ISBN: 978-628-95207-8-1. ISSN: 2414-6390. Digital Object Identifier: 10.18687/LACCEI2024.1.1.1772

# Detection and counting of vehicles using deep learning in a parking lot area of a university institution

Gian Franco Gonzales del Valle Romero, Ing[1], Walter Jesús Neyra Espinoza, Ing[2], and Pedro Huamaní-Navarrete, Dr[3]
[1,2,3]Ricardo Palma University, Perú, gian.gpmzalesdelvalle@urp.edu.pe, walter.neyra@urp.edu.pe, phuamani@urp.edu.pe

***Abstract–*** ***This article describes the implementation of two artificial neural networks with deep learning, with the objective of detecting and counting vehicles in the main parking area of a university institution in Lima-Peru, Universidad Ricardo Palma. In this way, it became possible to know the number of available parking spaces to reduce consultation time by the security personnel of said institution. Convolutional neural networks were used: You Only Look Once v5 (YOLO v5) and Faster Region-based Convolutional Neural Network (Faster R-CNN), and Transfer Learning was applied. The dataset was made up of images from the place and obtained from the web. To count vehicles, space limitation was used, by the line of interest (LOI) and region of interest (ROI). And, for the graphical user interface, the Tkinter library was chosen, which allowed numerical visualization of the detected vehicles and available spaces. Finally, for real-time implementation, a mobile phone was installed above the third gate of the university itself, using the university's Internet network and connected to a laptop; and, when performing the analysis for the two thresholds used, 0.35 and 0.55, and 27 parking spaces, the YOLO v5 neural network gave a lower average error equal to 5.33%, while the Faster R-CNN neural network did so with 7.82%; On the other hand, the same YOLO v5 network reached a higher average precision value, 81.62%, in training with 100 epochs.***

***Keywords–*** ***YOLO v5, Faster R-CNN, region of interest, vehicle counting, Tkinter library.***

# Detección y conteo de vehículos utilizando aprendizaje profundo en una zona del estacionamiento de una institución universitaria

Gian Franco Gonzales del Valle Romero, Ing[1], Walter Jesús Neyra Espinoza, Ing[2], and Pedro Huamaní-Navarrete, Dr[3]

[1,2,3]Ricardo Palma University, Perú, gian.gpmzalesdelvalle@urp.edu.pe, walter.neyra@urp.edu.pe, phuamani@urp.edu.pe

***Resumen*– Este artículo describe la implementación de dos redes neuronales artificiales con aprendizaje profundo, con el objetivo de detectar y contar vehículos en la zona principal del estacionamiento de una institución universitaria de Lima-Perú, Universidad Ricardo Palma. De esta manera, se hizo posible conocer el número de espacios disponibles de estacionamientos para reducir el tiempo de consulta, de parte del personal de seguridad de dicha institución. Se utilizaron las redes neuronales convolucionales: You Only Look Once v5 (YOLO v5) y Faster Region-based Convolutional Neural Network (Faster R-CNN), y se aplicó el Aprendizaje por Transferencia. El dataset estuvo conformado por imágenes propias del lugar y obtenidas de la web. Para el conteo de vehículos se utilizó la limitación de espacios, por la línea de interés (LOI) y región de interés (ROI). Y, para la interfaz gráfica de usuario, se optó por la librería Tkinter que permitió visualizar numéricamente los vehículos detectados y espacios disponibles. Finalmente, para la implementación en tiempo real, se instaló un teléfono móvil sobre la garita de la tercera puerta de la propia universidad, utilizando la red de internet de la misma y conectado a una laptop; y, al realizar el análisis para los dos umbrales utilizados, 0.35 y 0.55, y 27 espacios de estacionamientos, la red neuronal YOLO v5 otorgó un menor error promedio igual a 5.33%, mientras que la red neuronal Faster R-CNN lo hizo con 7.82%; por otro lado, la misma red YOLO v5 alcanzó un mayor valor de media de precisión, 81.62%, en el entrenamiento con 100 épocas.***

***Palabras claves*– YOLO v5, Faster R-CNN, región de interés, conteo de vehículos, librería Tkinter.***

## I. Introducción

Esta investigación tuvo como propósito utilizar los algoritmos de aprendizaje profundo, representados por las redes neuronales convolucionales, para la detección y conteo de vehículos en la zona principal del estacionamiento de la Universidad Ricardo Palma (URP), en Lima, Perú.

Pues, actualmente, los estudiantes, el personal administrativo y los docentes que hacen uso del estacionamiento de la universidad, deben esperar en algunas oportunidades demasiado tiempo para ingresar a través de las puertas vehiculares; además, no se cuenta con un monitoreo y control de los espacios disponibles del estacionamiento, lo que ocasiona congestión vehicular y en ciertas situaciones obliga al vigilante de la garita abandonarla, para ir en busca de los espacios libres lo cual da origen a un nuevo retraso de tiempo que puede alcanzar hasta los 10 minutos, en el momento del ingreso.

Sin embargo, teniendo como limitación el campo de visión de la cámara utilizada para la adquisición de los videos, la cual fue instalada sobre la garita de la tercera puerta del estacionamiento de la universidad, esta investigación vio conveniente trabajar en una zona o sección principal de dicho estacionamiento. Otra limitación fue la realización de una grabación durante las horas del día, con condiciones climáticas estables y ausencia de lluvia; tal es así que, se tuvo suficiente iluminación para el análisis de los videos capturados debido a la ubicación estratégica de la cámara empleada.

Adicionalmente, se limitó al uso de dos estrategias de conteo que restringieron el uso de las redes neuronales artificiales dentro del área de acción de la cámara de video. Y como también, la elección de las dos redes neuronales de aprendizaje profundo estuvo direccionado a la aplicabilidad del problema, en cuanto a reconocimiento de objetos, tal como se describe en la siguiente sección. Por último, las grabaciones fueron realizadas en un periodo de siete meses posteriores al retorno de la presencialidad, entre los meses de mayo a noviembre.

