Smart Parking: Sistema Inteligente para la Automatización y Supervisión de Parqueaderos

Daulin Riascos1 , Andrés Hurtado 2, Juan Cuero3

*Resumen*—La presente investigación busca mediante el reconocimiento de placas de vehículos lograr una herramienta de visión artificial que facilite las labores de gestión de los parqueaderos. El desarrollo del artículo se basó en la metodología para la revisión bibliográfica que sin duda fue fundamental para establecer un prototipo basado en OpenCV y Pytesseract versión 5. Su arquitectura incorpora retención adaptativa y modelo de red neuronal de memoria a corto plazo (LSTM) lo cual permitió alcanzar resultados con una exactitud del 90% sobre imágenes tomadas en un rango no superior a los 4 mts.

Palabras claves: Reconocimiento de caracteres, Visión artificial OpenCV , Pytesseact.

*Abstract*—The present investigation seek through the reconigtion of vehicle plates to achieve an artificial vision tool that facilitates the management of parking lots. The development of the article was based on the methodology for the bibliographic review that was undoubtedly essential to establish a prototype based on openCV and Pytesseract version 5. Its architecture incorporates adaptive retention and a neural network model of short-term memory (LSTM) which allowed to achieve results with anaccuracy of 90% on images taken at a range not exceeding 4 meters.

Keywords: Character recognition, artificial vision, openCV, Pytesseract.

# I. INTRODUCTION

L

a Inteligencia Artificial es un concepto que se ha venido posicionando en la esfera mundial, a partir del desarrollo tecnológico y por ende con la implementación de diferentes herramientas que en este sentido son constantemente promovidas en el mundo. La Inteligencia Artificial se presenta como el concepto macro del presente estudio, por medio del cual se busca diseñar un sistema para la automatización y supervisión de parqueaderos que logre favorecer a la población civil del Distrito de Buenaventura que cotidianamente acuden al centro de la ciudad para realizar diferentes labores socioeconómicas, dinamizadoras de la movilización colectiva ahí suscitada.

Para mayor comprensión del texto, se presenta al lector el abordaje teórico sobre la Inteligencia Artificial de manera que se logre avanzar hacia la comprensión de la necesaria utilización de estas tecnologías en el desarrollo humano, significando que su empleo cada vez se magnifica en el mundo.

Consecuentemente, se indica que el contexto de interés investigativo es el centro del Distrito de Buenaventura, escenario donde cotidianamente convergen múltiples ciudadanos, con la finalidad de llevar a cabo actividades económicas, políticas, educativas, laborales, médicas y comerciales por lo cual, se requiere contar con estacionamientos adecuados donde implementen la respectiva automatización, destacando así la manera como se evidencia a nivel internacional y nacional la población civil del Distrito de Buenaventura cada vez se adscribe al consumismo de vehículos particulares, lo que genera la necesidad de hacer efectivo el sistema de parqueo.

La detección y reconocimiento de placas vehiculares surge con la necesidad de garantizar el monitoreo y control de acceso a los parqueaderos y apoyar al proceso de vigilancia, sin embargo, en Buenaventura el registro más común de acceso de vehículos a estacionamientos es el acceso a través de un ticket generado por un personal de seguridad, el cual es válido para su entrada y verificado en la salida del mismo.

Esta forma tradicional de ingreso a un parqueadero mediante registro manual, conlleva al incremento del tiempo de espera en el acceso al estacionamiento generando aglomeración vehicular para los automóviles que ingresarán (formando colas y congestión), y un control limitado del aforo vehicular para los automóviles dentro del estacionamiento (generando problemas en el parqueo de vehículos). También ocasiona problemas de seguridad al no quedar registrado el propietario del vehículo, adicional se pueden generar daños por parte de conductores que tienen malas prácticas de parqueo.

En esta investigación se hace uso de la inteligencia artificial aplicada en técnicas de visión artificial para la detección de placas vehiculares con el objetivo de automatizar los procesos de los parqueaderos del centro de Buenaventura

También se consideró pertinente abordar algunas normas que a nivel internacional y nacional regulan el tema de la Inteligencia Artificial, donde el lector observe que es un aspecto regulado por los Estados para la debida supervisión de su aprovechamiento sin generar perjuicios humanos, ambientales y sociales.

