

UNIVERSIDAD RICARDO PALMA

FACULTAD DE INGENIERÍA

PROGRAMA DE TITULACIÓN POR TESIS

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**RECONOCIMIENTO AUTOMÁTICO DE PLACAS DE RODAJE
UTILIZANDO UNA RED NEURONAL CONVOLUCIONAL PARA
EL INGRESO DE VEHÍCULOS EN LA UNIVERSIDAD RICARDO
PALMA**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL
DE INGENIERO ELECTRÓNICO**

PRESENTADA POR

Bach. RAMIREZ MEJÍA, BRAYAN DE JESÚS

Bach. TITO APAZA, MACK ROLLY

Asesor: Dr. Ing. HUAMANÍ NAVARRETE, PEDRO FREDDY

LIMA-PERÚ

2020

DEDICATORIA

A mis padres que me apoyaron en toda la etapa universitaria, y por su dedicación durante toda mi formación como persona y como profesional. A mis maestros que aportaron en valores a mi crecimiento personal.

Brayan Ramirez Mejia

En primer lugar, doy gracias a Dios por darme fuerza y valor para culminar esta etapa de mi vida, además agradezco la confianza y el apoyo brindado por parte de mis padres, hermanos y sobrinos, asimismo a mi amada Olenka Sofia por ser el motivo a seguir adelante para poder concluir con éxito este proyecto de tesis y todos los anhelos futuros.

Mack Rolly Tito Apaza

AGRADECIMIENTO

A nuestra casa de estudios, que siempre nos brindó las mejores aulas y laboratorios para aprender, a nuestros docentes, quienes nos apoyaron constantemente en nuestro crecimiento profesional, a nuestros asesores, que nos brindaron su conocimiento para poder llevar a cabo exitosamente esta tesis y finalmente a nuestros compañeros de aula, con quienes hemos compartido años de carrera universitaria aprendiendo juntos.

Mack Tito y Brayan Ramirez

ÍNDICE GENERAL

RESUMEN.....	X
ABSTRACT	XI
INTRODUCCIÓN.....	12
CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA.....	14
1.1 Problemática	14
1.1.1 Problema General	14
1.1.2 Problemas Específicos	14
1.2 Objetivos.....	15
1.2.1 Objetivo General.....	15
1.2.2 Objetivos Específicos	15
1.3 Importancia y justificación	15
1.4 Alcances y limitaciones	16
1.4.1 Alcances.....	16
1.4.2 Limitaciones	16
CAPÍTULO II: MARCO TEÓRICO.....	17
2.1 Antecedentes del estudio de investigación	17
2.1.1 Antecedentes Nacionales	17
2.1.2 Antecedentes Internacionales	19
2.2 Bases teóricas relacionadas con el tema	21
2.2.1 Placa única de rodaje	21
2.2.2 Red neuronal convolucional	26
2.2.3 Base de datos	29
2.3 Definición de términos básicos	30
2.4 Metodología de la investigación	30
2.4.1 Variables de investigación.....	30
2.4.2 Tipo y método de investigación	30
2.4.3 Técnicas e instrumento de recolección de datos.....	31
2.4.4 Procedimiento para la recolección de datos	31
CAPÍTULO III: DISEÑO DE INGENIERÍA	32
3.1 Diagrama de bloques para el diseño de ingeniería.....	32
3.2 Elección del dispositivo para la toma de muestras (fotografías)	33
3.2.1 Cámara IP	33
3.2.2 Cámara dispositivo celular	34
3.2.3 Cámara Web	35

3.3	Entrenamiento y ejecución de la red neuronal convolucional a través del Toolbox Deep Learning	35
3.3.1	Algoritmo de pre-procesamiento para la detección de placas.	36
3.3.2	Modelo de red neuronal convolucional N° 1	40
3.3.3	Modelo de red neuronal convolucional N° 2	44
3.3.4	Modelo de red neuronal convolucional N° 3	46
3.4	Establecimiento de la base de datos utilizando DB Browser for SQLite	49
3.5	Diseño de la portada del proyecto a través del gadget App Designer	50
CAPITULO IV: PRUEBAS Y RESULTADOS		53
4.1	Validación de la Red Neuronal Convolucional N° 1	53
4.2	Validación de la Red Neuronal Convolucional N° 2	55
4.3	Validación de la Red Neuronal Convolucional N° 3	57
4.4	Validación de las redes neuronales convolucionales con un grupo externo....	60
4.5	Prototipo del módulo de reconocimiento de placas de automóviles.....	64
4.6	Costos del proyecto.....	65
CONCLUSIONES.....		67
RECOMENDACIONES		69
BIBLIOGRAFÍA		70
ANEXOS		72

ÍNDICE DE FIGURAS

Figura 1. Nueva codificación de placa de vehículos según el D.S. 017-2008-MTC. Parte 2.	22
Figura 2. Distribución de colores y representación gráfica de la placa única nacional de rodaje.	23
Figura 3. Ingreso del Estacionamiento inteligente. Reconocimiento de la placa y generación del ticket.	25
Figura 4. Esquema de un sistema de control de acceso.	25
Figura 5. Ejemplo de sistema de monitoreo de infracciones de tránsito. La cámara reconoce la placa del vehículo al pasar la luz roja.	26
Figura 6. Diagrama de bloques del proyecto.	32
Figura 7. Programación realizada en Matlab con cámara IP.	33
Figura 8. Conexión de aplicación IP Web Cam a través de celular.	34
Figura 9. Programación en Matlab utilizando la cámara web.	35
Figura 10. Procesamiento de la imagen.	37
Figura 11. Imagen cargada en programa Matlab.	37
Figura 12. Filtros de erosión y dilatación aplicados.	37
Figura 13. Imagen cargada en programa Matlab con filtros aplicados.	38
Figura 14. Programación que muestra el resultado de ubicación de la placa en la imagen.	38
Figura 15. Detección de la placa dentro de la imagen.	38
Figura 16. Detección final de la placa vehicular.	39
Figura 17. Almacenamiento de las imágenes de placas en la carpeta "entrenamiento".	39
Figura 18. Placas vehiculares que formaron parte del entrenamiento de la red neuronal.	40
Figura 19. Entrenamiento de red neuronal con 20 placas vehiculares.	41
Figura 20. Aplicación de capas convolucionales y maxpooling a la red.	41
Figura 21. Comandos de entrenamiento de la red neuronal convolucional.	42
Figura 22. Capas convolucionales de la red neuronal convolucional N°1.	42
Figura 23. Entrenamiento del Modelo de red neuronal N° 1.	43
Figura 24. Simulación de toma de fotografía para la detección de los caracteres de la placa de rodaje.	43

Figura 25. Entrenamiento de red neuronal convolucional con veinte placas vehiculares.	44
Figura 26. Capas convolucionales de la red neuronal convolucional N°2.....	45
Figura 27. Entrenamiento de la red neuronal convolucional N° 2.....	45
Figura 28. Índice de asertividad al 0.52.....	46
Figura 29. Capas convolucionales de la red neuronal convolucional N°3.....	46
Figura 30. Programación del modelo de red neuronal convolucional N° 3(Parte I, II y III).	47
Figura 31. Programación de la cámara web e índice de asertividad Parte I y Parte II (Línea 27).....	48
Figura 32. Detección de placa 'ARE-620' con el modelo de red neuronal N°3.....	49
Figura 33. Creación de una base de datos a partir de imágenes.....	49
Figura 34. Programación en Matlab para conectar con la base de datos.....	50
Figura 35. Programación de interfaz de la red con App Designer. Parte 1.....	50
Figura 36. Programación de interfaz de la red con App Designer. Parte 2.....	51
Figura 37. Programación de interfaz de la red con App Designer. Parte 3.....	51
Figura 38. Interfaz del programa creado mediante App Designer (I) y (II).....	52
Figura 39. Placa 'AZJ-188' detectada con el modelo de red neuronal convolucional N° 1	54
Figura 40. Placa 'B0K-104' detectada con el modelo de red neuronal convolucional N° 1	54
Figura 41. Placa vehicular 'BLD-051' no detectada.....	54
Figura 42. Placa vehicular 'AMD-299' no detecta.....	55
Figura 43. Placa 'B4Y-251' detectada con el modelo de red neuronal convolucional N° 2	56
Figura 44. Placa 'BLD-051' detectada con el modelo de red neuronal convolucional N° 2	56
Figura 45. Placa vehicular 'ARE-620' no detectada.....	56
Figura 46. Placa vehicular 'ATH-189' no detectada, se detectó la placa 'C7S-484'.....	57
Figura 47. Placa 'ATH-189' detectada con la red neuronal convolucional N° 3.....	58
Figura 48. Placa 'ARE-620' detectada con la de red neuronal convolucional N° 3.....	58
Figura 49. Placa vehicular 'HID-442' no detectada.....	59
Figura 50. Placa vehicular 'HID-442' detectada correctamente con el modelo de red neuronal convolucional N° 3.....	59

Figura 51. Resultado de evaluación de placa 'AXO-010' con la red neuronal convolucional N° 1.	61
Figura 52. Resultado de evaluación de placa 'AXO-010' con la red neuronal convolucional N° 2.	61
Figura 53. Resultado de evaluación de placa 'AXO-010' con la red neuronal convolucional N° 3.	61
Figura 54. Resultado de evaluación de la placa 'A2P-445' con la red neuronal convolucional N° 1.	62
Figura 55. Resultado de evaluación de la placa 'A2P-445' con la red neuronal convolucional al N° 2.	62
Figura 56. Resultado de evaluación de la placa 'A2P-445' con la red neuronal convolucional N° 1.	63
Figura 57. Simulación de ingreso vehicular al estacionamiento de la Universidad Ricardo Palma – Parte 1.	64
Figura 58. Simulación de ingreso vehicular al estacionamiento de la Universidad Ricardo Palma – Parte 2.	65
Figura 59. Resultado obtenido en App Designer – Reconocimiento correcto.	65
Figura 60. Cámara Web Full HD 1080p Webcam.	72

ÍNDICE DE TABLAS

Tabla 1. Nueva codificación de placa de vehículos menores según el D.S. 017-2008-MTC. Parte 1	21
Tabla 2. Zonas registrales para asignación de caracteres de la placa ordinaria.....	23
Tabla 3. Comparación de efectividad de las redes neuronales convolucionales.	60
Tabla 4. Resultado de la validación de placas ajenas con los modelos de red neuronal convolucional creados.	63
Tabla 5. Costos de materiales del proyecto de tesis.	66

RESUMEN

Las redes neuronales convolucionales (CNN) han tenido una gran evolución en su aplicación para la detección de imágenes, lo cual ha ayudado en la resolución de problemas que anteriormente se volvían complejos, aplicándose mayormente en sistemas de reconocimiento facial y de reconocimiento de placas vehiculares en bancos y centros comerciales que cuentan con un control de aforo vehicular. Por ello, se simuló el reconocimiento automático de placas de rodaje utilizando una CNN para el ingreso de vehículos en la Universidad Ricardo Palma. En primer lugar, se realizó la toma de fotografías de placas vehiculares para el entrenamiento de tres modelos de red, estos modelos fueron entrenados y evaluados con tres, cinco y ocho capas intermedias respectivamente, a través del Toolbox Deep Learning del Matlab, con seiscientas imágenes de placas vehiculares, pertenecientes a veinte vehículos, donde cuatrocientas cuarenta imágenes fueron para el entrenamiento interno de las CNNs y ciento sesenta para la validación de las mismas. A continuación, se corroboró la efectividad de las redes, obteniendo respectivamente en el primer, segundo y tercer modelo un porcentaje de efectividad del 40%, 75% y 95%. Con ese resultado, se eligió el tercer modelo de CNN para el reconocimiento de las placas vehiculares. Seguidamente, se procedió con la creación de la interfaz del proyecto a través del gadget App Designer y la creación de la base de datos con el software SQLite. Finalmente, se validó el funcionamiento del tercer modelo de red neuronal convolucional, logrando el correcto reconocimiento de las placas vehiculares.

Palabras claves: Red neuronal convolucional, Matlab, Toolbox Deep Learning, App Designer, SQLite.

ABSTRACT

Convolutional neural networks (CNN) have had a great evolution in their application for image detection, which has helped in solving problems that previously became complex, being applied mainly in facial recognition and license plate recognition systems in banks and shopping centers that have a vehicle capacity control. For this reason, the automatic recognition of license plates was simulated using a CNN to enter vehicles at the Ricardo Palma University. In the first place, vehicle license plates were taken for the training of three network models, these models were trained and evaluated with three, five and eight intermediate layers respectively, through the Matlab Deep Learning Toolbox, with six hundred images of license plates, belonging to twenty vehicles, where four hundred and forty images were for the internal training of CNNs and one hundred and sixty for their validation. Next, the effectiveness of the networks was corroborated, obtaining respectively in the first, second and third models an effectiveness percentage of 40%, 75% and 95%. With that result, CNN's third model was chosen for license plate recognition. Next, we proceeded with the creation of the project interface through the App Designer gadget and the creation of the database with the SQLite software. Finally, the operation of the third convolutional neural network model was validated, achieving the correct recognition of the license plates.

Keywords: Convolutional Neural Network, Matlab, Toolbox Deep Learning, App Designer, SQLite.

INTRODUCCIÓN

La presente investigación hace referencia al reconocimiento automático de placas de rodaje para permitir el ingreso vehicular al estacionamiento de la Universidad Ricardo Palma, que se puede definir como el reconocimiento de imágenes o también como la detección de caracteres de una placa de rodaje a través del entrenamiento y validación de una red neuronal convolucional. La característica principal del uso de esta red neuronal en la detección de placas vehiculares es para evitar la congestión vehicular en el acceso al estacionamiento, así como también mejorar el control del aforo dentro del estacionamiento de la Universidad Ricardo Palma. La causa principal de esta problemática es que el ingreso vehicular a la Universidad Ricardo Palma actualmente se realiza de la manera tradicional, la cual es mediante un registro manual, generando demoras en la validación de datos del conductor del vehículo.

En la presente investigación, se consideraron algunos antecedentes relacionados con el tema, de los cuales los más importantes fueron: Aldás Salazar, Collantes Vaca, & Reyes Vásquez, en su estudio denominado “Reconocimiento de placas vehiculares a través del procesamiento de imágenes con visión artificial para el control automático de acceso una entidad educativa superior en Ecuador” y Espinoza G., en su tesis titulada “Sistema de reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio”.

El interés para la realización de la presente investigación fue solucionar un problema constante que se presenta en la Universidad Ricardo Palma, el cual es la congestión vehicular que se genera en la avenida aledaña a la universidad, ocasionando insatisfacción a la comunidad universitaria dado que por años el acceso al estacionamiento de la casa de estudios se realiza de la manera clásica.

Las placas vehiculares utilizadas para el entrenamiento y validación de la red neuronal convolucional fueron obtenidas de vehículos pertenecientes al personal de la Universidad Ricardo Palma, es por ello que el tipo de muestra es no aleatorio, además el método de investigación es de tipo experimental, ya que se realizó el entrenamiento de tres modelos de redes neuronales convolucionales a fin de determinar el modelo de red más efectivo para finalmente realizar la validación y simulación de la detección de placas de rodaje.

La presente tesis, se divide en cuatro capítulos, de los cuales en el capítulo I se realiza el planteamiento y delimitación del problema, donde se observa la problemática, los objetivos, la importancia y justificación del proyecto y los alcances y limitaciones, en el capítulo II se realiza los antecedentes del estudio de investigación, las bases teóricas relacionadas con el tema y la metodología de la investigación. En el capítulo III se observa el diseño de ingeniería, el cual se compone por el diagrama de bloques, la elección del dispositivo para la toma de fotografías, el entrenamiento de los modelos de red neuronal, la creación de la base de datos, y finalmente el diseño de la interfaz del proyecto. Finalmente, en el capítulo IV se presenta las pruebas y los resultados obtenidos en la validación de las redes neuronales entrenadas.

Se concluye que al tener una mayor cantidad de capas convolucionales en el entrenamiento de la red neuronal, se obtendrán resultados más efectivos en la validación y detección de placas vehiculares. Además, se recomienda que las placas elegidas para la toma de datos se encuentren en condiciones óptimas y se sugiere tener la mayor cantidad de imágenes posibles de cada placa, a fin de que la red neuronal convolucional tenga un mejor entrenamiento y un correcto reconocimiento de las placas vehiculares.

CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA

1.1 Problemática

En la actualidad, 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, como es el caso de acceso a estacionamientos de centros comerciales, edificios, entidades públicas y privadas en su mayoría.

En la Universidad Ricardo Palma, el sistema que se utiliza para el ingreso vehicular al estacionamiento, se realiza también de la manera clásica mediante la generación de un ticket por parte del personal de seguridad. Esta forma de registro puede tener inconvenientes a futuro ya que no existe un reporte de entrada de los vehículos ni un monitoreo de los mismos. Este método de ingreso mediante registro manual, ocasiona un 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).