Es así como, el artículo fue estructurado de la siguiente manera. En la primera sección se presentó la introducción, luego en las dos siguientes secciones se continuó con los trabajos relacionados y el marco teórico, luego en la cuarta sección el desarrollo del prototipo; posteriormente, en la quinta sección se incluyen los resultados de las simulaciones, seguido de las conclusiones y las referencias bibliográficas utilizadas en esta investigación.

## II. Trabajos Relacionados

Igualmente, para el desarrollo de este trabajo se tomaron en cuenta tesis y artículos de investigación orientadas a la detección y conteo de personas en espacios cerrados utilizando estrategias basadas en visión artificial, así como también comparación de los modelos YOLO, SSD, Faster R-CNN, la de reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el estacionamiento de la Universidad Ricardo Palma, entre otras más.

Es así como, el uso de tres modelos de redes neuronales convolucionales permitió la detección del uso correcto de la mascarilla, previo re escalado de las imágenes para evitar el uso exhaustivo del rendimiento del hardware durante el entrenamiento [1]. De igual forma, el uso de otros tres modelos particulares de redes neuronales convolucionales, utilizando el Toolbox Deep Learning del MATLAB, permitió

la detección de placas de rodaje alcanzando un resultado del 95% de efectividad [2]. Luego, también fue utilizado el Toolbox Deep Learning del MATLAB para entrenar tres modelos de redes neuronales convolucionales, destacando la importancia de abundante y variedad de imágenes para el entrenamiento con la finalidad de evitar el sobreajuste [3].

Asimismo, una red neuronal convolucional Faster R-CNN fue utilizada para la selección de arándanos [4]; y, en [5] se utiliza una red neuronal multicapa, previo procesamiento de la imagen, para diferenciar los granos de trigo respecto a malezas, pero destacando la importancia de grupos de datos para el entrenamiento, la validación y la prueba.

De igual manera, en [6] se empleó una red neuronal artificial convolucional para contar las personas; sin embargo, el hardware utilizado no fue suficiente y más bien se sugirió el uso de una GPU para evitar falsos negativos. Y como también, en [7] se utilizó la red neuronal YOLO v4 para reconocer objetos en tiempo real para la conducción autónoma aplicando una cámara, y con cierto margen de error mínimo. No obstante, la detección de vehículos utilizando una red neuronal YOLO v5 fue lograda a partir del uso de un GPU [8].

Igualmente, en [9] se utilizó YOLO y Faster R-CNN en la aplicación de seguimiento de objetos; así como también, en [10] se emplearon los modelos YOLO v3, Faster R-CNN y SSD para la detección de pastillas en el área médica farmacéutica; y, una vez más la estructura YOLO v5 permitió detectar vehículos en carreteras desde vehículos aéreos no tripulados [11].

## III. Marco Teórico

Esta sección aborda los conceptos referentes al desarrollo de esta investigación, la cual fue producto de una sustentación de tesis para la obtención del título de Ingeniero Electrónico, en la Universidad Ricardo Palma, Lima, Perú [12].

### A. *Deep Learning*

Es un conjunto de algoritmos que intenta modelar abstracciones de alto nivel, a través de arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial [13].

### B. *Red neuronal convolucional*

Son tipos de redes neuronales artificiales utilizadas para el análisis y reconocimiento de imágenes, y sus múltiples capas de neuronas están conformadas por un grupo de filtros que convolucionan con ciertas entradas para generar un mapa de características en la salida, basado en patrones simples como una recta hasta complejos como el caso de una imagen [14].

### C. *Aprendizaje por transferencia*

Es un método empleado en el área de Deep Learning, en la que se utiliza una red neuronal capaz de resolver una gran cantidad de problemas gracias a una extensa base de datos, para que solucione un problema específico utilizando el re-entrenamiento [15]. Su uso en esta investigación permitió aprovechar el uso de modelos existentes y previamente entrenados, con la finalidad de mejorar la precisión.

### D. *YOLO v5*

Sistema conformado por una red neuronal artificial convolucional para la detección de objetos en tiempo real. Tiene una arquitectura general conformada por una troncal (Backbone), un cuello (Neck) y cabeza (Head) teniendo partes de CSPDarknet, PANet y YOLO respectivamente, tal como se puede observar en la Fig. 1; asimismo, en la troncal se tienen las capas de cuello de botella CSP y de Pooling espacial piramidal, en el cuello se tienen las capas de convolución 1x1 y 3x3, y funciones de concatenación; por último, la cabeza posee las capas de salida de convolución 1x1 [16]. Seguidamente, la Fig. 1 muestra la representación de la arquitectura de red simplificada perteneciente a YOLO v5.

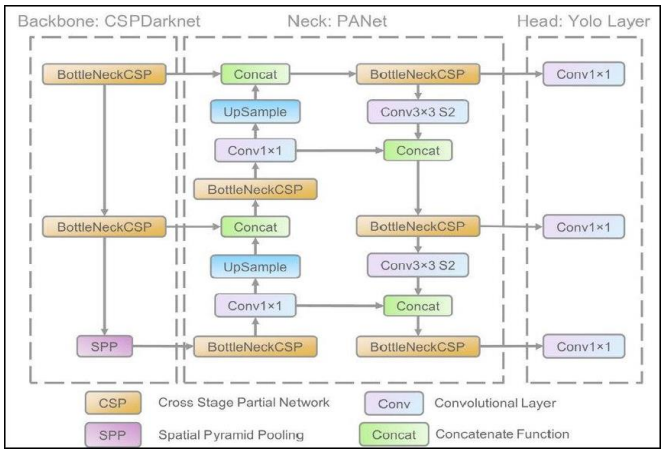


Fig. 1 Arquitectura de red neuronal YOLO v5 [16]

### E. *Faster R-CNN*

Red neuronal artificial convolucional profunda aplicada a la detección de objetos, y conformada por una Region Proposal Network (RPN) como algoritmo de propuesta de región y una Fast Region-based Convolutional Neural Network (Fast R-CNN) como red de detectores [17]. Seguidamente, en la Fig. 2, se muestra la arquitectura de red neuronal Faster R-CNN.