Después se establece el respectivo estado del arte como aspecto fundamental donde, a partir de la revisión bibliográfica previa se logró evidenciar la existencia de investigaciones realizadas en diferentes contextos pero que, a su vez, se relacionan estrechamente con el estudio adelantado mostrando la necesidad de diseñar e implementar parqueaderos inteligentes en otros escenarios sociales.

Consecuentemente, se hace mención de la metodología investigativa donde se plantea una investigación con enfoque en la revisión bibliográfica como técnicas para ser empleadas en el campo de estudio.

Finalmente, se describen detalladamente tanto los resultados obtenidos como las conclusiones a que se llegaron con la implementación del algoritmo.

II. METODOLOGÍA

La presente investigación se desarrolló bajo la metodología de revisión bibliográfica con el propósito de identificar las diferentes herramientas y arquitecturas usadas para el reconocimiento de caracteres.

La primera etapa fue la búsqueda de estudios relacionados con la aplicación de la visión artificial en la detección y reconocimiento de placas vehiculares ejecutadas durante los años 2017 – 2022 utilizando palabras claves como Smart parking, intelligence parking, vehicle recognition, sistema de parqueo inteligente, visión artificial, algoritmo, reconocimiento de placas, ciudades inteligentes en las bases de datos de lareferencia, Dialnet y Google académico. En total se analizaron más de 30 trabajos entre artículos, tesis teóricas, sitios web y demás fuentes necesarias sobre el tema en diferentes contextos.

Al cotejar los artículos académicos extraídos de la búsqueda inicial se exploraron las arquitecturas de visión artificial utilizadas por los investigadores con el objetivo de identificarlos y clasificarlos.

Posterior al proceso anterior, se escogieron los documentos más representativos y con mayor aporte a la solución de la problemática planteada en el artículo. en base a esto se organizó la segunda etapa de análisis de las arquitecturas utilizadas para reconocimiento y detección de placas vehiculares.

# III. DISCUSIONES Y RESULTADOS

Acudiendo al enfoque teórico de Corvalán (2017), la Inteligencia Artificial es mencionada como la “cuarta revolución industrial”. Por medio de su implementación, el mercado de las grandes empresas e industrias ha logrado potenciar su capacidad a nivel mundial, extendiendo sus aforos de oferta y demanda llegando a todo tipo de público y contexto en tiempo real sin importar las brechas geográficas que en el mercado tradicional significaron restricciones para la ampliación de las capacidades.

El machine Learning hace referencia a los sistemas que, mediante unas operaciones matemáticas son capaces de reconocer patrones en los datos, por otro lado, el Deep Learning es una disciplina específica del Machine Learning que incluye redes neuronales en capas consecutivas para aprender de los datos de forma iterativa. Garzón (2020).

La visión artificial son métodos que ayudan a adquirir, procesar y analizar una imagen convirtiéndola en información numérica para ser procesada en una computadora, Puente (2021). Igualmente, Gabriela (2017) indica que es un conjunto de métodos, técnicas y teorías la cual permite simular los procesos biológicos del ser humano de separar y analizar de forma automática la información de imágenes vistas.

OpenCV es una librería de visión artificial y de código abierto, fue desarrollada por INTEL y actualmente es la más utilizada en el procesamiento de imágenes, del mismo modo también se usa para la detección de movimientos, reconocimiento de objetos, robótica, reconocimiento de rostros, etc. Guadalupe (2017). Del mismo modo Marín (2020) afirma que es una multiplataforma compatible con lenguajes de programación como Python, Java y C++ y cuenta con más de 2500 algoritmos que se encuentran disponibles para fines académicos y comerciales.

Una CNN se podría definir como un grupo de redes que mediante un entrenamiento son capaces de reconocer variables de entradas eficientemente, como por ejemplo audios, patrones de imagen, entre otros. Bagnato (2020).