1.1.1 Problema General

¿Cómo utilizar una red neuronal convolucional para reconocer los caracteres de las placas de rodaje de vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma?

1.1.2 Problemas Específicos

A. ¿Cómo establecer una base de datos a partir de la captura de fotografías de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma?

B. ¿Cómo implementar y entrenar una red neuronal convolucional, con apoyo del Toolbox Deep Learning del Matlab, para la identificación de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma?

C. ¿Cómo validar la red neuronal convolucional, de manera offline, a través del reconocimiento de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma durante el horario matutino?

1.2 Objetivos

1.2.1 Objetivo General

Utilizar una red neuronal convolucional para reconocer los caracteres de las placas de rodaje de vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma.

1.2.2 Objetivos Específicos

A. Establecer una base de datos a partir de la captura de fotografías de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma.

B. Implementar y entrenar una red neuronal convolucional, con apoyo del Toolbox Deep Learning del Matlab, para la identificación de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma durante el horario matutino

C. Validar la red neuronal convolucional a través del reconocimiento de las placas de rodaje de los vehículos particulares que ingresan al estacionamiento de la Universidad Ricardo Palma.

1.3 Importancia y justificación

Para el diseño de la solución, previamente se estableció una base de datos a partir de la toma de fotografías a las placas de los vehículos particulares, algunos de los cuales ingresan a la Universidad Ricardo Palma, con ello se implementó y entrenó los modelos de red neuronal convolucional utilizando la herramienta Toolbox Deep Learning del Matlab. Una vez entrenado los modelos de red neuronal, se validó la efectividad y capacidad de cada uno en identificar los caracteres de la placa de rodaje de los vehículos que ingresan a la Universidad Ricardo Palma.

El beneficio de este proyecto de tesis, fue contar con un registro de acceso de los vehículos que ingresan a la Universidad Ricardo Palma, con el objetivo de monitorear y reducir significativamente la congestión vehicular que se genera al ingreso del estacionamiento, así como también monitorear y controlar el aforo vehicular, evitando inconvenientes en el parqueo de los vehículos particulares. Esta verificación se realizó a través de una simulación. Se pretende contar a futuro con un prototipo para lograr lo antes propuesto y además reducir el contacto físico entre el personal de seguridad de la Universidad Ricardo Palma con la persona que se identifica al ingresar al estacionamiento.

1.4 Alcances y limitaciones

1.4.1 Alcances

- Reducir el tiempo de espera para permitir el acceso vehicular al estacionamiento de la Universidad Ricardo Palma evitando así congestión vehicular.
- Generar un reporte de ingreso de vehículos al estacionamiento de la Universidad Ricardo Palma, con el fin de controlar el aforo del parqueo vehicular.
- Reducir de manera significativa el contacto entre el personal de seguridad y el conductor que desea ingresar a la Universidad Ricardo Palma.

1.4.2 Limitaciones

- El proyecto está orientado para el año 2020.
- Acceso restringido al estacionamiento de la Universidad Ricardo Palma, lo cual limita la captura de imágenes de las placas de rodaje de los vehículos que acuden diariamente.
- Características técnicas del dispositivo de captura de imágenes.
- Tiempo de procesamiento para el entrenamiento de la red neuronal en una computadora de escritorio.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedentes del estudio de investigación

2.1.1 Antecedentes Nacionales

Espinoza, G. (2014), en su tesis titulada “Sistema de reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio” sustentada en la Pontificia Universidad Católica del Perú, para obtener el título de Ingeniero informático, propone la creación de un software de reconocimiento de placas de rodaje el cual beneficiará a los sistemas de control de acceso utilizados en edificios. El principal problema que abordó en el trabajo es el referido a la detección automática de la placa vehicular del sistema peruano de placas vehiculares, lo cual implicó realizar un pre-procesamiento de la imagen el cual incluye la binarización y detección de bordes de la imagen, aplicación de programación C++ y finalmente la segmentación (detección) de la placa. La conclusión más relevante que tuvo en la tesis fue sobre la eficiencia que muestra el modelo algorítmico desarrollado, ya que en comparación con los demás algoritmos que se referencian en la tesis, expresada en tiempo de ejecución, a primera vista da señales de no ser suficiente para que este sea usado para analizar video en tiempo real. Sin embargo, se puede contrastar la eficacia, el índice de éxito, con el de otros modelos propuestos en conjunto con su velocidad de detección de placas. El aporte que tiene la investigación anterior con nuestra tesis es el desarrollo del proceso que tuvieron para la detección de la placa que fue de forma escalable, lo cual nos aportó como opción a seguir, lo que destaca nuestro proyecto de tesis con lo antes mencionado es que se realiza el algoritmo de detección a través del Toolbox Deep Learning herramienta del software Matlab, y se desarrolla una interfaz didáctica para su uso, además de tener una implementación de bajo costo.

Salazar, M (2014), en su tesis denominada “Desarrollo de un algoritmo para la localización automática de placas vehiculares peruanas usando técnicas de procesamiento de imágenes”, sustentada en la Pontificia Universidad Católica del Perú para obtener el título de Ingeniero Electrónico, nos indica que la necesidad de identificar un vehículo está relacionada con la seguridad ciudadana,

por lo que el desarrollo de un sistema automático de reconocimiento de placas vehiculares ayudaría a tener una ciudad más segura. Esta tesis plantea como solución el desarrollo de un algoritmo para la localización automática de placas vehiculares basado en técnicas de procesamiento digital de imágenes, en la tesis se hace una investigación de los métodos actuales de detección de las placas y la extracción de la región de interés de las placas para con ello desarrollar un algoritmo que mejore los anteriores. Finalmente, se concluye que la efectividad del método desarrollado tiene resultados positivos y es una solución robusta, pero depende básicamente de las condiciones que se adquirió la imagen; es decir, el estado de la placa y la calidad de la imagen. La investigación realizada por Salazar, apporto una visión más amplia en el desarrollo del método de análisis de detección de placas ya que en el estudio se indican los métodos existentes y como a través de ellos poder decidir el que se adapte mejor a nuestro fin.

Mundaca, G. (2016), en su tesis denominada “Detección de caracteres de placas de automóviles mediante técnicas de visión artificial”, realizada en la Universidad de Piura para obtener el título de Ingeniero Mecánico - Eléctrico, indica que el sistema de reconocimiento realizado está compuesto por etapas, y se empleó la herramienta " Video and Image Processing Toolbox" que interviene en base al software de MATLAB, donde se implementan varias técnicas para el óptimo desarrollo de la captura de caracteres de las placas de rodaje para mejorar el procesamiento digital de imágenes. En conclusión, los resultados evidenciaron que dependen en gran medida del estado en que se encuentran las placas de rodaje, y que no existan objetos cerca de las placas con fondo de un solo color y con relieve color negro ya que el programa suele no reconocer las placas en esos casos, además recomienda que la evaluación de la detección de la placa se lleve a cabo en escenas estáticas y a una distancia prudente para favorecer una detección de los mismos. Así mismo, Mundaca en su investigación nos brinda la conformidad que la técnica de identificación de imágenes se realiza por etapas, con ello pudimos tener un panorama más claro de la realización de nuestra red neuronal, la diferencia es que esta red neuronal será realizada a través del Toolbox Deep Learning.

2.1.2 Antecedentes Internacionales

Trejo, N. (2016), en su tesis denominada “Reconocimiento automático de matrículas de automóviles particulares mexicanas con información del color”, para obtener el grado de Maestro en Ciencias de la Computación, sustentado en la Universidad Autónoma del Estado de México, hizo un estudio en las matrículas de automóviles particulares extrayendo características de color de baja intensidad y de forma. Asimismo, propuso un algoritmo para el reconocimiento del registro de vehículos mexicanos para la segmentación, extracción y reconocimiento de caracteres (con base en sus características de color) ya que las placas mexicanas no son de fondo de un solo color, sino que tienen diseños de acuerdo al estado donde se encuentre. Los resultados obtenidos muestran que la técnica propuesta es robusta a ligeras variaciones de iluminación, marcas de agua y al color de los caracteres; siempre y cuando éstos conserven una baja intensidad, debido a que, en las pruebas de reconocimiento, aun cuando algunas de las técnicas empleadas sólo alcanzaron como máximo un 48 %, con el factor de correlación de pearson se alcanzó un 91.42 %, lo cual muestra que la segmentación de los caracteres fue adecuada. Además, la investigación realizada por Trejo aporta a nuestra investigación con la creación de una base de datos de la captura de placas para con ello poder entrenar a la red, el hacen la creación a través de cada carácter (número y letra) que conforma la placa. La diferencia es que nuestro trabajo será realizado con una base de datos de cada placa que ingrese a la Universidad Ricardo Palma, para con ello tener un mejor manejo y control de los mismos al ingreso a la entidad.

Aldás Salazar, Collantes Vaca, & Reyes Vásquez (2016), en su estudio denominado “Reconocimiento de placas vehiculares a través del procesamiento de imágenes con visión artificial para el control automático de acceso una entidad educativa superior en Ecuador”, partieron en su investigación desde la obtención de la imagen a través de la utilización de una cámara con Protocolo de Internet (cámara IP), se procesa y corrige la imagen con el desarrollo de algoritmos dentro del entorno de programación LABVIEW versión 2013, el mismo que utiliza la extracción de características de la región de interés mediante el reconocimiento óptico de caracteres (OCR) con un proceso de binarización de la imagen, lo que da como resultado, una cadena de caracteres

con los números y letras de la placa del vehículo que son comparados con una base de datos de autos registrados que ingresan al campus. Con los caracteres de la placa ingresados en el sistema y comparados con la base de datos se realiza un control de ingreso, y si el vehículo está registrado como usuario del Sistema de Control de Ingreso Vehicular de la entidad educativa enviado la instrucción para abrirse las barreras de entrada y salida. Las conclusiones que tienen en la investigación a través de las pruebas y resultados es que en las ocho muestras tomadas se puede observar una eficiencia de 87,19%, que es un buen resultado, debido a las condiciones climáticas presentadas y la resolución de la cámara que es baja. El aporte de la investigación hecha por los autores anteriores es que se debe realizar la detección de placas con una cámara de buena calidad, el beneficio de nuestro proyecto es la utilización de una cámara web, para la toma de fotografías de las placas a analizar, la misma que será utilizado para la detección de las placas esperando tener con ello un mayor índice de asertividad.

Muhammad Tahir Qadri y Muhammad Asif (2019) en su artículo titulado “Automatic number plate recognition system for vehicle identification using optical character recognition”, nos muestran como realizar el reconocimiento de una placa vehicular a través de una simulación del sistema ANPR o también conocido como reconocimiento automático de número de placa. Ellos hacen uso de una cámara conectada mediante puerto USB a un ordenador personal, el procedimiento de los métodos que utilizan fue en primer lugar la captura de la imagen, buscar píxeles amarillos, luego el filtrado de imágenes, la extracción de la región de la placa mediante el algoritmo de difuminado, separación de líneas mediante segmentación de filas, y finalmente la extracción de caracteres usando segmentación de columna para aplicar el reconocimiento de caracteres. El aporte del artículo anterior, es que se realiza la detección de las imágenes a través de una cámara conectada mediante un puerto USB, y no tienen inconvenientes con las fotografías obtenidas.

2.2 Bases teóricas relacionadas con el tema

2.2.1 Placa única de rodaje

Para el procesamiento de las imágenes como información a entrenar en la red neuronal convolucional, se hará uso de las capturas fotográficas de las placas vehiculares, es por ello que hemos visto beneficioso mencionar las características generales de la información a tratar en el entrenamiento de la red.

En la actualidad, el estado peruano se rige a una normativa que obliga a la homogenización de placas vehiculares según su código de región, categoría, entre otros. Con el fin de regularizar todos los inconvenientes que se presentaban con anterioridad como: placas borrosas, placas clonadas, inclusive vehículos que circulaban sin placa, los cuales no podían ser identificadas por tener los caracteres con dimensiones muy pequeñas e ilegibles. Según detalla el Ministerio de Transporte y Comunicaciones en su informe realizado en base al “Decreto supremo que aprueba el reglamento del sistema de placa única nacional de rodaje - exposición de motivos” (Ministerio de Transportes y Comunicaciones)

Por tal motivo en el decreto supremo N° 017-2008-MTC, se consideran algunos cambios aplicados a las placas vehiculares del territorio peruano tal como se muestra en la tabla 1 y se complementa en la figura 1.

Tabla 1. Nueva codificación de placa de vehículos menores según el D.S. 017-2008-MTC. Parte 1

ZONAS REGISTRALES	Departamentos	Placas para vehículos menores	
		Primer carácter	Segundo caracter
ZONA REGISTRAL N°I	Tumbes y Piura	P	Correlativo
ZONA REGISTRAL N°II	Lambayeque, Cajamarca y Amazonas	M	Correlativo
ZONA REGISTRAL N°III	San Martín	S	Correlativo
ZONA REGISTRAL N°IV	Loreto	L	Correlativo

ZONA REGISTRAL N°V	La Libertad	T	Correlativo
ZONA REGISTRAL N°VI	Ucayali	U	Correlativo
ZONA REGISTRAL N°VII	Ancash	H	Correlativo
ZONA REGISTRAL N°VIII	Junín, Huánuco y Pasco	W	Correlativo
ZONA REGISTRAL N°IX	Lima	A, B, C, D, F	Correlativo
ZONA REGISTRAL N°X	Cusco, Apurímac y Madre de Dios	X	Correlativo
ZONA REGISTRAL N°XI	Ica, Ayacucho y Huancavelica	Y	Correlativo
ZONA REGISTRAL N°XII	Arequipa	V	Correlativo
ZONA REGISTRAL N°XIII	Moquegua, Tacna y Puno	Z	Correlativo
Caracteres de reserva		G, I, J, K, N, O, Q, R	Correlativo

Fuente: (Ministerio de Transportes y Comunicaciones)



Figura 1. Nueva codificación de placa de vehículos según el D.S. 017-2008-MTC.

Parte 2.

Fuente: (Ministerio de Transportes y Comunicaciones)

El informe publicado en base al Decreto Supremo emitido hace referencia a todas las placas de los vehículos que se encuentran en territorio peruano: motos, camiones, buses interprovinciales, entre otros. Centrándonos en el elemento de estudio, que son las placas de vehículos particulares (autos), o también llamados

vehículos livianos, se planteó un cuadro de distribución de representaciones gráficas y de colores de las placas de rodaje como se detalla a continuación en la figura 2.

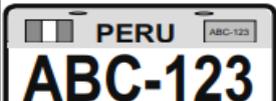
PLACA ÚNICA NACIONAL DE RODAJE				
Tipo de vehículo	Color del fondo	Color de la franja superior	Color de las letras y números	Gráfico
1. Placas ordinarias:				
1.2. Placas para vehículos livianos y pesados (Categorías M, N y O)				
Categoría M (Transporte de Personas)				
a) Vehículos Particulares (Categoría M)	Blanco	No lleva	Negro	
b) Taxis y Colectivos (Categoría M)	Blanco	Amarillo	Negro	

Figura 2. Distribución de colores y representación gráfica de la placa única nacional de rodaje.

Fuente: (Ministerio de Justicia y Derechos Humanos, 2008)

Para finalizar el análisis de las placas de rodaje a estudiar, el Ministerio de Justicia y Derechos Humanos, también realizó una tabla donde indica el tipo de carácter inicial de la placa de rodaje de acuerdo a la zona de registro de la misma, según se detalla a continuación en la tabla 2.

