



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO



Daniel Alejandro Contreras Véliz

Implementación de un sistema de identificación de patentes en RPI

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica
Facultad de Ingeniería**

Valparaíso, 19 de abril de 2018



Implementación de un sistema de identificación de patentes en RPI

Daniel Alejandro Contreras Véliz

Informe Final para optar al título de Ingeniero Civil Electrónico,
aprobada por la comisión de la
Escuela de Ingeniería Eléctrica de la
Facultad de Ingeniería de la
Pontificia Universidad Católica de Valparaíso
conformada por

Sr. Gabriel Hermosilla Vigneau
Profesor Guía

Sr. Sebastian Fingerhuth Massmann
Segundo Revisor

Sr. Jorge Mendoza Baeza
Director de Escuela

Valparaíso, 23 de mayo de 2018

Este trabajo está dedicado a:

Mis padres que, a pesar de las dudas, siempre me han apoyado.

A mis hermanos que siempre han visto en mí algo más que yo mismo.

A mis amigos, aquellos con quienes he compartido tantos buenos y malos ratos.

A mi familia que siempre han creído en mí, más que yo mismo.

A quienes fueron un pilar fundamental en mi vida y partieron mientras recorría este camino, mis abuelos Etelvina y Hernan, siempre serán una parte importante de mí.

Por último, al Daniel del futuro para que recuerdes que algo has logrado aunque te haya costado mucho, mucho, mucho.

Agradecimientos

Agradezco, por sobre todo, a mis padres por darme la oportunidad de ser quien soy, a pesar de nuestras diferencias y las dudas que pueden haberse presentado, siempre me han apoyado. A la “*Rosalba*” con quien las conversaciones por teléfono pueden durar horas hablando, riéndonos e incluso discutiendo en ocasiones, por ser un pilar fundamental en mi vida. Y, por supuesto, al “*Rulo*” con su apoyo en todo momento, por ser un papá apañador en las situaciones de diversión y a la hora de comer cosas nuevas. A ambos agradezco todo el esfuerzo que han hecho por sacar adelante todo lo que soy.

También les doy gracias a mis hermanos, al “*Cisco*” que a pesar de nuestras diferencias siempre nos hemos apoyado y lo seguiremos haciendo y al “*Enano*” que, aunque ya mida bastante más que yo, siempre será mi hermanito chico y quien me apaña en mis ideas ñoñas, locuras y en los momentos de hambre.

Mi familia, siempre entregándome su apoyo y una confianza que va mucho más allá de la que yo mismo siento. Unos más que otros han estado presentes en este camino, aun así a todos les doy infinitas gracias.

Gracias a aquellos que han formado parte de mi vida universitaria, tanto a quienes estuvieron conmigo en esas clases que a veces se hacían eternas y a los que conocí en mis tantos paseos por la ciudad.

Valparaíso, 28 de junio de 2018

Daniel Contreras Véliz

Resumen

El siguiente trabajo trata del estudio llevado a cabo para el desarrollo de un sistema de reconocimiento automático de patentes. Presenta una gran variedad de métodos y técnicas con las cuales es posible construir cada una de las fases de este sistema. Se presenta la problemática analizada, las opciones disponibles para dar solución a esta y la alternativa seleccionada para desarrollar el sistema.

A lo largo del primer capítulo se presentan la problemática, las propuestas de solución y un amplio compilado de trabajos de diferentes autores que desarrollaron distintas técnicas para lograr un sistema de identificación de patentes. En el recorrido por estos trabajos se obtuvo conocimientos y herramientas que fueron aplicadas en las tareas de este sistema.

El segundo capítulo está destinado a entregar una explicación técnica acerca de los métodos utilizados para el desarrollo del sistema. Este tipo de modelo consta de tres tareas básicas: detección de la patente, pre-procesamiento o segmentación de la placa e identificación de caracteres; por lo que se explica el funcionamiento, a grandes rasgos, de las tecnologías utilizadas en las tres técnicas seleccionadas para llevar a cabo la tarea de lectura de patentes.

El tercer capítulo está destinado a la explicación paso a paso del funcionamiento del sistema desarrollado, cómo este lleva a cabo la tarea de detección de patentes en la imagen que captura con la cámara, de qué manera se realiza la segmentación de caracteres de la patente detectada. Por último, cómo trabaja la etapa de reconocimiento de caracteres y la manera en que finalmente se entrega la información de la lectura por parte del sistema.

En el cuarto, y último, capítulo se presentan todos los experimentos realizados, tanto en laboratorio como pruebas de campo. Se realizaron una serie de experimentos para evaluar las etapas de detección de patentes y la identificación de caracteres. Se presentan resultados y análisis para estos, a experimentos desarrollados en computador y en la Raspberry. Además se realiza un paso a paso del cómo se realizaron las pruebas, considerando el número de muestras y la procedencia de estas y, por último, pasar a un capítulo dedicado a conclusiones obtenidas acerca del funcionamiento del sistema y sus partes.

Palabras claves: RPI (Raspberry Pi), ALPR (Automatic License Plate Recognition), Redes neuronales convolucionales, Visión por computadora.

Abstract

The following work deals with the study carried out for the development of an automatic patent recognition system. It presents a great variety of methods and techniques with which it is possible to build each of the phases of this system. The analyzed problem is presented, the options available to solve this and the selected alternative to develop the system.

Throughout the first chapter the problematic, the solution proposals and a wide compilation of works of different authors that developed different techniques to achieve a patent identification system are presented. In the course of these works, knowledge and tools were obtained that were applied in the tasks of this system.

The second chapter is intended to provide a technical explanation about the methods used for the development of the system. This type of model consists of three basic tasks: patent detection, pre-processing or segmentation of the plate and identification of characters; so it explains the operation, roughly, of the technologies used in the three techniques selected to carry out the task of reading patents.

The third chapter is intended for the step-by-step explanation of the functioning of the developed system, how it carries out the task of detecting patents in the image it captures with the camera, in which way the segmentation of characters of the detected patent is performed. Finally, how the character recognition stage works and how the information of the reading is finally delivered by the system.

In the fourth, and last, chapter, all the experiments performed are presented, both in the laboratory and in field tests. A series of experiments were carried out to evaluate the stages of patent detection and character identification. Results and analysis are presented for these, to experiments developed in computer and in the Raspberry. In addition, a step-by-step description of how the tests were carried out is made, considering the number of samples and their origin, and finally, a chapter dedicated to conclusions obtained about the operation of the system and its parts.

Key words: RPI (Raspberry Pi), ALPR (Automatic License Plate Recognition), Convolutional neural networks, Computer visión.

Índice general

Introducción.....	1
Objetivos generales.....	3
1 Marco Teórico	4
1.1 Problemática y propuesta de solución.....	4
1.2 Estado del arte	4
1.2.1 Detección en sistemas de identificación de patentes.	5
1.2.2 Procesamiento y segmentación de la matrícula.....	8
1.2.3 Identificación de caracteres	9
1.2.4 Sistema propuesto y algoritmos.....	9
2 Metodologías utilizadas	11
2.1 Visión por computador	11
2.1.1 Detección de la patente.	12
2.1.2 AdaBoost	12
2.2 Segmentación.....	13
2.2.1 OpenCV	14
2.2.2 Conversión a escala de grises y estandarización de tamaño	14
2.2.3 Sliding window	14
2.2.4 Redes neuronales	15
2.3 Deep learning	17
2.3.1 Capa convolucional.....	17
2.3.2 Función de activación.....	18
2.3.3 Capa de reducción (Pooling)	18
2.3.4 <i>Fullyconnected Layer</i>	19
2.3.5 Dropout	19
2.3.6 Keras	19
2.3.7 Tensorflow	19
2.4 Raspberry Pi 3.....	19
2.4.1 Implementación	22
3 El sistema ALPR propuesto	23

3.1 Detección de la patente.....	24
3.2 Segmentación de caracteres	25
3.3 Identificación de caracteres.....	26
3.3.1 MobileNet	27
3.3.2 Transfer Learning	28
3.4 Funcionamiento del sistema	29
3.5 Implementación en Raspberry	30
4 Resultados experimentales.....	31
4.1 Experimentos de laboratorio	31
4.1.1 Experimento 1: Validación de la etapa de identificación de caracteres.	31
4.1.2 Experimento 2: Análisis del sistema completo en laboratorio con archivos de vídeo e imágenes.	32
4.1.3 Experimento 3: Análisis del sistema completo en laboratorio con archivos de vídeo e imágenes con Raspberry Pi 3.....	35
4.2 Experimento en terreno	36
4.2.1 Experimento 4: Análisis del sistema completo en terreno con Raspberry Pi 3.	37
4.3 Análisis de resultados de los experimentos	39
Discusión y conclusiones.....	41
Bibliografía	44
A Códigos de funcionamiento del sistema.....	48
B Código de transfer learning para la red.....	51
C Detalle del desempeño de la red por caracter.....	52
D Tabla comparativa placas de desarrollo.....	54

Introducción

“Una imagen es una representación bidimensional de una escena del mundo tridimensional. La imagen es el resultado de la adquisición de una señal proporcionada por un sensor, que convierte la información del espectro electromagnético en codificaciones numéricas. Y su transformación en el formato elegido de representación de imágenes, que constituye información discreta tanto en los valores que puede tomar, como en los parámetros que la definen.” [1]

La cita anterior, extraída del libro “Conceptos y métodos en Visión por Computador”, hace referencia al área central en la que se encuentra el proyecto que se presenta en este informe. La visión por computador es un recurso muy presente en el diario vivir de las personas. Es posible encontrarla en las funciones de reconocimiento facial de teléfonos inteligentes y cámaras, las que toman una fotografía al momento de detectar una sonrisa en una imagen, como en lectores de huellas dactilares utilizados en centros de salud y entidades financieras. También en redes sociales, donde un claro ejemplo son las etiquetas a los rostros de las personas que aparecen en las fotografías. Sólo por nombrar algunos ejemplos cercanos, donde el procesamiento de imágenes y el reconocimiento de patrones, que implican la visión por computador, aparecen en nuestras vidas.

Dentro de los tantos usos de la visión por computador aparece el reconocimiento de patrones consistentes en la identificación de bordes, formas y configuraciones de estructuras repetitivas. Algunos ejemplos de patrones se encuentran presentes en las huellas digitales y retinas oculares que varían de persona a persona y también la clasificación de caracteres, presente en este trabajo. Y aunque, en estas últimas, puedan existir similitudes, todas tienen un conjunto de características únicas que las diferencian unas de otras. Lo que lleva al uso que se le puede dar a la identificación de caracteres, como lo es el reconocimiento automático de patentes de vehículos. Este no es un problema nuevo, es posible encontrar este tipo de sistemas en accesos a estacionamientos de centros comerciales y algunas empresas, como también en sistemas de cobro automático por uso de autopistas e identificación de vehículos robados entre otros.

Por lo tanto, la lectura de patentes es el enfoque de este trabajo. La principal razón para decidir el desarrollo de un sistema de lectura automático de patentes, es el implementar una solución de este tipo de bajo costo. Esto por medio del uso de hardware de fácil acceso y software de código libre. Además, desde un enfoque académico, la posibilidad de mejorar o variar los métodos de

funcionamiento de manera libre. Por esto, la propuesta a desarrollar es llevar a cabo un lector de patentes, basado en visión por computador y utilizando herramientas actuales.

Se presentan tres opciones, para llevar a cabo la implementación de un sistema como el planteado anteriormente: el desarrollo completo de cada una de sus partes desde cero, tomar un sistema completo ya desarrollado por alguien más y evaluar su funcionamiento y, por último, seleccionar técnicas enfocadas para tareas específicas para armar uno propio. Estas tareas específicas son detección de objetos (en este caso patentes), segmentación de caracteres e identificación de estos. La última opción presentada es la que fue elegida para ser utilizada.

Dentro de la documentación existente es posible encontrar diferentes maneras de realizar con éxito las distintas tareas del sistema, muchas cuentan con tasas de acierto altas en detección de placas, pre-procesamiento de patentes y/o segmentación de caracteres y reconocimiento de matrículas o lectura de letras y números. Entre todas las opciones hay, incluso, resultados con similares desempeños logrados con técnicas muy distintas, algunas sencillas y otras más complejas de implementar.

Con todo lo anteriormente mencionado se buscó una alternativa capaz de ser portada en hardware de bajo costo, que cuente con tecnologías existentes, las que no necesiten de un gran uso de recursos computacionales para funcionar y a la vez tengan un desempeño robusto. El sistema a implementar debía constar de una etapa de detección eficiente y eficaz, es decir, que logre acertar en la mayoría de los casos, logrando encontrar y extraer la placa patente de una imagen obtenida desde un sensor. También precisa una etapa de pre-procesamiento o segmentación con la que sea posible obtener muestras limpias de cada uno de los caracteres de importancia en la matrícula. Por último, una etapa de identificación de números y letras eficiente pero por sobre todo eficaz, con la cual obtener resultados precisos en un alto porcentaje, sobre el 90% de los casos.

Lo que fue mencionado con anterioridad acerca de eficiencia y utilización de recursos computacionales, por parte de cada una de las tareas y del sistema en su conjunto, guarda directa relación al hardware que se busca utilizar para la implementación final del modelo. Se busca un dispositivo portátil, de bajo costo y capaz de realizar el procesamiento requerido. Es por estas razones que un objetivo es la utilización de una tarjeta de desarrollo que cumpla con lo antes mencionado.

