

Development of an artificial vision system for the detection of free spaces in private parking lots in Lima

Yeefrin Kenedy Orellana Huamanchaqui
Universidad Tecnológica del Perú, Perú, U17303241@utp.edu.pe

Abstract– Private parking lots face significant challenges due to the increasing number of vehicles, requiring advanced technological solutions to optimize space use and improve user experience. The artificial vision system for spot detection is proposed as an innovative solution using machine learning techniques. The VDI 2206 methodology was adopted due to its suitability in the design of mechatronic systems, integrating various areas of knowledge for effective system control. The development of the system was divided into five key stages. In the first stage, "Requirements", quality images were collected to define architectures and select electronic and software components. In the second stage, the hyperparameters of the VGG16 model and the SVM classifier were adjusted, designing flowcharts and programming. The third stage included coding the detection model, developing live video streaming modules, and improving image post-processing. The fourth stage focused on the assembly and configuration of the hardware and camera, optimizing the code for accurate real-time detection. Finally, in the fifth stage, the data were validated, and the model was evaluated, reaching a precision of 0.90 and an accuracy of 0.92 in image processing in one second, ensuring the coherence and effectiveness of the process and achieving effective detection of parking spaces. Parking lot.

Keywords-- Artificial vision, convolutional, parking.

Desarrollo de un sistema de visión artificial con redes neuronales convolucionales para la detección de plazas libres en estacionamientos privados en Lima

Yeefrin Kenedy Orellana Huamanchaqui
Universidad Tecnológica del Perú, Perú, U17303241@utp.edu.pe

Resumen– Los estacionamientos privados enfrentan desafíos significativos debido al creciente número de vehículos, lo que requiere soluciones tecnológicas avanzadas para optimizar el uso del espacio y mejorar la experiencia del usuario. El sistema de visión artificial para la detección de plazas se propone como una solución innovadora, empleando técnicas de aprendizaje automático. Se adoptó la metodología VDI 2206 por su idoneidad en el diseño de sistemas mecatrónicos, integrando diversas áreas del conocimiento para un control eficaz del sistema. El desarrollo del sistema se dividió en cinco etapas clave. En la primera etapa, "Requerimientos", se recolectaron imágenes de calidad para definir arquitecturas y seleccionar componentes electrónicos y de software. En la segunda etapa, se ajustaron los hiperparámetros del modelo VGG16 y del clasificador SVM, diseñando diagramas de flujo y programación. La tercera etapa incluyó la codificación del modelo de detección, desarrollando módulos de transmisión de video en vivo y mejorando el pos-procesamiento de imágenes. La cuarta etapa se centró en el ensamblaje y configuración del hardware y la cámara, optimizando el código para una detección precisa en tiempo real. Finalmente, en la quinta etapa se validaron los datos y se evaluó el modelo, alcanzando una precisión de 0.90 y una exactitud de 0.92 en el procesamiento de imágenes en un segundo, asegurando la coherencia y efectividad del proceso y logrando una detección efectiva de plazas de estacionamiento.

Palabras clave - visión artificial, convolucional, estacionamiento

I. INTRODUCCIÓN

La adquisición de vehículos ha aumentado significativamente en muchos países debido al crecimiento poblacional. En Perú, el parque automotor creció, con autos de servicio público representando el 54% en congestión vehicular [1]. Este aumento provoca aglomeraciones en universidades, centros comerciales y playas de estacionamientos, como se observó en Shanghái, donde 9,558 vehículos entran y salen semanalmente de un supermercado [2]. La falta de disponibilidad de plazas causa incomodidad a los conductores, y un mal diseño de estacionamientos contribuye a la congestión [3]. Se propone el uso de inteligencia artificial para detectar vacantes y mejorar el control de estacionamientos.

Investigadores apoyan el uso de aprendizaje profundo y visión por computadora para mejorar la predicción de datos. Gao et al. destacan el uso de ConvGRU para predecir estacionamientos libres con alta velocidad y puntualidad [4]. Zou y otros emplearon una red convolucional CNN para detectar vehículos en 60 milisegundos con una eficiencia del 70 al 90% [5]. Ambos métodos muestran buenos resultados al clasificar y predecir datos cualitativos.

La visión artificial se ha utilizado para detectar plazas libres en estacionamientos. Bassam y Samann emplearon un modelo ORC para identificar caracteres en imágenes, mientras que Elmana et al. usaron un modelo convolucional para clasificar vehículos por tamaño [6]. Zhang y Yang implementaron reconocimiento aéreo de autos para controlar el tráfico [7]. Estos estudios requieren tecnología avanzada para ser efectivos.