Pytesseract es un motor con licencia libre que se utiliza para el reconocimiento óptico de caracteres (OCR), soporta una gran variedad de lenguajes y puede ser empleado para extraer texto fijado en las imágenes o en algo más sencillo como reconocer una línea de palabras Zelic (2021), también destaca sobre algunas de las funciones que ofrece y es la de obtener la orientación de una imagen de un documento y a su vez conseguir el texto presente en ella.

A continuación, se resumen algunas investigaciones que aportan soluciones en el contexto global de parqueaderos inteligentes.

Ramírez (2020), utiliza una CNN con unidad lineal rectificada (ReLU), la agrupación máxima, fullyConnected, la capa softmax y la capa de clasificación para la detección de placas vehiculares.

Fernández y Quinatoa (2017), implementaron un prototipo basado en dos etapas, la primera consistía en localizar la placa con los métodos de procesamiento de imágenes tales como: (Escala de grises, Umbralización, Detección de contornos) y la segunda en el reconocimiento de la placa vehicular mediante la segmentación de caracteres, además trabajaron con el lenguaje Python usando el IDE JetBrains PyCharn Coomunity que proporciona varias librerías entre las más importante OpenCV para procesamiento de imágenes y la utilización de algoritmos de Machine Learning de los cuales se empleó el método KNearest\_create().

No obstante Yilmaz, Güzel , Askerbeyli y Bostanci (2018), proponen un prototipo para entrenar un detector de vehículos usando métodos de aprendizaje profundo R-CNN, FASTER R-CNN , el desarrollo consta de 6 etapas que son: Carga del conjunto de datos, diseño de la red neuronal convolucional, configuración de opciones de entrenamiento, entrenamiento del detector de objetos Faster R-CNN y evaluación del detector entrenado y finalmente realizar una comparación de análisis experimentales con los resultados obtenidos.

Por otro lado, Erazo (2019), implementó un prototipo para detección de plazas de parqueo libres mediante la utilización de la librería OpenCV, el descriptor Haar y el microcontrolador Raspberry pi 2. De forma semejante Beaucamps (2018), desarrolló un software con lenguaje de programación Java haciendo uso de la librería OpenCV, el descriptor Haar y Adaboost como método de clasificación de los datos, para el entrenamiento utiliza 100 imágenes tomadas a los coches de un parking.

Así mismo Rojas (2017), propone implementar un sistema web empleando técnicas de visión artificial con lenguaje Java aplicando Tesseract OCR y ANPR para el reconocimiento de placas de vehículos robados.

Por último, Arnedo, Caicedo y Fuentes (2018), desarrollaron prototipos mediante la librería OpenALPR la cual funciona en dos etapas principales: detección de matrículas y reconocimiento de caracteres. Para detectar las placas utiliza el clasificador Haar de OpenCV y para el reconocimiento de caracteres Pytesseract, ambas herramientas deben ser entrenadas con imágenes de placas Orellana y Ortega (2019).

Cada una de las investigaciones y artículos relacionados tiene la característica de proponer una arquitectura para satisfacer necesidades relacionadas con la implementación de estacionamientos automatizados que respondan no solo al reconocimiento de placas sino también, al aprovechamiento de las tecnologías diseñadas para esta finalidad.

Finalmente se puede concluir que el campo de estudio de esta investigación es un tema de mucho impacto en el ámbito nacional e internacional pues es abordado por diferentes autores que implementaron diferentes soluciones y en diferentes contextos aplicando inteligencia artificial mediante el Deep Learning.

En este orden de ideas con el propósito de validar el estado actual de los procesos de aplicación de la CNN con Tesseract se identificaron los siguientes trabajos relacionados Qadri y Asif (2019), señalan como hacer el reconocimiento de placas vehiculares mediante una simulación del sistema ANPR, o dicho de otro modo como reconocimiento automático de placas. Los autores toman la imagen con una cámara conectada por medio del puerto usb, el procedimiento usado por ellos es, en primer lugar, la captura de la imagen, buscar pixeles amarillos, después hacen filtrado de imagen, realizan extracción de la región de la placa, separan líneas empleando segmentación de filas y por ultimo extracción de caracteres utilizando segmentación de columna para aplicar reconocimiento de caracteres.