Las tarjetas de desarrollo como las ofrecidas por Arduino, Raspberry o NVIDIA son kits de herramientas enfocadas a que desarrolladores y diseñadores tengan a su alcance formas de llevar a cabo sus proyectos. De este modo, estas han sido utilizadas en variados fines desde proyectos escolares, emuladores de consolas hasta proyectos más avanzados para investigación de bajo costo. Es por su expansión que existen cada vez más de estas placas con una capacidad de procesamiento mayor, con las que es posible implementar sistemas más avanzados, procesos específicos de cálculo computacional. Incluso existen algunas de estas que incluyen unidades de procesamiento gráfico con las cuales aumenta su capacidad de procesamiento y llegan a compararse a computadores personales de bajo rendimiento.

Con la utilización de estos microcomputadores, en cuanto a hardware la implementación de un sistema como el que se propone, se reducen mucho los costos en comparación a la utilización de un computador de escritorio o un servidor para el procesamiento de la información. Además, con el uso de librerías de código abierto es posible implementar procesos completos de visión por computador, las que incluyen detección, reconocimiento de objetos y procesamiento de imágenes.

Por tanto, y reuniendo las características antes mencionadas, el objetivo es la implementación de un sistema automático de reconocimiento de patentes en una Raspberry Pi 3. En el cuerpo de este informe se presentan las técnicas a utilizar en la investigación, el desarrollo y los experimentos.

Objetivos generales

- Detectar e identificar patentes de vehículos chilenos con el uso de un sistema embebido en un dispositivo Raspberry y aplicando la clasificación con deep learning.

Objetivos específicos

- Estudiar algoritmos de detección.
- Estudiar algoritmos de segmentación.
- Estudiar algoritmos de identificación.
- Búsqueda de configuración de red neuronal adecuada.
- Entrenamiento de red neuronal seleccionada.
- Realización de experimentos en laboratorio acerca de reconocimiento de caracteres.
- Realización de experimentos en laboratorio de lectura de patentes.
- Portar el sistema a la Raspberry ya preparada.
- Realizar pruebas en terreno buscando sobre 90% de acierto.

1 Marco Teórico

En este capítulo se plantea la problemática que dio origen a este proyecto, describiendo su desarrollo y los antecedentes de tecnologías utilizadas que permitieron analizar y especificar sistemas de procesamiento de imágenes y de redes neuronales para detección de rostros y patentes.

1.1 Problemática y propuesta de solución.

El problema a estudiar está enfocado en el control de acceso a recintos privados, tales como: estacionamientos de condominios, pequeñas empresas o estacionamientos para trabajadores de la universidad, es decir, como poder controlar quién ingresa o más específicamente quién está autorizado para acceder. La propuesta de solución a esta problemática está en la implementación de un sistema que lea, identifique y registre el contenido de las patentes de aquellos vehículos que ingresan a estos recintos.

Ahora bien, el implementar un sistema de identificación de patentes trae consigo problemas de costos por la instalación de cámaras, computadores, otros sensores y un software robusto que lleve a cabo la tarea. Para aminorar estos costos se propone la creación de un sistema de lectura e identificación de patentes embebido en una tarjeta de desarrollo, utilizando para este fin una Raspberry Pi 3 Model B, la cual combinada con otras técnicas proporcionen resultados de un 90% de efectividad. Un sistema que es comparable a los comerciales pero menos costoso. De allí nace la búsqueda de diversas maneras de llevar a cabo las tareas que implica un sistema como el señalado.

1.2 Estado del arte

Existe una gran variedad de metodologías para la detección y reconocimiento de patentes, algunos, como el propuesto en [2] donde se utilizan redes neuronales para detección de rostros y patentes. También existen casos como el presentado en [3] en el cual la Raspberry solo tiene la función de realizar la detección y el pre-procesamiento de la imagen para que en un computador se lleve a cabo la identificación y comparación con la base de datos de la información. Estos análisis son los que se presentan a continuación, de acuerdo a documentación existente acerca de la programación de sistemas de lecturas de patentes y las técnicas computacionales utilizados para lograr correctos desempeños.

1.2.1 Detección en sistemas de identificación de patentes.

Dentro de la literatura es posible encontrar variadas formas de realizar la tarea de detección de la patente en la escena que capta la cámara, para los sistemas ALPR (*Automatic License Plate Recognition*); algunas basadas en la binarización de la imagen completa de que utilizan la diferenciación de brillos con el fin de realizar la extracción de la placa. Otra forma es el procesamiento en escala de grises, ya sea de manera total como parcial de la imagen de entrada.

En el trabajo realizado en “License plate detection using Adabost” [4] se hace un estudio al respecto del funcionamiento de los clasificadores AdaBoost en conjunto con el algoritmo de Viola y Jones, esto con el fin de llevar a cabo la tarea de detección de matrículas dentro de una imagen. Para la validación del algoritmo utilizaron esta técnica para la detección de placas de automóvil, en un universo de 158 muestras, con lo que obtuvieron una tasa de detección de 95.6%, comentando que la mayor cantidad de falsos positivos generados en los experimentos se deben a imágenes en las que aparece vegetación. Concluyen que el método AdaBoost es una opción viable con el cual llevar a cabo esta tarea en un sistema de lectura de patentes.

Por otro lado está lo realizado en [5] donde se especifica que se utilizan características globales y locales para entrenar con el algoritmo AdaBoost y de aquí sacar clasificadores fuertes los cuales, por último, se utilizan para un clasificador en cascada. Específicamente utilizan descriptores similares a los de tipo Haar, ya que según explican esto los hace invariantes ante el brillo, color, tamaño y posición de la matrícula. La validación de este trabajo fue realizado sobre un set de datos de 460 imágenes positivas y 500 negativas. Para la etapa de entrenamiento del algoritmo se utilizaron dos grupos de 300 imágenes, uno con aquellas que contenían el objeto de estudio y otro con muestras sin placas presentes. Mientras que del total se usaron 160 muestras positivas para validación. Con lo que obtuvieron como resultado una tasa de acierto de 93.5%.

Por otra parte están aquellos estudios en que utilizan inteligencia computacional para llevar a cabo la primera de las tres tareas en el reconocimiento de placas patentes. Aquí aparece el trabajo realizado en [6] donde se utiliza un algoritmo genético codificado real. Usando este método es posible decidir umbrales para definir o buscar los colores de matrícula más probables, con el fin de llevar a cabo la detección, ya que el sistema aprende que el color de fondo, y de la escena completa, puede cambiar. Es posible usarlo para crear una función para detectar que sea robusta ante los cambios de iluminación. Los resultados de este trabajo se realizaron en base a 100 imágenes, se obtuvo 70% en cuanto a la tasa de acierto, de lo que concluyen que deben considerar un mayor rango de cambios de fondo desde días lluviosos hasta despejados con el fin de robustecer el sistema.

Dentro del mismo marco computacional tenemos el trabajo de [7] en el cual se utilizó un algoritmo genético distribuido. Con un data set de 70 imágenes se obtuvo una tasa de acierto del 92.8% en sus experimentos. Este trabajo se basa en segmentación, usando este tipo de algoritmos para extraer regiones de la placa. Según se comenta en el documento, lo consideran robusto al no verse afectado por falsos óptimos locales en el espacio de solución. Aunque los errores tienden a generarse para imágenes demasiado deformadas y en aquellas en las que se tiende a confundir

los colores entre la placa y el vehículo, por otra parte asumen que se puede mejorar en el aspecto de detección del color.

Para terminar con los métodos basados en clasificadores, se revisa el trabajo de [7] donde se aplica la utilización de redes neuronales mejoradas. Según la descripción que realizan, están compuestas por 27 nodos de entrada, 30 en la capa oculta y 6 de salida. Las características que importan en este estudio son: área, ancho y alto, histograma vertical con el fin de diferenciar una región de placa de uno que no lo es. En resumen es un método de ubicación de placas usando procesamiento de imágenes a color. Los experimentos se realizaron sobre 238 imágenes de vehículos con las cuales se obtuvo una tasa de acierto del 92.4%.

Además de los clasificadores, se tienen trabajos hechos realizados por medio del procesamiento de imágenes a color, donde primero se analizó el enfoque a las transformaciones del modelo de color, método expuesto en [8]. Se propone un algoritmo mediante el uso de un umbral adaptativo y una estrategia de búsqueda. El proceso es iniciado por un umbral adaptativo, que fue alcanzado por un proceso iterativo, aplicado a una imagen binarizada a partir de una en escala de grises donde finalmente se localiza la matrícula con la utilización de conocimientos previos acerca de la placa. El algoritmo fue probado en 100 muestras de 320 x 240 píxeles, en la cuales las placas variaron desde 41 x 13 a 80 x 25 píxeles, debido a posición y distancia de la cámara.

Dentro de este tipo de procesamiento, de imágenes a color, es que se encuentra el trabajo dirigido al procesamiento de histogramas presente en [9], basado en localización precisa de la matrícula con la utilización de la región de esta. También se usa el desplazamiento medio para segmentar y filtrar regiones candidatas de la escena a color. En este trabajo, se extraen tres características para decidir si la región candidata es o no una placa: rectangularidad, relación de aspecto y densidad de bordes. Para la validación del algoritmo propuesto se utilizaron 57 imágenes. Este método es robusto y preciso para vehículos que dieran al fondo un color distinto al de la matrícula.

El procesamiento de imágenes binarizadas es el siguiente método a analizar, lo que se encuentra en la documentación del trabajo en [10]. Se utiliza morfología matemática para el procesamiento de la imagen binarizada, con el fin de hallar la patente. La validación se realizó sobre una base de 105 muestras, con las cuales se obtuvo una tasa de acierto, en la detección, de 80.39%. El procedimiento consiste en realizar a la imagen una serie de operaciones morfológicas como aperturas, cierres, resta de fondo, entre otras.

También, en el ámbito de la binarización de la imagen de entrada, está lo realizado en [11], donde se consideran las variaciones al ángulo entre la cámara y el vehículo. Para la detección se utilizan gradientes verticales para detectar posibles regiones en las cuales encontrar la patente. Estas se evalúan, basándose en características geométricas: relación alto-ancho, tamaño y orientación. Con esto y un set de datos de 104 matrículas se obtuvo una tasa de acierto de 94.2%. Como conclusión expresan que el sistema funciona correctamente, teniendo en cuenta que la mayoría de este tipo de programas no considera grandes ángulos de giro al momento de realizar la detección de la matrícula.

El último documento relacionado a este enfoque que se revisó fue el de [12], donde al igual que en el anterior se utilizan operaciones morfológicas, pero también un algoritmo de proyección para extraer la matrícula. Por medio de estas operaciones se buscan regiones candidatas a contener la matrícula y luego busca al candidato más probable de acuerdo a la ubicación de la escena. Con una base de datos de 147 imágenes se obtuvo una tasa de detección de 93.2%.

A continuación se muestran los procedimientos llevados a cabo por medio del procesamiento de imágenes en escala de grises. En el primero se utiliza el procesamiento global de la imagen de entrada, metodología usada en [13]. Aquí a la imagen se le aplica, primero, un filtro gaussiano para disminuir el ruido y luego se extiende el histograma de la imagen con el fin de alcanzar todos los niveles de escala de grises. Para luego, escanear la imagen buscando alto contraste de gradientes y aislar la placa, del resto de la escena. La validación de este método dio como resultado una tasa de acierto del 99%, en cuanto a detección y segmentación de la placa.

Cambiando el enfoque, se realizó el estudio del procesamiento parcial de la imagen, como es el caso que se presenta en [14], el cual se basa en el análisis de la textura de características. Esta tarea se lleva a cabo en cuatro etapas, que son: pre-procesamiento, búsqueda de región aproximada, clasificación de regiones y corrección de inclinación. La tasa de acierto con este método es de un 96.1%, con una base de datos de 380 imágenes. Las causas de fallas reconocidas fueron: 1) existencia de otras fuentes de texto similar a la placa y 2) límite poco claro respecto a los bordes de la matrícula.

Por último, un trabajo que analiza la transformación de la imagen para poder procesarla parcialmente en escala de grises es el que se presenta en [15]. Según describen sus autores, el método que utilizan evalúa la simetría de las placas con el uso de la transformada de simetría generalizada. El procedimiento comienza con la búsqueda de intersecciones de bordes. Una vez ubicados, o sea una vez encontrada una esquina, se analizan vecindades hasta hallar las tres restantes con las que se forma la placa. Los resultados experimentales de este trabajo obtuvieron una tasa de acierto de 93.6% de un total de 330 imágenes que componían la base de datos.

También existen sistemas que implementan redes neuronales para procesar la imagen completa y buscar en ella la patente. Dentro de estos sistemas se encuentra el presentado en [16], donde se utilizan redes neuronales para encontrar texto dentro de la imagen y luego discriminar si es que este es parte de una matrícula o no.

Existen los casos aparte, embebidos en una Raspberry, que utilizan sensores ultrasónicos para detectar el vehículo y luego extraer con el uso de una webcam la información de la patente [17]. Y otros una combinación de fotografía y sensor como en [3]. Además existe, entre otros, el método utilizado en este proyecto, el cual involucra cascadas de características Haar o LBP entrenadas con el algoritmo Adaboost [4] y que es propuesto como alternativa además de ser entrenado en el trabajo realizado en [18].

1.2.2 Procesamiento y segmentación de la matrícula

Una vez se ha llevado a cabo la extracción de la patente de la imagen de entrada es necesario el procedimiento de segmentar los caracteres que esta contiene. De la misma manera en que se analizaron varios métodos para realizar la detección de la matrícula, se llevó a cabo la revisión de documentación con el fin de comparar, hallar u obtener ideas acerca del cómo realizarla en el proyecto. A continuación se presentan algunos enfoques y trabajos existentes al respecto.