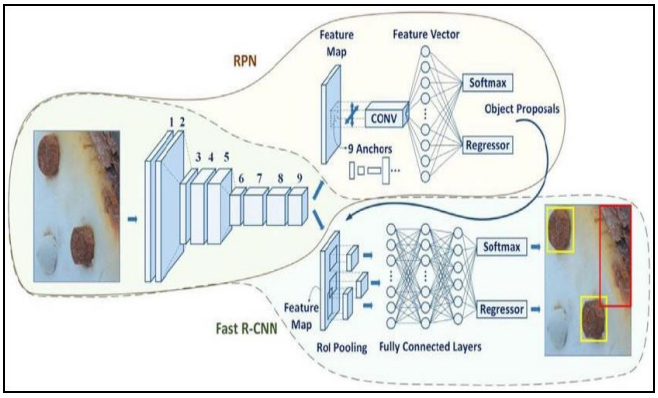


Fig. 2. Arquitectura de red neuronal Faster R-CNN [17]

**Digital Object Identifier:** (only for full papers, inserted by LACCEI).
**ISSN, ISBN:** (to be inserted by LACCEI).
**DO NOT REMOVE**

### *F. Estrategias de conteo*

Según [18], las estrategias de conteo más populares se encuentran divididas en dos grupos. Línea de interés (LOI) que está basada en el número de personas en movimiento que atraviesan cierta línea de la escena definida por el usuario. Y, Región de interés ROI) que está basada en el número de personas estáticas y en movimiento que se encuentran en determinada zona de una escena.

## IV. Desarrollo del Prototipo

Esta sección contiene el desarrollo del prototipo, el cual consiste en cinco etapas fundamentales, desde la grabación de los videos de un área del estacionamiento de la Universidad Ricardo Palma, pasando por los algoritmos de extracción de características en los fotogramas, hasta el entrenamiento y validación de las redes neuronales convolucionales propuestas en este trabajo.

### *A. Almacenamiento de los videos y extracción de fotogramas*

Para la recolección de los videos se utilizó una laptop Lenovo Ideapad Gaming 3 con un procesador Intel Core i5 (2.5 GHz hasta 4.20 GHz), un GPU NVIDIA GeForce GTX 4 GB GDDR5, una RAM 8 GB DDR4, un disco SSD de 512 GB, conectividad por Wi-Fi 6 2x2 AX y Bluetooth 5.0; así como también, un teléfono celular Xiamoi Mi 11 Lite con procesador Qualcomm Snapdragon 778G, 6 GB de RAM, capacidad de 128 GB, cámara frontal de 20 MB, y conectividad Wi-Fi 6 y Bluetooth 5.2. Si bien es cierto, que un teléfono móvil es una solución práctica e inmediata, no fue del todo efectivo porque estaba limitado al alcance y resolución de su propia cámara de video.

Por lo cual, para la obtención de la base de datos, se realizó la captura de videos de la zona principal del estacionamiento de la Universidad Ricardo Palma, utilizando el dispositivo Xiaomi Mi 11 Lite configurado a una resolución de 1920x1080 píxeles, y ubicado a 1.8 metros por encima de la garita de seguridad con altura de 7 metros. Al lado de dicho dispositivo, se instaló el router y la antena sectorial de Wi-Fi. Ver la Fig. 3.

Fig. 3. Fotografía mostrando la instalación de la cámara sobre la garita de la tercera puerta del estacionamiento de la Universidad.

Por lo tanto, dicha zona principal correspondió a 56 estacionamientos tal como se observa en la Fig. 4.

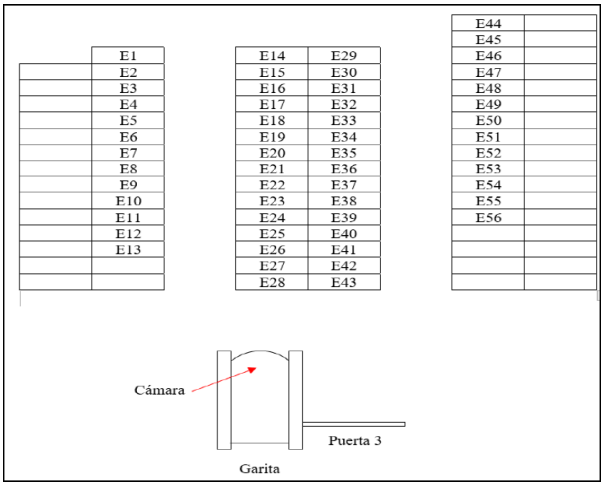


Fig. 4. Representación de la zona principal del estacionamiento de la Universidad Ricardo Palma con sus 56 espacios.

En cuanto a las grabaciones realizadas, es decir el conjunto de datos reales, fueron en promedio de 40 minutos por cada visita que se tuvo, de los cuales fueron extraídos de 7 a 10 muestras por cada video. Posteriormente, fueron redimensionados a una resolución de 640 x 640 pixeles, y almacenados en una carpeta dataset para luego crear las cajas rectangulares o cuadradas delimitadoras, conocido en el idioma inglés como bounding boxes. La aplicación del redimensionamiento no perjudicó la calidad de las imágenes debido a las características de la cámara del celular utilizado, en el proceso de captura.

