual se usa como referencia para la posición del objeto con respecto al robot.

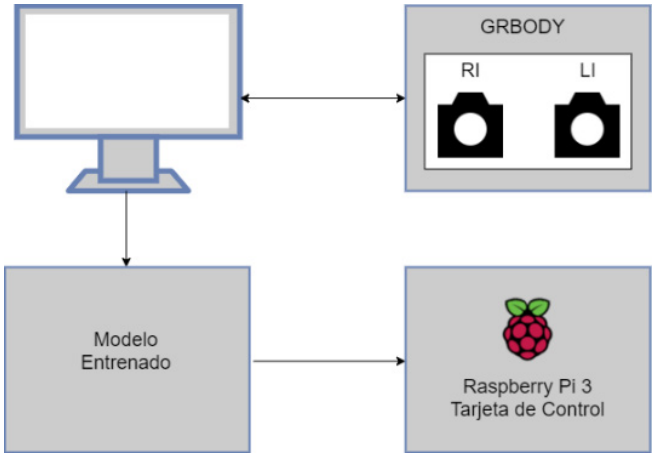
Figura 1. Sistema de referencia de la cabeza de InMoov



La captura inicial de las imágenes se realizó con las cámaras ubicadas en la cabeza del robot, las cuales se enviaron a un computador personal por medio de USB, tal como se describe en la Figura 2.

Aunque el robot tiene la capacidad de cambiar la orientación de las cámaras, así como girar su cabeza, para este artículo se decidió mantener en una posición y orientación fija tanto las cámaras como la cabeza. Esto con el fin de evaluar, primero, el desempeño de la red convolucional bajo estas condiciones para, posteriormente, analizar cuando la orientación de la imagen cambia por movimientos de la cabeza y de las cámaras. La definición de los sistemas de referencias está basada en el trabajo realizado por Valencia, et al. (2016), lo cual se explica en detalle en la siguiente sección.

Figura 2. Esquema de control de la cabeza de InMoov



2.2. Visión estereoscópica

La visión estereoscópica es una herramienta utilizada en la robótica humanoide que tiene como fin darle independencia a un sistema robótico a través de visión computacional por medio de dos cámaras desplazadas. Estas se encargan de tomar la captura de dos imágenes, generar una triangulación geométrica y, finalmente, estimar la ubicación de los objetos.

Esta técnica emplea dos cámaras ubicadas en cada uno de los ojos del robot InMoov. En la Figura 3 se muestra el esquema geométrico proyectado desde la vista superior para la captura y procesamiento de las imágenes de las cámaras. Esta figura contiene los parámetros fijos del sistema tales como: separación entre cámaras LI y RI de 62.27 mm, rangos del campo de visión de las cámaras LI y RI en el eje X los cuales son de 70°, restricciones determinadas por la distancia F al punto crítico de oclusión O de 107.167 mm, el cual está encargado de definir el área limitante para el rastreo de objetos LI-O-RI y, finalmente, los ángulos θ = 73.8° y α = 36.2, los cuales se forman por la geometría generada con la posición de las cámaras, el campo de visión y F. Por último, P es la distancia desde el punto de oclusión crítico y el objeto de interés (Valencia, et al. 2016).

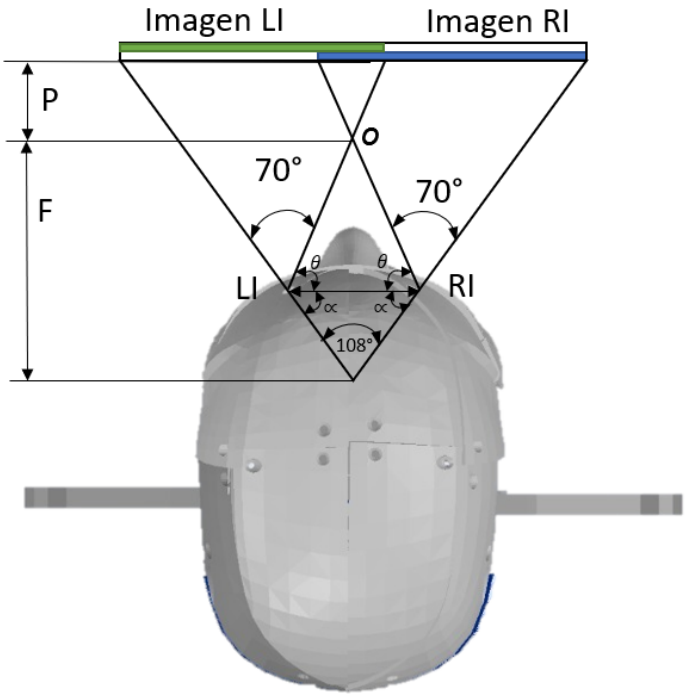
Estos datos fueron relevantes para el entorno controlado de trabajo, con el fin de deducir el área no rastreable de un objeto formado por el triángulo LI-O-RI y de esa manera definir las dimensiones del entorno de trabajo.

2.3. Redes neuronales convolucionales

La red neuronal convolucional o CNN por sus siglas en inglés (*Convolutional Neural Network)* es un tipo de red neuronal artificial que tiene la capacidad de aprender características, patrones o similitudes abstractas de objetos, escenas o cualquier información que esté en los datos de entrada (Qin, et al., 2018), los cuales, por lo general son, imágenes. Así que este tipo de red neuronal está muy relacionada con la visión por computadora (Wozniak, et al., 2018; Li et al., 2019).

Una CNN consta de un conjunto finito de capas de convolución que se encargan de extraer características de las imágenes. Tal que las primeras capas de convolución tendrán un nivel menor de abstracción, el cual irá aumentando con relación a la cantidad de capas que existan (Ghosh, et al., 2020). Por ejemplo, una CNN, para detectar un vaso, en las primeras capas aprenderá a detectar características simples como líneas, bordes, colores, entre otros y, en las últimas capas, detectará características o patrones complejos como siluetas, texturas, entre otros.