También Erazo (2019), diseñó un prototipo para detectar puestos libres de un parking utilizando OpenCV para realizar procesamientos de las imágenes y Raspberry PI 3 para conectar las cámaras de video y conectarse a la red Ethernet, luego de la implementación del prototipo el autor hizo 12 pruebas distribuidas en diferentes horas del día con variaciones de luminosidad y logró demostrar que el prototipo detectó las plazas libres y también las que estaban ocupadas dentro del estacionamiento.

Igualmente, Beaucamps (2018), utilizó la librería OpenCV con método de clasificación en cascada HAAR y Adaboost y lenguaje de programación Java para detectar vehículos en un parking, así como también las plazas disponibles, el sistema cumple en un 90% el objetivo planteado, pero en algunas ocasiones se queda algún vehículo sin ser detectado.

Del mismo modo Puente (2021), dio uso de las librerías OpenCV, Pytesseract (OCR), Tkinter, pil, método Canny, para el desarrollo del sistema y logró concluir que el sistema reconoce las placas con una eficiencia mayor al 90%.

Por otro lado, en el prototipo realizado por Ramírez (2020), se implementó un modelo CNN conformado por una capa de entrada, ocho capas intermedias y una capa de salida el cual fue entrenado y se comprobó que tiene una efectividad de 95% para el reconocimiento de los caracteres de las placas, se puede establecer que el modelo que utiliza CNN es más preciso que el modelo implementado con OpenCV y OCR.

Así mismo y analizando los aspectos de los resultados alcanzados con la aplicación de Pytsseract, OpenCV, Rojas (2017) logró obtener características relevantes para implementar un sistema de visión artificial que permita la detección y reconocimiento de placas vehiculares para la automatización de procesos en un parqueadero.

Por su parte González y Parrales (2022), aplicaron técnicas de visión artificial mediante el uso de OpenCV donde se definen varias fases entre las cuales resaltan las siguientes: primero realizan la captura del video, seguido de la detección del movimiento vehicular, luego definen una zona de interés para la detección de la placa (Transformación a escala de grises, reducción de ruido, detección de bordes con el método canny), y por último la lectura de la placa con la librería Easy OCR para obtener los caracteres, el sistema se probó con grabaciones de videos y se demostró que reconoce las placas de los vehículos que ingresan al parqueadero de la universidad en un 70% y 80% de confianza. El aporte que hace este trabajo es el de aclarar las diferentes etapas que sigue el algoritmo para reconocimiento de caracteres desde que toma la imagen de entrada y muestra los resultados.

Finalmente, Vaca, Palma, Torres y Granda (2020), emplean la librería OpenCV para procesamiento de la imagen, OCR para reconocimiento de caracteres, método ROI para la obtención de la zona de interés y el clasificador en cascada HAAR. Para la implementación del prototipo utilizaron la metodología ICONIX que tiene un enfoque iterativo e incremental que les permite incorporar nuevas funcionalidades. Una vez terminada la implementación del prototipo lograron demostrar que el sistema en un entorno controlado reconoce los caracteres de las placas con un 95% de precisión sin embargo el nivel de confianza del sistema disminuyó en un 2% cuando había condiciones de poca luz o las placas tenían polvo o estaban muy alejadas.

**Sistema Propuesto.**

El sistema fue implementado con el lenguaje de programación Python usando librerías tales como OpenCV, Pyteseract, Canny, PIL y numpy. Este proceso se compone de 4 etapas, **la primera etapa** consiste en la lectura de un video donde se captura la zona de interés en este caso la placa vehicular**. En la segunda etapa** se almacenan los fotogramas en una variable llamada frame para posteriormente convertirlos a una matriz de píxeles**. El tercer paso** se basa en extraer los píxeles de la zona de la placa. Finalizamos con **la cuarta etapa** donde se transfiere la imagen de la placa extraída en el paso anterior a Tesseract para procesarla y detectar los caracteres alfanuméricos que se encuentran escritos en la placa vehicular