La primera metodología hace referencia al uso de clasificadores con el fin de llevar a cabo la segmentación de un objeto del fondo de la escena, el que se presenta en [19], que hace uso de algoritmos de umbral con el fin de diferenciar el objeto de estudio con respecto a otros que lo rodean. Se realiza una clasificación de acuerdo a umbrales que determinan la posición de los objetivos en la imagen. Por otro lado en [64] proponen el seguimiento de contornos para realizar la segmentación de caracteres mediante la extracción de regiones, este también con la aplicación de umbral.

Se propone en [20] la utilización de cadenas ocultas de Markov, con el fin de segmentar una serie de caracteres en una imagen ruidosa y de baja resolución, en este caso la matrícula de vehículo. Con el uso del conocimiento previo sobre la patente, ya que se aplica el método sabiendo el número de caracteres y que existe cierta separación constante entre cada uno de ellos.

Similar a lo anterior, en [21] utilizan características de la placa como lo son el color y la forma de esta. Además se establece una relación de alto y ancho, de la placa y de los caracteres, que será una de las bases para realizar la segmentación. La que se lleva a cabo discriminando entre umbrales, de acuerdo a la intensidad en vecindades de píxeles. Después de una binarización se determinan los lugares dentro de la patente en que existe una mayor concentración de píxeles de negro. Por último si el área segmentada es demasiado grande, no comparable a un carácter, se descarta.

En la misma línea de binarización de la imagen está el trabajo de [22] que luego de separar la placa del resto de la imagen al binarizar, se limpia y se obtiene la proyección del histograma horizontal. Luego con la utilización de la proyección vertical se extraen los caracteres de la matrícula, todo esto se lleva a cabo en una imagen normalizada de la patente, con el fin de estandarizarla para toda imagen entrante al sistema. Por último, se encontró en [23] información acerca del uso de umbral adaptativo local con el que llevar a cabo la binarización. Con esto se obtienen histogramas tanto horizontales como verticales de los caracteres en la patente y, también, de manera adaptativa realizan la segmentación de acuerdo a las posiciones y tamaños obtenidos de las proyecciones de histogramas. Contando con una base de datos de 650 imágenes la tasa de acierto del sistema propuesto es de 96.76%.

1.2.3 Identificación de caracteres

Para la tarea de identificación existen técnicas como la aplicada en [18], donde se utiliza la comparación o *Template Matching* [24] para realizarla. Otro método utilizado con el objetivo de identificar caracteres es el presentado en [25], aquí se utilizan modelos de Markov, entrenados con ejemplos de cada uno de los caracteres, para realizar la clasificación.

Por otro lado se encuentra el trabajo basado en máquina de vectores de soporte que se presenta en [26]. El que presenta una tasa de acierto alrededor del 97.2%. Cada imagen de caracter extraído se ecualiza primero en histograma y se divide uniformemente en regiones de 8x8, luego se usan cuatro máquinas de vectores de soporte para reconocer los caracteres.

En el sistema mostrado en [18] se utiliza coincidencia de patrones o plantillas para la etapa de reconocimiento de caracteres, es decir, un sistema entrenado con la forma y borde de la letra o número, el cual al llevar a cabo una comparación elige el de mayor concordancia. Similar a este trabajo, en [27], luego de la extracción de algunos puntos de características, se utiliza operadores de coincidencia de plantillas para obtener una solución robusta en múltiples condiciones de adquisición, las pruebas realizadas en más de tres mil imágenes entregan una tasa de acierto de un 91% para este trabajo.

En el paper presentado por [28] se utilizan las redes neuronales artificiales para reconocimiento óptico de caracteres. Primero se aíslan las letras y números para luego ser entregadas a la red. La utilizada es una red neuronal probabilística, que son la implementación de redes neuronales de análisis discriminante.

Dentro de la documentación disponible es posible encontrar varios trabajos con un mismo enfoque, el cual es utilizar clasificadores de caracteres manuscritos para la etapa de reconocimiento de un sistema ALPR. Tales casos se encuentran en [28] donde proponen el “Método de comportamiento-conocimiento del espacio” que implica el uso de clasificadores grupales o múltiples, en vez de uno solo, con esto se agregan las decisiones obtenidas de los clasificadores individuales y derivar las mejores finales desde el punto de vista estadístico. Esta metodología se repite en [29], donde el enfoque del trabajo es, por sobre todo, aplicar esta idea y probar con varias combinaciones de clasificadores como lo son bayesianos, k-NN o de distancia.

Por último, en la misma línea anterior se encuentra [30], donde propone la utilización de varios clasificadores en combinación con redes neuronales. Los autores han desarrollado un algoritmo de red neuronal de agrupación incremental, con proceso de fusión y cancelación. Además de uno de extracción de características de histograma direccional modificado y un método de subclase con estrategia de rechazo neuronal de aprendizaje [31], [32].

1.2.4 Sistema propuesto y algoritmos

Una vez analizadas las distintas metodologías, para realizar cada una de las tareas que implica el funcionamiento de un sistema de lectura de patentes. Se propuso la implementación por medio de descriptores entrenados con Adaboost, con el uso del algoritmo de Viola y Jones, para la

detección de la placa en la escena capturada por el sensor. Este sistema fue previamente entrenado en la escuela su desempeño fue probado en [18].

El siguiente proceso es realizado por medio de un método basado en sliding window, para realizar la tarea de segmentación de los caracteres de la patente. Esto implica, considerando que la segunda etapa no implica un pre-procesamiento de la imagen ni la aplicación de técnicas de disminución de ruido, una fase de identificación robusta. Se eligió la utilización de redes neuronales convolucionales, específicamente la configuración “MobileNet” debido a su enfoque a sistemas móviles y embebidos. Todo esto implementado sobre una Raspberry Pi 3 Model B.

2 Metodologías utilizadas

Luego de revisar varias formas de llevar a cabo las tareas de detección de la patente, segmentación de la placa y reconocimiento de caracteres, se utilizó un sistema basado en AdaBoost para lo primero, sliding window para la segmentación de los caracteres y el reconocimiento de estos se realiza con el uso de una red neuronal. Todas estas tareas constituyen, en esencia, un sistema de visión por computador. Las técnicas seleccionadas conforman un sistema de reconocimiento de patentes que cumple con todas las características necesarias para un buen desempeño. A continuación se explica el funcionamiento de cada una de estas.

2.1 Visión por computador

La base para el desarrollo de este trabajo es la visión por computador, es decir, la capacidad de adquirir, procesar y “entender” (clasificar) información por parte de un computador sin la intervención de una persona que esté manejándolo. En otras palabras, es la capacidad de una máquina para identificar, extraer y procesar información que ingresa en forma de imágenes desde algún periférico diseñado para capturarlas.

Un sistema ALPR (*Automatic License Plate Recognition*) consta, generalmente, de 3 etapas fundamentales, presentadas en la Figura 2-1: detección de la placa, procesamiento de la información capturada y por último identificación de los caracteres. Nótese que en algunos trabajos se separa la parte de la segmentación, presentándola como una tarea más. Sin embargo también existen los casos en que la segunda tarea consiste específicamente o únicamente en la segmentación de los caracteres de la matrícula con el fin de realizar la clasificación.

En conjunto con la visión artificial son muy utilizadas las técnicas de procesamiento de imágenes y reconocimiento de patrones. Ya que, una vez obtenida la imagen, las primeras se utilizan para localizar y despejar las zonas de interés, mientras que las segundas sirven para identificar y clasificar las estructuras detectadas según sus características.

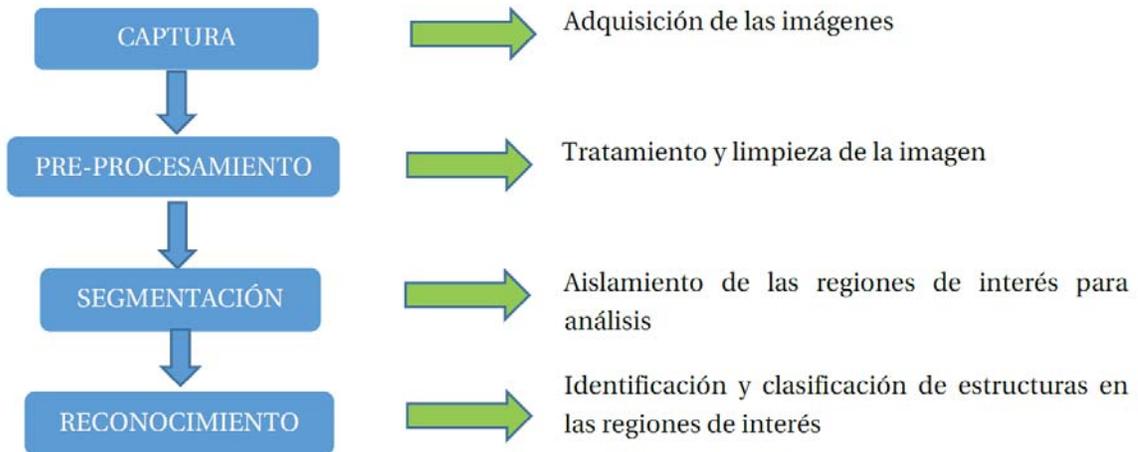


Figura 2-1 Etapas de la visión por computador.

2.1.1 Detección de la patente.

Dentro del proceso de detección, el cual se lleva a cabo sobre la imagen capturada, existen varios conceptos involucrados, los cuales es necesario conocer con el fin de comprender cómo se realiza. Estos corresponden al algoritmo desarrollado por Freund y Shapire [33], que implica la utilización de características Haar en combinación con el proceso de aprendizaje de AdaBoost y en conjunto del algoritmo desarrollado por Viola y Jones.

2.1.2 AdaBoost

Se trata de un algoritmo de clasificación, que permite encontrar una hipótesis fuerte a base de utilizar otras simples y débiles. El algoritmo Adaboost utiliza el boosting (clasificador) de manera iterativa, es decir, lo adapta (*Adaptive*) de manera que cada nuevo clasificador trabaje sobre los datos que el anterior clasificó erróneamente. Algoritmo formulado por Y. Freund y R. Shapire. AdaBoost selecciona el clasificador débil de mejor rendimiento de un conjunto de clasificadores débiles, cada uno actuando en una sola característica y, una vez entrenado, combina sus respectivos votos de forma ponderada. Este clasificador fuerte se aplica luego a las subregiones de una imagen que se escanea en busca de ubicaciones probables de matrículas.

La mejora introducida por Viola y Jones corresponde a implementar una cascada de clasificadores fuertes. El algoritmo propuesto por ellos es el primero de detección de objetos en tiempo real. Fue orientado, inicialmente, a la detección de rostros pero por medio de la extracción de características es posible extrapolarlo a un sinnúmero de objetivos.

Clasificadores en cascada Adaboost

Es un método de combinación de clasificadores fuertes, de configuración Adaboost, por etapas sucesivas de complejidad. Esta configuración ideada, por Viola y Jones [34], permite extraer características de un objeto. Esto conlleva a la construcción de clasificadores eficientes para la detección del objeto, con el uso de la cascada de clasificadores.

Durante las primeras etapas del entrenamiento, los clasificadores más débiles, mediante el rechazo de la mayor parte de las sub-ventanas de búsqueda que no contienen el objeto, extraen las características más básicas. Luego dan paso a clasificadores cada vez más fuertes. El procedimiento explicado está ejemplificado en la Figura 2-2; en donde se muestra que, mientras más clasificadores haya reconociendo características del objeto de estudio, en este caso patentes, más posibilidades existen de que la detección final sea limpia de falsos positivos.

Descriptores Haar

Para los clasificadores se utilizaron descriptores de características del tipo Haar; se trabaja sobre una imagen integral, la que se define como la suma de los píxeles que se encuentran en la parte superior izquierda, considerando desde el inicio de un punto (x,y) de la imagen original, incluyendo el punto en sí. Las características Haar corresponden a un conjunto de funciones para la extracción de características de textura de un objeto. Estas se calculan como la diferencia entre la suma de intensidades de los píxeles que se encuentran en cada una de las áreas rectangulares adyacentes que se posicionan dentro de la ventana de detección.

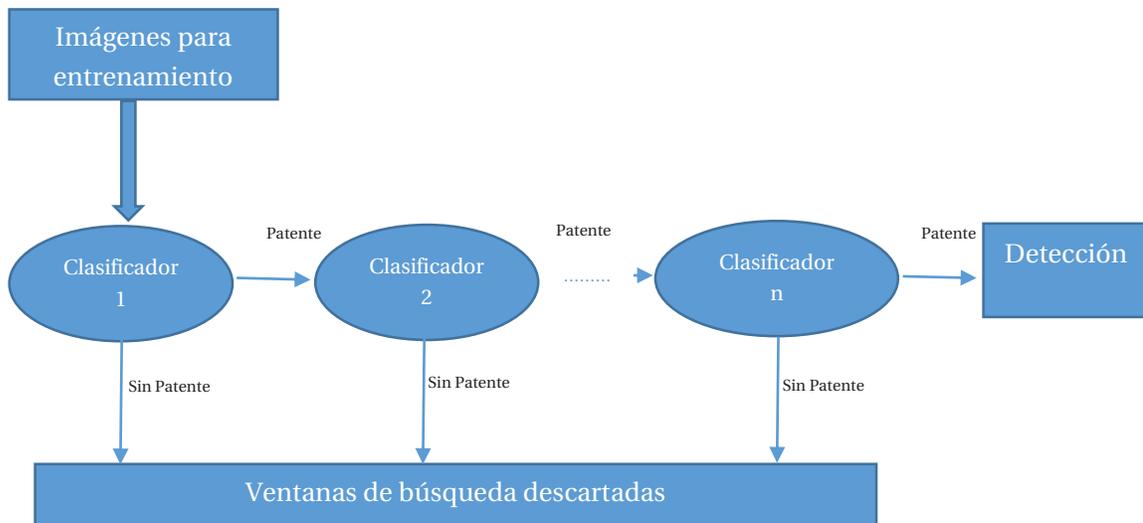


Figura 2-2 Clasificador de algoritmo Adaboost

2.2 Segmentación

Una vez hallada en la imagen la patente, la cual dependiendo de la distancia a la que se realizó la captura tendrá tamaños variables, es necesario una estandarización de esta a un valor determinado de píxeles. Las dimensiones a las que se lleva son: 350 píxeles de ancho por 100 de alto. Las tareas se desarrollarán con el uso de la librería para visión por computadora OpenCV.