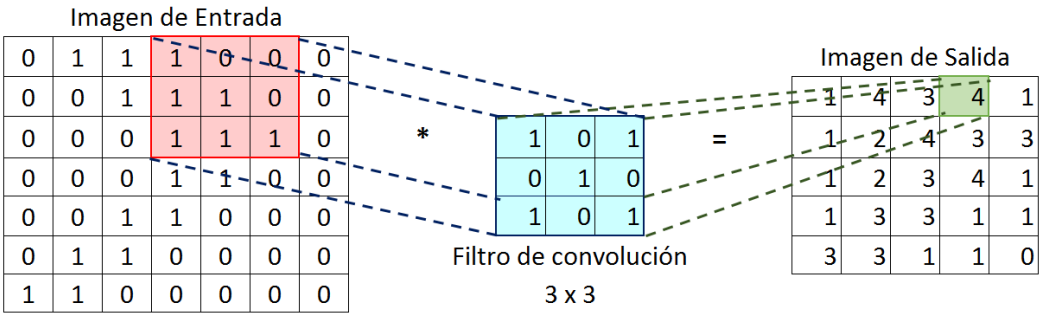
Figura 3. Sistema de referencia de los ojos



Posteriormente, para implementar la técnica por el método de visión estereoscópica es necesario realizar la captura y el acondicionamiento de las imágenes del objeto detectado.

La extracción de información por cada capa de convolución se hace a través de filtros de un tamaño inferior al tamaño de la imagen. Sin embargo, los tamaños más usados son 3x3, 5x5 y 7x7 (Chansong y Supratid, 2021). Estos filtros recorren toda la imagen de izquierda a derecha y de arriba hacia abajo, realizando un producto escalar con la matriz de píxeles de la imagen, con el fin generar una nueva imagen convolucionada. La Figura 4 es un ejemplo del funcionamiento del filtro de convolución 3x3 en una matriz de 7x7.

Figura 4. Ejemplo del proceso de convolución



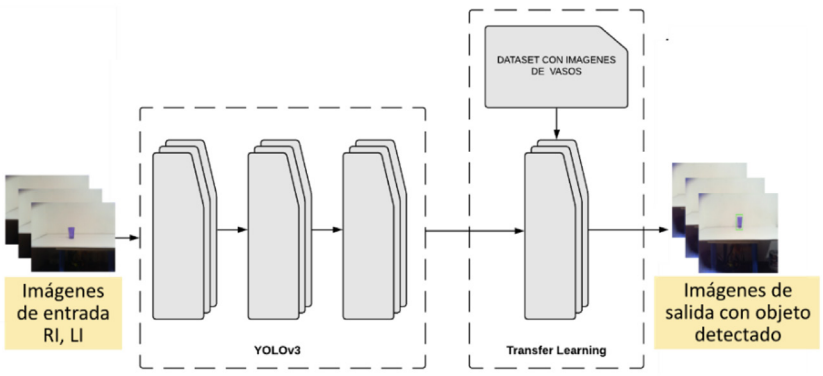
La imagen de salida luego pasará por otro filtro llamado aplanamiento o *pooling*, el cual se encarga de disminuir el tamaño espacial de la imagen con el fin de reducir la potencia computacional requerida para procesar las matrices multidimensionales.

2.4. Transfer Learning

Como estrategia para estimar la posición del objeto, se utilizó *Transfer Learning*, el cual consiste en tomar un modelo pre entrenado y adaptarlo con nuevos datos. El modelo empleado es YoloV3 (Redmon y Farhadi, 2018), el cual es una CNN que detecta y segmenta 80 clases de objetos distintos en una caja. Si bien, este modelo es suficientemente robusto se escogió como clase única a detectar 'vaso', debido a que este objeto reúne características geométricas simples para que la CNN de YoloV3 pueda aprender de manera más sencilla y, también, porque este objeto es bastante utilizado por las personas en los entornos domésticos. Así que el robot solo podrá detectar ese objeto de interés dentro la escena. Esto, se realizó de esta manera para que la CNN de percepción espacial logre identificar correctamente el objeto y sus coordenadas. De manera que, la implementación de YoloV3 es el cimiento para que el robot logré discernir el objeto de interés cuando esté rodeado por otros objetos.

Por lo tanto, en la última capa de YoloV3 se anexa un Dataset con 1 000 imágenes diferentes de vasos y etiquetados con la herramienta computacional LabelImg (Tzutalin, 2015), con el fin de entrenar esa última capa. Finalmente, se pone a correr el modelo para observar cómo detecta el objeto. La Figura 5 muestra la adaptación de YoloV3 con las imágenes extraídas del robot InMoov y, así mismo, el resultado luego de utilizar *Transfer Learning*.

Figura 5. Arquitectura de algoritmo YoloV3 adaptada por Transfer Learning



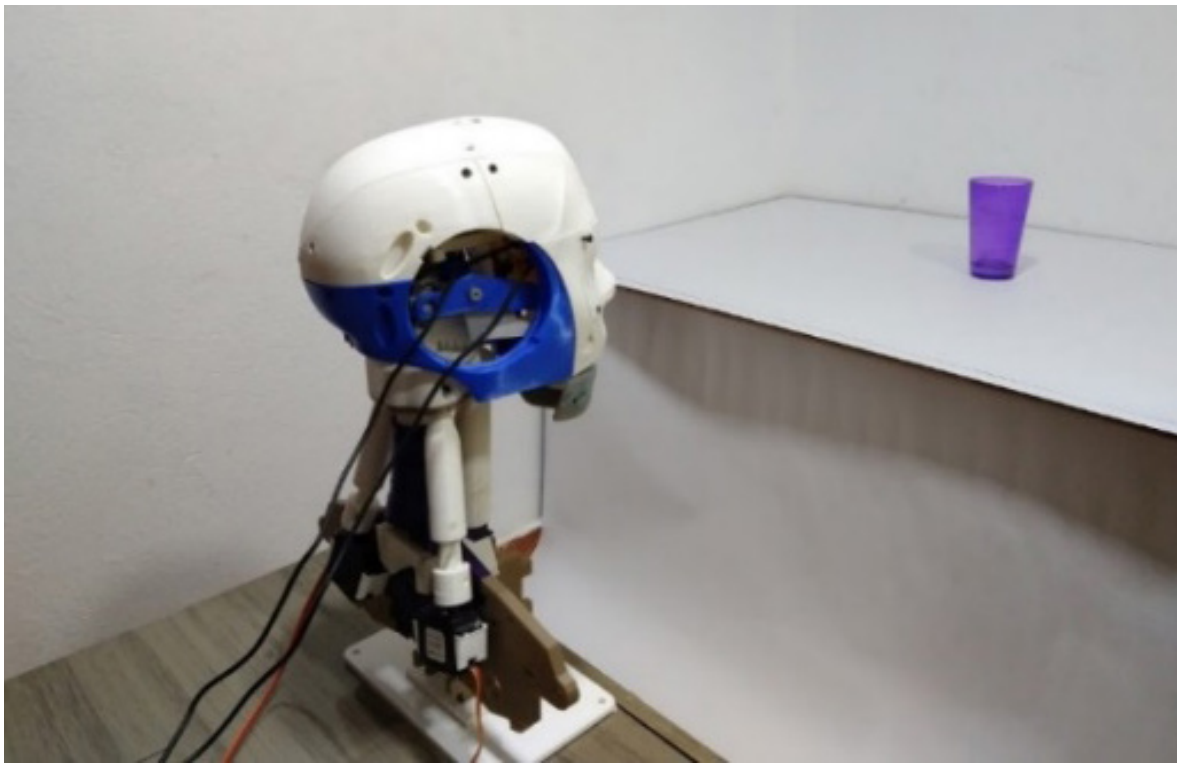
Con las imágenes capturadas por el método de *Transfer Learning* se construye el conjunto de datos para la CNN de percepción espacial, la cual se explica a continuación junto con los parámetros y condiciones iniciales que se implementaron.