Métodos de detección de bordes son esenciales en la identificación de objetos. Chen y otros usaron el método Sobel para eliminar ruido en imágenes 2D, aunque no es eficaz en imágenes 3D [8]. El método Prewitt, más eficaz en detectar bordes verticales y horizontales, es ideal para plazas en aparcamientos al aire libre [9]. Prasetyo y otros compararon ambos métodos en 2023, encontrando que Prewitt es similar a Sobel, pero más eficiente.

Modelos actuales mejoran la precisión en la localización de objetos. Paidi y Fleyeh implementaron Fast-RCNN, aunque con deficiencias en datos y tiempo [10]. Barhamtoshy y otros usaron RCNN lineal para evitar pérdida de datos [11]. Naufal et al. combinaron ajuste de contraste con RCNN para mejorar detección en estacionamientos [12]. Aunque estos métodos son efectivos, requieren bases de datos extensas y sistemas computacionales avanzados.

Según las referencias anteriores, el aprendizaje profundo en inteligencia artificial permite predicciones precisas comparado con el aprendizaje automático. La visión artificial se aplica en reconocimiento de caracteres, clasificación de vehículos y localización en tiempo real, optimizando recursos tecnológicos. Sin embargo, falta un algoritmo sencillo y preciso para detectar plazas libres con una base de datos pequeña. Este trabajo busca diseñar una red CNN que detecte y cuente plazas en estacionamientos con una base de datos mínima.

II. METODOLOGÍA

La metodología VDI 2206 es un marco de trabajo ampliamente reconocido para el desarrollo de sistemas mecatrónicos, enfocándose en asegurar que los sistemas sean eficientes, seguros y confiables desde su concepción hasta su implementación. Se estructura en diversas etapas que abarcan desde la definición de requisitos hasta la validación final del sistema. En el contexto del trabajo de investigación sobre el desarrollo de un sistema de visión artificial para la detección de plazas libres en estacionamientos privados, la metodología

VDI 2206 fue fundamentada. En un principio se validó el requerimiento de la base de datos, en donde se plantearon dos subtemas, los cuales fueron abordados en la detección de las plazas en un estacionamiento privado, así también se identificó y se concretó la creación del script con Python en el entorno de GoogleColab. En la segunda etapa de diseño, se estableció los hiperparámetros para el ajuste adecuado y mejorar significativamente el rendimiento del modelo junto el diseño del diagrama de flujo del sistema propuesto y la programación, los cuales se siguió en el entorno de GoogleColab teniendo en cuenta los requerimientos y el diseño previo, para después realizar la integración en la cuarta etapa. Por último, se realizó la verificación de los datos empleadas y el modelo entrenado.

III. RESULTADOS

A. Resultados de requerimiento

En seguida, Dentro de la etapa de requerimiento se logró establecer las bases para todas las fases que siguen se cumplan. La definición del modelo RCNN para la detección permitió crear un Sistema personalizado, donde la arquitectura VGG16 y SVM se complementan aumentando la generalización de detección de múltiples plazas con distintos tamaños de vehículos. El VGG16 al ser una arquitectura pre-entrenada y utilizar 16 capas profundas, aseguro un procesamiento uniforme de las características; con su diseño sencillo y eficaz, capturo detalles finos y patrones complejos de los distintos vehículos que están presente en la base de datos tal como se ve en la figura 1. Gracias a su preentrenamiento en grandes conjuntos de datos como ImageNet, VGG16 pudo generalizar bien a una variedad de distintos escenarios en el prototipo diseñado de estacionamiento y mostrando buenos resultados en cada una de ellas, todo mientras mantiene un buen equilibrio entre precisión y costo computacional. Su disponibilidad como modelo pre-entrenada en bibliotecas como TensorFlow y PyTorch facilita el uso del transfer learning.