Por otro lado, para los datos artificiales se restringió a la recolección de imágenes de vehículos desde la plataforma de búsqueda de Google, por medio de la extensión Image Downloader de Chrome; por lo cual, no fue considerada la expansión del dataset con imágenes de estacionamientos de otros centros universitarios por la limitación de solicitud de permiso, y porque el tipo de vehículo que participó en este trabajo fue el clásico automóvil o camioneta SUV (vehículo utilitario deportivo). De esta manera, se consideró dicho grupo de imágenes como representativo para el entorno real de un estacionamiento en un centro universitario.

### *C. Creación de los boundings boxes*

La creación de etiquetas se realizó utilizando la plataforma web de Make Sense, lo cual consistió en subir las imágenes de vehículos para realizar el bounding boxes para cada una, teniendo como única etiqueta el tipo de auto, tal como se visualiza en la Fig. 5.

Posteriormente, se optó por descargar los datos de las bounding boxes tanto en formato ".xml" y ".txt" para ambos algoritmos de aprendizaje profundo, Faster R-CNN y YOLO v5, respectivamente. Finalmente se organizaron las imágenes y sus respectivos archivos en dos carpetas separadas: "train",

para el entrenamiento de la red neuronal convolucional; y "val", para la validación de estas.

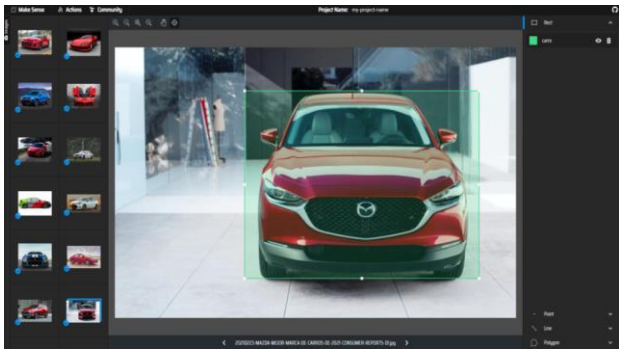

Fig. 5. Datos artificiales y reales utilizados en la plataforma Make Sense para la creación de bounding boxes.

*D. Aprendizaje or transferencia con YOLO v5*

Como es de conocimiento, cada variante de YOLO v5 se diferencia en la profundidad de neuronas, por lo cual se eligió la variante YOLO v5x porque posee el mejor porcentaje de predicción debido a la profundidad de cada capa, teniendo como promedio una precisión de 68.9 % y una velocidad de 766 milisegundos, lo cual no afectó al desarrollo de esta investigación [19].

De esta manera, para el aprendizaje por transferencia se aplicó el congelamiento de las primeras 20 capas dejando las últimas 4 capas de convolución, de la sección head, para el reentrenamiento y teniendo como configuración de etiqueta la categoría de auto. Este reentrenamiento se realizó en la plataforma de Google Colab por lo mismo que se encuentra en la nube y facilita su uso para el entrenamiento de redes neuronales. Para dicho reentrenamiento se utilizaron las imágenes recolectadas y las bounding boxes obtenidas en formato .txt, utilizando la configuración de 150 épocas y resolución de 640 pixeles con un número de muestras procesadas (batch size) de 16.

*E. Aprendizaje or transferencia con Faster R-CNN*

Se utilizó el método de Fine Tuning para el re-entrenamiento de la red Faster R-CNN. Esto se logró manteniendo los pesos iniciales de la red pre entrenada y haciendo un re-entrenamiento a toda la red, modificando la capa de softmax regression, encargada de calcular la probabilidad de predicción de cada clase, y la capa de salida encargada de la clasificación de dos clases: background y vehículos.

*F. Implementación del algoritmo de conteo por ROI y LOI*

Consistió en delimitar el área de acción de la red neuronal convolucional para detectar solamente los vehículos ubicados entre las líneas de interés (LOI), o dentro de las regiones de interés (ROI). Para ello, se utilizó la biblioteca de visión computacional y procesamiento de imágenes OpenCV, en Python, para redimensionar los fotogramas por interpolación, y dibujar las líneas delimitadoras LOI y los polígonos que conforman las ROI.

Para las LOI se utilizó la función *line()*, que permitió trazar seis líneas para delimitar los espacios de estacionamiento, con la finalidad que la red neuronal convolucional cuente solo los vehículos que traspasen las líneas determinadas. Y, para las ROI se utilizó la función *fillPoly()*, que permitió dibujar un polígono relleno cuyas coordenadas fueron los vértices de los tres polígonos que delimitan los espacios de estacionamiento. Con ello, se logró crear una imagen de máscara con pixeles iguales a 1 en la zona de interés, y pixeles iguales a 0 en la zona del fondo de la imagen.

Luego, se realizó una operación AND entre los valores de los píxeles de la nueva máscara y los píxeles del fotograma, donde se aprecia solo los espacios de estacionamiento delimitados por las ROI.

A continuación, la Fig. 6 muestra el diagrama de bloques que resume el procedimiento descrito anteriormente, desde el fotograma hasta la obtención de la imagen final utilizada por las redes neuronales convolucionales en el proceso de entrenamiento, así como la creación de sus respectivos bounding boxes y del conteo automático.