2.5. Conjunto de datos

Para facilitar el entrenamiento de una CNN se plantea adquirir los datos de imágenes estereoscópicas de un entorno controlado, lo cual facilita la adaptación del conjunto de datos a la arquitectura planteada en una fase inicial y después pretender su escalabilidad a entornos con perturbaciones.

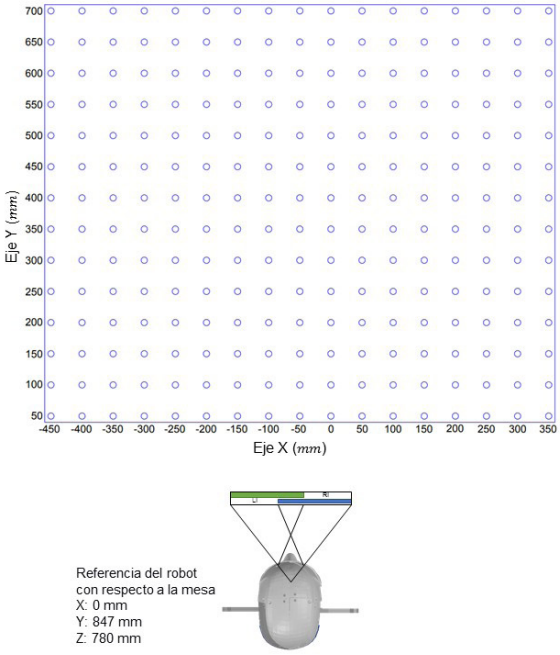
El entorno se realizó en una mesa con una cuadricula de puntos separados cada 50 mm a una distancia frente al robot de 847mm, tal como se ve en la Figura 6. Estos puntos sobre la mesa representan la posición que tomará el objeto dentro del escenario (Demby's, et al., 2019). Aparte de esto, la luminosidad dentro del entorno es constante.

Figura 6. Entorno controlado de recopilación de datos



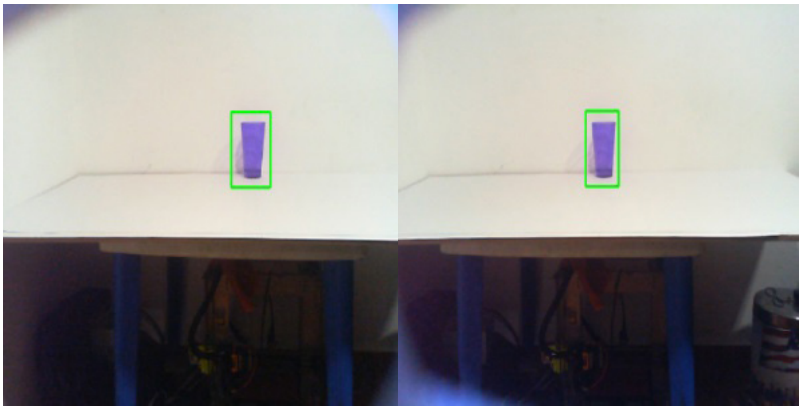
La altura Z del robot es fija y la posición del objeto sobre la mesa en el eje X y Y varía en los puntos designados de la cuadrícula, por lo que la variación en Z del escenario se realizó elevando la mesa 10 mm en 7 ocasiones cada que el objeto recorriera todos los puntos de la cuadricula. La distribución de coordenadas sobre la mesa y la posición del robot con respecto a la misma se evidencian en la Figura 7. Para hallar la coordenada en X en el sistema de Coordenadas GR Body no es necesario hacer alguna transformación en las medidas de la cuadricula, mientras que para los valores del eje Y se les suma una distancia de 840mm y en el eje Z se suma una distancia de 780mm. Estos valores representan la distancia del escenario con respecto a GR Body.

Figura 7. Distribución de coordenadas sobre la mesa de entrenamiento



El método para recopilar el conjunto de entrenamiento de las imágenes de ambos ojos se realizó moviendo manualmente el objeto en la escena y almacenando LI y RI con las coordenadas actuales del objeto en el sistema de referencia GR Body como, por ejemplo, [$x\_y\_z.png$]. Esta manera de nombrar las imágenes de entrenamiento fue extraída del conjunto de datos UTKFace (Zhang, Song y Qi, 2017) que tiene la utilidad de simplificar el procesamiento de las imágenes y, a la vez, correlacionar las etiquetas con las imágenes de una manera sencilla. Por otro lado, las imágenes se recopilaron con una dimensión de ($416x416x3$) pixeles, lo cual significa la anchura, altura y cantidad de canales RGB, correspondientemente. En la Figura 8 se presenta un ejemplo de LI y RI.