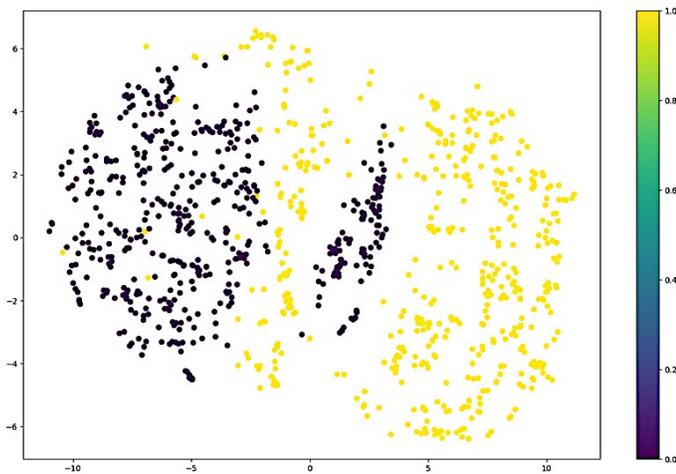


Fig. 1. Datos extraídos con el VGG16

Por otro lado, el uso del clasificador SVM resulto ser especialmente efectivo en el prototipo del estacionamiento, donde el conjunto de datos tiene un gran número de características o variables de entrada. Esto gracias a que encontró el hiperplano que se muestra en la figura 2, con una función lineal, que maximiza el margen entre las clases, mejorando la generalización, a pesar de que los datos de entrenamiento a menudo no son linealmente separables en su espacio original. Esto significa que no se puede trazar una línea recta (o un hiperplano en dimensiones superiores) que separe perfectamente las clases, en este caso, plazas libres y ocupadas.

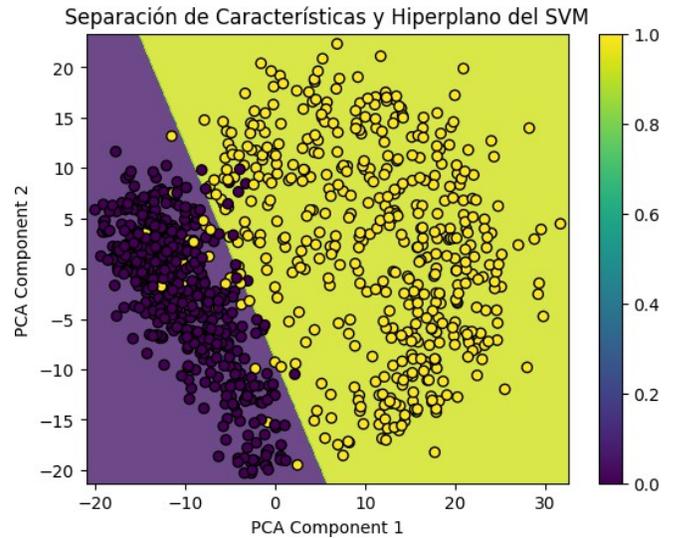


Fig. 2. Hiperplano del clasificador SVM

Además, aunque el entrenamiento puede ser computacionalmente costoso, la predicción se realiza en un tiempo de 0.86 segundos resultando ser rápida, tal como se muestra en la figura 3, lo que es crucial para sistemas en tiempo real. El SVM funciona bien con un conjunto de datos pequeños disponible para el escenario específico como el monitoreo de plazas de estacionamiento, controlando el sobreajuste mediante regularización. También pudo manejar clases desbalanceadas gracias al ajuste de los pesos para diferentes clases, dado que la detección de plaza libre u ocupada es un problema de clasificación binaria.

```
import time

def medir_tiempo_clasificacion(modelo, datos):
    start_time = time.time()
    prediccion = modelo.predict(datos)
    end_time = time.time()
    tiempo_transcurrido = end_time - start_time
    return tiempo_transcurrido, prediccion

# Ejemplo de cómo usar la función para medir el tiempo de clasificación
tiempo_prediccion, predicciones = medir_tiempo_clasificacion(svm_model, validation_features)
print(f"Tiempo de clasificación: {tiempo_prediccion} segundos")
```

32/32 [=====] - 0s 5ms/step
Tiempo de clasificación: 0.8604607582092285 segundos

Fig. 3. Tiempo de la clasificación con el SVM

De igual manera, se obtiene un promedio similar, al hacer una medición del tiempo que se demora en clasificar el SVM en 20 intentos tal como se muestra en la Tabla I

TABLA I
TIEMPO DE CLASIFICACIÓN EN 20 SEGUNDOS

Intento	Tiempo de Clasificación (segundos)
1	1.030437708
2	0.975208759
3	0.904649973
4	1.055740356
5	0.936021805
6	0.909917593
7	0.992678404
8	1.342455626
9	1.479554653
10	1.307539701
11	1.160335302
12	0.995288372
13	0.629985809
14	0.75925684
15	0.883471251
16	0.630591393
17	0.745626211
18	0.763018847
19	0.591813326
20	0.767165184