2.2.1 OpenCV

Es una librería, de visión por computador y machine learning, de código abierto. Desarrollada por Intel, está escrita en C++ y cuenta con interfaces en Java, C, C++ y Python [1]. Es soportada por sistemas operativos basados en IOs, Windows, Android y GNU-Linux.

2.2.2 Conversión a escala de grises y estandarización de tamaño

La imagen es capturada a color (RGB), contiene información que no es relevante, ya que el análisis de características para la identificación, se realiza sobre una extracción de información acerca del valor numérico de intensidad de cada pixel y no de acuerdo al color que representa. De esta forma la primera transformación es convertir la imagen a escala de grises utilizando un umbral adaptativo, incluido por defecto en la librería, ya que la matriz en la que cada valor dentro de esta representa el nivel de gris del pixel que le corresponde. Esto será realizado con comandos de la librería OpenCV.

2.2.3 Sliding window

Sliding Window o ventana deslizante es un algoritmo utilizado en el procesamiento de imágenes para buscar, generalmente, un objeto específico. Se realiza un barrido con una ventana, de un tamaño definido menor al de la imagen a analizar. Para este caso se utilizó el concepto de ventana deslizante, con el que se obtuvo una serie de capturas para un mismo caracter. Se usa una ventana que se mueve a través de la imagen con el fin de seleccionar sólo esas áreas para posteriormente guardarlas. Para ejemplificar el procedimiento se presenta la Figura 2-3.

El procedimiento utilizado posiciona una ventana de 50 pixeles de ancho por 100 de alto en el punto de inicio de la imagen. Luego de obtener el primer corte a la patente, la ventana se desplaza 3 pixeles de forma horizontal para obtener la segunda y continúa moviéndose hasta hacer 4 cortes, correspondientes al primer caracter; para los siguientes caracteres las posiciones de inicio son los pixeles 45, 120, 165, 245 y 290. Para cada uno se realiza el mismo procedimiento de corte y desplazamiento de la ventana. Está considerado dentro del entrenamiento de la etapa de identificación el solapamiento de aproximadamente 5 pixeles en los que podrían aparecer fragmentos, ya sea del caracter anterior como del que se encuentra a continuación.

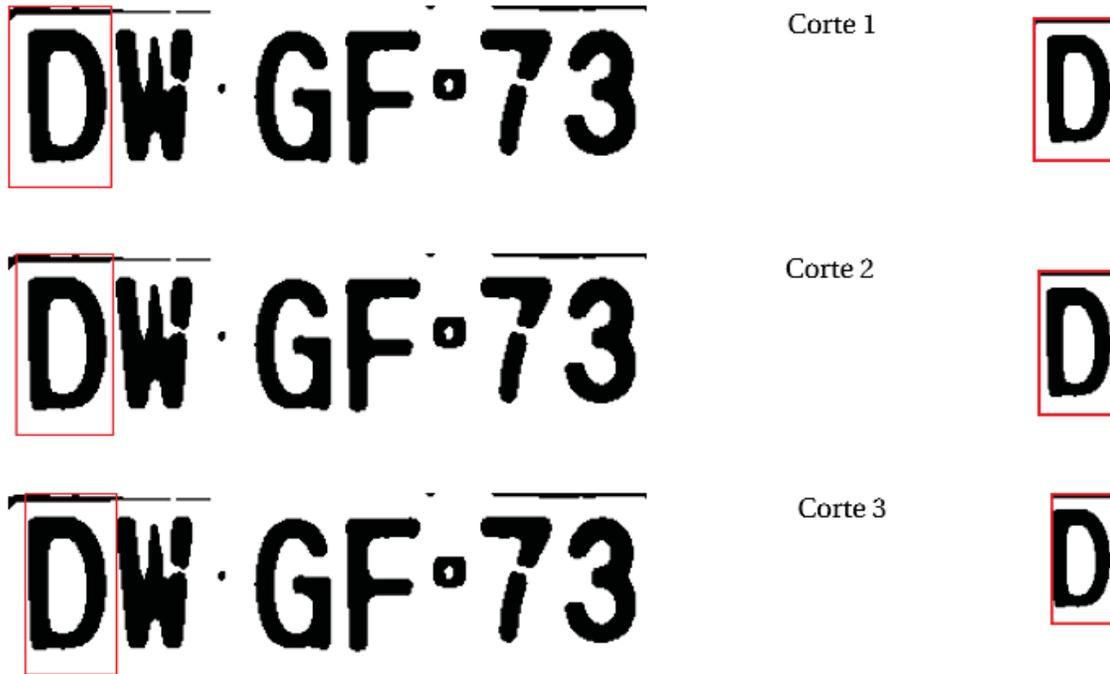


Figura 2-3 Ejemplo del sliding window aplicado.

2.2.4 Redes neuronales

Son modelos de aprendizaje automático inspirados en el funcionamiento del cerebro. En este caso cada una de las neuronas es llamada perceptrón y trabajan interconectadas, ya que cada una es una unidad de procesamiento. En la Figura 2-4 se presenta la estructura simplificada de una red neuronal, donde el movimiento de la información de entrada parte desde los píxeles de la imagen a las neuronas de las capas de la red hasta llegar a la salida, ejemplificada en un solo perceptrón.

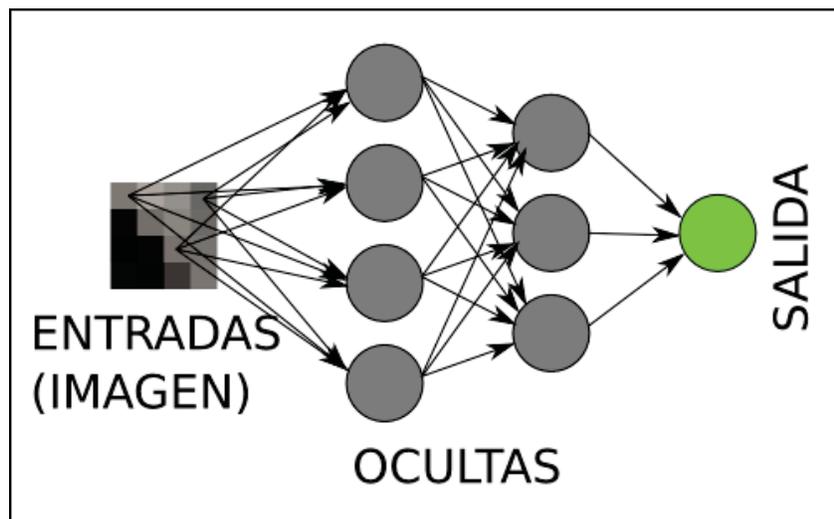


Figura 2-4 Estructura simplificada de una red neuronal.

Funcionamiento de las neuronas

Cada neurona recibe parte de la información que entra a la red, la cual multiplica por un peso determinado. Luego calcula la suma, del producto de cada entrada por su peso correspondiente. A este peso se le agrega generalmente un valor de bias o corrección. Por último el valor resultante se aplica a una función que, dependiendo de si se supera un cierto umbral o no, activa una siguiente neurona o entrega una salida.

Composición de la red

Las redes se componen por capas, las que procesan en cadena la información que entra a la red. El tipo de función de activación y el número de capas determina la complejidad de los problemas que es capaz de resolver. Dentro de esta interconexión de capas existe tanta la posibilidad de que todas las neuronas de una capa se conecten con todas las de la siguiente (*fully connected*), como también que exista una conexión selectiva de perceptrones (*partially connected*). Ejemplos de estas son: la red neuronal dinámica, las que pueden ser con retardos en el tiempo, recurrentes (total o parcialmente). También se tienen las redes neuronales jerárquicas, probabilísticas y convolucionales, entre muchas otras.

El trabajo de reconocimiento de patentes realizado en [18], utilizaba para hacer la tarea de identificación de caracteres la técnica “*template matching*”. Método que consiste en guardar, durante el entrenamiento, un patrón de la imagen que se quiere reconocer, para después realizar un *matching* con la imagen que entra al sistema. Se realiza una comparación o resta con todos los patrones, que conoce el modelo, y el de menor peso es aquel que entrega como resultado de identificación.

Por otro lado el funcionamiento de la red neuronal convolucional aunque requiere de un mayor uso de recursos tanto computacionales como de tiempo de entrenamiento y armado que la técnica de *template matching*, tiene un funcionamiento más robusto ya que cuenta con múltiples capas que trabajan en conjunto con un mismo propósito. Comenzando desde las primeras capas con la extracción de características; también hacen un reconocimiento de algunas particularidades de clases, donde se incluyen formas y bordes. Además detectan variaciones de posición, iluminación, escalas de tamaño, etc. La información que se extrae para cada clase se va condensando hacia una sola salida, o una posición en un vector de clasificación. En las últimas capas se encuentran reunidas las características que se buscan dentro de las imágenes que entran al sistema y así realizar la identificación.

Proceso de aprendizaje

Debido a la estructura multicapa de la red, no sólo el peso de cada una de las neuronas en combinatoria con la información de entrada avanza hacia la salida sino que también lo hace el error que pueda cometer cualquiera de las capas. La manera de contrarrestar esto para evitar que influya en el resultado del entrenamiento es con la utilización del algoritmo de backpropagation.

El proceso de aprendizaje de la red se lleva a cabo con el uso del algoritmo de backpropagation; este tiene dos fases, la primera consistente en el paso de patrones en la información de entrada

capa a capa hasta llegar a la salida de la red. Estos valores de salida se comparan con el resultado esperado obteniendo así el error que comete la red y ajustando los pesos mediante la propagación hacia atrás. Así se modifican los pesos de aprendizaje de toda la red [35].

2.3 Deep learning

El aprendizaje con redes neuronales convolucionales se realiza por medio de la extracción de características comunes dentro de las imágenes de entrada, a través de aplicar filtros a los píxeles de estas. Dependiendo del tamaño del filtro que se utilice variará el detalle de las características que se extraigan de las imágenes. Se mencionó anteriormente la existencia y, de manera ligera, función de algunas capas que componen el sistema de aprendizaje, a continuación se hará referencia al resto de su estructura. Y también se comentará sobre las etapas por las que pasa la información que ingresa a la red, que son:

1. Convolución
2. ReLU
3. *MaxPooling*
4. *Dense*
5. *Dropout*

2.3.1 Capa convolucional

El primer destino de la información que ingresa a la red corresponde a la capa convolucional, la cual se constituye por filtros, cuya cantidad y tamaño son definidos por quien programa la red. Estos filtros cumplen con cubrir la profundidad del volumen de entrada, realizan la extracción de las características de la imagen, ya que convolucionan a través de la sección transversal de esta y en todos los canales que la compongan. Mientras se realiza un producto punto entre el filtro y la sección de la imagen de entrada en la que se encuentre. Por último estos resultados son sumados con valores configurados para hacer que coincidan o se acerquen a los esperados en el umbral, estos se denominan bias.

Con esto se transforma la matriz de píxeles que corresponde a la imagen en una matriz de características. Donde cada cuadrante no solo tiene la información de un pixel de la imagen sino de una vecindad de ese. En la Figura 2-5 se presentan las etapas por las cuales para la información dentro de una red neuronal convolucional.

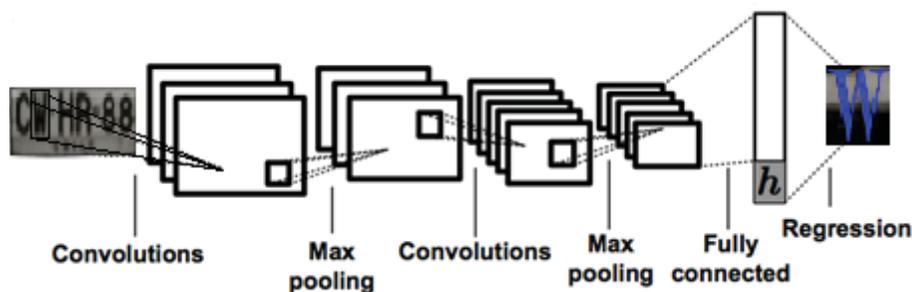


Figura 2-5 Pasos que sigue la información en la red

2.3.2 Función de activación

La red utiliza ReLU (Rectified linear unit) para la activación de las neuronas. Esta función tiene un gradiente de 0 cuando la entrada es igual o menor a 0 y de 1 cuando la entrada es mayor a cero. Esto quiere decir que, es una función que de acuerdo al valor recibido de la neurona anterior y considerando el peso sináptico de la neurona que recibe la información, genera un resultado. El cual si se encuentra por sobre el umbral la neurona se activa y genera el paso de la información a la siguiente capa. Por el contrario si el valor que recibe el perceptrón no alcanza el umbral la información no continua a las demás capas. La Figura 2-6 presenta la gráfica que toma esta función.