Tabla 2. Zonas registrales para asignación de caracteres de la placa ordinaria

ZONAS REGISTRALES	Departamentos	Placas para vehículos livianos y pesados		Placas para vehículos menores	
		Primer carácter	Segundo carácter (2)	Primer carácter	Segundo carácter (2)
ZONA REGISTRAL N°I	Tumbes y Piura	A	Correlativo	A	Correlativo
ZONA REGISTRAL N°II	Lambayeque, Cajamarca y Amazonas	B	Correlativo	B	Correlativo
ZONA REGISTRAL N°III, IV Y VI	San Martín	C	Correlativo de A a H	C	Correlativo
	Loreto		Correlativo de I a Q	P	Correlativo
	Ucayali		Correlativo de R a Z	Q	Correlativo

ZONA REGISTRAL N°V	La Libertad	D	Correlativo	D	Correlativo
ZONA REGISTRAL N°VII	Ancash	F	Correlativo	F	Correlativo
ZONA REGISTRAL N°VIII	Junín, Huánuco y Pasco	G	Correlativo	G	Correlativo
ZONA REGISTRAL N°IX	Lima	H, I, J, K	Correlativo	H, I, J, K	Correlativo
ZONA REGISTRAL N°X	Cusco, Apurímac y Madre de Dios	L	Correlativo	L	Correlativo
ZONA REGISTRAL N°XI	Ica, Ayacucho y Huancavelica	M	Correlativo	M	Correlativo
ZONA REGISTRAL N°XII	Arequipa	N	Correlativo	N	Correlativo
ZONA REGISTRAL N°XIII	Moquegua, Tacna y Puno	O	Correlativo	O	Correlativo
Caracteres de reserva		P, Q, R, S, T, U, V, W, X, Y y Z	Correlativo	P, Q, R, S, T, U, V, W, X, Y y Z	Correlativo

Fuente: Ministerio de Justicia y Derechos Humanos (2008)

2.2.1.1 Sistemas actuales de identificación de placas

- Estacionamientos inteligentes

Hoy en día encontrar estacionamiento en zonas comerciales es muy difícil y el problema aumenta cuando no recordamos la ubicación donde se encuentra el vehículo al salir, esto genera delitos como el parqueo en zonas prohibidas, pueden ser las veredas, áreas verdes, frente a garajes, etc. y también pérdida de tiempo y aumento de estrés en las personas al no recordar donde se realizó el parqueo de su vehículo. Para enfrentar este problema se ha creado un sistema de parqueo inteligente que consiste en la colocación de una cámara inteligente en el ingreso, que le permita leer el número de la placa, un tablero donde indican los lugares libres y se genere un ticket cuando el conductor realiza el pago

del estacionamiento, un software reconoce el recibo y se muestra en la pantalla del tablero la posición del vehículo tal como se muestra en la figura 3. (Salazar Marquez, 2014)

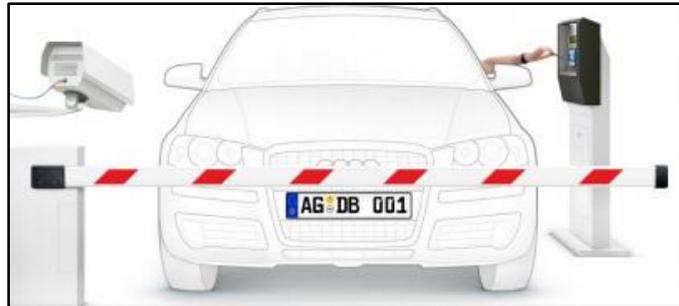


Figura 3. Ingreso del Estacionamiento inteligente. Reconocimiento de la placa y generación del ticket.

Fuente: (Salazar Marquez, 2014)

- Control acceso zonas restringidas

La inseguridad en nuestro país es proporcional al incremento de vehículos y ello hace que existan más robos de los mismos; sin embargo, la tecnología también está en crecimiento, por lo cual los sistemas automatizados en el control de acceso a áreas restringidas en base a la identificación de matrículas sean aún más seguros. En la figura 4 se muestra un esquema del sistema de control de acceso mediante el uso de reconocimiento del número de placas de vehículos a través de una cámara convencional y una base de datos para validar el permiso de ingreso. (Salazar Marquez, 2014)

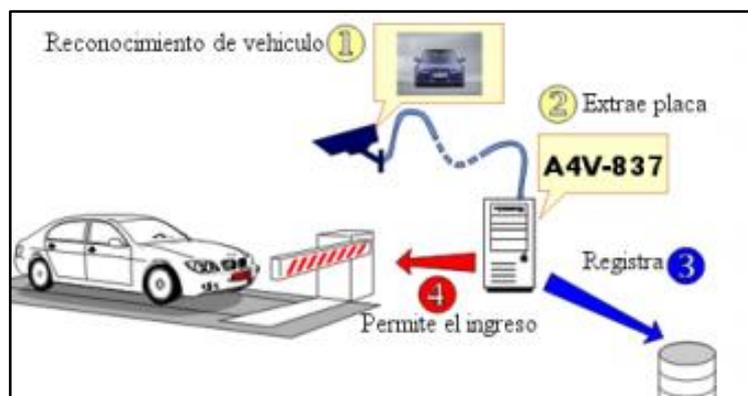


Figura 4. Esquema de un sistema de control de acceso.

Fuente: (Salazar Marquez, 2014)

- Control policial de infracciones de tránsito

Un factor principal de los accidentes de tránsito es la excesiva velocidad en la que los conductores incurren, debido a que infringen las reglas de tránsito, ya sea por la velocidad no permitida o cruzar las pistas cuando el semáforo está en rojo. Por ello, a pesar de que se tienen sistemas que eviten la excesiva velocidad de los vehículos, como son los badenes o rompe muelles, tachones u ojo de gato, es necesario sistemas eficientes de control y monitoreo de los vehículos en base a la identificación de placas para contener las infracciones de tránsito como se muestra en la figura 5. (Salazar Marquez, 2014)



Figura 5. Ejemplo de sistema de monitoreo de infracciones de tránsito. La cámara reconoce la placa del vehículo al pasar la luz roja.

Fuente: (Salazar Marquez, 2014)

2.2.2 Red neuronal convolucional

Una red neuronal convolucional en inglés Convolutional Neural Networks – CNN. Se puede definir como un conjunto de redes que a través de un entrenamiento pueden reconocer una variable de entrada efectivamente, como son por ejemplo patrones de imagen, audios entre otros. (Bagnato, s.f.) Recuperado el 8 de agosto del 2020.

Además, según Bagnato para entender de forma correcta el funcionamiento de una red neuronal convolucional, nos debemos poner un ejemplo práctico. Supongamos que queremos reconocer un rostro humano, para poder reconocer debemos conocer como son los ojos, nariz, labios, entre otros patrones y características que conforman el conjunto a identificar denominado rostro, con la información que tengamos de los ojos, nariz, labios se entrena la red neuronal para adquirir conocimiento y una vez tengamos el entrenamiento necesario

podremos reconocer de manera correcta el rostro. Bagnato, indica que las redes neuronales convolucionales consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo causal no-lineal. Como cualquier red empleada para clasificación, al principio estas redes tienen una fase de extracción de características, compuesta de neuronas convolucionales, luego hay una reducción por muestreo y al final tendremos neuronas de perceptrón más sencillas para realizar la clasificación final sobre las características extraídas. La fase de extracción de características se asemeja al proceso estimulante en las células de la corteza visual. Esta fase se compone de capas alternas de neuronas convolucionales y neuronas de reducción de muestreo. Según progresan los datos a lo largo de esta fase, se disminuye su dimensionalidad, siendo las neuronas en capas lejanas mucho menos sensibles a perturbaciones en los datos de entrada, pero al mismo tiempo siendo estas activadas por características cada vez más complejas.

Así también Bagnato indica que la red neuronal convolucional en resumen tiene la siguiente estructura: Como datos de entrada, toma los píxeles de la imagen, de donde si es a color tiene tres capas para los colores rojo, azul y verde. Por ello es conveniente trabajar con imágenes en blanco y negro, este procesamiento se realiza en el computador a través de un software, en nuestro caso se realizará a través del software Matlab con ayuda del Toolbox Deep Learning. Además, tiene las capas de convolución en donde se realiza el proceso de las entradas aplicándole filtros y entrenando a la red. Acompañada de las capas de convolución se da el muestreo donde se realiza una reducción de las dimensiones de la imagen en ancho y alto, pero se mantiene la profundidad. Finalmente, una vez entrenada la red se realiza la validación con una entrada y se obtiene los resultados.

2.2.2.1 Matlab – Toolbox Deep Learning

En la aplicación de la red neuronal convolucional se hace uso del Toolbox de Matlab denominado Deep Learning (Aprendizaje Profundo), esta herramienta es parte vital de la detección de imágenes e inclusive ha ido evolucionando con el paso de los años, como es de conocimiento el Deep

Learning se utiliza en la industria automotriz, aplicado en los autos sin conductor donde se debe tener una gran precisión para la detección de transeúntes y vehículos cercanos, en el campo aeroespacial para la identificación de objetos voladores no identificados, entre otros. (What Is Deep Learning? 3 things you need to know, s.f.) Recuperado el 05 de agosto del 2020.

La mayoría de los métodos de aprendizaje profundo utilizan arquitecturas de redes neuronales, por lo que los modelos de aprendizaje profundo a menudo se denominan redes neuronales profundas. El término "profundo" generalmente se refiere al número de capas ocultas en la red neuronal. Las redes neuronales tradicionales solo contienen 2-3 capas ocultas, mientras que las redes profundas pueden tener hasta 150. (What Is Deep Learning? 3 things you need to know, s.f.)

Además, se indican las tres formas más comunes en que las personas usan el aprendizaje profundo para realizar la clasificación de objetos son:

Entrenamiento desde cero – Para el entrenamiento de este tipo de red se reúne una gran cantidad de datos y se diseña una arquitectura de red neuronal que aprenderá las características y los modelos, normalmente este tipo de entrenamiento es el menos común ya que comprende un entrenamiento de días o incluso semanas.

Transferencia de aprendizaje – Este tipo de entrenamiento es el más usado por las aplicaciones de aprendizaje profundo ya que previamente existe una red como por ejemplo AlexNet o GoogleNet y a estas se les alimenta de información con ellos en vez de realizar un entrenamiento con muchos datos, realizan la clasificación de elementos en particular.

Extracción de características – Este tipo de entrenamiento es el menos común ya que extrae características de las capas conforme van aprendiendo en el proceso de capacitación, estas funciones pueden ser utilizadas para un modelo de aprendizaje automático.

2.2.3 Base de datos

Por lo general, una eficiente administración de datos requiere el uso de una base de datos de computadora. Una base de datos es una estructura computarizada compartida e integrada que guarda un conjunto de:

- Datos del usuario final, es decir, datos sin elaborar que son de interés para el usuario final.
- Metadatos, o datos acerca de datos, por medio de los cuales los datos del usuario final son integrados y manejados.

Los metadatos dan una descripción de las características de los datos y del conjunto de relaciones que enlaza los datos encontrados dentro de la base de datos. (Coronel, Morris, & Rob, 2011)

SQLite es el nombre dado a un SBD (Sistema de base de datos) de código abierto cuya principal diferencia es que en lugar de ser un SBD estándar, está contenido en una biblioteca escrita en lenguaje C. Un SBD clásico es un programa que crea y gestiona los datos de una base de datos al que se accede externamente por programas que requieren este tipo de datos, SQLite, sin embargo, no es un programa, sino una biblioteca de código que implementa una base de datos y es manipulada por el programa mediante llamada a funciones. SQLite no requiere instalación, basta con integrarlo de forma convencional como una biblioteca de código escrito para el programa, lo que permite incorporarlo al sistema. Esto es especialmente útil cuando tenemos que entregar sistemas con base de datos, pero no hay ninguna posibilidad de organizar un servidor de base de datos para acceso. (Benitez & Arias, 2017)

Benítez y Arias indican que por estas razones y por su tamaño relativamente pequeño, SQLite se utiliza en la mayoría de las aplicaciones que requieren almacenar y acceder a datos de manera rápida, pero no pueden acceder a una base de datos externa (por varias razones, incluyendo la velocidad y la convivencia (como: navegadores, clientes de correo electrónico, smartphone (android), los sitios web, los programas que necesitan almacenar datos temporales y etc. SQLite es también una buena opción para aquellos que quieren utilizar un SBD simple para la enseñanza de base de datos.

2.3 Definición de términos básicos

- Red neuronal artificial- Conjunto de elementos de procesamiento de la información conectados, que son capaces de aprender con la información que se alimenta. aplicables a reconocimiento de imágenes, reconocimiento de voz, entre otros.
- Procesamiento digital de imágenes - Conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información que posibilite efectuar operaciones del procesamiento sobre ella.
- Placa única nacional de rodaje - Elemento de identificación vehicular durante la circulación de los vehículos por las vías públicas terrestres, estableciendo su clasificación y características con el fin de alcanzar los estándares de seguridad internacional para evitar su falsificación.
- Red neuronal convolucional – Conjunto de redes que a través de un entrenamiento basado en una información brindada pueden reconocer una variable de entrada efectivamente.
- Convolución de imágenes – Aplicación de filtros a los pixeles de una imagen con el fin de realizar cambios en la misma.

2.4 Metodología de la investigación

2.4.1 Variables de investigación

Variable dependiente	:	Reconocimiento de caracteres de la placa de rodaje.
Variable independiente	:	Red neuronal convolucional.

2.4.2 Tipo y método de investigación

El tipo de investigación que se utiliza es del tipo aplicada, el método de investigación a utilizar es el método experimental, ya que se realizará el entrenamiento de la red neuronal convolucional para con ello detectar los caracteres de las placas, mientras la red neuronal se entrene con más información, podrá detectar con más exactitud los caracteres de las placas de rodaje de los autos particulares.

2.4.3 Técnicas e instrumento de recolección de datos

La técnica de recolección de datos a usar es la técnica de observación experimental, ya que la toma de información se realizará con el uso de una cámara web para con ello tomar fotografías a las placas vehiculares y así tener un registro y orden de los datos obtenidos. El instrumento a emplear será una cámara web de alta resolución de píxeles, con la cual se verificará la placa del vehículo y con ello validar la información a través de la red neuronal convolucional.

2.4.4 Procedimiento para la recolección de datos

En primer lugar, se determinó los veinte vehículos particulares que formaron parte del entrenamiento de la red neuronal convolucional, se pretendió que estos estén integrados por placas vehiculares de personal que tenga acceso al estacionamiento de la Universidad Ricardo Palma, una vez determinado los vehículos, se procedió a realizar la toma de fotografías a las placas vehiculares y a su vez archivando las fotos en una carpeta denominada “entrenamiento”. Por consiguiente, la información obtenida fue utilizada para entrenar y validar los modelos de red neuronal convolucional con el fin de que se evalué que modelo de red fue más efectivo, para que finalmente se simule la detección de las placas de vehículos que ingresan a la Universidad Ricardo Palma por la puerta N° 4.

CAPÍTULO III: DISEÑO DE INGENIERÍA

3.1 Diagrama de bloques para el diseño de ingeniería

En el diagrama que se muestra a continuación, detalla las etapas que comprende el proyecto. Como se puede observar en primer lugar se realizó la adquisición de datos a través de la toma de fotografías, con ello se creó una base de datos, las cuales fueron utilizadas para el entrenamiento de los modelos de red neuronal.

Una vez entrenados los modelos de red neuronal convolucional, se evaluó y determinó el modelo de red más efectivo en comparación a los otros dos. Luego, se creó la base de datos para registro a través del software SQLite y se creó la interfaz del proyecto con el gadget App Designer, finalmente se evaluó los resultados.

El proyecto sigue el diagrama de bloques mostrado en la figura 6.

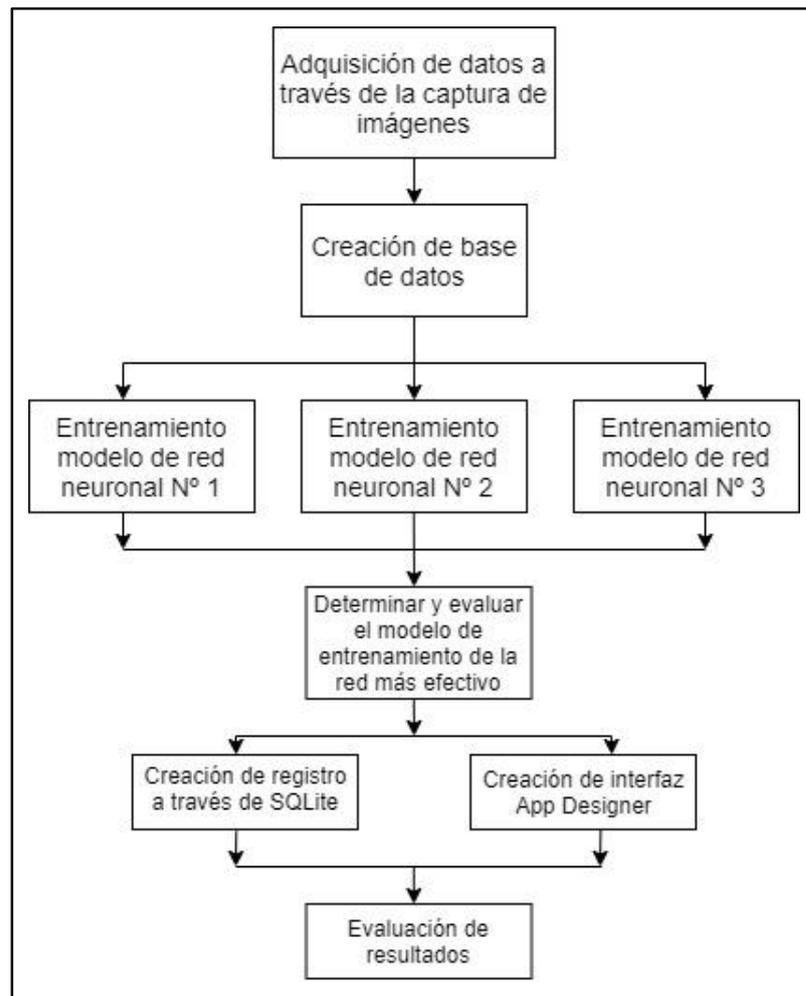


Figura 6. Diagrama de bloques del proyecto.