Continuo a ello, el etiquetado preciso de las imágenes dentro de plaza libre como ocupada, mejoro en parte la exactitud y rendimiento del modelo, y aseguro la adaptación a diversas condiciones de iluminación, clima y ocupación, así como a las características particulares del estacionamiento. Esto redujo el ruido y optimizo los recursos computacionales, facilitando un entrenamiento más eficiente. Además, mejoro la generalización del modelo al cubrir casos específicos y balancear adecuadamente las clases de datos como se muestra en la figura 4.

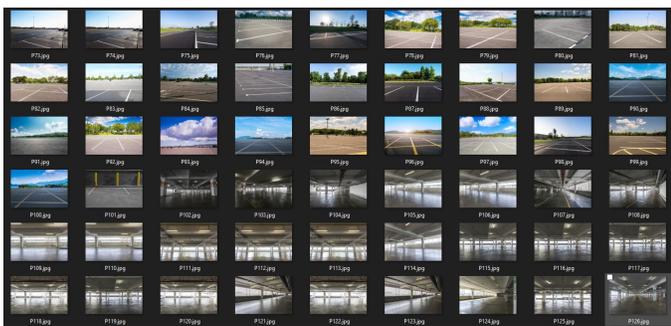


Fig. 4. Diversidad de iluminación en la base de datos

B. Resultados de Diseño

En esa etapa se da a conocer los resultados de diseñar y definir bien los hiperparámetros de un sistema de detección de plazas libres, desde la optimización del sistema al reducir falsos positivos y negativos, se mejoró el rendimiento y la confiabilidad del modelo. Esto se traduce en una mayor eficiencia operativa, con ahorro de recursos computacionales y tiempos de respuesta más rápidos, lo cual puede mejorar la experiencia del usuario al mostrarse en la interfaz las plazas libres, reduciendo el tiempo y la frustración de los conductores. Además, el sistema al ser afinado y tener bien claro los hiperparámetros que están en la Tabla II, se puede adaptar y escalar, permitiendo la flexibilidad ante cambios y expansión sin comprometer el rendimiento, lo que reduce costos operativos y de infraestructura. Finalmente, facilita la integración con otros sistemas de gestión y automatización del estacionamiento, promoviendo la interoperabilidad y mejorando la comunicación de datos.

TABLA II
PARÁMETROS UTILIZADOS PARA EL ENTRENAMIENTO

Hiperparámetros	Valor	Descripción
img_width	390	Ancho de las imágenes de entrada en píxeles.
img_height	260	Alto de las imágenes de entrada en píxeles.
batch_size	16	Tamaño del lote utilizado en el entrenamiento.
rotation_range	40	Rango de rotación en grados para el data augmentation.
width_shift_range	0.2	Rango de desplazamiento horizontal para el data augmentation.
height_shift_range	0.2	Rango de desplazamiento vertical para el data augmentation.
shear_range	0.2	Rango de operaciones de corte para el data augmentation.
zoom_range	0.2	Rango de operaciones de zoom para el data augmentation.
horizontal_flip	True	Aplicación de volteo horizontal en el data augmentation.
fill_mode	'nearest'	Modo de llenado para los píxeles fuera del límite después de transformaciones.
svm_kernel	'linear'	Tipo de kernel utilizado en el clasificador SVM.
train_accuracy	Calculated train accuracy	Precisión del modelo SVM en datos de entrenamiento.
validation_accuracy	Calculated validation accuracy	Precisión del modelo SVM en datos de validación.

Por otra parte, el diagrama de flujo del funcionamiento del sistema proporcione claridad visual sobre cada etapa del proceso, facilitando la identificación y resolución de problemas durante el desarrollo y la optimización del sistema. Además, se usó como documento crucial para la comunicación y colaboración entre equipos, mejorando la comprensión y el análisis del desempeño del sistema. La estructura del diagrama también ayudo en la integración efectiva de la programación con otro diagrama de flujo, desde el entrenamiento del modelo de detección hasta la captura del entorno del estacionamiento, facilitando la comprensión del flujo de trabajo y permitiendo una depuración sistemática y una optimización precisa. Además, se estructuro modularmente el desarrollo, permitiendo ajustes específicos en cada fase, desde el preprocesamiento de datos hasta la evaluación del rendimiento, asegurando una integración fluida y eficaz, facilitando la identificación y eliminación de cuellos de botella, redundancias y áreas de mejora como se ve en la figura 5.