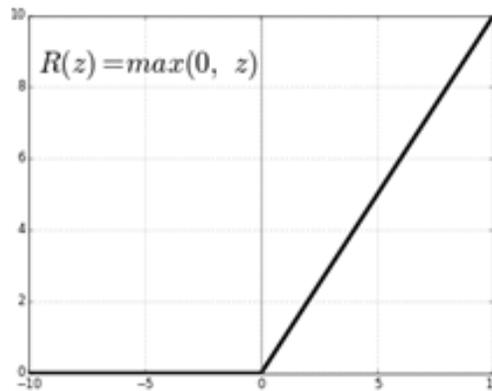


Figura 2-6 Función de activación (ReLU)

2.3.3 Capa de reducción (Pooling)

La función de esta capa es evitar el *overfitting* (sobre ajuste) de las capas convolucionales y de la red, reduciendo progresivamente las características particulares de una muestra dentro de una clase, es decir, elimina características que son, más bien, propias de una imagen en cuestión y no de una clase en general. La pooling layer opera de manera independiente en cada pieza de profundidad de la entrada y modifica el tamaño espacial, usando la operación MAX, Figura 2-7, la cual extrae el máximo valor de la región en la que se está ejecutando.

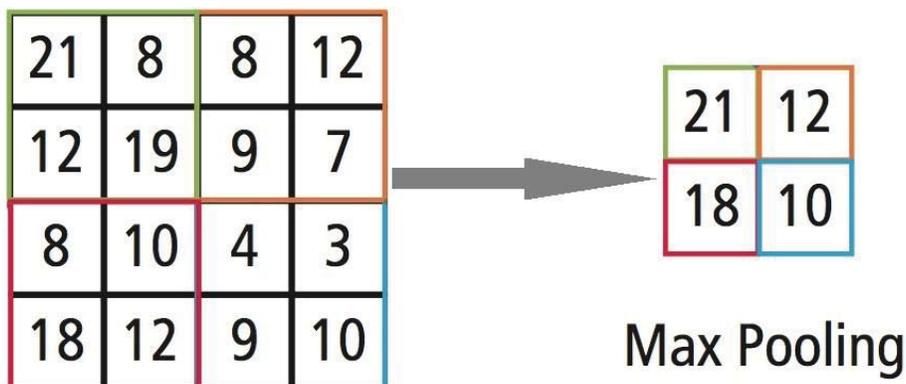


Figura 2-7 Max pooling.

2.3.4 *Fullyconnected Layer.*

Esta capa es denominada, en el código del programa, como *DenseLayer* o capa densa, es la encargada de realizar una conexión completa entre una capa y la siguiente. Es una capa clasificadora totalmente conectada, se encuentra al final de las capas convolutivas y de reducción. Estas contienen y consideran una neurona por cada píxel de entrada. La red tendrá, en este caso, tantas neuronas de salida como clases, ya que se encuentran a continuación de una capa que guarda todas las características de una clase en un arreglo o vector. También en medio de las capas existe aquella que involucra el concepto de *dropout*, lo que implica la eliminación de algunas características con el fin de evitar el *overfitting* (sobre entrenamiento) de la red.

2.3.5 Dropout

Es una técnica que consiste en la desconexión aleatoria de neuronas por iteración, con el fin de prevenir sobre entrenamiento, de alguna característica, en cualquiera de ellas. Esta capa permite evitar el *overfitting* a través de una técnica que omite aleatoriamente algunos de los filtros extractores de características en cada iteración de entrenamiento. Esto disminuye la situación en que una neurona extrae una característica específica y solo si las otras extraen otra determinada [36]. Todo lo anterior fue programado sobre la interfaz de programación de aplicaciones (API) Keras, la cual utiliza como *backend* a Tensorflow.

2.3.6 Keras

Es una API para redes neuronales de alto nivel, desarrollada en Python y capaz de ejecutarse sobre Tensorflow, CNTK o Theano. Fue ideada con el fin de llevar a cabo las tareas pensadas en redes neuronales de una manera rápida, sin grandes entramados de programación. Admite configuraciones de redes neuronales convolucionales, recurrentes y combinaciones de ambas.

2.3.7 Tensorflow

Es una biblioteca de software de código libre para computación numérica de alto rendimiento. Su arquitectura flexible permite una fácil implementación de computación en plataformas de CPU, GPU o TPU. Desarrollado por Google, cuenta con un sólido respaldo para el aprendizaje automático y profundo. Su núcleo de computación numérica flexible se utiliza en muchos otros dominios científicos. Como se ha mencionado anteriormente, en el manejo de la red neuronal por parte de la API Keras, Tensorflow se utiliza como *backend*, o sea, hace el manejo de bases de datos y computación numérica en el entrenamiento.

2.4 Raspberry Pi 3

Raspberry Pi es una organización benéfica que trabaja para poner el poder de la producción digital en manos de personas de todo el mundo [37]. Pero, específicamente, lo que importa en este caso es que proveen de micro computadoras de bajo costo y alto rendimiento, conocidas también como tarjetas de desarrollo. Es decir, Raspberry Pi también es una placa computador de

bajo costo. Las características técnicas, que son de importancia para el proyecto, de la Raspberry Pi 3 (Figura 2-8), son:

- Procesador: Chipset Broadcom BCM2387 de cuatro núcleos ARM Cortex-A53 que funcionan a 1.2 GHz
- 1 GB LPDDR2 en RAM
- Terminales de Ethernet 10/100 BaseT, 4 USB 2.0, HDMI audio/video, Conector GPIO

¿Por qué utilización de la versión 3 modelo B, en desmedro de sus antecesoras? Es, por una parte, su mayor capacidad de procesamiento con un procesador de cuatro núcleos que funciona 1200 millones de ciclos por segundo. Aunque es la característica que más sobresale frente a las otras en la Tabla 2-1 se especifican otras. La comparación se realiza con respecto a los dos modelos más cercanos a la placa seleccionada que es la Pi 3 modelo B, los que son la Pi 2 modelo B y la Pi Zero W.



Figura 2-8 Raspberry Pi 3 model B.

Tabla 2-1 Comparación entre modelos de Raspberry

Modelo	2 Modelo B	3 Modelo B	Zero W
Característica			
SoC	BCM2836	BCM2837	BCM2835
CPU	900MHz Quad ARM Cortex-A7	1.2 GHz Quad ARM Cortex-A53	1GHz ARM1176JZF-S
RAM	1 GB	1 GB	512 MB
USB	4	4	1 Micro
Boot	MicroSD	MicroSD	MicroSD
Red	Ethernet	Ethernet, Wifi, BT	Wifi, BT
Consumo	800mA/4W/5V	2.5A/12.5W/5V	160mA/0.8W/5V

Con respecto a la primera, la utilizada presenta ventaja con respecto a la velocidad de procesamiento de 900Mhz que proporciona el procesador A7, de la 2 modelo B, frente a los 1.2 GHz del A53 que posee la Pi 3. Además de esto, el segundo (A53) soporta 64bits, mientras que el primero solo 32bits. Por último el procesador de la Pi 3 proporciona una aceleración en reproducción de medios, es decir, puede ejecutar 128 bits a la vez mientras que el de la Pi 2 no tiene esta capacidad.

La mayor diferencia entre la Pi 3 modelo B y la Pi Zero se encuentra en la cantidad de núcleos de procesamiento que poseen, ya que la primera cuenta con 4, y la segunda tiene solamente uno. Una vez claras las diferencias, por lo menos entre placas de la misma marca y precios, relativamente, cercanos ya que estas se encuentran en el rango entre 10 y 35 dólares es posible especificar lo que fue necesario para que el sistema funcionara en la Raspberry.

Como parte del hardware también está considerada la webcam, una c270 de Logitech cuyas características son:

- Modo de vídeo de 720p.
- 3 Mega Píxeles.
- Corrección de iluminación.

Para el caso de las pruebas en terreno se utilizó una *powerbank* PHILIPS modelo DLP10403, cuyas especificaciones aparecen en la Tabla 2-2 Características Powerbank. Este banco de baterías

otorga una autonomía de 1 hora de uso continuo para la Raspberry. Además se utilizó un *selfiestick* con el fin de estandarizar la altura de captura a 50 cm, de manera vertical.

Tabla 2-2 Características Powerbank

Energía	
Salida	5 V, 3.1 A
Entrada de energía	5 V, 2.1 A
Capacidad de la batería	10400 mAh

2.4.1 Implementación

El programa probado en laboratorio, comprende los scripts de mando, segmentación y el archivo de configuración entrenado, el modelo de red neuronal “MobileNet_bigd_Reload”. Esta última debe encontrarse en una carpeta con el nombre CNN, todo esto dentro de un mismo directorio.

Una vez estos archivos son portados a la Raspberry, en la cual deben encontrarse instaladas las librerías necesarias para el manejo de imágenes y otros procesos del programa, como lo son OpenCV, Tensorflow y Keras. La instalación tiene una duración aproximada de 8 horas.

La placa posee una memoria RAM de 1 GB, por lo que es necesaria una expansión por medio del uso de “swap” para el S.O. Raspbian, esto con el uso de un dispositivo flash con formato FAT 32. Con esto es posible correr el programa del sistema.

3 El sistema ALPR propuesto

La propuesta inicial comprendía tres opciones, la primera utilizar solo técnicas existentes para desarrollar el sistema era la primera, la segunda formar un sistema desde cero con el cual llevar a cabo las tareas del sistema y la tercera opción, utilizar técnicas existentes para conformar un sistema de reconocimiento automático de patentes. Esta última fue la que se desarrolló utilizando esta técnica para dar forma a un sistema robusto, en base a los recursos y conocimientos existentes.

Las tareas y técnicas que finalmente componen el sistema serán presentadas a lo largo de este capítulo, ver Figura 3-1. La detección de patentes realizada con clasificadores en cascada, la segmentación de caracteres realizada por medio del uso de *sliding window* y, por último, la identificación de letras y números realizada con el uso de una red neuronal convolucional (CNN).

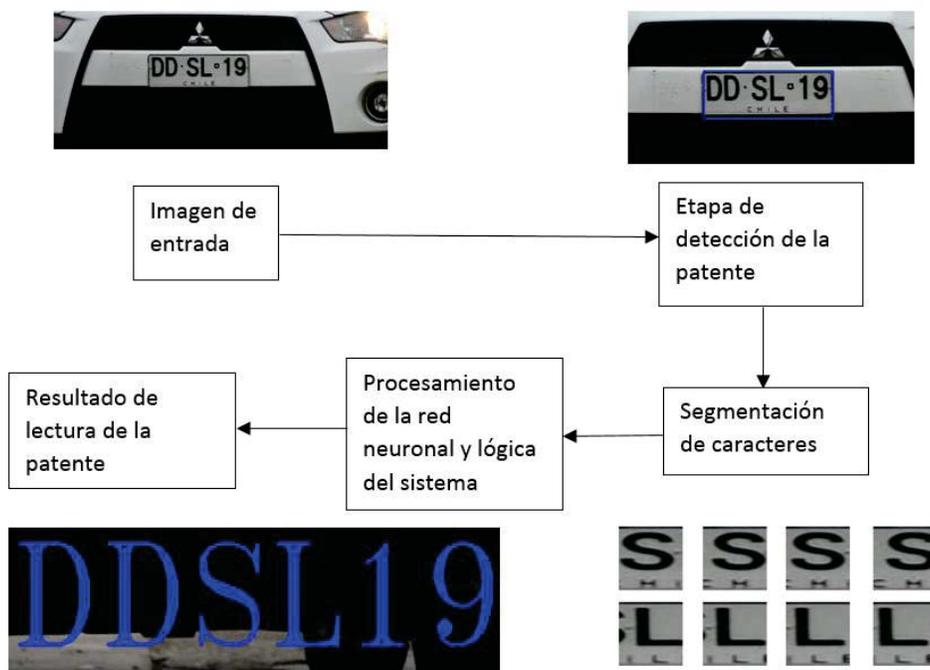


Figura 3-1 Sistema propuesto.

3.1 Detección de la patente

La primera etapa utilizada es la detección, que consiste en rastrear y marcar la placa en la imagen obtenida, desde la cámara. Con el uso de OpenCV se implementa un código en clasificadores en cascadas, entrenados por medio del algoritmo de Viola y Jones, de descriptores del tipo Haar.

El proceso en el sistema comienza con el ingreso de la imagen, desde una fotografía, un vídeo o captura por medio de webcam. Luego el algoritmo realiza un análisis multi-resolución de la imagen y se extraen sub-ventanas que son revisadas en busca de la patente. Estas son entregadas al clasificador en cascada, el cual determina si existe o no el objeto en la ventana. Por último si es encontrada dentro de la imagen la patente se marca con rectángulo y luego se aísla para realizar el procesamiento de la información, tal como se ilustra en la Figura 3-2, imagen obtenida de [18]. El resultado de la detección se presenta en la Figura 3-3.