Fuente: Elaboración propia

3.2 Elección del dispositivo para la toma de muestras (fotografías)

Para la elección del dispositivo adecuado, que se utilizó a lo largo del proyecto se analizaron tres opciones, trabajar con una cámara IP, trabajar con una cámara de dispositivo celular y trabajar con una cámara web.

A continuación, se analizó las características de cada una de ellas con el fin de obtener el dispositivo más adecuado para el fin propuesto.

3.2.1 Cámara IP

En primera instancia se consideró usar una cámara IP, específicamente optamos por usar la cámara de una marca reconocida para no tener inconvenientes. Esta fue de la marca HIKVISION modelo H.265+, tiene una resolución de 4K (4096 x 2160 megapíxeles), cuenta con la interface de red RJ45 y se alimenta con 12 VCD.

Esta cámara al ser probada con la programación realizada en Matlab tenía múltiples inconvenientes, el más reiterado fue que se desconectaba con frecuencia de la red wifi con la que se trabajaba y no tenía buen reconocimiento de imágenes, al cambiar el tipo de conexión a través de LAN con un cable de red conectado hacia la laptop usada, no se logró la conexión entre el Matlab y la cámara IP, por tanto, se concluyó que el formato de streaming de la cámara no era compatible con el Matlab. En la figura 7 se muestra la configuración realizada en el software Matlab para enlazar la cámara IP con el programa.



```
Editor - C:\Users\DELL-MACK\Desktop\camar.m
+2 train.m x train2.m x camar.m x camara.m x prueba.m x tomarfotos.m x registrarD
1 - close all;
2 - clc;
3
4 - cam = ipcam('http://192.168.5.110/mjpg/video.mjpg', 'admin', 'Hik12345')
5 - img = snapshot(cam);
6 - pause(1);
7
8 - img = imresize(img, [200 200]);
9 - img = rgb2gray(img);
10
11 %YPredicted = predict(net, img);
```

Figura 7. Programación realizada en Matlab con cámara IP.

Fuente: Creación propia

3.2.2 Cámara dispositivo celular

Cuando se tuvo el inconveniente de la cámara IP, se optó por buscar opciones, se decidió por un elemento práctico como fue el uso de los dispositivos celulares para la captura e identificación de imágenes.

Se utilizó la aplicación IP Web Cam que se encuentra en Play Store para su descarga gratuita, esta aplicación permite que el dispositivo móvil funcione como una cámara desde donde se detectaría la placa vehicular y se analizaría de acuerdo a la base de datos a crear con el fin de reconocer las placas vehiculares. Cuando se puso en práctica el uso del celular como dispositivo de captura de imágenes como se muestra en la figura 8, si bien esta vez se logró conectar al Matlab, una vez más tuvimos el inconveniente de fallas y pérdidas de conexión de red, esto debido a que el dispositivo móvil cambiaba la conexión de las direcciones IPs a las que se conectaba ya que en el ambiente simulado existían repetidores de red con diferentes direcciones LAN, tal como es el escenario de la Universidad Ricardo Palma. Esto nos llevó a realizar cambios en la programación del Matlab, por lo que decidimos buscar una variante a esta solución.

En cuanto a la programación empleada para lograr la comunicación entre la cámara del dispositivo celular y el Matlab, fue muy similar a la empleada en el caso anterior.

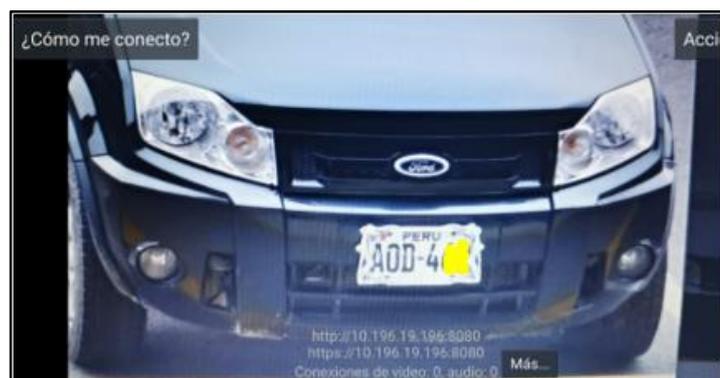


Figura 8. Conexión de aplicación IP Web Cam a través de celular.

Fuente: Creación propia

3.2.3 Cámara Web

Finalmente, se decidió el uso de una cámara web para los fines descritos anteriormente. La cámara que se utilizó fue la Cámara Web Full HD 1080p Webcam de marca genérica, la cual tiene las siguientes características principales: Sensor de imagen CMOS, interfaz USB 2.0 y 3.0, alimentación 5V DC, formato de video AVI, resolución: 1920x1080 pixeles, ángulo de la cámara: 120 grados.

Esta cámara se conectó a través del puerto USB a la laptop y se tuvo indicadores correctos de detección de imágenes tomadas, con claridad y sin caídas de conexión. En la figura 9, se muestra la programación para enlazar la cámara web con el software Matlab.

```
try
    cam=webcam('USB CAMERA'); %nombre de la webcam a ser utilizada
catch exception
end

cam.Resolution='1920x1080';
img = snapshot(cam);
pause(1);

img2 = imresize(img, [200 200]);
img2 = rgb2gray(img2);
YPredicted = predict(net,img2);
YPredicted

max = 0;
```

Figura 9. Programación en Matlab utilizando la cámara web.

Fuente: Creación propia

Al probar los tres métodos para la detección de placas vehiculares, se concluyó con la elección de la cámara web para el uso del proyecto, ya que es la que tiene una conexión más estable en comparación con la cámara IP y la cámara de dispositivo móvil.

3.3 Entrenamiento y ejecución de la red neuronal convolucional a través del Toolbox Deep Learning

Una vez creada la base de datos y elegido el dispositivo a utilizar para la captura de imagen (placa vehicular), se procedió con la creación y entrenamiento de la red neuronal a través del Toolbox Deep Learning del Matlab. Este proceso empezó con la designación de capas convolucionales a utilizar, así como la cantidad de placas vehiculares que formaron parte del proceso de entrenamiento de la red.

3.3.1 Algoritmo de pre-procesamiento para la detección de placas.

Para iniciar la creación de los modelos de redes neuronales para el proyecto, se dio inicio a un algoritmo de pre-procesamiento para la detección de las placas dentro de la imagen. Seguidamente, se procedió con la lectura del archivo de la imagen, se realiza el cambio de resolución de la imagen, se transformó el formato de color, cambiando de RGB a tonos de gris, se eliminó el ruido de la imagen a través de un filtro mediana con un tamaño de ventana de 5 x 5 píxeles. (Romero Marra, 2015)

Posteriormente, según lo que indica Romero Marra, se segmentó la placa del automóvil utilizando el comando ZoomPlate, este comando utiliza la siguiente programación:

```
function [ F ] = ZoomPlate( img )
BW = edge(img, 'canny',0.5);
%figure;imshow(BW); title('BW CANNY');
[M,N] = size(BW)
B = zeros(M,1)
for i=1:1:M
aux=1;
pixel = zeros(3,1);
for j=1:1:N
pixel(aux)=BW(i,j);
aux=aux+1;
if aux > 3
aux = 1;
if ( pixel(1) == 0 && pixel(2) == 0 && pixel(3) == 1)
B(i) = B(i) + 1;
end
if ( pixel(1) == 1 && pixel(2) == 0 && pixel(3) == 1)
B(i) = B(i) + 1;
end
pixel = zeros(3,1);
end
end
end
%figure;plot(B,M:-1:1);
[maxv,maxi]=max(B);
Lindex = maxi-81;
Hindex = maxi+81;
Corte=[0 Lindex N Hindex]; %Determina coordenadas de corte
F=imcrop(img,Corte);
%figure;imshow(F);
end
```

Con el comando ZoomPlate se realizó la segmentación de la imagen como se muestra en la figura 10, y se tiene como resultado en la figura 11.

```

%% Carga de la imagen %%
clear all;
clc;
close all;
img = imread('C:\Users\DELL-MACK\Desktop\tito\1.jpg');
%% Preprocesado %%
img=imresize(img,[480 640]);
figure;imshow(img);
% Rgb to gray
img=rgb2gray(img);
%figure;imshow(img);

% Eliminación de ruido median filter
img = medfilt2(img,[5 5]);
%figure;imshow(img);
%% Segmentacion %%
img = ZoomPlate(img);
figure;imshow(img);title('Segmentacion');

```

Figura 10. Procesamiento de la imagen.

Fuente: Creación propia



Figura 11. Imagen cargada en programa Matlab.

Fuente: Creación propia

En seguida, se procedió con los filtros de erosión, dilatación y borde con operador Canny, tal como se muestra a continuación en la figura 12. Y, se obtuvo como resultado lo visualizado en la figura 13.

```

17 - img = ZoomPlate(img);
18 - figure;imshow(img);title('Segmentacion');
19 - orig = img;
20 - %% Operaciones Morfológicas %%
21 - se = [0 0 1 0 0; 0 1 1 1 0; 1 1 1 1 1; 0 1 1 1 0; 0 0 1 0 0];
22
23 % erosion
24 - erosion = imerode(img,se);
25
26 % dilatacion
27 - apertura = imdilate(erosion,se);
28
29 % cierre
30 - img = apertura - erosion;
31 - %figure;imshow(img); title('Transformaciones');
32

```

Figura 12. Filtros de erosión y dilatación aplicados.

Fuente: Creación propia

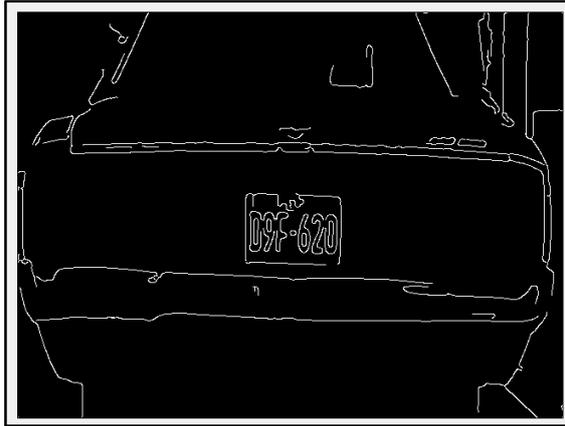


Figura 13. Imagen cargada en programa Matlab con filtros aplicados.

Fuente: Creación propia

Con los filtros ya aplicados, se procedió a la detección de las coordenadas de ubicación de la placa de rodaje por cada imagen analizada (figura 14). En la figura 15 se representa la detección de la ubicación para una imagen en particular.

```
%% Mostrar resultado %%
X=E.*[1 0 0 0]; X=max(X); %Determina eje X esquina superior Izq. Placa
Y=E.*[0 1 0 0]; Y=max(Y); %Determina eje Y esquina superior Der. Placa
W=E.*[0 0 1 0]; W=max(W); %Determina Ancho Placa
H=E.*[0 0 0 1]; H=max(H); %Determina Altura placa
Corte=[X Y (W-2) (H-7)]; %Determina coordenadas de corte
IMF=imcrop(orig,Corte);
I1 = orig(:,:,1);
[M,N] = size(IMF);
figure; imagesc(IMF);colormap gray; axis equal;
```

Figura 14. Programación que muestra el resultado de ubicación de la placa en la imagen.

Fuente: Creación propia



Figura 15. Detección de la placa dentro de la imagen.

Fuente: Creación propia

Finalmente, se detecta la placa y se extrae de la imagen, obteniendo lo que se muestra a continuación en la figura 16.



Figura 16. Detección final de la placa vehicular.

Fuente: Creación Propia

Para las evaluaciones realizadas, todas las redes neuronales convolucionales fueron entrenadas con veinte placas vehiculares, cada una de las cuales tuvo treinta fotos. Estas imágenes de las placas vehiculares, fueron almacenadas en una carpeta denominada “entrenamiento” las cuales fueron utilizadas para el entrenamiento de las redes neuronales convolucionales como se muestra en la figura 17.

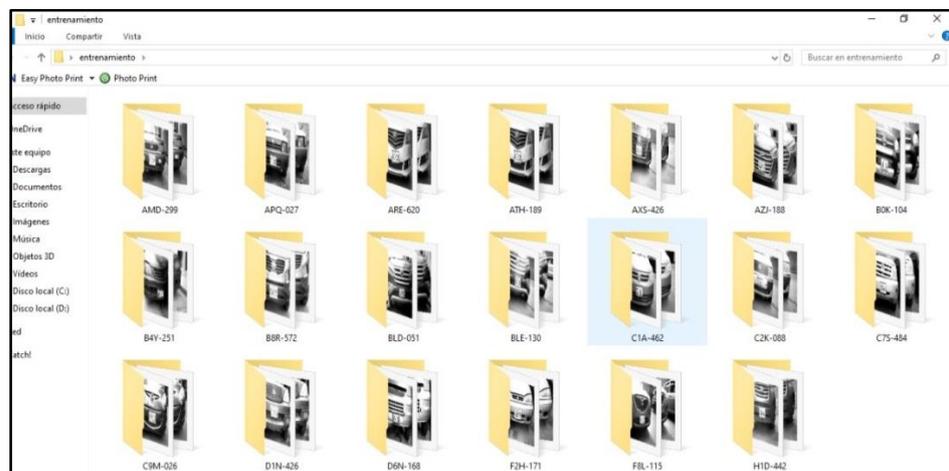


Figura 17. Almacenamiento de las imágenes de placas en la carpeta "entrenamiento"

Fuente: Creación propia

Para el entrenamiento de los modelos de red neuronal convolucional se vio conveniente realizar la evaluación con las veinte placas vehiculares. Estas placas fueron las siguientes: 'AMD-299', 'APQ-027', 'ARE-620', 'ATH-189', 'AXS-426', 'AZJ-188', 'B0K-104', 'B4Y-251', 'B8R-572', 'BLD-051', 'BLE-130', 'C1A-462', 'C2K-088', 'C7S-484', 'C9M-026', 'D1N-426', 'D6N-168', 'F2H-171', 'F8L-115', 'H1D-442'.

A continuación, en la figura 18 se muestra cuatro placas vehiculares que formaron parte del entrenamiento y validación de la red neuronal convolucional.



Figura 18. Placas vehiculares que formaron parte del entrenamiento de la red neuronal.

Fuente: Creación propia

3.3.2 Modelo de red neuronal convolucional N° 1

Con la simulación del algoritmo de detección de placas realizado, se procedió a la creación de un modelo de red neuronal convolucional con el cual se logró la detección de placas vehiculares a través de la captura de imágenes obtenidas con la cámara web, teniendo en cuenta que ya se tenían todas las fotos de las placas vehiculares almacenadas en una carpeta para su análisis al cual se le denominó “entrenamiento”. Para ello, se realizó el modelo de red con veinte diferentes placas vehiculares, donde se entrenó la red neuronal con treinta imágenes por cada placa vehicular, utilizando veintidós imágenes para el entrenamiento interno y ocho imágenes para la validación interna del entrenamiento de la red tal como se muestra en la figura 19.

```

cat = {'AMD-299','APQ-027','ARE-620','ATH-189','AXS-426','AZJ-188', ...
      'BOK-104','B4Y-251','B8R-572','BLD-051', ...
      'BLE-130','CIA-462','C2K-088','C7S-484', ...
      'C9M-026','D1N-426','D6N-168','F2H-171', ...
      'F8L-115','H1D-442'};

rootFolder = 'C:\Users\DELL-MACK\Desktop\entrenamiento';

imds = imageDatastore(fullfile(rootFolder,cat),...
    'LabelSource','foldernames');

numTrainFiles = 22;
[imdsTrain, imdsValidation] = splitEachLabel(imds, numTrainFiles, 'randomize');

```

Figura 19. Entrenamiento de red neuronal con 20 placas vehiculares.

Fuente: Creación propia

A continuación, se procedió con la configuración de las capas de red neuronal convolucional a aplicar, la capa de entrada (image input layer), dimensiona la imagen en una resolución de 200x200 pixeles y en escala de grises. La red neuronal convolucional estuvo integrada por un conjunto de capas al cual se le denomina: “capas intermedias”. Estas “capas intermedias” estuvieron conformadas por cuatro capas: Capa convolucional, capa de normalización por lotes, capa de unidad lineal rectificadora (ReLU) y capa de agrupación máxima, las cuales fueron aplicadas tres veces en este modelo de red, finalmente se coloca la capa de salida, la cual estuvo compuesta por la capa fullyConnected, la capa Softmax y la capa de clasificación, como se muestra en la figura 20.

```

%capa de entrada
imageInputLayer([200 200 1])

%Capas intermedias
convolution2dLayer(3,8,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,16,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,32,'Padding','same')
batchNormalizationLayer
reluLayer

%Capa de salida
fullyConnectedLayer(20)
softmaxLayer
classificationLayer

```

Figura 20. Aplicación de capas convolucionales y maxpooling a la red.