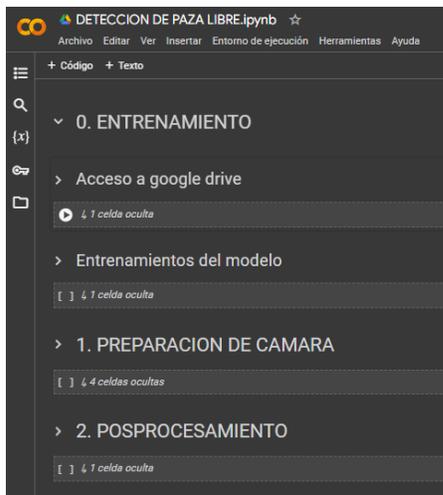


Fig. 5. Secuencia de diseño

C. Resultados de integración

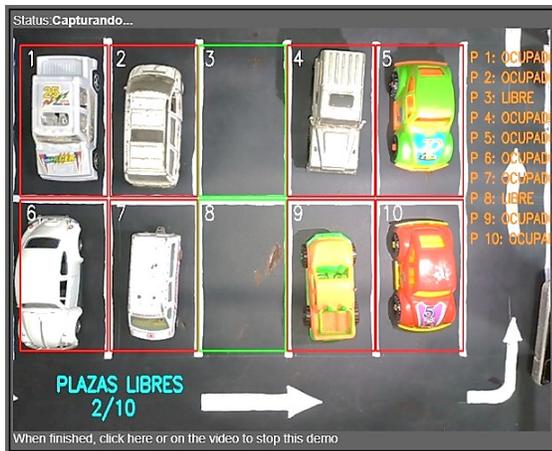


Fig. 6. Detección del primer escenario

En la segunda detección se realizó en un escenario oscuro con iluminación, simulando una detección del sistema en la noche, como se ve en la figura 7, los 4 vehículos expuestos a la luz son detectados con normalidad, sin embargo, los vehículos alejados de la luz no son detectados, en consecuencia, el contador de plaza libre es incorrecta.

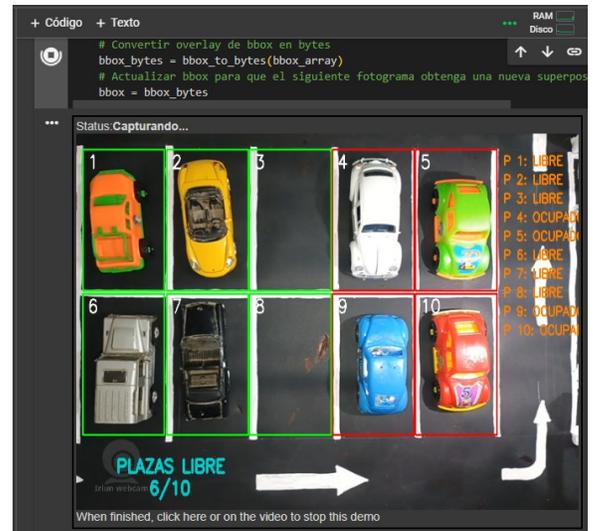


Fig. 7. Detección del segundo escenario

D. Resultados de Validación

Siguiendo la idea de la programación, durante la integración se obtuvieron aspectos importantes del sistema de detección de plaza, pero como punto de partida, utilizando la fórmula de la figura 8 y los datos de una matriz de confusión, el modelo presenta una precisión de 0.901, esto quiere decir, que de todas las veces que el modelo predice que una instancia es positiva, el 90.1% de esas predicciones son realmente correctas.

$$Precisión = \frac{VP}{VP + FP} = \frac{476}{476 + 52} * 100 = 90.15\%$$

Fig. 8. Formula de precision

Posterior a la sustitución de los datos de la matriz de confusión en la fórmula de la figura 9, se obtiene una exactitud del 0.924, esto significa que el modelo está haciendo predicciones correctas en el 92.4% de los 10 plazas libre sobre el conjunto de datos en el que se está evaluando. En términos simples, de todas las predicciones que el modelo ha hecho, el 92.4% son correctas.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} = \frac{476 + 448}{476 + 52 + 448 + 24} * 100 = 92.4\%$$

Fig. 9. Formula de exactitud

Como primer escenario, el sistema de detección de plaza libre se implementó en un día soleado con 8 vehículos, en un prototipo de estacionamiento rectangular de 80 cm de largo y 60 cm de ancho, además, se utilizó un dispositivo móvil como cámara web con buena calidad en la parte superior y central de las plazas con un soporte, tal como se ve en la figura 10.