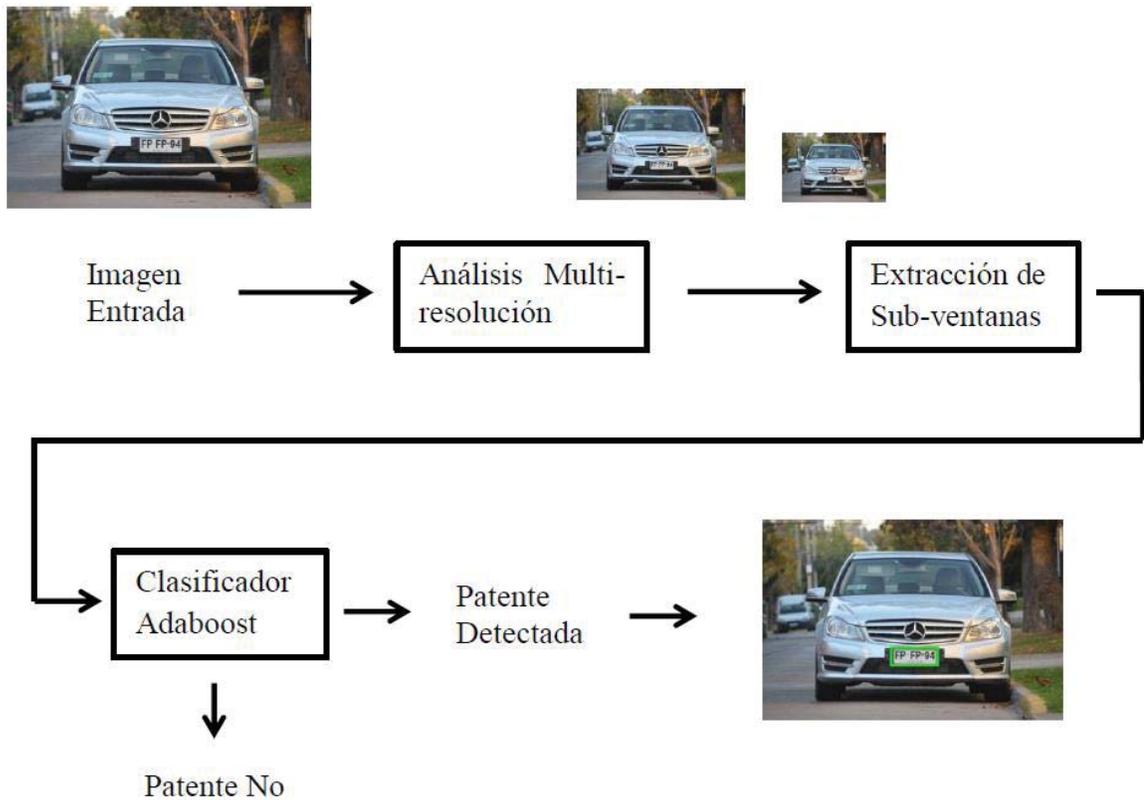


Figura 3-2 Esquema de detección de patentes [18]



Figura 3-3 Detección de la placa

3.2 Segmentación de caracteres

Una vez aislada el área en la cual se encuentra la placa, se reduce parte de los bordes, con el fin de conservar solo la zona en la que se encuentran los caracteres. Es decir, se intenta eliminar la mayor cantidad de información que no forma parte de letras ni números de identificación, como las orillas de la matrícula o las letras inferiores, además de bordes que puedan aparecer de una captura movida.

Con el fin de obtener una o más imágenes, que pudiese usar la red neuronal para realizar la identificación, es que con el conocimiento previo acerca de la ubicación y tamaño de letras y números, se configura esta etapa. Cada caracter tiene dimensiones de 40x50 pixeles, así se realizó un *sliding window* desde el pixel 0 en el eje x, con una ventana de 50x100 pixeles que se moviera 3 pixeles por captura, con el fin de obtener la imagen del caracter, sin importar si este se encontraba o no, cien por ciento centrado en su ubicación original.

Antes de realizar este procedimiento, se agregó una etapa de escalamiento de grises a la imagen. Ya que la característica a utilizar es la intensidad de los pixeles. De esta imagen inicial, Figura 3-4, se extrae un set de muestra por cada carácter obteniendo el resultado que se presenta en la Figura 3-5. Con esta forma de obtener las letras y números es posible entregar, a la siguiente etapa, caracteres más completos en una imagen mal capturada. De esta manera es posible tener un conjunto de muestras que puedan ser procesadas por la siguiente etapa del sistema y así comparar los resultados obtenidos con cada una de ellas, para generar una respuesta certera. Estos cortes también pueden ser utilizados para enseñar al sistema de identificación, la clase de posibles muestras a obtener desde la etapa de segmentación. Esto, debido a que incluyen imágenes de letras y números que están movidos dentro de la imagen, además de las imágenes completas de los caracteres necesarios.

Tal como se aprecia en las Figuras 3-4 y 3-5, algunos cortes pueden contener, al inicio o al final de la letra (en este caso la U), segmentos del anterior o del que le sigue. Cabe destacar que son el tipo de imágenes que se obtienen de la detección y la razón por la que se entrega más de una muestra a la etapa de identificación. Es necesario precisar que el algoritmo de reconocimiento de

caracteres, dentro de su set de entrenamiento, cuenta con muestras de este tipo, haciéndola robusta ante situaciones como las anteriormente presentadas.

Por último, los cortes obtenidos de la imagen son redimensionados al tamaño de las muestras utilizadas para entrenamiento, 224x224 píxeles, para ser entregadas a la etapa de identificación.



Figura 3-4 Patente cortada desde el frame completo.

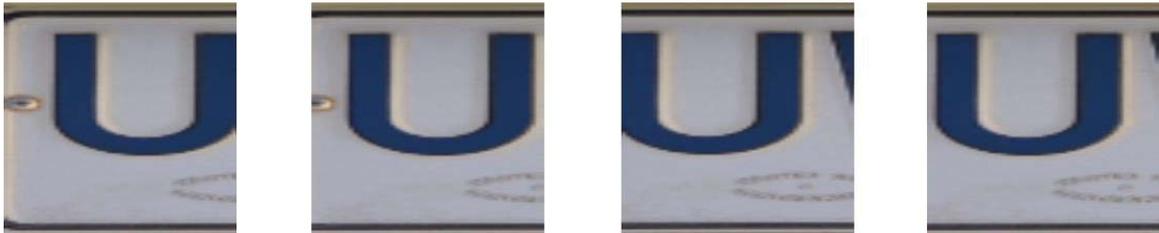


Figura 3-5 Set de muestras obtenidas desde la etapa de segmentación.

3.3 Identificación de caracteres

Esta tarea es llevada a cabo por la red neuronal convolucional, a la cual se le entrega un set de imágenes por cada carácter, esta a su vez entrega un conjunto de predicciones por cada muestra dentro del set, las que se llevan a una etapa de comparación. Esto último se realiza por medio de un empate de predicciones con el fin de obtener una suerte de moda dentro de los resultados entregados por la red por cada uno de los caracteres.

Cada uno de los cortes obtenido de la etapa de segmentación pasa por la configuración de la red entrenada. Donde se desglosan sus características de bordes y formas para llegar a la capa de clasificación que reúne el conjunto de estas y las compara con la información que tiene almacenada con respecto a cada una de las clases, con las que fue entrenada. La configuración del deep learning seleccionada es la estructura MobileNet, ya que está orientada a sistemas embebidos y móviles, por lo que su utilización de recursos de procesamiento es baja.

3.3.1 MobileNet

La última tarea y la que, en este caso, debe ser la más robusta, se lleva a cabo con una red neuronal convolucional. Se analizaron algunas alternativas dentro de las cuales entran las configuraciones del tipo Inception, también Xception y por último la de la MobileNet. Todas estas fueron re-entrenadas con el proceso de transferencia de aprendizaje para adaptar su etapa de clasificación, ya que su desempeño en las áreas de reconocimiento de objetos ya se encuentra documentado en [29] y [30].

Esta red, también denominada red neuronal eficiente para aplicaciones móviles, está orientada a sistemas móviles y embebidos. Al igual que la Xception basa su arquitectura convolutiva en las capas profundas de convoluciones separables, esto con la finalidad de disminuir el uso de recursos computacionales o, en otras palabras, construir redes livianas. Con la excepción de la primera capa que es de convolución completa.

“Todas las capas son seguidas por un normalizador de batch y usan la función de activación ReLU a excepción de la última que es una capa totalmente conectada con función softmax para la clasificación” [34]. Además se disminuye el sampling con una serie de convoluciones profundas así como en la primera capa. Finalmente aplicar un Average Pooling para reducir la resolución espacial a uno antes de la capa completamente conectada.

Considerando las capas de convolución profunda y las puntuales, en total la red tiene 28 capas. En la Figura 3-6 se presentan las diferencias entre capas convolucionales estándar con las presentes en la MobileNet. A la izquierda se presentan las capas normales en una etapa de la iteración de una red, con una convolución normal de 3x3, una normalización de batch y una función de activación. Al lado derecho se tienen las etapas de una sección convolutiva de la MobileNet, con una convolución en profundidad y con normalización de batch función de activación para una convolución de 1x1 otra normalización de batch y una función de activación para la siguiente etapa o neurona.

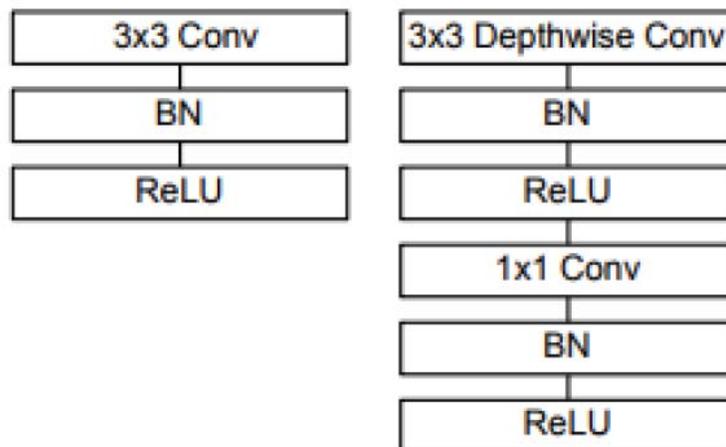


Figura 3-6 Serie de capas estandar (izq.) versus MobileNet (der.)

3.3.2 Transfer Learning

La red fue originalmente entrenada para el reconocimiento de objetos, es decir, que es capaz de reconocer las diferencias que existen entre los bordes y las formas de una gran variedad de cosas. Por medio del *transfer learning* se le especifica que, los objetos a clasificar están compuestos por una cantidad determinada de clases, en este caso los caracteres. Además esta aprende que cada una de estas clases tiene formas distintivas unas de otras. Así realiza la clasificación de letras y números.

A la red pre-entrenada se le agregaron algunas capas, con el fin de realizar el Transfer Learning a su etapa de clasificación, La configuración resultante se muestra en la Figura 3-7, la transferencia de aprendizaje, es un recurso de investigación centrado en utilizar el conocimiento adquirido con un fin, en la solución de un problema distinto pero, en cierta medida, relacionado. Es decir, en este caso, una red entrenada en el reconocimiento de objetos, bordes y formas se utiliza para la identificación de caracteres, a través del reacondicionamiento de sus capas de clasificación.

En la Figura 3-7 se presenta la configuración final con la que se trabaja. Especifica que existe una capa de entrada, luego viene la que contiene la configuración pre-entrenada de la MobileNet, luego una capa de dropout y por último la capa en la que se condiciona a clasificar las características de la información que ingresó en 32 clases, correspondientes al número de caracteres distintos de las patentes de vehículos. La capa “input_2” es la primera de la configuración en la cual se determina la cantidad de las imágenes de entrada, el tamaño de estas (alto y ancho) y por último los canales que las componen. La siguiente capa corresponde a la configuración de la red para entrenamiento. Le sigue una capa de dropout y por último se configura la capa de salida de la estructura a la cual se le agrega el parámetro de la cantidad de clases, en este caso 32.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, None, None, 3)	0
mobilenet_1.00_224 (Model)	(None, 1024)	3228864
dropout_1 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 32)	32800
=====		
Total params: 3,261,664		
Trainable params: 3,239,776		
Non-trainable params: 21,888		

Figura 3-7 Capas entrenables y parámetros.

3.4 Funcionamiento del sistema

El resumen del funcionamiento del sistema es mostrado en la Figura 3-8. En esta se presenta la entrada del sistema, representada por la configuración de una patente completa. El sensor óptico, la cámara, obtiene una imagen que corresponde a la escena completa y que contiene el vehículo con su entorno y, por supuesto, la matrícula que se extrae esta última por medio del funcionamiento de la fase de detección. Se aplica la limpieza mencionada en el apartado 3.2 para luego proceder con la etapa de segmentación de caracteres, con la que se obtiene el set de muestras. Para, finalmente, entregar estos datos a la etapa de identificación de la que se obtendrá el conjunto de predicciones por set de imágenes, las que serán comparadas para determinar el contenido de la patente.

Esta comparación es el último paso, dentro del conjunto de tareas que componen el sistema de identificación de patentes. El cual consiste en tomar las predicciones, agrupadas según el carácter al que corresponden, y compararlas. Esto con la finalidad de evitar que una predicción realizada a un corte mal ubicado al principio del set o al final de este, se entregue como respuesta. Es por esto que es necesario que empaten dos respuestas entregadas por la red para un carácter, para que el sistema muestre esta predicción como resultado final.

La lógica utilizada en la comparación de los caracteres es analizar en orden de ingreso la coincidencia de las predicciones, es decir, la primera con la segunda si estas no son iguales se realiza nuevamente entre la segunda y la tercera y lo mismo para la cuarta, sin embargo si las predicciones concuerdan se toma esta respuesta como resultado. El análisis se realiza de esta manera debido a que la mayor cantidad de errores, provocados por la mala detección de la patente, se dan en los primeros caracteres de la placa o en los últimos lo que se corrige con el método implementado.