Fuente: Creación propia

Una vez creadas las “capas intermedias” y “capa de salida”, se realiza los comandos para el entrenamiento y clasificación de la red como se muestra en la figura 21.

```
options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',15, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain, layers, options);

YPred = classify(net, imdsValidation);
YValidation = imdsValidation.Labels;

accuracy = sum(YPred == YValidation)/numel(YValidation);
```

Figura 21. Comandos de entrenamiento de la red neuronal convolucional.

Fuente: Creación propia

A continuación, se muestra el diagrama de las capas convolucionales que componen el modelo de red neuronal convolucional N°1, tal como se muestra en la figura 22.

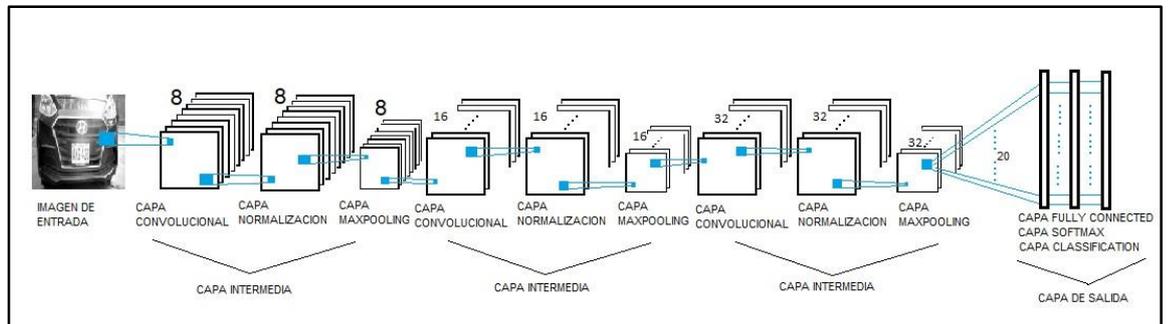


Figura 22. Capas convolucionales de la red neuronal convolucional N°1.

Fuente: Creación propia

Con los comandos ya creados, se procedió a entrenar la red neuronal convolucional. El entrenamiento duró cuatro minutos y treinta y tres segundos, el cual fue con cuarenta y cinco iteraciones como se observa en la figura 23.

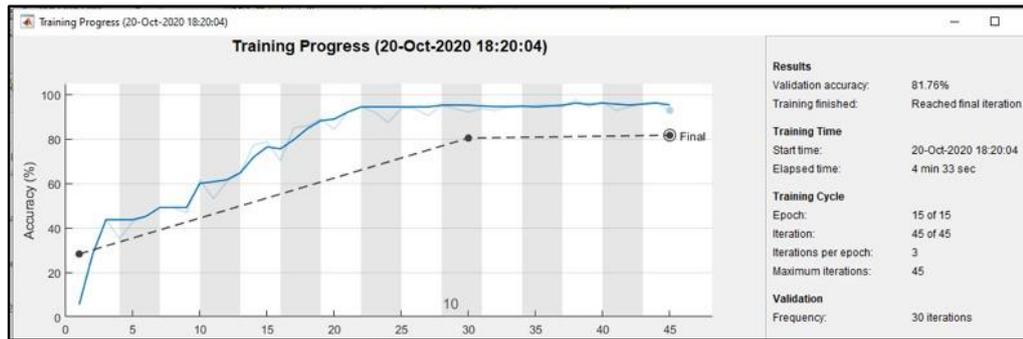


Figura 23. Entrenamiento del Modelo de red neuronal N° 1.

Fuente: Creación propia

Una vez entrenada la red neuronal convolucional, se realiza la validación del correcto funcionamiento de la misma, en esta ocasión se extrae una imagen ajena a la carpeta de imágenes entrenada (carpeta “entrenamiento”) para con ello validar que la detección de los caracteres de la placa sea correcta. Esta validación se puede observar en la figura 24.

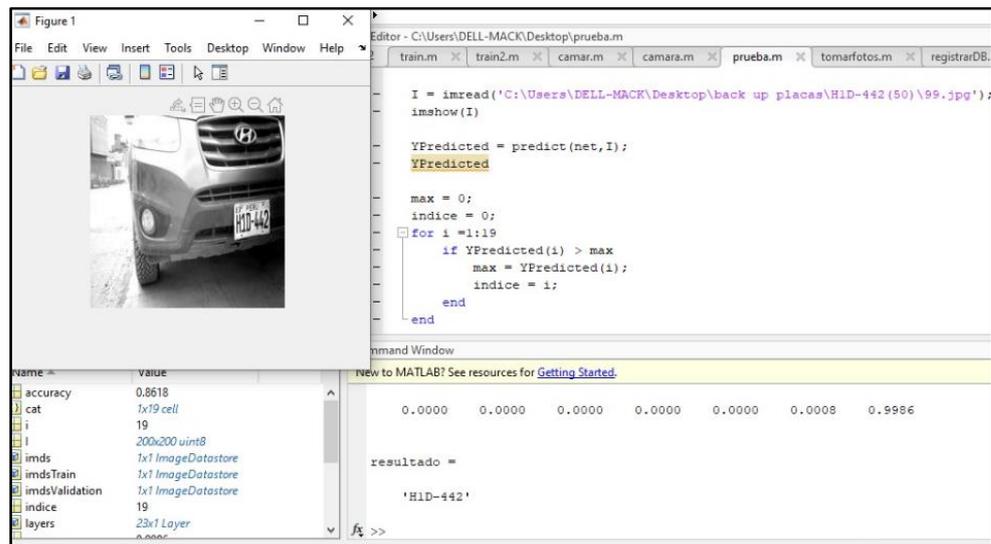


Figura 24. Simulación de toma de fotografía para la detección de los caracteres de la placa de rodaje.

Fuente: Creación propia

Este modelo de red neuronal fue analizado con veinte tipos de placas vehiculares, utilizando treinta fotografías para el entrenamiento de cada placa y con una red de tres “capas intermedias”, en la validación se obtuvieron resultados correctos y equivocados. Pero al validar los resultados totales, estos fueron en su mayoría erróneos dado que no existía congruencia entre la imagen

de la placa evaluada y el resultado obtenido del entrenamiento de la red neuronal convolucional.

3.3.3 Modelo de red neuronal convolucional N° 2

Este modelo de red neuronal convolucional fue creado con el fin de disminuir los errores obtenidos en la evaluación de resultados del modelo anterior, en esta ocasión la validación del entrenamiento de la red neuronal se realizó de igual manera con veinte placas vehiculares distintas de las cuales se utilizó treinta fotos de cada una para el entrenamiento de la red neuronal convolucional.

En la figura 25, se muestra la programación realizada para las veinte placas vehiculares, donde se aplica una cantidad de cinco “capas intermedias” con el fin de obtener mejores resultados en la detección de placas vehiculares.

```
clear all;
clc;
close all;

cat = {'AMD-299','APQ-027','ARE-620','ATH-189','AXS-426','AZJ-188', ...
      'BOK-104','B4Y-251','B8R-572','BLD-051', ...
      'BLE-130','C1A-462','C2K-088','C7S-484', ...
      'C9M-026','D1N-426','D6N-168','F2H-171', ...
      'F8L-115','H1D-442'};

rootFolder = 'C:\Users\DELL-MACK\Desktop\entrenamiento';

imds = imageDatastore(fullfile(rootFolder,cat),...
    'LabelSource','foldernames');

numTrainFiles = 22;
[imdsTrain, imdsValidation] = splitEachLabel(imds, numTrainFiles, 'randomize');

layers = [
    %capa de entrada
    imageInputLayer([200 200 1])
```

Figura 25. Entrenamiento de red neuronal convolucional con veinte placas vehiculares.

Fuente: Creación propia

A continuación, en la figura 26 se muestra el diagrama de las capas convolucionales que componen el modelo de red neuronal convolucional N°2, considerando que cada capa intermedia está compuesta por un conjunto de capas antes indicado.

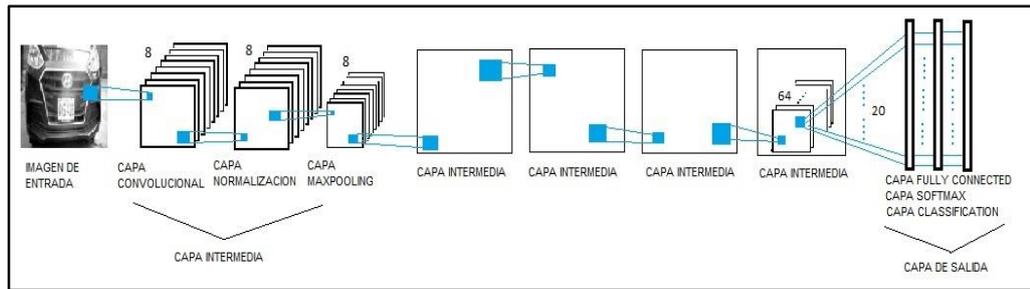


Figura 26. Capas convolucionales de la red neuronal convolucional N°2.

Fuente: Creación propia

Una vez concluido con la programación de la red neuronal convolucional, se procedió a entrenarla como se muestra en la figura 27. Con el entrenamiento de la red neuronal convolucional terminado, se realizó la validación de detección de la placa a través de la cámara web, los resultados obtenidos fueron parcialmente positivos. Esto quiere decir que existían placas que sí eran detectadas correctamente, pero otras que en ocasiones eran detectadas correctamente pero en otras ocasiones no lo eran, con ello se llegó a la conclusión de que a pesar de entrenar la red con una mayor cantidad de filtros convolucionales, esta todavía no tenía resultados exactos.

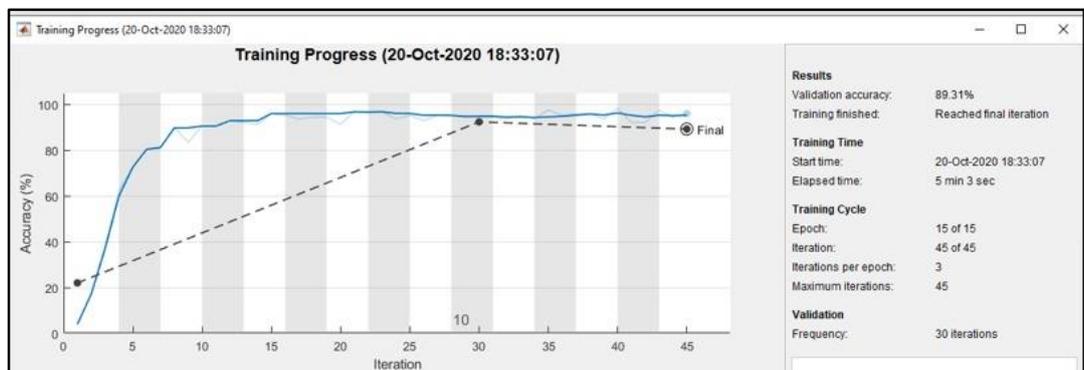


Figura 27. Entrenamiento de la red neuronal convolucional N° 2.

Fuente: Creación propia

La solución a este inconveniente era aumentar la cantidad de imágenes a entrenar o aumentar la cantidad de capas de entrenamiento, además observamos que al cambiar el rango del índice de actividad la cantidad de errores de detección de placas disminuía, este índice se muestra en la figura 28. Estos cambios se verán reflejados en el modelo de red siguiente.

```

27 -     if max < 0.52
28 -         indice = 0;
29 -     end
30
31
32 -     if indice == 1
33 -         resultado='AMD-299'
34 -     elseif indice == 2
35 -         resultado='APQ-027'
36 -     elseif indice ==3
37 -         resultado='ARE-620'
38 -     elseif indice == 4
39 -         resultado='ATH-189'
40 -     elseif indice == 5
41 -         resultado='AXS-426'

```

Figura 28. Índice de asertividad al 0.52

Fuente: Creación propia

3.3.4 Modelo de red neuronal convolucional N° 3

El modelo de red neuronal N° 3 fue creado en base al modelo anterior, con las variaciones de la cantidad de “capas intermedias” y del índice de asertividad, esto ya que en el modelo de red neuronal N° 3 se aplicará ocho “capas intermedias” para el entrenamiento de la red neuronal convolucional, lo cual genera un mejor y más efectivo entrenamiento de la red neuronal. En la figura 29 se muestra el diagrama de redes neuronales convolucionales que componen el modelo N°3.

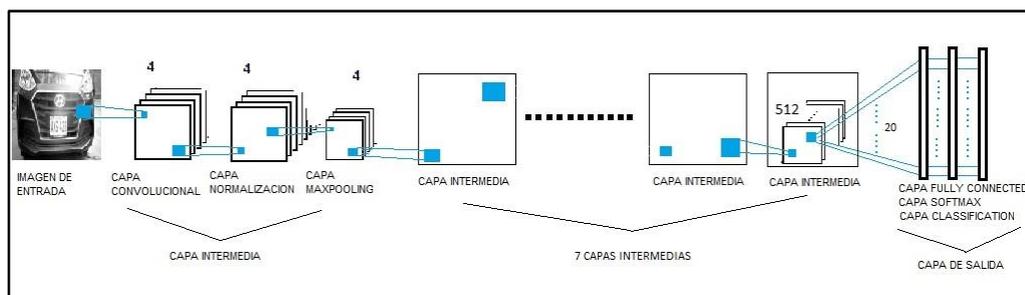


Figura 29. Capas convolucionales de la red neuronal convolucional N°3.

Fuente: Creación propia

En referencia al índice de asertividad, este determina el nivel de precisión en la detección de las imágenes a evaluar obtenida de la cámara web. Si este índice es más cercano al uno, la red neuronal convolucional se vuelve más precisa en la detección de imágenes. Se continuó usando veinte placas vehiculares para el

entrenamiento de la red, y se aplicó los ocho filtros como se muestra en la figura 30 parte I, II y III.

<pre> %Capas intermedias convolution2dLayer(3,4,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,8,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,16,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,32,'Padding','same') batchNormalizationLayer </pre>	<pre> reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,64,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,128,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) convolution2dLayer(3,256,'Padding','same') batchNormalizationLayer reluLayer maxPooling2dLayer(2,'Stride',2) </pre>
---	---

(I)

(II)

```

convolution2dLayer(3,512,'Padding','same')
batchNormalizationLayer
reluLayer

%Capa de salida
fullyConnectedLayer(20)
softmaxLayer
classificationLayer
];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',15, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain, layers, options);

```

(III)

Figura 30. Programación del modelo de red neuronal convolucional N° 3(Parte I, II y III).

Fuente: Creación propia

Con la red entrenada, se procedió a validar la programación realiza en Matlab en referencia a la captura de imagen con la cámara web, donde se realizó el cambio del indicador de asertividad a 0.71, como se detalla en la figura 31.

```

2 - close all;
3
4 - try
5 -     cam=webcam('USB CAMERA'); %nomb:
6 -     catch exception
7 -     end
8
9 -     cam.Resolution='1920x1080';
10 -    img = snapshot(cam);
11 -    pause(1);
12
13 -    img2 = imresize(img, [200 200]);
14 -    img2 = rgb2gray(img2);
15 -    YPredicted = predict(net,img2);
16 -    YPredicted
17
18 -    max = 0;
19 -    indice = 0;
20 -    for i =1:20
21 -        if YPredicted(i) > max
22 -            max = YPredicted(i);
23 -            indice = 1;
24 -        end
25 -    end
26
27 -    if max < 0.71
28 -        indice = 0;
29 -    end

```

(Parte I)

```

27 -    if max < 0.71
28 -        indice = 0;
29 -    end
30
31
32 -    if indice == 1
33 -        resultado='AMD-299'
34 -    elseif indice == 2
35 -        resultado='APQ-027'
36 -    elseif indice ==3
37 -        resultado='ARE-620'
38 -    elseif indice == 4
39 -        resultado='ATH-189'
40 -    elseif indice == 5
41 -        resultado='AXS-426'
42 -    elseif indice == 6
43 -        resultado='AZJ-189'
44 -    elseif indice == 7
45 -        resultado='BOL-104'
46 -    elseif indice == 8
47 -        resultado='BOL-104'
48 -    elseif indice == 9
49 -        resultado='BOL-104'
50 -    elseif indice == 10
51 -        resultado='BOL-104'
52 -    elseif indice == 11
53 -        resultado='BOL-104'
54 -    elseif indice == 12
55 -        resultado='BOL-104'
56 -    elseif indice == 13
57 -        resultado='BOL-104'
58 -    elseif indice == 14
59 -        resultado='BOL-104'
60 -    elseif indice == 15
61 -        resultado='BOL-104'
62 -    elseif indice == 16
63 -        resultado='BOL-104'
64 -    elseif indice == 17
65 -        resultado='BOL-104'
66 -    elseif indice == 18
67 -        resultado='F2H-171'
68 -    elseif indice == 19
69 -        resultado='F8L-115'
70 -    elseif indice == 20
71 -        resultado='H1D-442'
72 -    else
73 -        resultado='Desconocido'
74 -    end
75
76 -    preview(cam)
77 -    imshow(img2)
78 -    if strcmp(resultado,'Desconocido')
79 -    else
80 -        registrarDB(resultado)
81 -    end
82

```

(Parte II)

Figura 31. Programación de la cámara web e índice de asertividad Parte I y Parte II (Línea 27).