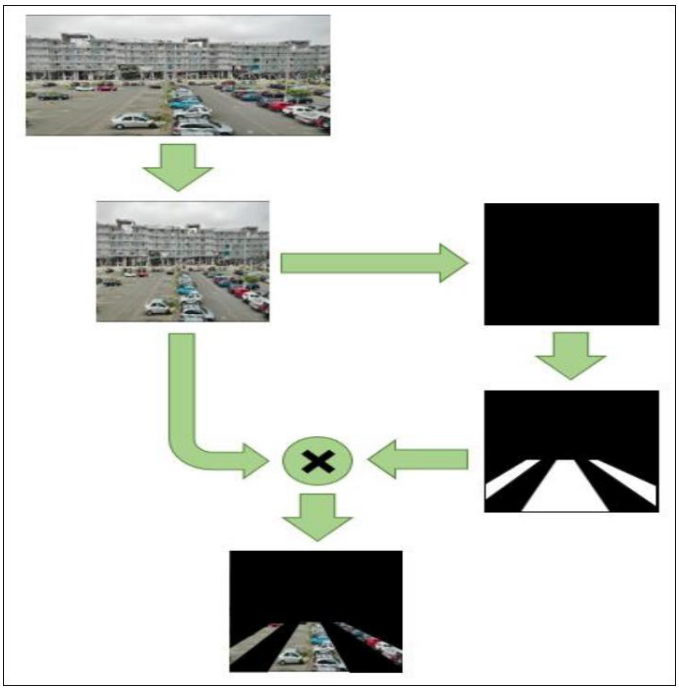

Fig. 6. Diagrama de bloques del procedimiento de conteo por ROI.

*G. Aplicativo GUI para la interfaz gráfica*

Una vez aplicado el aprendizaje por transferencia, se optó por implementar una interfaz gráfica para visualizar el resultado de la aplicación del aprendizaje profundo. Para ello, se utilizó la librería Tkinter del lenguaje de programación Python, debido a que es intuitiva porque permite de una manera fácil y rápida comprenderla; por otro lado, es funcional porque reacciona a la entrada del usuario realizando los cambios del programa.

Sin embargo, por el alto consumo de recursos del procesador de la memoria de video, se tuvo que agregar un botón que permitió iniciar y detener el proceso de detección y conteo de vehículos; complementariamente, se agregaron casillas para visualizar el número de vehículos estacionados y los espacios libres de estacionamiento.

A continuación, la Fig. 7 muestra una representación de la interfaz gráfica implementada cuando se utilizó YOLO v5.

## V. Resultados de las Simulaciones

Para la comparación de los dos modelos de redes neuronales se utilizaron la media de precisión promedio (mean average precision o mAP), y tomando solo las detecciones que tienen una intersección sobre unión (IoU) mayor a 0.5, que es lo convencional para considerar una detección como válida. Por lo tanto, durante el entrenamiento de la red neuronal convolucional YOLO v5, utilizando 100 épocas, la red alcanzó un mAP de 81.615%; mientras que la Faster R-CNN, para el mismo número de épocas, logró un mAP de 81.595%, ligeramente menor al anterior modelo de red.


Fig. 7. Interfaz gráfica de usuario utilizando Tkinter.

De esta manera, para el conteo usando las LOI se delimitó el área a través de líneas para detectar los vehículos, señalizados con un punto rojo en el centro de estos; sin embargo, el contador aumentó progresivamente llegando a contar con un alto error. Por otro lado, para el conteo usando las ROI se extrajeron las regiones llegando a contar solamente los rectángulos ubicados en dichas áreas, por lo cual el contador no incrementó indefinidamente y más bien se mantuvo estable oscilando entre los valores de 22 a 25 vehículos.

Asimismo, para observar el rendimiento de ambos modelos de redes neuronales, se realizaron dos pruebas. Una para reconocer 56 espacios en el estacionamiento de la Universidad, y otra para 27. Adicionalmente, se optó por utilizar los valores de umbral 0.35 y 0.55 para estudiar la estabilidad del algoritmo al limitar la cantidad de detecciones, entregadas al contador, en cada caso; así como también, se agregó un contador de cuadros por segundo (FPS, siglas en inglés) para observar la velocidad de la red neuronal en el momento de la detección de un vehículo. Por tal razón, el video, los bounding boxes, el contador de vehículos de FPS se visualizaron a través del GUI implementado.

Durante la ejecución de las pruebas se utilizó la red Wi-Fi de la propia universidad y una laptop para la recepción de los datos enviado por el celular, posicionándose próxima a la garita con la finalidad de aprovechar al máximo la señal de internet; pero, aun tomando en cuenta estas consideraciones, al realizar la medida de latencia desde la laptop al celular conectado a la red Wi-Fi, se encontró una demora de aproximadamente medio segundo. Ver la Fig. 8.


Fig. 8. Prueba de latencia y velocidad de la señal de internet realizado desde la laptop al celular conectado a la red Wi-Fi de la URP.

### *A. Obtención de FPS por los modelos de redes neuronales*

Para ello, se optó por extraer la cantidad de FPS en una grabación de video de un minuto y medio, con la finalidad de analizar una posible diferencia de velocidad de procesamiento, entre los dos modelos de redes neuronales propuestos al realizar cambios del valor de umbral, para los casos de 56 y 27 espacios de estacionamiento

Por lo tanto, para el caso de 56 espacios de estacionamiento, se tomaron muestras de los FPS extraídos en cada segundo durante la duración del video de prueba, para cada uno de los modelos y sus correspondientes valores de umbral. Es así como, haciendo la comparación, se observa que la red neuronal YOLO v5 alcanzó la mayor tasa de FPS que fue igual a 1.44 en promedio. También se percibe que el cambio de umbral no afecta significativamente la velocidad.

A continuación, la Fig. 9 y Fig. 10 muestran la evolución de los FPS obtenidos con el modelo YOLO v5 y Faster R-CNN en los 90 segundos de duración del video para un umbral de 0.55 y 56 espacios de estacionamiento.

Del mismo modo, se realizó el procedimiento de extracción de los FPS para el caso de 27 espacios de estacionamiento; por lo cual, al realizar la comparación entre ambos modelos de redes neuronales, se observó valores ligeramente superiores a las pruebas realizadas con 56 espacios de estacionamiento. Además, la red neuronal YOLO v5 obtuvo nuevamente la mayor tasa que fue igual a 1.86 en promedio, para el caso de un umbral igual a 0.35 e inferior para el umbral de 0.55.