Figura 8. Imagen captada por ambos ojos izquierdo y derecho respectivamente



De esta forma, se obtuvieron dos carpetas o directorios con 1 547 imágenes en cada uno. Ambas carpetas tienen en común las coordenadas del objeto en las 1 547 posiciones, pero se

diferencian en la distribución de pixeles intrínsecos de las imágenes. Esto quiere decir que hay dos imágenes de entrada (izquierda y derecha) y 3 datos de salida (X, Y, Z). Así que la arquitectura que se planteará en las próximas secciones es de tipo MIMO (*Multiple Input-Multiple Output*).

La Tabla 1 es una muestra de la cabecera del Dataframe, en él se encuentran las entradas y salidas del sistema indexado y estructurado. Por otra parte, el Dataframe se dividió al 70 % para el entrenamiento y el 30 % restante para validación y prueba del modelo. Esto, con el fin de verificar que los resultados obtenidos a través del aprendizaje no estén sobre ajustados.

Tabla 1. Dataframe del conjunto de entrenamiento

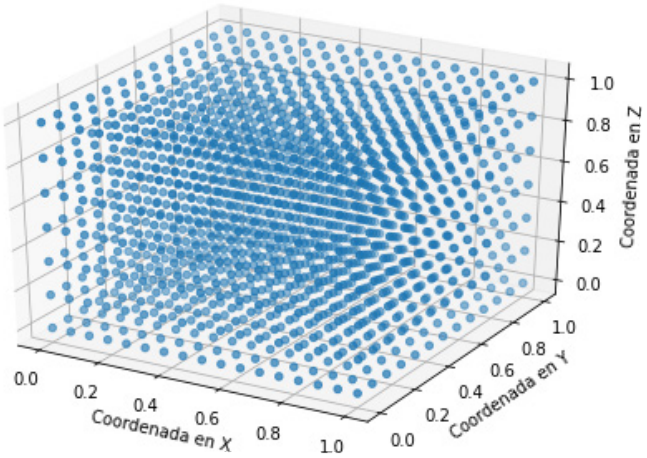
	X	Y	Z	Carpeta cámara izquierda	Carpeta cámara derecha
0	-300	1 197	800	CamIzq/-300_1197_800.png	CamDer/-300_1197_800.png
1	250	997	790	CamIzq/250_997_790.png	CamDer/250_997_790.png
2	50	1 047	780	CamIzq/50_1047_780.png	CamDer/50_1047_780.png
3	-150	897	830	CamIzq/-150_897_830.png	CamDer/-150_897_830.png
4	350	1 147	810	CamIzq/350_1147_810.png	CamDer/350_1147_810.png

Tanto para el preprocesamiento en las imágenes como en las coordenadas de la posición del vaso se hizo una normalización al rango de 0 hasta 1, cambiando los valores de cada característica para que el valor mínimo sea 0 y luego dividiendo por el valor máximo (Geron, 2019b; Smola y Vishwanathan, 2008).

$$z = \frac{z\text{-}min\ (x)}{[max(x)\text{-}min\ (x)]} \tag{1}$$

La Figura 9 es una representación tridimensional del conjunto de entrenamiento normalizado. Por otro lado, no se utilizó el aumento de datos enfocado en las imágenes (Lee y Saitoh, 2018) para no modificar su estructura interna y la organización de pixeles que contienen cada una de las imágenes, ya que al modificarlo el valor recopilado de la coordenada en dicha imagen no tendría coherencia con el valor real de la coordenada.

Figura 9. Dataset de las coordenadas de entrenamiento normalizado

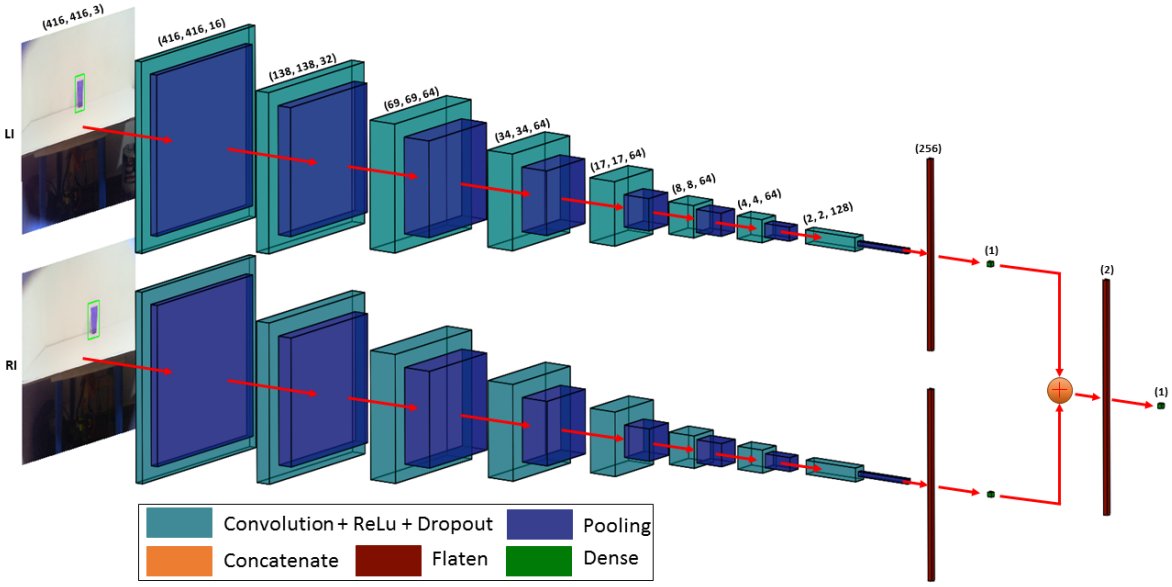


2.6. Arquitectura CNN de percepción espacial

La arquitectura CNN de percepción espacial utilizada en esta investigación se basa en UTKFace (Zhang, Song y Qi, 2017), la cual tiene como característica principal distinguir entre 3 etiquetas (edad, genero, raza). Y en la arquitectura propuesta por Radu Enucă (2019), la cual recibe dos imágenes de espectrogramas (corriente y voltaje) para predecir entre 11 etiquetas útiles. Por lo tanto, se extrajeron las particularidades de estas dos arquitecturas y se interpolaron al enfoque de esta investigación.