Fuente: Creación propia

Este modelo de red neuronal convolucional, tuvo mejores resultados en comparación con los resultados obtenidos en los dos modelos anteriores, se reconoció correctamente las placas evaluadas. Estos resultados serán indicados en el capítulo correspondiente a resultados, donde además se incluirá la interfaz usada en el proyecto. En la figura 32, se muestra el reconocimiento correcto de una placa vehicular tomada al azar utilizando la red neuronal convolucional N° 3.

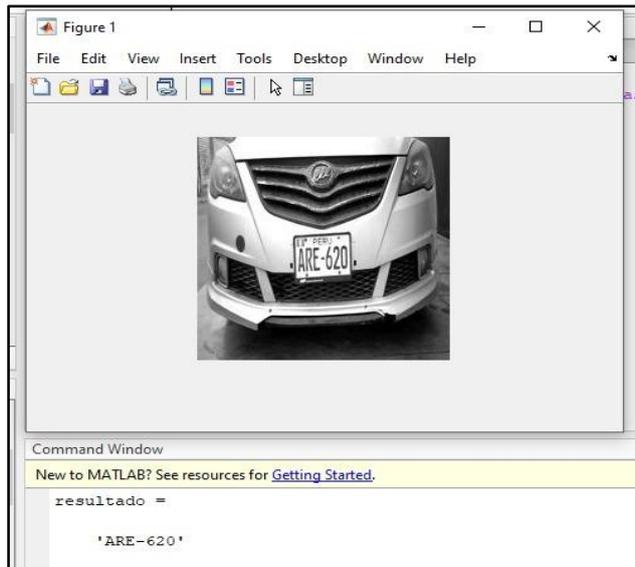


Figura 32. Detección de placa 'ARE-620' con el modelo de red neuronal N°3.

Fuente: Creación propia

3.4 Establecimiento de la base de datos utilizando DB Browser for SQLite

Las capturas de imágenes realizadas a las placas vehiculares, en primera instancia estuvieron siendo archivadas en una carpeta del escritorio del computador. Por un tema de orden y fácil accesibilidad a la información se vio conveniente crear una base de datos adicional, que sirva como registro de los vehículos que ingresan a la Universidad Ricardo Palma y por ende que fueron validadas por la red neuronal convolucional a través de la captura de imágenes de la cámara web, se realizó la creación de una tabla donde se indicó el registro de hora, fecha y placa vehicular tal como se muestra en la figura 33.

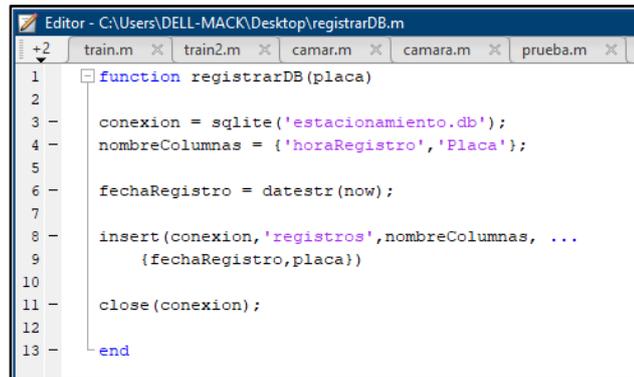
 A screenshot of the DB Browser for SQLite application. The window title is "DB Browser for SQLite - C:\Users\DELL-MACK\Desktop\estacionamiento.db". The "Table:" dropdown is set to "registros". The table has two columns: "horaRegistro" and "Placa". The data rows are as follows:

	horaRegistro	Placa
35	18-Sep-2020 19:07:48	H1D-442
36	18-Sep-2020 19:08:08	H1D-442
37	18-Sep-2020 19:08:17	D1N-426
38	18-Sep-2020 19:08:38	D1N-426
39	18-Sep-2020 19:50:38	B4Y-251
40	18-Sep-2020 19:50:59	B4Y-251
41	18-Sep-2020 19:51:30	H1D-442
42	18-Sep-2020 19:51:36	H1D-442

Figura 33. Creación de una base de datos a partir de imágenes.

Fuente: Creación propia

Esta base de datos, obtiene los resultados acertados de la validación de placas realizada por la red neuronal convolucional. Esto se realiza con la programación en el software Matlab como se muestra en la figura 34 para la conexión entre la red neuronal con la base de datos creada en el software SQLite.



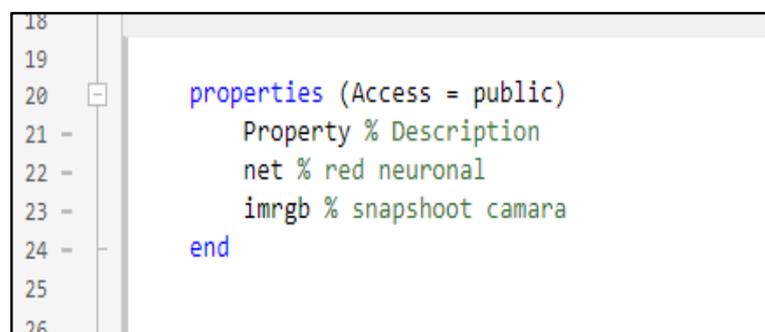
```
1 function registrarDB(placa)
2
3     conexion = sqlite('estacionamiento.db');
4     nombreColumnas = {'horaRegistro','Placa'};
5
6     fechaRegistro = datestr(now);
7
8     insert(conexion,'registros',nombreColumnas, ...
9           {fechaRegistro,placa})
10
11     close(conexion);
12
13 end
```

Figura 34. Programación en Matlab para conectar con la base de datos.

Fuente: Creación propia

3.5 Diseño de la portada del proyecto a través del gadget App Designer

Una vez obtenido el modelo de red neuronal convolucional con resultados correctos, se procede a crear la interfaz del proyecto, esto se realizó a través del gadget App Designer que nos brinda el software Matlab, donde se debe indicar en primera instancia los botones que se tendrá en el perfil del proyecto. Esto se logra con la librería de los componentes, tal como se muestra en la figura 35, en nuestro caso estos botones son: “entrenamiento de la red” y “captura de imagen” (se muestra en Matlab como ‘net’ e ‘imrgb’ respectivamente).



```
18
19
20 properties (Access = public)
21     Property % Description
22     net % red neuronal
23     imrgb % snapshot camara
24 end
25
26
```

Figura 35. Programación de interfaz de la red con App Designer. Parte 1.

Fuente: Creación propia

En seguida, se introduce la programación para obtener un mensaje al momento del reconocimiento de la placa vehicular, tal como se muestra en la figura 36: “Se reconoció el vehículo con placa...” o en su defecto “No se pudo reconocer ningún vehículo”.

```
if strcmp(placa,'Desconocido')
    mensaje = "No se pudo reconocer ningun vehiculo";
    app.lblMensaje.Text = mensaje;
    app.RegistroLamp.Color = 'red';
else
    mensaje = "Se reconoció el vehiculo con placa ";
    mensaje = strcat(mensaje,placa);
    app.lblMensaje.Text = mensaje;
    app.RegistroLamp.Color = 'green';
end

app.Snap.ImageSource = imagen;
```

Figura 36. Programación de interfaz de la red con App Designer. Parte 2.

Fuente: Creación propia

Luego, se configura la asociación del botón “entrenar red” con la programación realizada en Matlab que realiza el entrenamiento de la red neuronal convolucional creada, se considera que una vez entrenada la red neuronal, el botón debe desaparecer de la interfaz ya que ese procedimiento solo se realiza una sola vez. Se hace lo propio con la captura de imagen, como se muestra en la figura 37.

```
% Button pushed function: btnEntrenar
function btnEntrenarButtonPushed(app, event)
    app.net = entrenarRed;
    app.btnEntrenar.Visible = 'off';

    try
        cam = webcam('USB CAMERA');
    catch ME
    end

    while 1
        app.imrgb = snapshot(cam);
        app.Video.ImageSource = app.imrgb;
    end
```

Figura 37. Programación de interfaz de la red con App Designer. Parte 3.

Fuente: Creación propia

Con la programación realizada, el diseño de la interfaz quedaría tal y como se muestra en la figura 38, se considera una ventana donde se muestra la imagen en tiempo real, y una ventana donde se realizará la captura de la placa vehicular a través del botón registrar.



(I)



(II)

Figura 38. Interfaz del programa creado mediante App Designer (I) y (II).

Fuente: Creación propia

CAPITULO IV: PRUEBAS Y RESULTADOS

En el presente capítulo se muestran los resultados obtenidos en las validaciones realizadas con los tres modelos de red neuronal convolucional implementados en el capítulo anterior, para posteriormente realizar la comparación de efectividad entre ellas, utilizando la expresión matemática que mide la efectividad en porcentaje. Esta expresión se muestra en la ecuación 1.

$$Efectividad\ de\ la\ red(\%) = \frac{Placas\ detectadas\ con\ acierto}{Total\ de\ placas\ evaluadas} \times 100\% \quad (1)$$

Donde:

- Placas detectadas con acierto: Son las placas detectadas por el modelo de red neuronal convolucional con resultado acertado.
- Total de placas evaluadas: Es el total de placas evaluadas con el modelo de red neuronal.

A continuación, se muestra la validación de las tres redes neuronales convolucionales.

4.1 Validación de la Red Neuronal Convolucional N° 1

La implementación de la red neuronal convolucional N° 1, se realizó utilizando tres “capas intermedias”. Por lo tanto, para la validación se ha considerado un grupo de veinte imágenes capturadas con ángulos diferentes y ajenos al grupo de entrenamiento. De esta manera, se tuvo resultados en su mayoría errados en la detección de placas vehiculares. De las veinte placas evaluadas, doce de ellas fueron erradas y solo ocho tuvieron resultados acertados en la detección. En las figuras 39 y 40 se muestran dos de las placas detectadas correctamente en la evaluación, mientras que en las figuras 41 y 42 se muestran dos de las placas detectadas con resultado: “desconocido”.

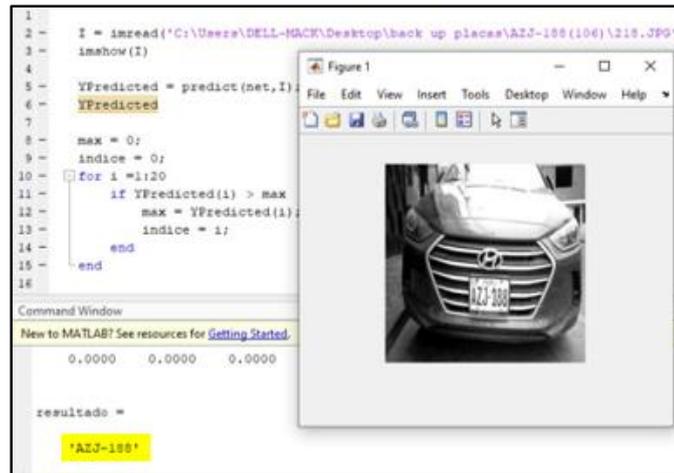


Figura 39. Placa 'AZJ-188' detectada con el modelo de red neuronal convolucional N° 1

Fuente: Creación propia

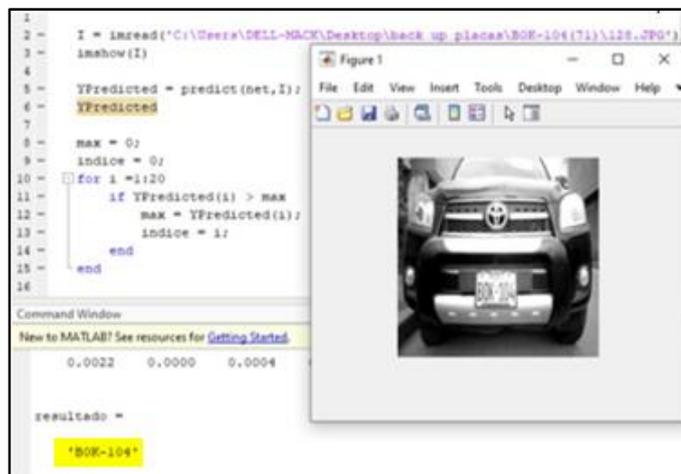


Figura 40. Placa 'BOK-104' detectada con el modelo de red neuronal convolucional N° 1

Fuente: Creación propia

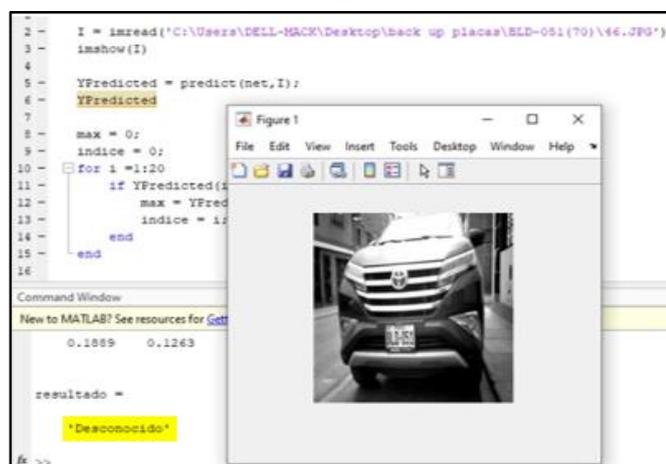


Figura 41. Placa vehicular 'BLD-051' no detectada

Fuente: Creación propia

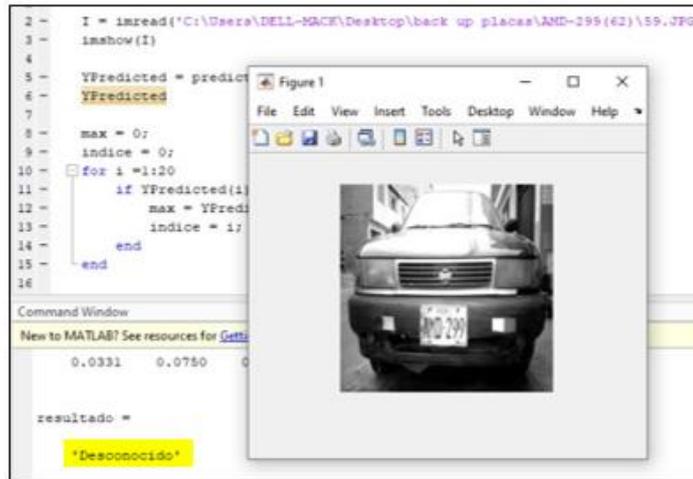


Figura 42. Placa vehicular 'AMD-299' no detecta

Fuente: Creación propia

Teniendo en cuenta que todas las placas entrenadas fueron evaluadas con la red neuronal convolucional creada, se pudo obtener el porcentaje de efectividad de la red neuronal convolucional a través de la ecuación número 1, obteniendo lo siguiente:

- Placas vehiculares detectadas correctamente: 8 placas vehiculares
- Placas vehiculares evaluadas en total: 20 placas vehiculares.

$$Efectividad\ de\ la\ red = \frac{8}{20} \times 100\% = 40\% \text{ de efectividad}$$

4.2 Validación de la Red Neuronal Convolucional N° 2

El entrenamiento de la red neuronal convolucional N° 2 se realizó con cinco “capas intermedias”, esto debido a que se optimizó el modelo de red anterior a fin de obtener mejores resultados. Se evaluó de la misma forma las veinte placas vehiculares teniendo un resultado mejor que el anterior, esta vez las placas reconocidas efectivamente fueron quince, dos de ellas se muestran en la figura 43 y la figura 44. Mientras que cinco de ellas no fueron detectadas o tuvieron como resultado un número de matrícula distinto al evaluado como se muestra en la figura 45 y figura 46 respectivamente.