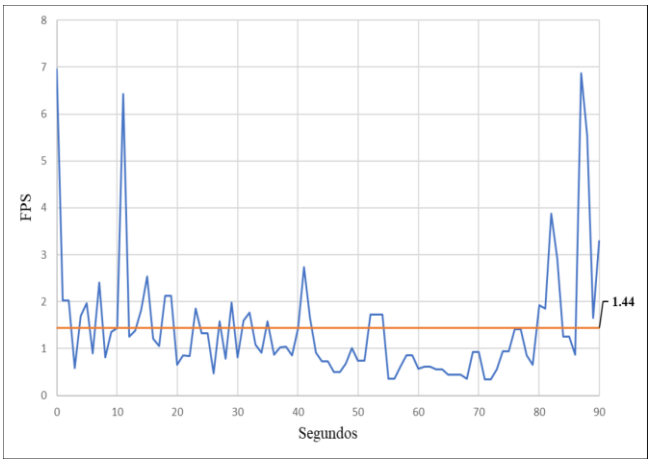

Fig. 9. Representación gráfica de la evolución de FPS con YOLO v5, umbral 055 y 56 espacios de estacionamiento.

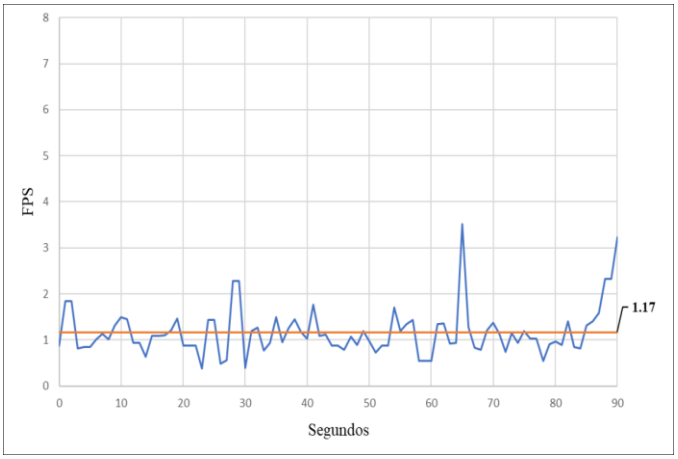

Fig. 10. Representación gráfica de la evolución de FPS con Faster R-CNN, umbral 055 y 56 espacios de estacionamiento.

*B. Prueba de precisión y estabilidad del contador*

Para la realización de esta prueba se procedió a extraer el número de espacios disponibles por cada segundo durante una grabación de video de un minuto y medio, con la finalidad de observar la variación del contador en el tiempo con respecto al número de espacios disponibles reales para determinar su estabilidad, así como también observar la cercanía de los valores del contador al número real de espacios disponibles para determinar la precisión.

Es así como, después de registrar por cada segundo la cantidad de espacios disponibles otorgada por cada modelo de red neuronal, cuando el número real fue dos, se observó que la red neuronal YOLO v5 para el caso de 56 espacios de estacionamientos otorgó un valor de 12 en promedio para el umbral de 0.35, y 30 en promedio para el umbral de 0.55. No obstante, la estabilidad fue mejor debido a que presentó menor variación durante la cuenta de espacios. Por otro lado, la red neuronal Faster R-CNN resultó más precisa porque otorgó un valor de 2 en promedio para el umbral de 0.35, y 9 en promedio para el umbral de 0.55. No obstante, se obtuvieron falsos positivos porque detectó vehículos donde no los había.

De la misma manera, se obtuvieron los resultados de la prueba de precisión y estabilidad del contador cuando se consideraron 27 espacios de estacionamiento. Entonces, cuando el número real fue uno, se observó que la red neuronal YOLO v5 otorgó un valor de 2 en promedio para el umbral de 0.35, y 7 en promedio para el umbral de 0.55. No obstante, la estabilidad fue mejor debido a que presentó menor variación durante la cuenta de espacios. Por otro lado, la red neuronal Faster R-CNN resultó más precisa porque otorgó un valor de 1 en promedio para el umbral de 0.35, y 3 en promedio para el umbral de 0.55. Nuevamente, se obtuvieron falsos positivos porque detectó vehículos donde no los había.

De esta manera, ambos modelos de redes presentaron menor precisión cuando se aumentó el umbral de 0.35 a 0.55, pero a la vez mostraban mayor estabilidad.

*C. Cálculo del promedio del porcentaje de error*

El cálculo del promedio del porcentaje de error fue realizado por cada segundo durante la duración del video. Por lo cual, este procedimiento se determinó para los umbrales de 0.35 y 0.55 en ambos modelos de redes neuronales.

Es así que, al analizar 56 espacios de estacionamiento, se obtuvieron 19.39% y 52.67% para el caso de la red YOLO v5 con umbrales de 0.35 y 0.55, respectivamente; mientras que para el caso de la red Faster R-CNN se hallaron 5.07% y 14.15% con umbrales de 0.35 y 0.55, respectivamente. A continuación, la Fig. 11 muestra una captura de pantalla de la interfaz gráfica desarrollada cuando el umbral fue 0.35 en el caso de la red neuronal Faster R-CNN.