En el siguiente diagrama se describe el proceso de funcionamiento del sistema desarrollado

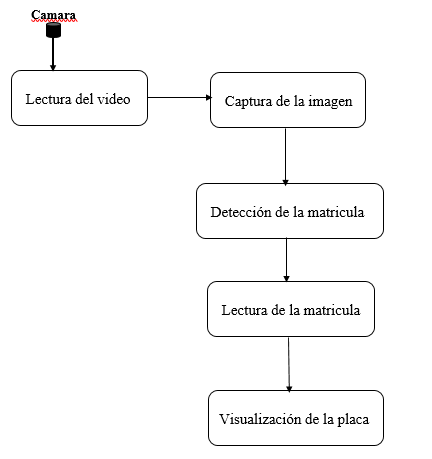


Figura 1: Sistema propuesto reconocimiento matricula

**Lectura del video.**

Para que fuera posible realizar el algoritmo y hacer la prueba en tiempo real, se realizó la grabación de vídeos de varios vehículos particulares al momento en que ingresaban y salían del parqueadero, la toma del video se hace a 70 cms de altura debido a que nos interesa más que todo visualizar la parte frontal y trasera donde está ubicada las placas de los vehículos. El proceso se llevó a cabo utilizando una cámara de una celular marca Xiaomi redmi note 10 con lente de 48 megapíxeles a una resolución de 1280 x 720.

**Captura de la imagen del vehículo.**

Cuando se reproduce el video, se captura una parte específica de la imagen y se calcula el ancho y alto del fotograma y mediante operaciones aritméticas, se determina cual es el centro del video y se dibuja un rectángulo que encierra el área del vehículo a la cual se le llama zona de interés y consiste básicamente en capturar los fotogramas de esa zona en particular, para luego convertirla en una matriz de pixeles y a partir de allí se extraen los pixeles que corresponden a la placa, es decir se recorta una parte de la imagen donde contiene la placa para posteriormente realizar el procesamiento.

**Detección de la matrícula:**

La detección de la matrícula se basa en tres procesos que son:

**Binarización de la imagen:** este proceso consiste en convertir la imagen a escala de grises para facilitar el procesamiento, en este caso utilizamos el método treshold y THRESH\_BINARY que proporciona OpenCv, después de obtener la imagen en binario se procede con operaciones para eliminar el ruido para evitar que se creen falsos contornos.

**Detección de contornos.**

En este paso se detectan y se extraen los contornos de la imagen utilizando el método findContours de OpenCV, esto con el fin de que se cierren los bordes donde está la placa, después de que se detectan todos los contornos se selecciona el que cumpla con las características para este caso se dibujara el contorno que tenga un valor en el área comprendido entre 500 y 5000 que sería la parte donde está la placa. Luego de haber detectado la placa se procede a realizar el proceso inverso, es decir, la matriz con la placa detectada anteriormente se convierte nuevamente a imagen mediante la función Image.fromarray para posteriormente procesarla empleando OCR.

**Lectura de la placa**

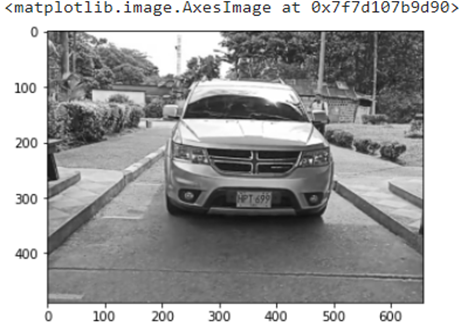
Para poder leer la imagen recortada en los pasos anteriores correspondiente a la placa se usa la librería Pytesseract OCR de Python a la cual se le pasa la imagen de la placa detectada en los pasos anteriores y como resultado arroja los caracteres que están escritos en la misma, después de completar este proceso los caracteres obtenidos son comparados con la información guardada en la base de datos y determinar si el vehículo está registrado en el parqueadero.