La Figura 10 es una representación simplificada de la arquitectura diseñada, en la cual se ilustra únicamente una sola coordenada para fines demostrativos. La arquitectura es una red neuronal convolucional supervisada, tal que consta de dos canales principales correspondientes a LI y RI, los cuales son las imágenes de entrada. Cada canal se divide en tres ramas (X, Y, Z) que son las etiquetas o datos de entrenamiento. De esta manera, la CNN aprende características, similitudes y patrones únicos de cada imagen y coordenada por separado con el fin de que, en capas posteriores, se concatene la información que ha aprendido de cada coordenada por separado con su igual de la otra imagen. Esto quiere decir que toda la arquitectura es una sola CNN, por lo que el entrenamiento de cada rama en cada canal se hace de manera paralela. De igual forma, el método que se usó para calcular el gradiente de la función de error con respecto a todos los pesos de la red fue Backpropagation (Lillicrap, et al., 2020).

Figura 10. Arquitectura propuesta de dos canales, una rama. La salida es una sola coordenada



Las convoluciones de la capa oculta usan un filtro estándar de tamaño 3x3 que recorre toda la imagen y la minimiza a lo alto y a lo largo, pero aumenta su profundidad. Mientras que el filtro que usa el *pooling* para agrupar y reducir el tamaño de la imagen es de 2x2.

Luego de cada convolución se utiliza la técnica de regularización *Dropout* para disminuir el sobreajuste de la CNN una vez que entrene (Poernomo y Kang, 2018), omitiendo aleatoriamente por cada capa oculta el 25 % de las neuronas. Por último, cada neurona de esta capa

utiliza una función de activación tipo ReLu, la cual anula los valores negativos y deja intacto los valores positivos de cada neurona.

$$R_{(z)} = \max(0,\ z) \tag{2}$$

La activación que se utiliza en las capas densas es de tipo lineal con el fin de obtener un resultado numérico análogo a la salida, por lo tanto, esto indica que el modelo planteado es de regresión lineal. La métrica usada para la medición del error en el entrenamiento y validación de la CNN fue MAE (*Mean Absolute Error*). La Ecuación 3 es la métrica que halla el promedio de la diferencia absoluta entre el valor esperado y el valor predicho por el modelo. La ventaja de utilizar este tipo de métrica en comparación con MSE (*Mean Square Error*) es que mejora la precisión del modelo en términos del error promedio, siempre y cuando los datos de entrada no presenten mucho ruido (Qi, et al., 2020).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i\text{-}\hat{y}_i| \tag{3}$$

2.7. Experimentos

Los valores de los parámetros utilizados en el entrenamiento se asignaron de manera heurística con un total de 23 ensayos al entrenar la red neuronal. La cantidad de neuronas asignadas a las capas de la CNN de percepción espacial 16_{conv1}, 32_{conv2}, 64_{conv3}, 64_{conv4}, 64_{conv5}, 64_{conv6}, 64_{conv7}, 128_{conv8}.

Mientras que la tasa de aprendizaje o LR por sus siglas en inglés (*Learning Rate*) puede tomar el rango de valores presentados en la Ecuación 5 para que converja el algoritmo del descenso del gradiente a un mínimo global en cada una de las coordenadas.

$$9.3e^{\text{-}4} > lr < 1e^{\text{-}5} \tag{4}$$

Sin embargo, el valor que arrojaron los resultados expuestos en la siguiente sección fue $lr = 9.6e^{\text{-}4}$. Por otro lado, se utilizó el descenso del gradiente por mini-lote a 32 (Khirirat, Feyzmahdavian y Johansson, 2017), lo cual significa que se tomarán esas cantidades de muestras por cada iteración de entrenamiento. Por último, todo el entrenamiento se hizo con un total de 100 épocas.

2.8. Plataforma experimental

La plataforma experimental desarrollada consta de una cabeza robótica de InMoov junto con dos cámaras internas de referencia Microsoft LifeCam HD- 3000 con una resolución máxima para la captura de fotos de 1 280 x 720 píxeles y una distancia focal de 140 mm. Estas cámaras se encuentran ubicadas en los ojos del robot tal como se describe en la Figura 1 y Figura 3.

Además, tanto la programación y el entrenamiento de la CNN de percepción espacial se realizó usando el servicio en la nube Google Colaboratory con los *frameworks* de Tensorflow y Keras para Python. El servidor remoto para el entrenamiento fue una GPU Nvidia Tesla P100 PCIe, la cual tiene 16 GB de VRAM. Mientras que el modelo entrenado se migró para correrlo, fi-

nalmente, en un Raspberry Pi 3 de 1Gb de RAM, usando Tensorflow Lite. El video de las cámaras se transfiere a la red neuronal usando OpenCV.

3. Resultados

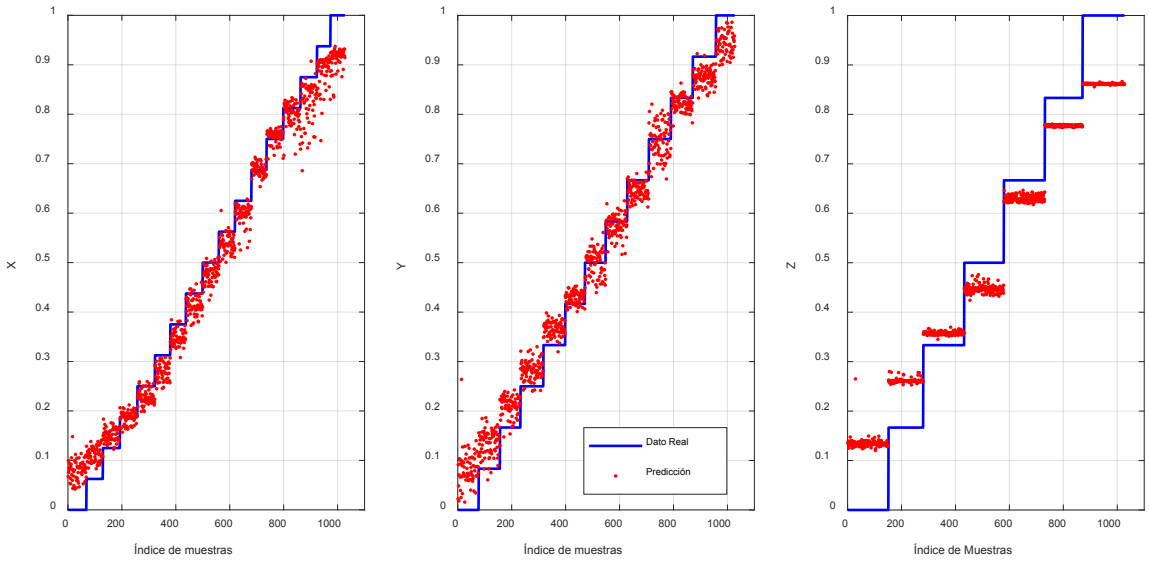
Como primer resultado del modelo propuesto se observa en la Figura 11 el comportamiento de la CNN para estimar la posición en los ejes (X, Y, Z) de acuerdo con las posiciones reales y con un total de 1 000 muestras. Además, el eje de las ordenadas de las tres gráficas está normalizado, tomando como referencia los valores del Dataset tridimensional de la Figura 9 y la Ecuación 3 para mantener la misma relación y coherencia entre los valores reales.