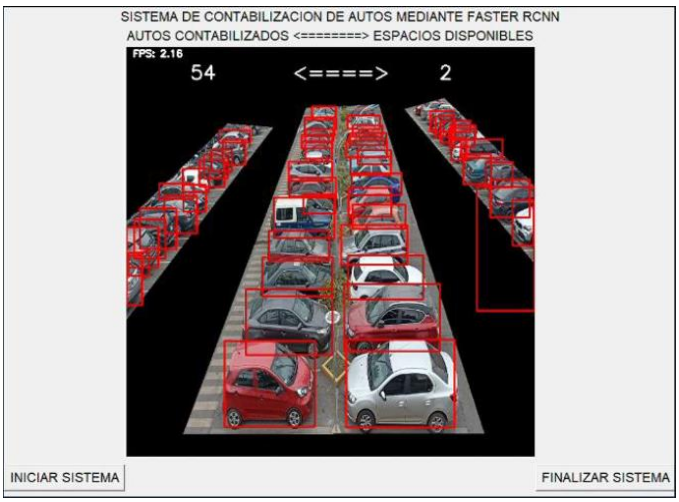

Fig. 11. Cantidad mínima de vehículos contabilizados por el modelo de red Faster R-CNN con umbral de 0.35 y 56 espacios de estacionamiento.

Igualmente, al analizar el caso de 27 espacios de estacionamiento, se obtuvieron 5.33% y 22.02% para el caso de la red YOLO v5 con umbrales de 0.35 y 0.55, respectivamente; mientras que para el caso de la red neuronal Faster R-CNN se hallaron 7.82% y 8.07% con umbrales de 0.35 y 0.55, respectivamente. Seguidamente, la Fig. 12 muestra una captura de pantalla de la interfaz gráfica

desarrollada cuando el umbral fue 0.35 en el caso de la red neuronal Faster R-CNN.


Fig. 12. Cantidad mínima de vehículos contabilizados por el modelo de red YOLO v5 con umbral de 0.35 y 27 espacios de estacionamiento.

### *D. Comparación de los modelos YOLO v5 y Faster R-CNN*

A continuación, en la Tabla I, se presenta la comparación de las métricas entre los dos modelos de redes neuronales convolucionales, a partir de la información recopilada de las secciones anteriores. Esta comparación permitió establecer una conclusión en cuanto al modelo de red que presentó mejor rendimiento.

TABLA I
RESUMEN DE LAS MÉTRICAS PARA LOS DOS MODELOS DE REDES NEURONALES

| | | 56 espacios | | 27 espacios | |
|---|---|---|---|---|---|
| | | Umbral 0.35 | Umbral 0.55 | Umbral 0.35 | Umbral 0.55 |
| YOLO v5 | FPS (promedio) | 1.33 | 1.44 | 1.86 | 1.26 |
| | Contador (promedio) | 12 | 30 | 2 | 7 |
| | Contador (máximo) | 15 | 33 | 4 | 9 |
| | Contador (mínimo) | 9 | 27 | 2 | 4 |
| | % Error promedio | 19.39% | 52.67% | 5.33% | 22.02% |
| Faster R-CNN | FPS (promedio) | 1.31 | 1.17 | 1.80 | 1.59 |
| | Contador (promedio) | -2 | 9 | -1 | 3 |
| | Contador (máximo) | 2 | 13 | 1 | 5 |
| | Contador (mínimo) | -7 | 5 | -4 | 2 |
| | % Error promedio | 5.07% | 14.15% | 7.82% | 8.07% |

Y, en la Tabla II, se observa la comparación resumen en cuanto a las métricas de mAP, FPS, porcentaje de error promedio y menor cantidad de falsos positivos, considerando la información otorgada por la Tabla I. Por lo cual, el modelo de red neuronal YOLO v5 resultó como la mejor.

TABLA II
TABLA COMPARATIVA DE LOS MODELOS DE REDES NEURONALES

| Métricas | Red neuronal Faster R-CNN | Red neuronal YOLO v5 |
|---|---|---|
| *mAP@0.5* | ✓ | |
| Cantidad de FPS | ✓ | |
| % Error Promedio | | ✓ |
| Menor cantidad de falsos positivos | ✓ | |

Asimismo, la aplicación de los algoritmos de delimitación por líneas de interés y región de interés, para el conteo de vehículos, fueron aplicados en tiempo real por el cual, debido a la morfología de los estacionamientos seleccionados y la menor tasa de error en las detecciones de los espacios de estacionamiento, se optó por elegir el método de selección por región de interés delimitando las secciones a contabilizar mediante el procesamiento de imágenes.

## VI. CONCLUSIONES

Se realizó la operación de aprendizaje por transferencia sobre los modelos YOLO v5 y Faster R-CNN a partir de los métodos de congelamiento de capas y fine tuning, respectivamente. Asimismo, se utilizó un dataset conformado por 460 imágenes y en base a los resultados alcanzados en cuanto a precisión, cantidad de FPS, porcentaje de error promedio y menor cantidad de falsos positivos, para los dos umbrales de detección, 35% y 55%, y en las dos situaciones de 27 y 56 espacios de estacionamiento, el modelo de YOLO v5 otorgó mejores resultados con respecto a la detección de vehículos, tal como se observó en la Tabla II.

Además, se logró implementar una interfaz gráfica de usuario para ambas redes neuronales artificiales convolucionales, el cual tiene la funcionalidad de inicializar la cámara y la red neuronal visualizando los resultados de la contabilización de espacios de estacionamiento; asimismo contó con un botón para finalizar la ejecución del sistema de detección y conteo de vehículos.

Como recomendaciones se precisa el uso de una cámara con un campo de visión de 180° y para exteriores, con la finalidad de ampliar el número de detecciones en el estacionamiento. Adicional a ello, optar por una red de internet dedicada con un mínimo de 10 Mbps de velocidad simétrica, evitando así retardos en el intercambio de datos; y, como también, aplicar técnicas de balanceo de datos para mejorar el porcentaje de detecciones de vehículos, y utilizar la versión GPU de OpenCV para elevar la tasa de FPS y alcanzar a utilizar múltiples cámaras a la vez.