**Visualización de los resultados**

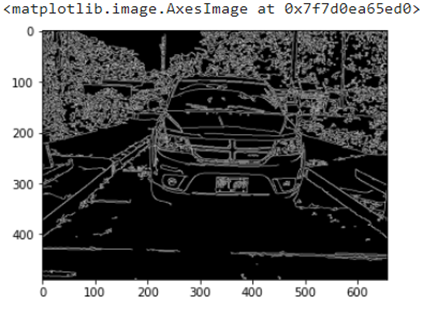
Para visualizar los resultados se utiliza la función cv2.imshow que permite reproducir el video en tiempo real mientras se ejecuta el algoritmo y de esta forma mostrar los resultados.

A continuación se presenta una serie de imágenes donde se detalla el pre procesamiento que lleva a cabo el algoritmo para detectar y reconocer los caracteres de las matriculas vehiculares.

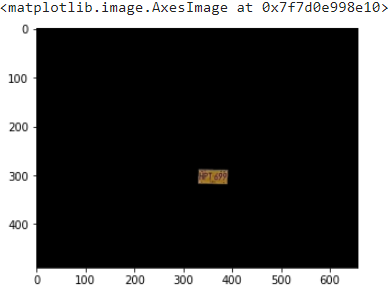


** Figura 2: Cuadro de imagen zona de interés.**

**Figura 3. imagen del vehículo escala de grises**

****

**Figura 4. Detección de contorno del vehículo**

****

**Figura 5. Extracción de placa.**

****

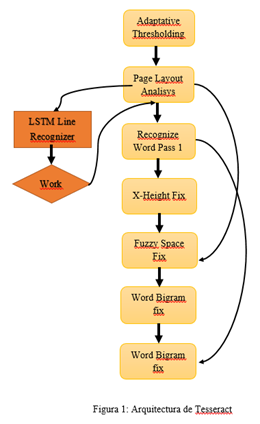
**Figura 6. Marco detectado de placa recortada**

****

**Figura 7. Visualizacion de resultados**

**Arquitectura del sistema Tesseract**

Para la implementación del sistema de detección de las placas se empleó la librería Pytesseract que provee OpenCV en el siguiente diagrama describe la arquitectura del sistema y el proceso que se lleva a cabo para el reconocimiento de caracteres.



**Figura 8. Arquitectura del sistema Tesseract**

Considerando la arquitectura de que Tesseract se puede observar que a la imagen de entrada se le aplica retención adaptativa que producirá una imagen binaria, posteriormente se definen los contornos del texto, luego se organizan esos contornos para formar las palabras y finalmente Tesseract reconocerá la palabra formada y la envía a la salida.

**Pruebas**

El sistema se implementó con el lenguaje Python y para las pruebas se utilizó un laptop con procesador Intel(R) Core(TM) I5-7200U, CPU 2.50 GHz de velocidad, memoria Ram de 8 GB y SO Windows 11.

El algoritmo se va a ejecutar desde la distribución libre de miniconda 3 de Python versión 3,97 docs.conda.io (2022) y el código estará alojado en el repositorio público de github (<https://github.com/Dayoripa/RecognitionPlate> ) para garantizar disponibilidad en cualquier momento y además para poder ser mejorado aprovechando que el sistema github permite trabajar en equipo y llevar el control de versiones.

Para los datos de entrada se tomó un video con celular el cual disponía de una cámara de 16 MP, se registraron las placas de 20 vehículos ingresando al parqueadero de las cuales el sistema detecta 16 de forma eficiente, con un porcentaje de efectividad del 80%, además el algoritmo con relación a la Base de datos determina si el propietario o conductor está registrado en el parqueadero, así como también a cuanto equivale el valor del parqueo.

La arquitectura propuesta en este artículo solo reconoce los caracteres de las placas de carros particulares con fondo amarillo y cuando el vehículo se encuentra a una distancia determinada, para este caso de 4 metros.

La implementación del sistema en un ambiente real presento algunos problemas afectando así el desempeño del algoritmo, principalmente las dificultades ocurrieron porque no se había definido a que distancia y altura debe ser ubicada la cámara para que enfoque el número de la placa y poder ser identificada por sistema tesseract. Otro de los inconvenientes presentados fue cuando existían placas con algún grado de deterioro ocasionando que el sistema no pudiera identificar correctamente los caracteres.