En general, se puede observar que la CNN realiza una buena predicción en cada una de las posiciones reales de los objetos. El error promedio de cada coordenada se puede evidenciar en la Tabla 2 que en la que se presentan los datos experimentales.

Tabla 2. Media de la predicción de la CNN

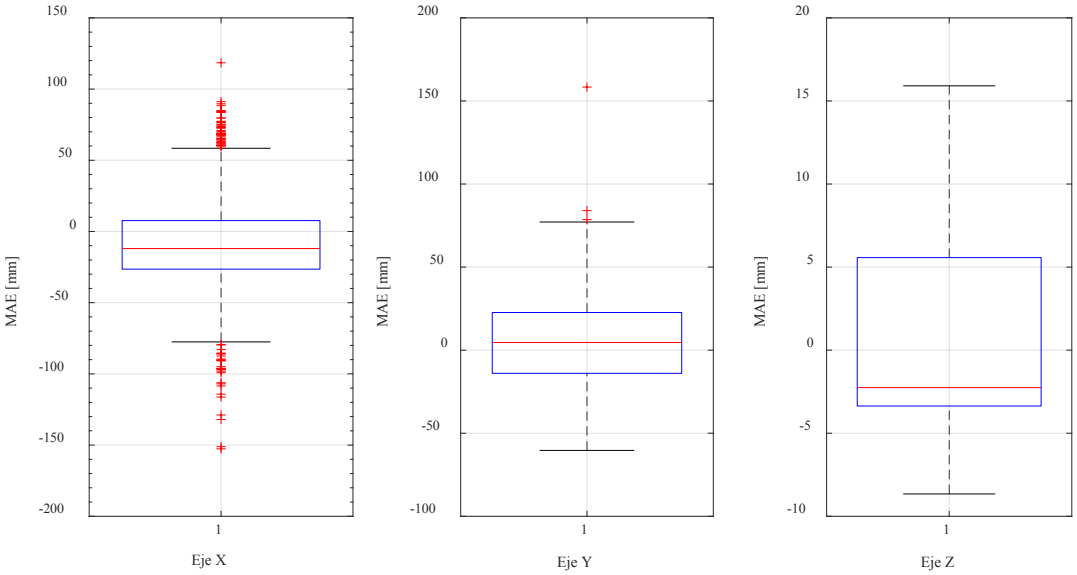
Eje	Error promedio [mm]
X	27.5229
Y	21.26912
Z	4.6170

Figura 11. Comportamiento de la predicción de la arquitectura CNN en cada coordenada normalizada en el eje de las ordenadas



Sin embargo, se presentaron valores atípicos que se evidencian de mejor manera en la Figura 12, la cual es un diagrama de caja. Este indica para el caso de la coordenada X valores atípicos en el límite inferior y superior del diagrama, así como en la coordenada Y donde los valores atípicos rondan solo en el límite superior (Marmolejo-Ramos y Siva Tian, 2010).

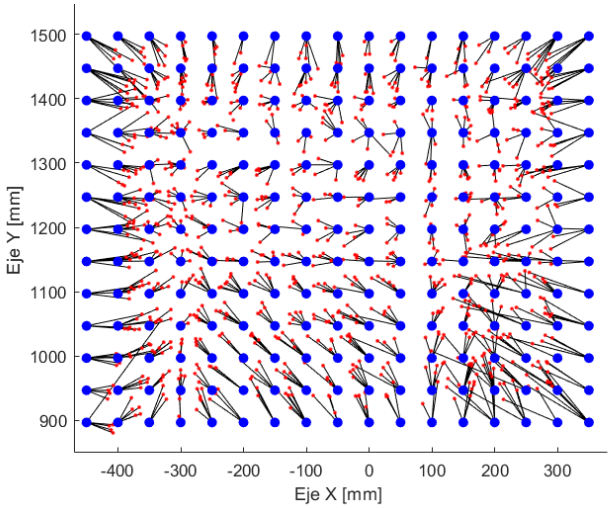
Figura 12. Diagrama de caja de los datos predichos