Fig. 10. Primer escenario

Como respuesta de la implementación en el escenario planteado, se tiene una exactitud de 0.896, ligeramente inferior a la precisión, indica que el sistema tiene un buen balance entre las predicciones correctas de plazas ocupadas y libres, aunque hay un margen de error que podría mejorarse, ya que, la plaza número 10 del primer tiempo representado por el color rosa cambia su predicción en el segundo tiempo representado por el color azul. Sin embargo, la precisión es de 0.943, lo cual indica que una gran proporción de las predicciones de plazas ocupadas son correctas tal como se muestra en la figura 11.

Posicion	Prediccion	Confianza	Brillo	Precision	Exactitud
1	OCUPADO	0.530351996	147.2519067	0.943478261	0.896491228
2	OCUPADO	0.647554576	139.6663649	0.943478261	0.896491228
3	LIBRE	0.059135959	127.4879561	0.943478261	0.896491228
4	OCUPADO	0.683042645	102.9549246	0.943478261	0.896491228
5	OCUPADO	0.835593224	130.8866941	0.943478261	0.896491228
6	OCUPADO	0.672602177	131.2888615	0.943478261	0.896491228
7	OCUPADO	0.689537585	129.9605213	0.943478261	0.896491228
8	LIBRE	0.105195068	106.5986831	0.943478261	0.896491228
9	OCUPADO	0.652790189	116.8697668	0.943478261	0.896491228
10	LIBRE	0.472084999	123.602524	0.943478261	0.896491228
1	OCUPADO	0.531239212	147.2019204	0.943478261	0.896491228
2	OCUPADO	0.640786469	139.5925377	0.943478261	0.896491228
3	LIBRE	0.057080571	127.400439	0.943478261	0.896491228
4	OCUPADO	0.659910262	102.8780796	0.943478261	0.896491228
5	OCUPADO	0.717309713	130.6840878	0.943478261	0.896491228
6	OCUPADO	0.675262809	131.2718244	0.943478261	0.896491228
7	OCUPADO	0.697544456	129.9138272	0.943478261	0.896491228
8	LIBRE	0.116562068	106.5603567	0.943478261	0.896491228
9	OCUPADO	0.669842362	116.7967901	0.943478261	0.896491228
10	OCUPADO	0.513403714	123.4721811	0.943478261	0.896491228

Fig. 11. Primer escenario

La variabilidad en el brillo es mayor de 100 como se ve en la figura 12, por lo que el sistema manejar bien manteniendo una alta precisión y exactitud, ya que las condiciones de luz afectan la imagen capturada. Por otra parte, las variaciones en la confianza indican que el sistema está menos seguro en algunas predicciones, especialmente en las plazas libres. Esta variabilidad podría deberse a factores como las sombras, o diferencias en los vehículos estacionados.



Fig. 12. Brillo del primer escenario

Como segundo escenario, se utilizó 8 vehículos de distintos colores distribuidos en las plazas de estacionamiento, menos en el número 3 y 8, además el prototipo se implementará en la oscuridad con iluminación, simulando una detección en la noche. Como resultado, la baja precisión de 0.281 y exactitud de 0.488 que se muestra en la figura 13 indica que una proporción significativa de las predicciones de plazas ocupadas no son correctas, además sugiere un balance deficiente en las predicciones correctas de plazas ocupadas y libres.