Además, se evidencio que el algoritmo presentó inconvenientes para distinguir entre los caracteres “I”, “1”, debido a que estos poseen patrones parecidos, así mismo en ocasiones no lograba identificar correctamente las letras “Q”, “O”, “D”.

En los resultados de la investigación se puede apreciar que la aplicación funciona bastante bien, 80% de efectividad es un porcentaje alto teniendo en cuenta que no se tiene ayuda para controlar el entorno. Vaca, Palma, Torres y Granada (2020), utilizaron OpenCV, OCR y el clasificador HAAR para el diseño de un prototipo que tenía una eficiencia del 95% para reconocimiento de caracteres si se implementa en un entorno controlado y un 92% si había condiciones de poca luz, se evidencio que el modelo de los autores mencionados anteriormente tuvo mejores resultados que el propuesto en el presente artículo.

De igual forma Ramírez (2020), implementó una CNN conformado por una capa de entrada, ocho capas intermedias y una capa de salida la cual fue entrenado y se comprobó que tiene una efectividad de 95% para el reconocimiento de los caracteres de las placas, se puede establecer que el modelo que utiliza CNN es más preciso que el modelo implementado con OpenCV y OCR, por consiguiente, para una futura oportunidad se podría implementar un prototipo basado en una red neuronal convolucional que de seguro se tendría mejores resultados como lo expone Ramirez.

IV. CONCLUSION

Se logró desarrollar un sistema de detección y reconocimiento de placas de vehículos en parqueaderos mediante el uso de visión artificial donde se usa OpenCV para la detección de placas, y Tesseract para el reconocimiento de los caracteres. El sistema fue probado en algunos parqueaderos consiguiendo una eficiencia del 80%. Es preciso señalar que el algoritmo fue probado en grabaciones durante el día aprovechando la luz natural y excluyéndolo de condiciones climáticas adversas.

En conclusión, se puede decir que el resultado fue exitoso porque se consiguió cumplir los objetivos planteados en un inicio. Para ello fue necesario analizar diferentes investigaciones y trabajos relacionados con el objeto de estudio del presente artículo para adquirir el conocimiento necesario a cerca de las técnicas y librerías de visión artificial utilizadas en la solución de problemas similares y en base a eso poder construir la arquitectura del prototipo utilizado.

Aunque se obtuvieron resultados exitosos se considera que el sistema necesita mejorarse en algunos aspectos. Uno de ellos podría ser que pueda reconocer placas sin importar el color de fondo que tengan puesto que solo reconoce las que tienen fondo amarillo.

Por otra parte, para poder aumentar el porcentaje de eficiencia es necesario utilizar cámaras de mayor calidad que no sean tan afectadas por los cambios que ocurren en el entorno como la iluminación, movimientos de los vehículos etc.

Para terminar, se puede decir que el sistema propuesto necesita de más tiempo y conocimientos para ser productivo, a pesar de eso se puede decir que el prototipo desarrollado funciona bien y deja una base para empezar con un proyecto más ambicioso.

Referencias.

Bagnato, J.(2020) Healthg Big Data: Consultado el 18 de abril del 2022. https://www.juanbarrios.com/redes-neurales-convolucionales/

Beaucamps (2018), Implementación de software para la gestión de un parking mediante el uso de técnicas de visión artificial, Universidad politécnica de Valencia. <https://riunet.upv.es/handle/10251/111197>.

Blanco (2017), Sistema de reconocimiento de

matrículas y monitorización en entornos urbanos,

Universidad de Almería. http://hdl.handle.net/10835/6617.