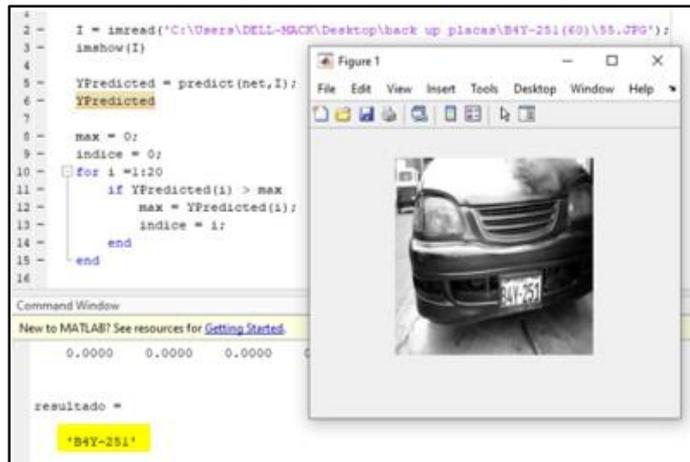


Figura 43. Placa 'B4Y-251' detectada con el modelo de red neuronal convolucional N° 2

Fuente: Creación propia

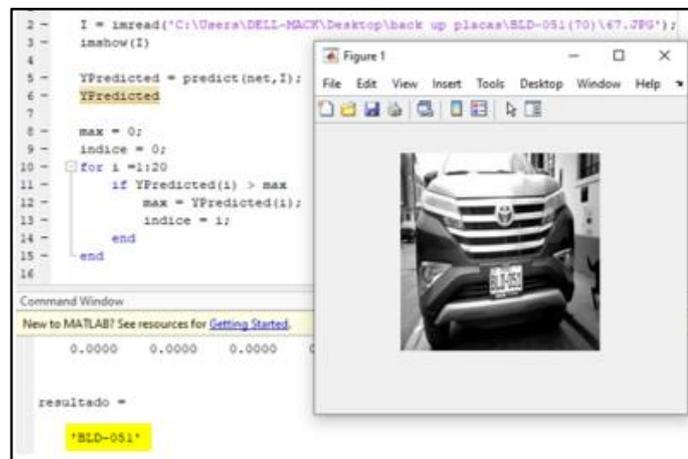


Figura 44. Placa 'BLD-051' detectada con el modelo de red neuronal convolucional N° 2

Fuente: Creación propia

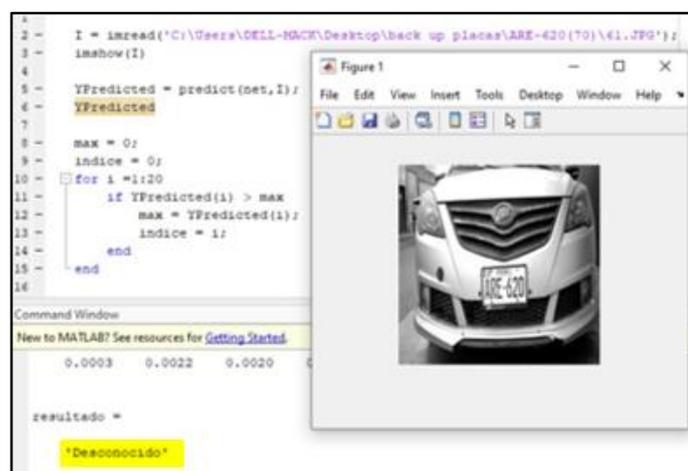


Figura 45. Placa vehicular 'ARE-620' no detectada.

Fuente: Creación propia

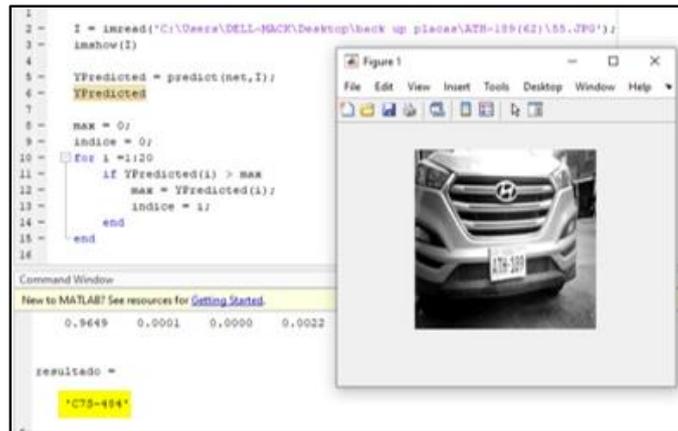


Figura 46. Placa vehicular 'ATH-189' no detectada, se detectó la placa 'C7S-484'

Fuente: Creación propia

Como se puede observar, el modelo de red neuronal convolucional N° 2 presenta mejoras en la detección de las placas vehiculares, un ejemplo es en la detección de la matrícula 'BLD-051' que en el modelo anterior no fue detectada. Se validó que una de las placas fue detectada, pero con el resultado erróneo ya que este resultado fue una matrícula diferente a la evaluada.

Con ese resultado, se evaluó el porcentaje de efectividad según lo siguiente:

- Placas vehiculares detectadas correctamente: 15 placas vehiculares
- Placas vehiculares evaluadas en total: 20 placas vehiculares.

$$Efectividad\ de\ la\ red = \frac{15}{20} \times 100\% = 75\% \text{ de efectividad}$$

4.3 Validación de la Red Neuronal Convolucional N° 3

La red neuronal convolucional N° 3 fue entrenada con ocho “capas intermedias”, además se realizó el cambio del índice de asertividad haciéndolo más cercano a la unidad para obtener así mejores resultados. En la evaluación, se obtuvo los resultados esperados ya que, de las pruebas realizadas con las veinte placas vehiculares, diecinueve de ellas fueron detectadas efectivamente a la primera validación, mientras que una de ellas tuvo que ser evaluada nuevamente para que su detección sea correcta. Las mejoras se observan en la figura 47 y figura 48 donde se detectó la placa vehicular 'ATH-189' y 'ARE-620' respectivamente, placas que no pudieron ser detectadas con la red neuronal convolucional N° 2.



Figura 47. Placa ‘ATH-189’ detectada con la red neuronal convolucional N° 3

Fuente: Creación propia

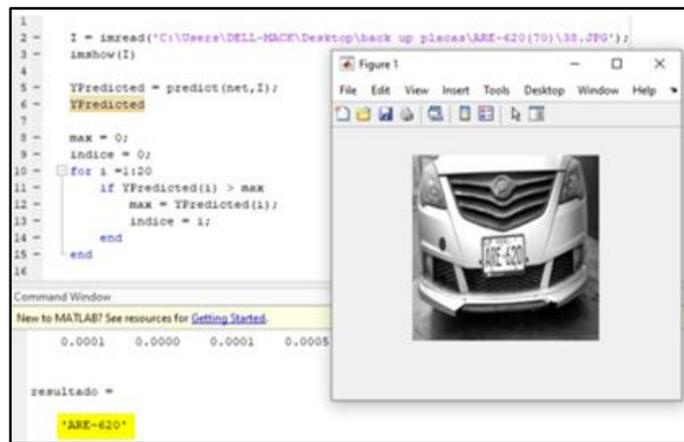


Figura 48. Placa ‘ARE-620’ detectada con la de red neuronal convolucional N° 3

Fuente: Creación propia

A continuación, se muestra en la figura 49 la evaluación de la red neuronal convolucional N° 3 con la detección de la placa ‘HID-442’ donde el resultado es “desconocido”, mientras que luego de una validación más a la misma placa vehicular se obtuvo una respuesta correcta como se muestra en la figura 50.

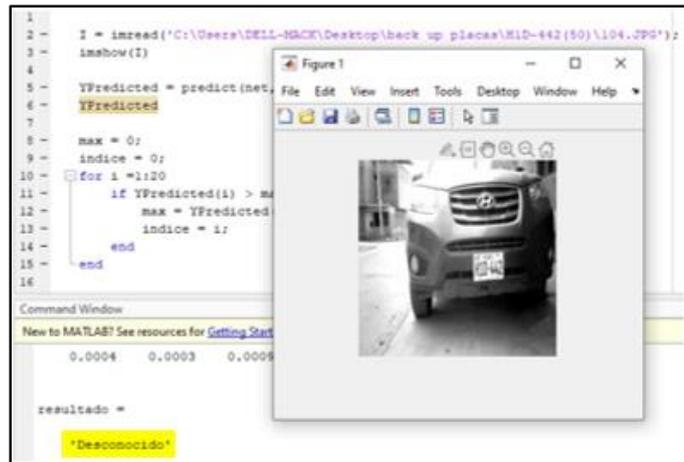


Figura 49. Placa vehicular 'HID-442' no detectada

Fuente: Creación propia

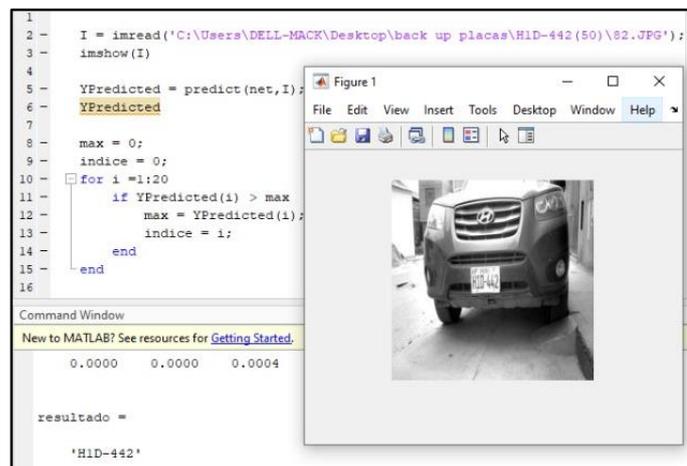


Figura 50. Placa vehicular 'HID-442' detectada correctamente con el modelo de red neuronal convolucional N° 3

Fuente: Creación propia

Finalmente, con ese mayor nivel de detección de placas en relación a las otras redes neuronales convolucionales evaluadas, se obtiene el porcentaje de efectividad de la red, como se muestra a continuación:

- Placas vehiculares detectadas correctamente: 19 placas vehiculares
- Placas vehiculares evaluadas en total: 20 placas vehiculares.

$$\text{Efectividad de la red} = \frac{19}{20} \times 100\% = 95\% \text{ de efectividad}$$

Con los resultados obtenidos de la validación de las tres redes neuronales convolucionales realizadas, se obtiene en la siguiente tabla:

Tabla 3. Comparación de efectividad de las redes neuronales convolucionales.

Validación de red	Red neuronal convolucionaria 1 N° 1	Red neuronal convolucionaria 1 N° 2	Red neuronal convolucionaria 1 N° 3
Placas detectadas correctamente	8	15	19
Placas detectadas erróneamente	0	1	0
Placas no detectadas "desconocido"	12	4	1
Nivel de efectividad de la red	40%	75%	95%

Fuente: Creación propia

4.4 Validación de las redes neuronales convolucionales con un grupo externo

Los modelos de redes neuronales convolucionales mostrados anteriormente fueron validados con placas vehiculares que pertenecían al grupo de placas entrenados.

Para verificar el comportamiento de los modelos de redes neuronales al ser validados con placas vehiculares ajenas al grupo de entrenamiento se realiza la validación con diez placas vehiculares.

A continuación, se muestra un resumen de los resultados obtenidos en las validaciones de las redes neuronales convolucionales.

En la figura 51, se muestra el resultado de la evaluación de la placa 'AXO-010' con la red neuronal convolucional N° 1, donde se obtiene un resultado de "desconocido", en

la figura 52 y figura 53 se muestra el resultado obtenido con el modelo de red neuronal convolucional N° 2 y N° 3 respectivamente.

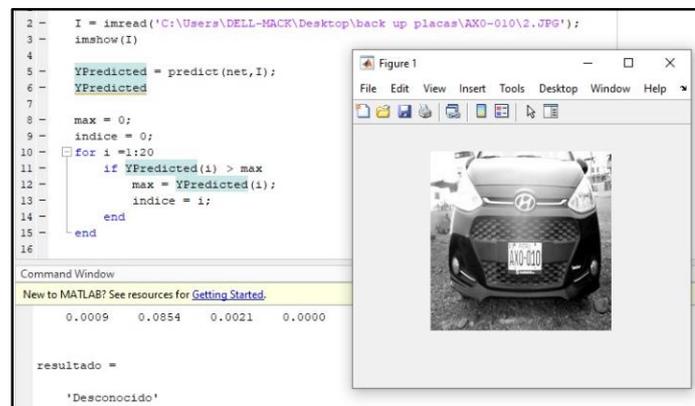


Figura 51. Resultado de evaluación de placa 'AXO-010' con red neuronal convolucional N° 1.

Fuente: Creación propia

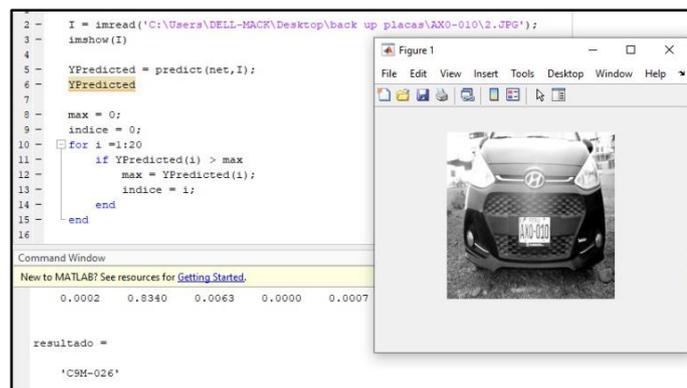


Figura 52. Resultado de evaluación de placa 'AXO-010' con red neuronal convolucional N° 2.

Fuente: Creación propia

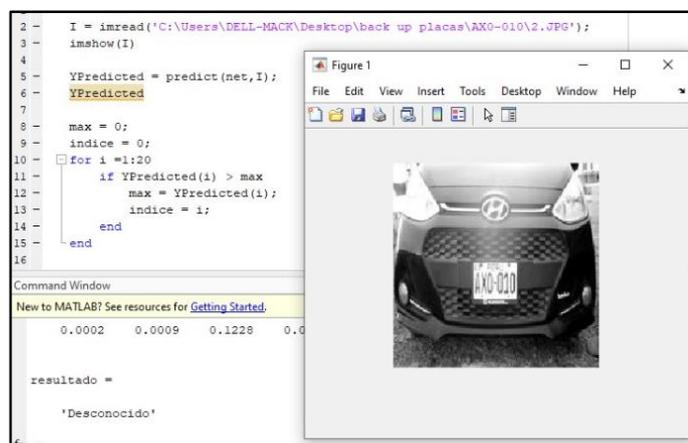


Figura 53. Resultado de evaluación de placa 'AXO-010' con red neuronal convolucional N° 3.

Fuente: Creación propia

Se observa, que al validar la primera placa ajena al grupo de entrenamiento ‘AXO-010’, esta tiene diferentes resultados en comparación con los tres modelos de redes neuronales convolucionales, validado con la red neuronal convolucional N° 1 y red neuronal convolucional N° 3 tiene como resultado “desconocido”, mientras que al validarlo con la red neuronal convolucional N° 2 este tiene como resultado la placa ‘C9M-026’ como se observan en las figuras anteriores.

En seguida, se procedió a realizar la evaluación con otra placa externa a las placas entrenadas. Esta vez la validación fue con la placa ‘A2P-445’, la cual es validada con los tres modelos de red neuronal convolucional como se muestra en las figuras 54,55 y 56 a continuación:

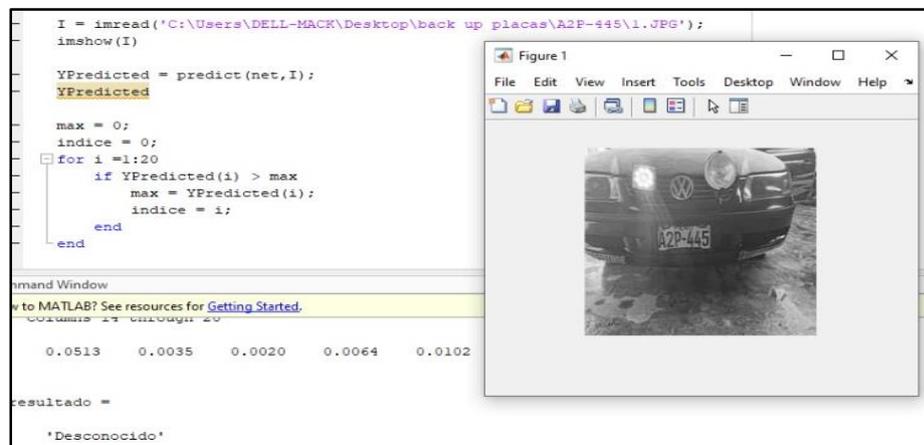


Figura 54. Resultado de evaluación de la placa ‘A2P-445’ con la red neuronal convolucional N° 1.