Finalmente, como trabajo futuro, se plantea el uso de la versión GPU de OpenCV para la elevar la tasa de FPS, y así aplicar nuevos filtros para la visualización en espacios con

poca iluminación o bien en situaciones nocturnas, así como también emplear múltiples cámaras a la vez que permitan abarcar completamente la zona de estacionamiento de la Universidad Ricardo Palma, en Lima-Perú. Y, la principal recomendación es la de optar por una red de internet dedicada para la aplicación desarrollada, con un mínimo de 10 Mbps de velocidad simétrica para evitar retardos en el envío de información.

## AGRADECIMIENTOS

Se agradece a la Oficina de Servicios Administrativos y Mantenimiento de la Universidad Ricardo Palma, así como al personal de seguridad y vigilancia que labora en las garitas de control de acceso vehicular, por el apoyo brindado al permitirnos realizar las capturas de imágenes y videos desde la parte más alta de una de las garitas, así como también por las sugerencias y demás consideraciones para la realización de este trabajo de investigación.

## REFERENCIAS

[1] X. Chirinos y P. Calero, "Detección del uso correcto de mascarillas utilizando una red neuronal convolucional para el ingreso de personas a un laboratorio de una universidad", Tesis para la obtención del Título de Ingeniero Electrónico, Facultad de Ingeniería, Universidad Ricardo Palma, Lima, Perú, 2021.

[2] B. Ramírez y M. Tito, "Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la Universidad Ricardo Palma", Tesis para la obtención del Título de Ingeniero Electrónico, Facultad de Ingeniería, Universidad Ricardo Palma, Lima, Perú, 2020.

[3] N. Cayllahua y J. Suárez, J, "Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio", Tesis para la obtención del Título de Ingeniero Electrónico, Facultad de Ingeniería, Universidad Ricardo Palma, Lima, Perú, 2019.

[4] W. Narciso y E. Manzano, "Sistema de visión artificial basado en redes neuronales convolucionales para la selección de arándanos según estándares de exportación", Campus, vol. 26, no. 32, pp. 155-166, 2021

[5] P. Rios, "Diseño y entrenamiento de una red neuronal empleando procesamiento de imágenes para diferenciar granos de trigo respecto a malezas", Tesis para optar el Título de Ingeniero Electrónico, Facultad de Ingeniería, Universidad Tecnológica del Perú, Lima, Perú, 2022.

[6] J. Jokela, "Person counter using real-time object detection and a small neural network", Bachelor's Thesis, Information and Communications Technology, Turku University of Applied Sciences, Finlandia. 2018.

[7] A. Sarda, S. Dixit & A. Bhan, "Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm", in Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, pp. 1370-1374, 2021, DOI: 10.1109/ICICV50876.2021.9388577

[8] W. Tian-Hao, W. Tong-Wen & L. Ya-Qi, "Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network", in 3rd World Symposium on Artificial Intelligence (WSAI), Guangzhou, China, pp. 24-28, 2021, DOI: 10.1109/WSAI51899.2021.9486316

[9] R. Deepa, E. Tamiselvan, E. Abrar. & S. Shrinivas, "Comparison of Yolo, SSD, Faster R-CNN for Real Time Tennis Ball Tracking for Action Decision Networks", in International Conference on Advances in Computing and Communication Engineering (ICACCE), Sathyamangalam, India, pp. 1-4, 2019, DOI: 10.1109/ICACCE46606.2019.9079965

[10] L. Tan, T. Huangfu, L. Wu & W. Chen, "Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification", BMC Medical Informatics and Decision Making, 2021, DOI: 10.1186/s12911-021-01691-8

[11] A. Panthakkan, N. Valappil, S. Al-Mansoori and H. Al-Ahmad, "AI based Automatic Vehicle Detection from Unmanned Aerial Vehicles (UAV) using YOLOv5 Model," in 5th International Conference on Image Processing Applications and Systems (IPAS), Genova, Italy, pp. 1-5, 2022, DOI: 10.1109/IPAS55744.2022.10053056.

[12] G. Gonzales del Valle y W. Neyra, "Implementación de una red neuronal convolucional para la detección y conteo de vehículos en una sección del estacionamiento de la Universidad Ricardo Palma", Tesis para la obtención del Título de Ingeniero Electrónico, Facultad de Ingeniería, Universidad Ricardo Palma, Lima, Perú, 2022.

[13] Y. Bengio, A. Courville, & P. Vincent, "Representation Learning: A Review and New Perspectives", IEEE Trans. PAMI, special issue Learning Deep Architectures, vol. 35, pp. 1798-1828, 2014, DOI: 10.1109/TPAMI.2013.50

[14] J. Gelvez, "Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática". (julio, 2019). ResearchGate [Online], DOI: 10.13140/RG.2.2.10855.39843

[15] ResearchGate. (junio, 2023). The network architecture of YOLO v5 [Online]. Available: https://www.researchgate.net/figure/The-network-architecture-of-Yolov5-It-consists-of-three-parts-1-Backbone-CSPDarknet_fig1_349299852

[16] Medium. (Agosto 2019). Faster R-CNN for object detection [Online]. Disponible: https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46

[17] Medium. (Abril 2019). Faster R-CNN object detection [Online]. Disponible: https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4#:~:text=Faster%20RCNN%20is%20an%20object,SSD%20(%20Single%20Shot%20Detector

[18] A. Luna y N. Rodríguez, "Detección y conteo de personas en espacios cerrados utilizando estrategias basadas en visión artificial", Tesis, Departamento de Electrónica, Pontificia Universidad Javeriana, Bogotá D.C., Colombia, 2017.

[19] Github. (2024). Ultralytics Yolov5 [Online]. Disponible: https://github.com/ultralytics/yolov5