La Figura 3-8 presenta la lógica utilizada por el sistema y sigue el siguiente procedimiento: se tiene la patente detectada representada por "AB*CD*01", luego se presenta el modelo de segmentación que utiliza el sistema consistente en el *sliding window* de la letra "A", después cada uno de los parches que se obtiene se ingresa a la red neuronal la cual entrega una predicción por cada imagen. Las predicciones se nombran en el diagrama de la figura como P1 (Predicción 1), P2, P3 y P4, mientras que luego de realizar la comparación si hay una coincidencia entre dos consecutivas el sistema entrega como respuesta una PT (Predicción Total). Por último, se agrupan las predicciones totales de cada uno de los caracteres y se compone la lectura realizada por el sistema. En resumen, para la etapa de identificación, basta con que existan dos predicciones iguales seguidas para tener una respuesta de lectura por parte del sistema.

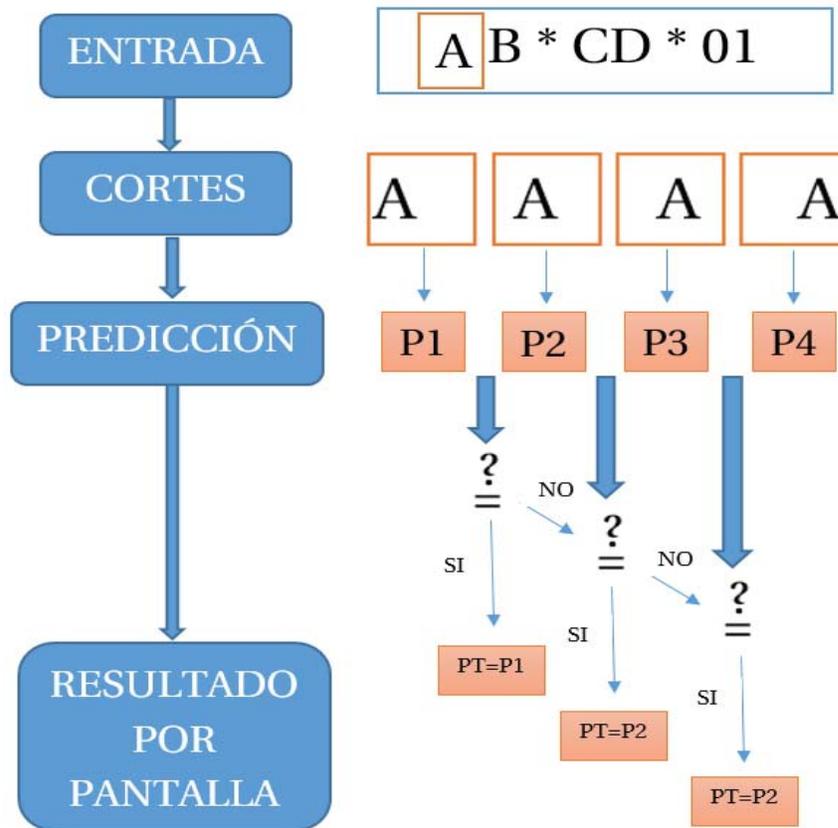


Figura 3-8 Funcionamiento del sistema de identificación de patentes.

3.5 Implementación en Raspberry

En la etapa final del trabajo, el objetivo fue el de portar el programa probado en laboratorio, a la tarjeta de desarrollo. Este proceso implicó la instalación de las librerías necesarias para ejecutarse sin problemas. Aquellas implicadas en el proceso de detección, el procesamiento a las imágenes como lo es OpenCV, la que tarda alrededor de 4 horas en estar disponible para utilizarse. Y por último el funcionamiento de la tarea de detección con el uso de la red neuronal, lo que implica la instalación de Tensorflow que funcionará como backend y Keras que será la API a utilizar para el manejo de la red neuronal.

4 Resultados experimentales

Para corroborar el funcionamiento del sistema se llevaron a cabo 4 experimentos, tres de los cuales fueron realizadas en laboratorio y uno en terreno. Las cuatro experiencias se llevaron a cabo con el fin de obtener las tasas de acierto con bajos tiempos de procesamiento. Además de encontrar las condiciones óptimas de trabajo del sistema en cuanto a iluminación y otros campos importantes.

El primer experimento se realizó con la idea de probar el funcionamiento de la red neuronal basada en MobileNet por medio del reconocimiento de caracteres. Los siguientes tres experimentos se enfocaron en evaluar el desempeño del sistema realizando detección y reconocimiento de patentes en videos, fotografías y vehículos, tanto en el laboratorio como en terreno. Los procedimientos y resultados se presentan a continuación.

4.1 Experimentos de laboratorio

En los siguientes apartados se presentan los experimentos realizados en laboratorio. El primero consistente en la validación de funcionamiento de la red neuronal “MobileNet_bigd_Reload”, de la etapa de reconocimiento de caracteres. Mientras que los dos siguientes abarcan el funcionamiento de las etapas de detección e identificación de patentes mediante un computador y la Raspberry Pi.

Note que para todos los experimentos se utilizó un computador de escritorio (PC) con las siguientes características: un procesador Intel Core i7 de 8 núcleos (3.40GHz), 8Gb de RAM, una tarjeta NVIDIA GeForce GTX 1050 Ti de 7Gb de memoria y Windows 8 Pro de 64 bits. Mientras que las características de la placa se detallan en el apartado 2.4.

4.1.1 Experimento 1: Validación de la etapa de identificación de caracteres.

El experimento consiste en tomar un total de 4288 imágenes de caracteres provenientes de patentes con desplazamientos y aplicarlas a la red neuronal para conocer el desempeño de clasificación. Las imágenes consisten en 4 parches por cada caracter con un corrimiento de 3 pixeles entre uno y otro, dando un total de 148 muestras por clase, que son entregadas a la red para obtener sus respectivas predicciones. Estas lecturas fueron analizadas, una a una, para

obtener el comportamiento de la configuración completa de la red entrenada. Una tabla que muestra el detalle del reconocimiento de los caracteres se presenta en el Apéndice A.

Es necesario especificar que no fue posible reunir la misma cantidad de muestras para la totalidad de las clases que componen la base de datos de la red. Aquellas que contaron con 37 muestras fueron: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, B, C, D, F, G, H, J, K, L, P, R, S, T, V, W, X, Y y Z. Mientras que la letra N contó con 31 muestras, la U con 5 y, por último, para la A y la E no se encontró ejemplos con los cuales realizar una evaluación, esto último debido a las pocas existencias.

Resultados del experimento 1

Del total de predicciones realizadas por la red neuronal, el desempeño fue de 4201 sobre 4288. Es decir la tasa de acierto de la red es 97.97%. Los caracteres sin errores en su predicción fueron: 0, 1, 2, 3, 5, D, H, P, S y Z, o sea una tasa de acierto de 100%. Aquellos con mayor cantidad de errores son la B con una tasa de acierto de 94.6% y la G con un 91.9% de respuestas correctas; lo que implica 8 y 12 errores, respectivamente. Dentro de las 148 predicciones realizadas las confusiones más comunes fueron para la B la E, el 8 y 2 mientras que para la G los errores más frecuentes fueron con el 0 y la C. El detalle de la clasificación por caracter se encuentra disponible en la Tabla 0-2, en los apéndices del informe.

Los resultados obtenidos con respecto al funcionamiento de la configuración de red neuronal entrenado son satisfactorios, las tasas de acierto para todas las clases probadas, ya fuesen con las 148 muestras o las 20 estuvieron por sobre el 90%, lo que demuestra un correcto accionar de predicción por parte del modelo. De manera más específica y considerando solamente aquellos caracteres igual número de muestras (148), todas las tasas de acierto se encuentran por sobre el 91%. En comparación con otros sistemas basados en redes neuronales, como son los casos de [38], con 95.24% de precisión, y [16], con 97.56%, el rendimiento es bajo, pero en ambos casos el uso de redes neuronales también se aplica al proceso de detección.

4.1.2 Experimento 2: Análisis del sistema completo en laboratorio con archivos de vídeo e imágenes.

La base de datos utilizada para este experimento consta de un total de 126 imágenes de vehículos con sus respectivas patentes, contenidas en 2 archivos de vídeo con 66 autos y 60 fotografías disponibles para test, las dimensiones de estas últimas son 986 por 653 píxeles. La Figura 4-1 presenta un ejemplo de las imágenes que ingresan al sistema en la secuencia de vídeo, cuyas especificaciones son: 854 por 480 píxeles, en tamaño, a 30fps. Los vídeos muestran vehículos que avanzan de frente a la cámara, de modo de obtener una imagen limpia de la matrícula de cada auto. La finalidad del experimento consiste es analizar el desempeño del sistema completo, es decir, la detección y clasificación de las patentes, sin el uso de la cámara en el laboratorio.

Para realizar las pruebas con vídeos se reemplaza la entrada de la cámara al programa por la ubicación del material audiovisual. El experimento consiste en la lectura de patentes utilizando un computador de escritorio cuyas características fueron mencionadas al principio del capítulo. Una vez realizada la detección y captura de la patente, además de la segmentación de caracteres,

las imágenes resultantes son entregadas a la red neuronal, donde se obtiene una lectura. que se comparó con una base de datos. La manera de informar si la patente en cuestión se encontraba dentro de la base de datos, era entregar por consola la lectura de la placa. Por otro lado, para el caso de que la patente no se encontrara en la base de datos, el sistema entregaba un “No” antecediendo a la lectura realizada, tal como se muestra en la Figura 4-2 donde se presenta un diagrama del sistema, donde “Lectura” hace referencia al resultado entregado por la red.



Figura 4-1 Ejemplo muestra de test

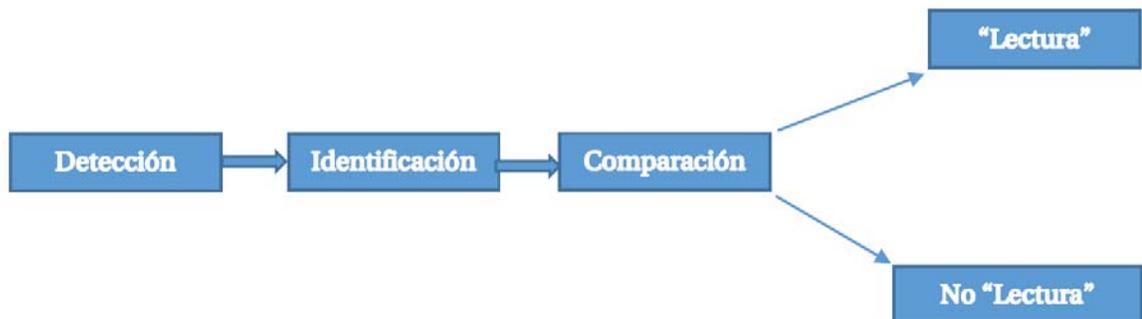


Figura 4-2 Lógica de funcionamiento del sistema

Resultados del experimento 2

En base al universo de 126 muestras, la respuesta que la etapa de detección de patentes del sistema entregó fue de 119 resultados verdaderos positivos (VP), correspondientes a aquellas placas correctamente detectadas; además de 7 falsos negativos (FN), situación que se da cuando una matrícula no es detectada por el sistema, estando en la escena.

Utilizando como nuevo universo de muestras el total de 119 patentes correctamente detectadas. Con respecto a la tarea de identificación de caracteres, la cantidad de patentes que fue leída

correctamente de manera completa fue de 108, también existe una cantidad de 11 lecturas incorrectas. Dentro de los errores de lectura de las patentes detectadas aparecen algunos tales como leer un 1 como si fuera un 7, o una B como E; esto debido en gran medida a la captura corrida de la placa detectada. El tiempo medio de procesamiento de cada patente una vez es detectada es de 0.38 segundos.

Los resultados presentados en la Tabla 4-1 se deben interpretar de la siguiente manera: la totalidad de las muestras las componen 2 tipos de archivo que combinadas representan el tamaño de la base de datos de patentes analizadas para la detección, de lo que se obtiene una efectividad del sistema de 94.44%. Mientras que las identificaciones correctas se basan en el total de verdaderos positivos en detección (119), de este nuevo total de muestras los aciertos obtenidos en identificación entregan una tasa de 90.75% correspondientes a 108 respuestas correctas y, por tanto, la tasa del sistema.

Con respecto a las malas identificaciones, fueron producto de una imagen tomada con una iluminación desfavorable, es decir, situaciones en que se oscureció tanto la placa de la escena que no se realizó detección o se llevó a cabo de manera corrida si es que un sector de la patente estaba más iluminada que otra. En la Figura 4-3 se encuentra una representación de la respuesta del sistema a patentes dentro y fuera de la base de datos.

Tabla 4-1 Resultados porcentuales

Muestras		Detección		Identificación [%]	
Tipo	Cantidad	VP	FN	Correctas	Incorrectas
Vídeo	66	60	6	50	10
Imagen	60	59	1	58	1
Totales	126	119	7	108	11
Porcentual	100	94.44	5.56	90.75	9.25

FRHD48 No FKS Y66
DZPX74 No VH9294

Figura 4-3 Patentes en el sistema (izq.). Patentes fuera del sistema (der.).

4.1.3 Experimento 3: Análisis del sistema completo en laboratorio con archivos de vídeo e imágenes con Raspberry Pi 3.

Utilizando las mismas muestras que en el experimento anterior se realizó una prueba utilizando el sistema completamente implementado en la Raspberry. El experimento consistió en utilizar el sistema incluyendo el sensor y proyectando las muestras con el fin de obtener el desempeño en laboratorio de la placa, en cuanto a procesamiento de las imágenes y detección de las patentes.