Posicion	Estado	Prediccion	Confianza	Brillo	Precision	Exactitud
1	OCUPADO	LIBRE	0.075249851	75.3829904	0.28125	0.4888889
2	OCUPADO	LIBRE	0.197430685	103.9387929	0.28125	0.4888889
3	LIBRE	LIBRE	0.08595667	85.15078189	0.28125	0.4888889
4	OCUPADO	OCUPADO	0.874705672	158.4986283	0.28125	0.4888889
5	OCUPADO	LIBRE	0.396426916	132.2820576	0.28125	0.4888889
6	OCUPADO	LIBRE	0.151859552	89.53462277	0.28125	0.4888889
7	OCUPADO	LIBRE	0.037099659	74.93245542	0.28125	0.4888889
8	LIBRE	LIBRE	0.187216565	104.9288066	0.28125	0.4888889
9	OCUPADO	OCUPADO	0.686181843	133.0457064	0.28125	0.4888889
10	OCUPADO	OCUPADO	0.629551172	142.2009602	0.28125	0.4888889
1	OCUPADO	LIBRE	0.068603784	73.8514952	0.28125	0.4888889
2	OCUPADO	LIBRE	0.17723988	101.7942387	0.28125	0.4888889
3	LIBRE	LIBRE	0.084011152	84.50611797	0.28125	0.4888889
4	OCUPADO	OCUPADO	0.884549201	160.183594	0.28125	0.4888889
5	OCUPADO	OCUPADO	0.501126409	138.5961043	0.28125	0.4888889
6	OCUPADO	LIBRE	0.132804275	87.23508916	0.28125	0.4888889
7	OCUPADO	LIBRE	0.031192409	72.2066118	0.28125	0.4888889
8	LIBRE	LIBRE	0.144909352	100.2058436	0.28125	0.4888889
9	OCUPADO	OCUPADO	0.675992727	132.8676818	0.28125	0.4888889
10	OCUPADO	OCUPADO	0.6470927	144.4855693	0.28125	0.4888889

Fig. 13. Segundo escenario

La variabilidad del brillo que oscila entre 72.20 y 160.18 en ambos tiempos. refleja las diferentes condiciones de iluminación durante la noche, donde no hay una correlación clara entre el brillo y la precisión de las predicciones, sugiriendo que la variación en brillo puede estar afectando la capacidad del sistema para hacer predicciones precisas. Por otro lado, la gran variabilidad en la confianza que varía considerablemente, desde tan baja como 0.03 hasta tan alta como 0.88., sugiere que el sistema es menos seguro en algunas predicciones, especialmente para plazas libres. Las predicciones con alta confianza son generalmente más precisas, pero las predicciones con baja confianza son menos fiables.

IV. CONCLUSIONES

A lo largo de la investigación, se realizaron pruebas que incluyeron la recopilación de datos, ajuste de modelos de aprendizaje profundo y evaluación de la precisión del sistema. Los resultados demuestran que el sistema de visión artificial basado en redes neuronales convolucionales para la detección de plazas libres en un estacionamiento privado es efectivo y cumple con los requisitos de precisión y eficiencia en tiempo real. El diseño se desarrolló mediante transferencia de aprendizaje, validando así la hipótesis de que se puede crear un sistema eficaz para esta tarea.

La combinación de VGG16 y SVM ha proporcionado un modelo robusto, ya que se ha podido extraer las características relevantes de una base de datos limitada para entrenar el clasificador SVM y, junto al hardware utilizado, ha sido suficiente para procesar los datos en tiempo real y lograr una clasificación en un tiempo de 0.86 segundos. Además, la configuración de la estructura informática no solo reduce la complejidad del proceso de entrenamiento, sino que también permite una mayor flexibilidad y adaptabilidad del sistema en diferentes entornos y condiciones operativas. Por lo tanto, se valida la primera hipótesis específica de que estos requerimientos son esenciales para el desarrollo del sistema.

La simplicidad en la configuración de hiperparámetros facilitó la implementación y ajuste del modelo, además, el uso de diagramas de flujo ha facilitado la compilación y adaptación del modelo a nuevos datos, mostrando alta precisión en la detección de plazas libres. La documentación clara y detallada ha permitido un proceso de desarrollo más robusto y escalable. Por lo tanto, se valida la segunda hipótesis específica de que un diseño bien estructurado y documentado mejora la facilidad de compilación y la adaptabilidad del sistema.

La integración del módulo de captura de imágenes, procesamiento de datos en tiempo real y una interfaz de usuario eficiente ha permitido una gestión efectiva de los espacios disponibles en el estacionamiento. El sistema ha demostrado ser capaz de capturar, procesar y visualizar los resultados de manera eficiente. Además, el uso de técnicas de transferencia de aprendizaje ha sido particularmente efectivo en este contexto, permitiendo aprovechar modelos preentrenados en grandes conjuntos de datos y adaptarlos a nuestra aplicación específica con un conjunto de datos más reducido. Este enfoque no solo aceleró el proceso de desarrollo, sino que también mejoró significativamente la precisión del sistema de detección de plazas. Por lo tanto, se valida la cuarta hipótesis específica de que la integración de estos componentes es crucial para el funcionamiento del sistema.