Cervera, (2020). GPU-Accelerated Vision for Robots: Improving System Throughput Using OpenCV and CUDA. https://ieeexplore.ieee.org/abstract/document/9047171

Corvalán, J. (2017). Inteligencia Artificial y derechos humanos (Parte I). Diario DPI Cuántico, Diario Constitucional y Derechos Humanos, http://www.scielo.org.pe/pdf/pyr/v7n2/a21v7n2.pdf

Erazo (2019) Prototipo de detección de aparcamientos libres mediante visión artificial en un parqueadero de la universidad técnica del norte. Universidad Tecnica del Norte. http://repositorio.utn.edu.ec/handle/123456789/9075

Fernandez, Quinatoa (2017), Reconocimiento de placas vehiculares en tiempo real por medio de visión artificial, Universidad central del ecuador. <http://www.dspace.uce.edu.ec/handle/25000/11755>

Franco Calderon, J. A. (2020). Modelado de la gestión entre parqueaderos y vehículos mediante la implementación de agentes inteligentes. Escuela Colombiana de ingeniería Julio Garavito. <https://repositorio.escuelaing.edu.co/handle/001/1227>

Gonzalez, G. (2017) Reconocimiento de objetos utilizando OpenCV y Python en una Raspberry pi 2 en una tlapalería. Universidad autónoma del estado de México. http://ri.uaemex.mx/handle/20.500.11799/68150

Hernández (2018) El crecimiento de la ciudad y su impacto en el transporte público. Consultado el 02 de mayo de 2022. <http://www.trcimplan.gob.mx/blog/el-crecimiento-de-la-ciudad.html>

Hsu, S., & Chuang, C.-H. (2018). Vehicle Detection using Simplified Fast R-CNN, 3–5.

Lara (2020). Desarrollo de un Sistema de Parqueadero Inteligente Mediante una Red LPWAN. Universidad de las fuerzas armadas. <http://repositorio.espe.edu.ec/bitstream/21000/22398/1/T-ESPE-043751.pdf>

Marín, R. (2020) Instituto Europeo de Estudios Empresariales S.A. ¿Qué es OpenCV?. Revista digital. https://revistadigital.inesem.es/informatica-y-tics/opencv/

Mendoza, Pacheco, Fuentes (2018) Desarrollo De Un Prototipo Para Identificación De Placas Vehiculares Y Reconocimiento De Caracteres En Tiempo Real Implementado En Raspberry Pi 3 Para La Universidad Del Sinú Seccional Cartagena Sede Santillana. Universidad Del Sinú Elías Bechará Zainúm Seccional Cartagena. http://repositorio.unisinucartagena.edu.co:8080/xmlui/handle/123456789/45

Puente (2021), Desarrollo de un sistema de reconocimiento de placas para la obtención de información en tiempo real, en apoyo a las operaciones de la empresa UNACEM, SAA. Universidad Tecnológica del Perú. https://repositorio.utp.edu.pe/handle/20.500.12867/4700

Qadri, Asif, (2019). Automatic Number Plate Recognition System For Vehicle Identification Using Optical Character Recognition. https://ieeexplore.ieee.org/abstract/document/5169511

Ramirez (2020), Reconocimiento Automático De Placas De Rodaje Utilizando Una Red Neuronal Convolucional Para El Ingreso De Vehículos En La Universidad Ricardo. Universidad Ricardo Palma. http://repositorio.urp.edu.pe/handle/URP/3646

Ramirez, Apaza, Rolly (2020), Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la universidad ricardo palma. <http://repositorio.urp.edu.pe/handle/URP/3646>.

Rojas (2017), Desarrollo De Un Sistema De Reconocimiento De Placas Y Su Influencia En La Detección De Vehículos Robados En La Municipalidad De San Isidro. Universidad Inca Garcilaso de la vega. http://repositorio.uigv.edu.pe/handle/20.500.11818/1359

Viera, M (2017), Procesamiento De Imágenes Usando OpenCV Aplicado En Raspberry Pi Para La Clasificación Del Cacao. Universidad de Piura. https://pirhua.udep.edu.pe

Yilmaz, Güzel, Askerbeyli y Bostanci, (2018), A Vehicle Detection Approach using Deep Learning Methodologies, Computer Engineering Department Ankara University. https://arxiv.org/abs/1804.00429

Zelic (2021). OCR in Python with Tesseract, OpenCV and Pytesseract. https://nanonets.com/blog/ocr-withtesseract/#preprocessingfortesseract