Fuente: Creación propia

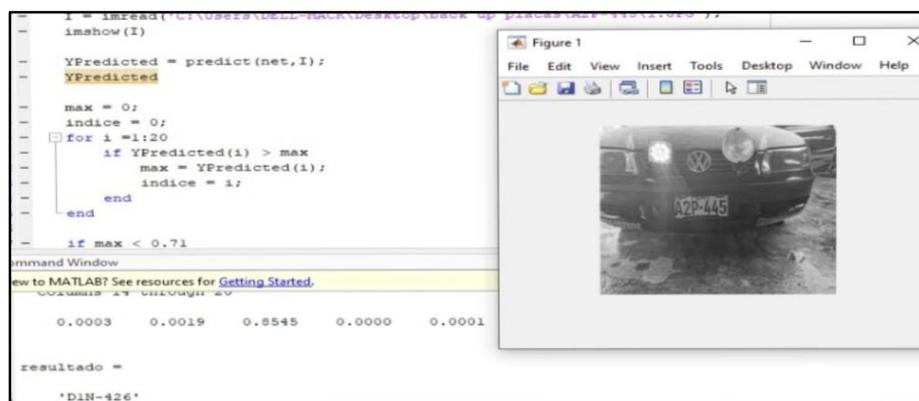


Figura 55. Resultado de evaluación de la placa ‘A2P-445’ con la red neuronal convolucional al N° 2.

Fuente: Creación propia



Figura 56. Resultado de evaluación de la placa ‘A2P-445’ con la red neuronal convolucional N° 1

Fuente: Creación propia

A continuación, en la Tabla 4 se muestra los resultados obtenidos en la validación de las placas adicionales con las redes neuronal convolucionales (RNC) modeladas.

Tabla 4. Resultado de la validación de placas ajenas con los modelos de red neuronal convolucional creados.

N°	PLACAS VEHICULARES	REDES NEURONALES CONVOLUCIONALES		
		RESULTADOS		
		RNC 1	RNC 2	RNC 3
1	AX0-010	Desconocido	Detecto placa C9M-026	Desconocido
2	A2P-445	Desconocido	Detecto placa D1N-426	Desconocido
3	ALA-233	Desconocido	Desconocido	Desconocido
4	C1G-509	Desconocido	Desconocido	Desconocido
5	BMJ-242	Desconocido	Desconocido	Desconocido
6	D5L-377	Detecto placa B4Y-251	Desconocido	Desconocido
7	F4C-179	Desconocido	Desconocido	Desconocido
8	AAE-436	Desconocido	Detecto placa D6N-168	Desconocido
9	A0D-462	Detecto placa F8L-115	Desconocido	Desconocido
10	A7Q-293	Desconocido	Desconocido	Desconocido

Fuente: Creación propia.

4.5 Prototipo del módulo de reconocimiento de placas de automóviles

Por el contexto en el cual estamos actualmente, no fue posible realizar un módulo e implementarlo dentro de la garita de seguridad en la Universidad Ricardo Palma, por ello se hizo la simulación del mismo, esta simulación fue realizada en primera instancia mediante el uso de una mesita de madera donde se colocó la laptop y con ella la cámara web, luego con el trípode donde se colocó solamente la cámara web, este trípode estuvo colocado a la entrada de la puerta del estacionamiento vehicular de la Universidad Ricardo Palma. La cámara web fue conectada mediante puerto USB a la laptop, desde donde se utilizó el modelo de red neuronal N° 3 programado con Matlab. Desde allí se simuló el ingreso de tres vehículos al estacionamiento de la Universidad Ricardo Palma, utilizando un indicador de pare vehicular como tope para que el auto se detenga y realizar la captura de la imagen, finalmente se validó los resultados acertados de detección de placas en los tres casos. En las figuras 57 y 58 se muestran las simulaciones de ingreso vehicular al estacionamiento, y en la figura 59 se observa el resultado obtenido en la laptop mediante el software Matlab con el gadget App Designer desarrollado en este proyecto.



Figura 57. Simulación de ingreso vehicular al estacionamiento de la Universidad Ricardo Palma

– Parte 1.

Fuente: Creación propia



Figura 58. Simulación de ingreso vehicular al estacionamiento de la Universidad Ricardo Palma
– Parte 2.

Fuente: Creación propia.

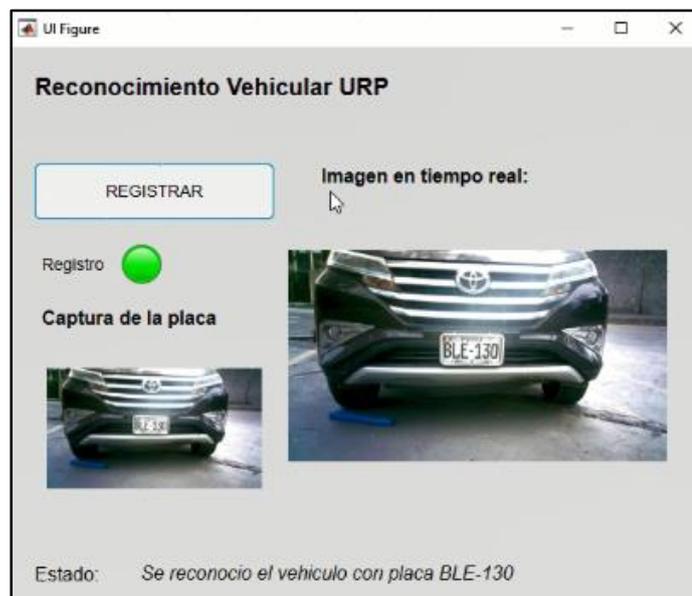


Figura 59. Resultado obtenido en App Designer – Reconocimiento correcto.

Fuente: Creación Propia

4.6 Costos del proyecto

A continuación, se muestra en la tabla 5 los costos de materiales utilizados en la elaboración del proyecto de tesis: “Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la Universidad Ricardo Palma.”

Tabla 5. Costos de materiales del proyecto de tesis.

COSTOS DE MATERIALES				
ITEM	DESCRIPCION	CANTIDAD	COSTO POR UNIDAD	COSTO TOTAL
1	CÁMARA IP	1	S/. 270.00	S/. 270.00
2	CÁMARA WEB	1	S/. 80.00	S/. 80.00
3	SOFTWARE MATLAB	1	S/. 0.00	S/. 0.00
4	LAPTOP DELL	1	S/. 0.00	S/. 0.00
5	TRIPODE DE CÁMARA	1	S/. 70.00	S/. 70.00
6	INDICADOR DE PARE VEHICULAR	1	S/. 15.00	S/. 15.00
7	TRANSPORTE	3	S/. 25.00	S/. 75.00
8	COSTO TOTAL			S/. 510.00

Fuente: Creación propia.

CONCLUSIONES

1. Se estableció una base de datos a partir de la elección del dispositivo de toma de fotografías, este se dio evaluando tres posibles opciones: cámara IP, cámara de dispositivo celular y cámara web, donde finalmente se escogió la cámara web tal como se indica en la sección 3.2, ya que fue la que obtuvo mejor resultado a comparación de las otras dos. A través de la cámara web, se estableció una carpeta denominada “entrenamiento” como se muestra en la figura N°17, donde se almacenaron seiscientas imágenes, es decir treinta imágenes de cada placa, teniendo en cuenta que son veintidós placas, utilizadas para los entrenamientos y ocho placas para la validación de los modelos de red neuronal convolucional.
2. Se entrenó tres modelos de red neuronal convolucional a través del Toolbox Deep Learning del Matlab como se evidencia en la sección 3.3. Para la validación de imágenes se formularon tres modelos de redes convolucionales, estos modelos estuvieron compuestos por una capa de entrada, una capa de salida y capas intermedias. El primer modelo de red neuronal se entrenó con tres capas intermedias demostrado en la sección 3.3.2, el segundo modelo de red se entrenó con cinco capas intermedias como se muestra en la sección 3.3.3 y finalmente en la sección 3.3.4 se muestra el tercer modelo de red neuronal convolucional que se entrenó con ocho capas intermedias, donde se evaluó y comprobó que el modelo más efectivo fue el modelo de red neuronal N°3, ya que al tener más capas intermedias hizo que la red neuronal convolucional tuviera mayor efectividad al momento de reconocer los caracteres de las placas de rodaje teniendo menor índice errores, además se consideró un índice de asertividad más cercano a uno, haciendo a la red más efectiva en la toma de decisiones, por lo tanto la red neuronal convolucional fue más precisa en el reconocimiento de caracteres al comparar con los dos modelos de redes neuronales convolucionales anteriores, como se muestra en la sección 4.3. Finalmente se concluye que se implementó el modelo de red neuronal convolucional N°3 para la identificación de placas de rodajes de vehículos particulares.

3. Se realizó la creación de una interfaz a través del gadget App Designer del Matlab desde donde se visualizó la cámara web en tiempo real y la captura de fotografía. Se simuló el ingreso vehicular de tres autos particulares al estacionamiento de la Universidad Ricardo Palma y se visualizó el correcto funcionamiento de la red neuronal convolucional N°3 como se muestra en la sección 4.5, ya que en la interfaz se logró observar el resultado acertado en la detección de las placas de rodaje. Asimismo, en la tabla N°2 se evaluó las otras diecisiete placas de rodaje teniendo como resultado final un 95% de efectividad y 5% de margen de error debido a diferentes factores como luminosidad, ángulo de captura de imagen, estado de las placas, entre otros. Por lo tanto, se concluye que se logró validar el funcionamiento de la red neuronal convolucional a través del reconocimiento de las placas de rodaje de los vehículos que simularon el ingreso a la Universidad Ricardo Palma.

RECOMENDACIONES

1. Se recomienda que se realice la captura de imágenes con el mismo dispositivo con el que serán detectadas las placas de rodaje para la creación de la base de datos, a fin de que la red neuronal pueda reconocer con mayor efectividad los caracteres.
2. Se sugiere que las placas elegidas para la toma de datos se encuentren en condiciones óptimas, ya que si presentan imperfecciones que involucren a los caracteres, no habrá un adecuado reconocimiento de las placas de rodaje.
3. Para la validación del reconocimiento de placas vehiculares se recomienda realizar el mayor número de filtros posibles a fin de equilibrar el tiempo de procesamiento del entrenamiento de la red neuronal convolucional con la efectividad en el reconocimiento de las placas de rodaje.
4. Para un futuro estudio de investigación se recomienda tener la mayor cantidad de imágenes posibles de cada placa de rodaje, para que la red neuronal convolucional tenga un mejor entrenamiento y un correcto reconocimiento de las placas vehiculares, además se sugiere realizar el entrenamiento de la red neuronal convolucional teniendo la misma cantidad de imágenes por placa vehicular.

BIBLIOGRAFÍA

- Aldás Salazar, D. S., Collantes Vaca, S. M., & Reyes Vásquez, J. P. (2016, Julio-Diciembre). PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE PLACAS VEHICULARES. *Revistas Científicas*. Recuperado el Setiembre 26, 2020, de <https://revistasdigitales.upec.edu.ec/index.php/sathiri/article/download/15/28/106>
- Bagnato, J. (s.f.). *Healthg Big Data*. Recuperado el Agosto 8, 2020, de Healthg Big Data: <https://www.juanbarrios.com/redes-neurales-convolucionales/>
- Benitez, M., & Arias, A. (2017). *Curso de Introduccción a la Administración de Base de Datos* (segunda ed.). IT Campus Academy. Recuperado el Noviembre 14, 2020, de <https://books.google.com.pe/books?id=35YSDgAAQBAJ&pg=PA93&dq=base+de+datos+sqlite&hl=es-419&sa=X&ved=2ahUKEwi03aeru7rsAhVQJrkGHdAdAO8Q6AEwAXoECAQQAg#v=onepage&q=base%20de%20datos%20sqlite&f=false>
- Carioto, E. G. (2003). *VIAJES Y VIAJEROS EN EL MUNDO ANTIGUO*. (S. d. Publicaciones, Ed.) La Mancha, España: Ediciones de la Universidad de Castilla-La Mancha.
- Coronel, C., Morris, S., & Rob, P. (2011). *Base de datos - Diseño, implementación y administración* (novena ed.). Cengage Learning Editores. Recuperado el Noviembre 16, 2020, de https://books.google.es/books?hl=es&lr=&id=KINC0Gc_RREC&oi=fnd&pg=PR6&dq=base+de+datos&ots=V4d82zWBLx&sig=ceJI2Lz_QhErPsQ9UqoXYWi3Ld0#v=onepage&q=base%20de%20datos&f=false
- ENCICLOPEDIA BRITANICA. (1976). *Enciclopedia Británica*. Recuperado el Agosto 03, 2020, de Enciclopedia Británica: <https://global.britannica.com/technology/automotive-industry>
- Espinoza Vásquez, G. (2014). *Sistema de Reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio*. Tesis, Pontificia Universidad Católica del Perú, Lima, Lima. Recuperado el Agosto 03, 2020
- Flores Lopez, R., & Fernandez Fernandez, J. M. (2008). *Las redes neuronales artificiales-Fundamentos teóricos y aplicaciones prácticas*. La Coruña, España: NETBIBLO, S.L. Recuperado el Agosto 05, 2020, de <https://books.google.com.pe/books?id=X0uLwi1Ap4QC&pg=PA11&dq=red+neuronal&hl=es-419&sa=X&ved=2ahUKEwjd8K7Qh4XrAhWnI7kGHcCnA-QQ6AEwBHoECAUQA#v=onepage&q=red%20neuronal&f=false>
- Lumingo.com*. (s.f.). Recuperado el octubre 13, 2020, de https://www.lumingo.com/producto/camara-web-full-hd-1080p-webcam-con-microfono-usb-pc-laptop/p/000000000004755781?gclid=CjwKCAjwrKr8BRB_EiwA7eFapgSmM

f6Ab6gzNDTb4dJj3kw99PD6Pt1wfmMYeJmvsdDneMBD1hr_RoC1GUQAv
D_BwE

- MathWorks*. (s.f.). Recuperado el Agosto 05, 2020, de MathWorks:
<https://www.mathworks.com/discovery/deep-learning.html>
- Ministerio de Justicia. (2008). *DECRETO SUPREMO N° 017-2008-MTC*. Decreto Nacional, Ministerio de Justicia, Lima. Recuperado el Agosto 04, 2020, de http://transparencia.mtc.gob.pe/idm_docs/normas_legales/1_0_1488.pdf
- Ministerio de Justicia y Derechos Humanos. (2008). *ANEXO I : CUADRO DE DISTRIBUCIÓN DE COLORES Y REPRESENTACIÓN GRÁFICA DE LA PLACA ÚNICA NACIONAL DE RODAJE*. Lima. Recuperado el Agosto 05, 2020, de <http://spij.minjus.gob.pe/Graficos/Peru/2008/Abril/19/DS-017-2008-MTC.pdf>
- Ministerio de Transportes y Comunicaciones. (s.f.). *DECRETO SUPREMO QUE APRUEBA EL REGLAMENTO DEL SISTEMA DE PLACA ÚNICA NACIONAL DE RODAJE*. Lima. Recuperado el Agosto 04, 2020, de <https://portal.mtc.gob.pe/transportes/terrestre/documentos/Exposici%C3%B3n%20de%20motivos%20-%20Placas.pdf>
- Mundaca Vidarte, G. (2016). *Detección de caracteres de placas de automóviles mediante técnicas de visión artificial*. Piura. Recuperado el Agosto 02, 2020, de https://pirhua.udep.edu.pe/bitstream/handle/11042/2578/IME_203.pdf?sequence=1&isAllowed=y
- Pérez Villaba, J. E., & Brito Moncayo, G. D. (2014). *Reconocimiento de placas vehiculares mediante procesamiento de imágenes para optimizar el acceso a los parqueaderos de la UTA, Campus Huachi*. Universidad Técnica de Ambato, Ambato, Ecuador. Recuperado el Setiembre 26, 2020, de <https://repositorio.uta.edu.ec/handle/123456789/7792>
- Qadri, M. T., & Asif, M. (2019, Setiembre 29). AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM FOR VEHICLE IDENTIFICATION USING OPTICAL CHARACTER RECOGNITION. *IEEE Explore*. Recuperado el Noviembre 25, 2020, de <https://ieeexplore.ieee.org/document/8969068>
- Romero Marra, J. (2015, Junio 25). *Romerobots Blog*. Recuperado el Julio 10, 2020, de <https://jjromeromarras.wordpress.com/2015/06/25/deteccion-de-matriculas/>
- Salazar Marquez, M. (2014). *Desarrollo de un algoritmo para la localización automática de placas vehiculares peruanas usando técnicas de procesamiento de imágenes*. Pontificia Universidad Católica del Perú, Lima, Lima. Recuperado el Agosto 03, 2020, de <http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/6215>
- Servicio de Administracion Tributaria de Lima. (2019, Diciembre 31). *Servicio de Administracion Tributaria de Lima*. Recuperado el Agosto 03, 2020, de Servicio de Administracion Tributaria de Lima: <https://www.sat.gob.pe/websitev9/Inicio/DatosAbiertos>

ANEXOS

- Cámara web usada en el proyecto modelo genérico, fue obtenida por compra online en la página web Lumingo.com



Figura 60. Cámara Web Full HD 1080p Webcam.

Fuente: (*Lumingo.com, s.f.*)