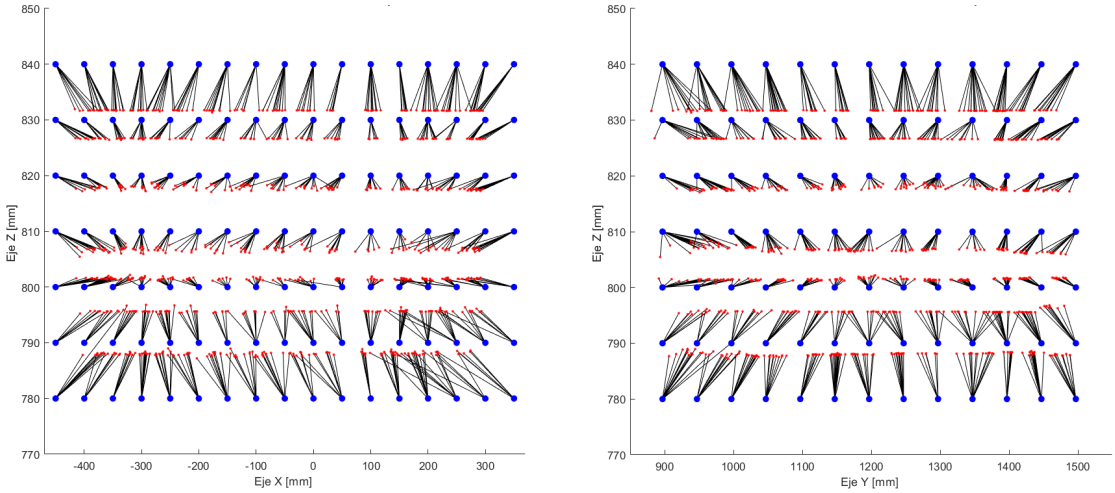


La CNN con la arquitectura propuesta logró aprender la visión periférica de la visión estereoscópica. Esta, tiene la particularidad de que vuelve progresivamente la imagen más borrosa en la periferia de la vista, mientras que mantiene en el centro de la mirada la imagen nítida (Pramod, Katti y Arun, 2018). Por lo tanto, este tipo de visión está relacionada con el reconocimiento y localización de objetos a los que se fija la atención.

La visión periférica que se presenta a raíz de los datos predichos se puede evidenciar de mejor manera en la Figura 13. En esta, se encuentran los valores de las posiciones reales tanto del conjunto de entrenamiento como de validación y los valores de las posiciones estimadas por la CNN relacionadas junto al vector de error encargado de dimensionar estas dos posiciones gráficamente.

Figura 13. Posiciones reales vs posiciones estimadas del vector de error en las coordenadas X vs Y, X vs Z y Y vs Z, respectivamente





En la Figura 13 se puede evidenciar que en el centro del escenario de entrenamiento los datos estimados son más precisos con respecto a las posiciones reales. Mientras que en las orillas de este escenario los vectores de error tienden a aumentar por la separación entre la posición real con respecto al valor real. Esta variación del error son los valores que se presentan como atípicos en la Figura 12.

Esta visión periférica que desarrolló el algoritmo se debe a que el robot siempre mantuvo una posición estática, por lo tanto, el punto de enfoque que se evidencia en la Figura 3 siempre apuntó al centro del escenario del entorno controlado.

4. Discusión

En investigaciones similares de percepción espacial, como los de Demby's, et al. (2019) y Leitner, et al. (2013) se plantean métodos de obtención de datos y arquitecturas diferentes al de esta investigación. La Tabla 3 compara los resultados obtenidos en esas investigaciones con el resultado obtenido con la arquitectura de esta investigación.

Tabla 3. Comparativa de resultados de investigaciones similares

	X Predicha	Y Predicha	Z Predicha	Cámara empleada
Investigación 'A' (Demby's,et al., 2019)	1,9	4,7	-	Módulo de cámara RP V2
Investigación 'B' (Leitner, et al., 2013)	15,9	43,1	37,3	Dragon Fly2 DR2-03S2C-EX-CS
Investigación hecha por los autores	27,5	21,26	4,6	Microsoft LifeCam 3000 HD

La investigación 'A' es la que obtiene mejores resultados del valor real con respecto al valor predicho. Sin embargo, ese error tan bajo se debe a que utilizaron pocos datos de entrenamiento y, a la vez, la variación de distancia entre los puntos dentro del escenario era pequeña. Aparte de esto, el enfoque que esta investigación plantea no contempla que la estimación en Z y la resolución máxima de la foto es inferior, al ser de 640 x 480 pixeles en comparación con la

resolución de la cámara de este proyecto (1 280x720 píxeles). Por otro lado, la investigación 'B' sí contempla la predicción de la coordenada Z y, a la vez, la cantidad de datos que utilizaron para el entrenamiento y validación son similares a la cantidad utilizada en esta investigación. Por lo tanto, se puede observar que los resultados del método propuesto son ligeramente mejores a la investigación 'B' en términos del error de predicción, teniendo presente, además, la resolución y la distancia focal similares (misma resolución y distancia focal).

5. Conclusiones y trabajos futuros

En el futuro se está planteando implementar las coordenadas (X, Y, Z) obtenidas a partir de la arquitectura propuesta en una réplica del brazo robótico de InMoov, el cual ya está completamente construido. Esto se planea hacer introduciendo esas coordenadas como valores de entrada en las ecuaciones de cinemática inversa del brazo del robot InMoov. De tal manera, que no solo tenga la capacidad de localizar un objeto utilizando la percepción espacial, sino también poder alcanzarlo y agarrarlo. Esto se realiza con el fin de que, en un futuro próximo, el robot pueda alcanzar objetos en un entorno cotidiano para ayudar a personas con discapacidad. Aparte de esto, se planea recopilar un conjunto de datos más amplio y con entornos o fondos dinámicos para probarlo con la arquitectura propuesta en esta investigación y comprobar su adaptabilidad y escalabilidad.

Pese a que no se tuvo en cuenta el efecto del modelo cinemático de la cabeza del robot InMoov en la percepción espacial, el resultado del algoritmo obtuvo una precisión comparable o mejor a los artículos guía de esta investigación, tal como se puede evidenciar en la Tabla 3. Por lo tanto, la arquitectura propuesta aprendió de manera satisfactoria a localizar objetos dentro del entorno controlado con un error promedio en cada coordenada de 27.5mm en X, 21.26mm en Y y 4.6mm en Z.

Por último, los resultados muestran que la arquitectura que se diseñó se adapta bien a la visión estereoscópica del robot InMoov, a tal punto de aprender la visión periférica según los parámetros de entrada LI y RI. Este resultado se observa de mejor manera en la Figura 13, en la cual existen dos patrones relevantes: el primero es que en las coordenadas más alejadas al centro del escenario de entrenamiento se evidencia un mayor error en estas predicciones; y segundo, los datos que se encuentran al centro del escenario obtuvieron una mejor estimación. Estos dos patrones son características inigualables de la visión periférica natural. Por lo tanto, incrementando las pruebas experimentales en entornos dinámicos y con una mayor variedad de objetos mejorarán la adaptabilidad y autonomía de este algoritmo.