La validación del sistema, realizada mediante la comparación de las predicciones del modelo con datos de referencia en tiempo real, ha demostrado que el sistema desarrollado alcanza una precisión del 90.15% y una exactitud del 92.4%.

Asimismo, el sistema ha sido capaz de detectar plazas libres en un tiempo promedio de 1 segundo, logrando una eficiencia del 90% en condiciones de iluminación y con diferentes tipos de vehículos, lo que subraya la robustez y fiabilidad del modelo. Además, este nivel de eficiencia es comparable al de otros sistemas de visión artificial más complejos que tienen acceso a conjuntos de datos mucho más extensos. Por lo tanto, se valida la última hipótesis específica, que establece que el sistema cumple con los estándares de precisión y eficiencia esperados.

REFERENCIAS

- [1] R. Corrales, M. Silvera, y F. Campos, "Evaluation of the microscopic behavior of motorcycle taxis by adapting the SSAM model at an urban intersection", 2022 Congreso Internacional de Innovación y Tendencias en Ingeniería, CONIITI 2022 - Conference Proceedings, 2022, doi: 10.1109/CONIITI57704.2022.9953701.
- [2] H. Brožová y M. Růžicka, "The prediction of parking space availability", *Transport*, vol. 35, no 5, pp. 462–473, dic. 2020, doi: 10.3846/TRANSPORT.2020.14016.
- [3] A. N. Abbood, A. R. I. Ahmed, y H. K. K. Ajam, "Evaluation of parking demand and future requirement in the urban area", *Civil Engineering Journal (Iran)*, vol. 7, no 11, pp. 1898–1908, nov. 2021, doi: 10.28991/CEJ-2021-03091767.
- [4] L. Gao, W. Fan, Z. Hu, y W. Jian, "Prediction of Vacant Parking Spaces in Multiple Parking Lots: A DWT-ConvGRU-BRC Model", *Applied Sciences (Switzerland)*, vol. 13, no 6, mar. 2023, doi: 10.3390/APP13063791.
- [5] Y. Zou, F. Ye, A. Li, M. Munir, E. Hjelseth, y S. F. Sujan, "A Digital Twin prototype for smart parking management", *eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 14th European Conference on Product and Process Modelling, ECPPM 2022*, pp. 250–257, 2023, doi: 10.1201/9781003354222-32.
- [6] R. Bassam y F. Samann, "Smart Parking System based on Improved OCR Model", *IOP Conf Ser Mater Sci Eng*, vol. 978, no 1, dic. 2020, doi: 10.1088/1757-899X/978/1/012007.
- [7] F. Zhang, C. Li, y F. Yang, "Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation", *Sensors (Switzerland)*, vol. 19, no 3, feb. 2019, doi: 10.3390/S19030594.
- [8] W. Chen, Q. Liu, H. Hu, J. Liu, S. Wang, y Q. Zhu, "Novel laser-based obstacle detection for autonomous robots on unstructured terrain", *Volume 20, Issue 18, Pages 1 - 18*, vol. 20, no 18, pp. 1–18, ene. 5048, doi: 10.3390/s20185048.
- [9] C. Geng, W. Shi, Z. Liu, H. Xie, y W. He, "Nondestructive Surface Crack Detection of Laser-Repaired Components by Laser Scanning Thermography", *Volume 12, Issue 11*, vol. 12, no 11, ene. 5665, doi: 10.3390/app12115665.
- [10] V. Paidi y H. Fleyeh, "Parking occupancy detection using thermal camera", *VEHITS 2019 - Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, pp. 483–490, 2019, doi: 10.5220/0007726804830490.
- [11] H. M. Al-Barhamtoshy, K. K. Abdullah, M. K. Dauda, y T. F. Himdi, "Detecting Available Parking Spaces in Smart Cities", *1st International Conference in Advanced Innovation on Smart City, ICAISC 2023 - Proceedings, 2023*, doi: 10.1109/ICAISC56366.2023.10085386.
- [12] A. A. Naufal, C. Fatichah, y N. Suciati, "Preprocessed Mask RCNN for Parking Space Detection in Smart Parking Systems", *International Journal of Intelligent Engineering and Systems*, vol. 13, no 6, pp. 255–265, 2020, doi: 10.22266/IJIES2020.1231.23.