Referencias

Chansong, D., y Supratid, S. (2021). Impacts of kernel size on different resized images in object recognition based on convolutional neural network. *9th International Electrical Engineering Congress (iEECON)*: 448-451. https://doi.org/10.1109/ieecon51072.2021.9440284

Demby's, J., et al., (2019). Object detection and pose estimation using CNN in embedded hardware for assistive technology. *IEEE Symposium Series on Computational Intelligence (SSCI)*. https://doi.org/10.1109/ssci44817.2019.9002767

Enucă, R. (2019) *Dual-input CNN with Keras*. https://medium.datadriveninvestor.com/dual-input-cnn-with-keras-1e6d458cd979

Geron, A. (2019b). *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Ghosh A., et al., (2020) Fundamental Concepts of Convolutional Neural Network. En Balas V., Kumar R., Srivastava R. (eds) Recent Trends and Advances in Artificial Intelligence and Internet of Things. Intelligent Systems Reference Library. Springer. https://doi.org/10.1007/978-3-030-32644-9_36

González, J., et al., (2008). La Silla RobÓTica SENA. Un enfoque basado en la interacción hombre-máquina. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(2): 38-47. https://doi.org/10.1016/s1697-7912(08)70143-2

Hassan, H. F.; Abou-Loukh, S. J., y Ibraheem, I. K. (2020). Teleoperated robotic arm movement using electromyography signal with wearable Myo armband. *Journal of King Saud University-Engineering Sciences*, 32(6): 378-387. https://doi.org/10.1016/j.jksues.2019.05.001

Huang, B., et al., (2020). Improving head pose estimation using two-stage ensembles with top-k regression. *Image and Vision Computing*, 93(103827): 103827. https://doi.org/10.1016/j.imavis.2019.11.005

Jardón, A., et al., (2008). Asibot: Robot portátil de asistencia a discapacitados. Concepto, arquitectura de control y evaluación clínica. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(2): 48-59. https://doi.org/10.1016/s1697-7912(08)70144-4

Kerstens, H., et al. (2020). Stumbling, struggling, and shame due to spasticity: a qualitative study of adult persons with hereditary spastic paraplegia. *Disability and Rehabilitation*, 42(26): 3744-3751. https://doi.org/10.1080/09638288.2019.1610084

Khirirat, S., Feyzmahdavian, H. R., & Johansson, M. (2017). Mini-batch gradient descent: Faster convergence under data sparsity. *IEEE 56th Annual Conference on Decision and Control (CDC)*. https://doi.org/10.1109/cdc.2017.8264077 \

Langevin, G. (2012). *InMoov -open-source 3D printed life-size robot*. https://inmoov.fr

Lee S., y Saitoh T. (2018) Head Pose Estimation Using Convolutional Neural Network. En Kim K., Kim H., Baek N. (eds) *IT Convergence and Security 2017. Lecture Notes in Electrical* Engineering. Springer. https://doi.org/10.1007/978-981-10-6451-7_20

Leitner, J., et al., (2013). Artificial neural networks for spatial perception: Towards visual object localisation in humanoid robots. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. https://doi.org/10.1109/ijcnn.2013.6706819

Li, J., et al., (2021). An integrated approach for robotic Sit-To-Stand assistance: Control framework design and human intention recognition. *Control Engineering Practice*, 107(104680): 104680. https://doi.org/10.1016/j.conengprac.2020.104680

Li, T., et al., (2019). CNN and LSTM based facial expression analysis model for a humanoid robot. *IEEE access: practical innovations, open solutions*, 7: 93998-94011. https://doi.org/10.1109/access.2019.2928364

Lillicrap, T. P., et al., (2020). Backpropagation and the brain. *Nature Reviews. Neuroscience*, 21(6): 335-346. https://doi.org/10.1038/s41583-020-0277-3

Ministerio de Salud y Protección Social de Colombia. (2019). *Sala situacional de las personas con discapacidad*. https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/MET/sala-situacional-discapacidad2019-2-vf.pdf

Miseikis, J., et al., (2018). Robot localisation and 3D position estimation using a free-moving camera and cascaded convolutional neural networks. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. https://doi.org/10.1109/aim.2018.8452236

O'Mahony N. et al. (2020) Deep Learning vs. Traditional Computer Vision. En Arai K., Kapoor S. (eds) *Advances in Computer Vision. CVC 2019. Advances in Intelligent Systems and Computing*. Springer. https://doi.org/10.1007/978-3-030-17795-9_10

Poernomo, A., y Kang, D.-K. (2018). Biased Dropout and Crossmap Dropout: Learning towards effective Dropout regularization in convolutional neural network. *Neural Networks: The Official Jour-*

nal of the International Neural Network Society, 104: 60-67. https://doi.org/10.1016/j.neunet.2018.03.016

Pramod, R. T., Katti, H., & Arun, S. P. (2018). *Human peripheral blur is optimal for object recognition.* http://arxiv.org/abs/1807.08476

Qi, J., et al., (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, 27: 1485-1489. https://doi.org/10.1109/lsp.2020.3016837

Qin, Z., et al., (2018). How convolutional neural networks see the world. A survey of convolutional neural network visualization methods. *Mathematical Foundations of Computing*, 1(2): 149-180. https://doi.org/10.3934/mfc.2018008

Redmon, J., y Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. http://arxiv.org/abs/1804.02767

Smola, A., y Vishwanathan, S. (2008). *Introduction to Machine Learning*. https://alex.smola.org/drafts/thebook.pdf

Tzutalin. (2015). *LabelImg Free Software: MIT License*. https://github.com/tzutalin/labelImg

Valencia, N. O., et al., (2016). Movement detection for object tracking applied to the InMoov robot head. *XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*. https://doi.org/10.1109/stsiva.2016.7743328

Wozniak P., et al., (2018) Scene Recognition for Indoor Localization of Mobile Robots Using Deep CNN. En Chmielewski L., et al., (eds) *Computer Vision and Graphics. ICCVG 2018. Lecture Notes in Computer Science*. Springer. https://doi.org/10.1007/978-3-030-00692-1_13

Xu, Y., y Wang, Z. (2021). Visual sensing technologies in robotic welding: Recent research developments and future interests. *Sensors and Actuators. A, Physical*, 320(112551), 112551. https://doi.org/10.1016/j.sna.2021.112551

Zhang, Z.; Song, Y., y Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.463