El procedimiento para llevar a cabo esta experiencia consistió en montar el sistema completo con un soporte fijo frente a una pantalla en la cual se proyectaran las muestras. Las secuencias de vídeo se presentaron a velocidad normal en consideración del movimiento de vehículos, mientras que las imágenes se pasaron en presentación para la lectura con la cámara. Las condiciones de funcionamiento constaron de una habitación completamente a oscuras, con la finalidad de reducir al máximo la posibilidad de que aparecieran reflejos que afectasen la detección de las patentes. La cámara se posicionó a una distancia y altura tal de emular la ubicación del sistema frente a vehículos y sus matrículas. Las capturas de la cámara están programadas a 10 fotografías por segundo, debido a que el tiempo de procesamiento del sistema por cada imagen que ingresa tarda alrededor de 28 segundos. La información de cada una de las lecturas se entrega por consola además de, al finalizar el programa, almacenarse en un archivo de texto nombrado "Patentes.txt". En la Figura 4-4 se muestran ejemplos del funcionamiento del sistema.



Figura 4-4 Ejemplos de funcionamiento

Resultados del experimento 3

Los resultados entregados por este experimento dan cuenta de que los tiempos en los que se presentan los vehículos frente a la cámara son importantes. Esto debido a lo que demora el procesamiento por imagen (28 [s]), es por esto que como medida paliativa se manejó la cantidad de *frames* por segundo. Esto se refleja mayormente en el número de detecciones que realizó el sistema sobre las secuencias de vídeo, donde de un total de 66 patentes que aparecen se

detectaron 24. Por otro lado el tiempo de exposición de las imágenes es manejable por lo que aumenta el número de detecciones, donde de 60 muestras se realizaron correctamente 50 detecciones. Del total de 126 muestras utilizadas para el experimento se detectaron 74 patentes. Para el caso de la etapa de identificación de caracteres, sobre el total de patentes detectadas (74), se identificaron de manera correcta 66, dando una tasa de acierto para el sistema de 89.2 %. Todo esto se presenta en la Tabla 4-2.

Los resultados de la detección, como se menciona anteriormente son debidos al tiempo de procesamiento de por imagen que tiene el sistema. Cada vez que se detecta una patente esta imagen es procesada, pasando por cada una de las etapas que compone el proceso de lectura lo que implica 28 segundos entre detección y detección, los vehículos que aparecen frente al sensor en medio de ese proceso no se capturan. Por ende la tasa de detección versus la cantidad de vehículos que se presentan frente a la cámara es baja si estos no se detienen el tiempo suficiente para que se procese cada uno. Al igual que en el experimento anterior las identificaciones se ven afectadas por la forma en que se captura la imagen de la patente.

Tabla 4-2 Resultados experimento 3

Muestras		Detección		Identificación	
Tipo	Cantidad	VP	FN	Correctas	Incorrectas
Vídeo	66	24	42	22	2
Imagen	60	50	10	44	6
Totales	126	74	52	66	8
Porcentual [%]	100	58.7	41.3	89.2	10.8

4.2 Experimento en terreno

La última etapa del trabajo contemplado como parte del proyecto fue la de realizar pruebas a vehículos directamente en calles, estacionamientos, etc. Para esto se portó el programa a la tarjeta de desarrollo, Raspberry Pi 3 y se implementó el sistema con el resto de los periféricos para completar el sistema, las características de estas herramientas están especificadas en el capítulo 2, la Figura 4-5 presenta un ejemplo de cómo se tomaron las muestras en terreno.



Figura 4-5 Toma de muestras en terreno

4.2.1 Experimento 4: Análisis del sistema completo en terreno con Raspberry Pi 3.

El experimento consistió en el uso del sistema completo para la lectura de las matrículas a vehículos en la calle. Utilizando vehículos en movimiento y autos estacionados como las muestras para realizar el experimento. La cantidad de patentes que conforman la base de datos es 100, divididas en 20 de automóviles en movimiento y 80 de vehículos estáticos. Las muestras consideradas para censar fueron diversas, ya que se incluyeron algunas patentes en mal estado. También aquellas comprendidas dentro de la configuración antigua, vale decir “AB*01*23”, al igual que las pertenecientes a la nueva, o sea “AB*CD*01”. También se incluyó la nueva fuente utilizada en la fabricación de las patentes, todo esto con el fin de revisar el desempeño ante diversas posibilidades de patentes presentes en las calles.

El procedimiento consistió en ubicar la cámara en un lugar en el que se acercaran de frente los automóviles y a una altura de 50 cm, con el menor ángulo posible entre el sensor y la patente, con el fin de evitar errores de detección y captura. Lo anterior, debido a que la cascada de características admite un pequeño ángulo de orientación para buscar las características del objeto. Para la parte de la prueba que se realizó con vehículos estacionados, se ubicó la cámara frente a los autos a una distancia de un metro y medio y un soporte con una altura de 50 centímetros. La Figura 4-6 representa un ejemplo del posicionamiento de la cámara, respecto a la matrícula para tomar las lecturas. Como se mencionó anteriormente, la limitación en la detección de placas considerando la variación del ángulo entre esta y la cámara es causal, de que las pruebas se hallan realizado mayoritariamente con tomas frente a los vehículos. Dejando como

factores de error a la iluminación de la escena y el color de fondo a la patente, o sea que se pierdan los límites entre la matrícula y el vehículo.

La realización de la experiencia comenzó por encontrar un lugar dónde colocar la cámara tal que cumpliera con las condiciones de posición para realizar la detección. El sitio donde se ubicó la cámara fue en a la entrada del condominio Alto Placeres, en el cerro Los Placeres de Valparaíso. Donde de intentó aprovechar el viraje que realizan los vehículos para hacer ingreso al estacionamiento para obtener una toma de frente a la patente de estos. Mientras que las tomas a vehículos estacionados fueron realizadas en el plan de Valparaíso. En la parte del experimento con muestras estáticas se posicionó la cámara frente al vehículo a una distancia estándar de un metro y medio y a una altura de 50 cm, para localizarla frente a la patente. El resto de ambas pruebas es esperar, una vez se realiza la detección, el procesamiento de las imágenes y la entrega de la información de la lectura.

Resultados experimento 4

El primer aspecto del experimento fue la evaluación de la detección de placas con el uso de la cámara. Las características en cuanto a posición de la cámara, son las descritas anteriormente e involucran el ángulo existente entre el foco de la cámara y el frente del vehículo. Las condiciones presentes para la prueba son de buena iluminación de un día soleado y nublado pero claro. Los resultados presentan, con respecto a la detección 86 verdaderos positivos. Contando 8 detecciones correctas para vehículos en movimiento sobre 20 muestras, mientras que 78 de las muestras de vehículos estacionados de 80 fueron correctamente detectadas. Donde se cuentan para las muestras móviles 12 resultados falsos positivos, mientras que para aquellas estáticas 2.

La siguiente información de interés obtenida del experimento es la tasa de identificación, la cual se trabajó sobre el total de las detecciones correctas. El número de lecturas completas de patentes de manera correcta fue de 81, con cinco errores, sobre un total de 86 muestras disponibles para realizarlas.

Es de consideración que la mayoría de las tomas en movimiento consideraron un ángulo de aproximadamente 15° que se disminuyó todo lo posible y que implicó un conteo de acierto de solo 8 detecciones exitosas. En la Tabla 4-3 se presentan la cantidad de muestras y la tasa de acierto para la detección del sistema. Además el tiempo de procesamiento fue un factor a considerar a la hora de llevar a cabo el desarrollo final del sistema, ya que los tiempos de espera en un ingreso de vehículos o el reconocimiento de una matrícula en una escena con autos en movimiento, requieren de un funcionamiento tanto rápido como robusto. El sistema implementado en la Raspberry tiene un tiempo medio de procesamiento por matrícula detectada de 29.48 segundos.

La Tabla 4-3 presenta los datos considerando la suma de las muestras móviles y estáticas como total. Se muestran los verdaderos positivos de las capturas en movimiento también los falsos negativos para estas. Además de los resultados correctos e incorrectos para las muestras de vehículos estáticos. El total de detecciones correctas pasa a ser el 100% (86) de las muestras para el análisis de la etapa de identificación de caracteres del sistema. De esto último se obtiene una

tasa de acierto de la etapa de identificación de patentes de 94.19% correspondiente 81 placas correctamente leídas, con un error de 5.81%.

Cabe destacar que el proceso de detección fue evaluado en la tesis presentada en [18] por lo que no será materia de análisis en esta oportunidad, además de que la versatilidad de esta tarea está sujeta a condiciones de iluminación, calidad en la captura de la cámara.

Los resultados obtenidos reflejan el desempeño del sistema con luz de día, tanto con un cielo nublado como despejado. Por otra parte, con respecto a las pruebas a vehículos en movimiento es necesario buscar una ubicación para la cámara, que optimice el no molestar el tránsito de los autos y que sea compatible con el funcionamiento del detector.

Tabla 4-3 Resultados del sistema de lectura de patentes

Muestras		Detecciones		Identificación	
Tipo	Cantidad	VP	FN	Correctas	Incorrectas
Móviles	20	8	12	6	2
Estáticas	80	78	2	75	3
Total [%]	100	86	14	94.19	5.81

4.3 Análisis de resultados de los experimentos

Dentro de los experimentos realizados es posible realizar una comparación entre el 2 y el 3, ya que estos fueron realizados con la misma base de datos, consistente en 2 secuencias de vídeo además de imágenes. Aunque un punto de comparación que deja en evidencia la diferencia en las características del hardware con las que fueron realizadas son los tiempos de procesamiento obtenidos, donde el primero en PC tarda 0.38 segundos y el llevado a cabo en la Raspberry tarda 28 segundos por cada patente detectada.

Debido a la diferencia en el *frame rate*, ya que las secuencias están a 30fps. y la cámara está configurada a 10fps. por tanto la cantidad de imágenes detectada por uno y otro son muy distintas. Pero en cuanto al procesamiento de las imágenes detectadas el resultado es comparable entre ambos experimentos dado que el procesamiento de las imágenes independiente de la cantidad es el mismo. Es por esto que se realiza de manera porcentual independiente del tamaño de la base de datos que se obtiene después de la etapa de detección de patentes. En la tabla se presenta la comparación de resultados entre las dos experiencias que evaluaron la misma cantidad de muestras en laboratorio. Cuantas de las muestras fueron correctamente detectadas y, en base a las detecciones de cada una, el porcentaje de identificación o tasa de acierto que demuestra cada una.

Los resultados que muestra la Tabla 4-4 indican que a pesar de la diferencia en cuanto a cantidad de detecciones realizadas a las muestras ingresadas directamente y a las adquiridas por medio del sensor los resultados globales de identificación de patentes son cercanos en ambos casos. La mayor diferencia detectada es con respecto a la cantidad de detecciones provocada por la diferencia en los tiempos de procesamiento de cada uno de los casos. En una base de 126 muestras. La columna FP dentro de la tabla correspondiente a los falsos positivos indica el porcentaje de muestras en las que se detectó en la imagen una patente siendo que no existía tal dentro de la escena, se presenta por el hecho de que implica el procesamiento de información innecesaria, pero no afecta al momento de analizar la tasa de acierto del sistema y están expresados en base a la cantidad de muestras del experimento.

Tabla 4-4 Tabla de comparación Experimentos 2 y 3

Experimento	Detecciones [%]			Identificaciones [%]		Tiempo [s]
	VP	FN	FP	Correctas	Incorrectas	
2 PC	94.44	5.56	3.97	90.75	9.25	0.38
3 RPI	58.7	41.3	0.79	89.2	10.8	29.48

Los resultados presentados en la tabla resumen el funcionamiento del sistema en los experimentos 2 y 3, pero el análisis de los resultados mostrados en sus respectivos apartados demuestra que la mayor diferencia de detección de patentes es en los casos de vídeos; donde la velocidad de captura en la grabación es de 30 fps. y la captura de la cámara se encuentra configurada a 10 fps. Específicamente dentro del universo de muestras que representa el 100% de las imágenes 94.44% versus 58.7% es una gran diferencia en cuanto a las tasas de detección de patentes para cada método. Considerando por separado los universos de imágenes detectadas para la etapa de identificación, el primero de los experimentos entrega un 90.75% de acierto mientras el segundo da un 89.2% lo cual muestra que la identificación de caracteres y el posterior proceso de comparación funcionan de manera consistente en ambos casos.

Discusión y conclusiones

En este trabajo se presentó un sistema automático de reconocimiento de, compuesto por tres tareas de procesamiento de imágenes. Las técnicas utilizadas se enfocan, primero, en la detección de la placa, luego en la segmentación de la patente con tal de obtener los caracteres y, finalmente, la identificación de estos para conseguir una lectura acorde a la imagen que fue capturada por una cámara. De los métodos seleccionados el de detección de patentes se eligió por los resultados obtenidos con él en el trabajo presentado en [18]; el de segmentación dada la facilidad de implementación y realización, y para la identificación de caracteres el uso de redes neuronales debido a lo innovador de la idea en su uso académico, además de resultar ser una técnica robusta en la clasificación.

Con respecto a la red neuronal que lleva a cabo la etapa de identificación de caracteres, la combinación entre el método de transferencia de aprendizaje y el uso de una configuración, para reconocimiento de objetos, orientada a aplicaciones móviles y embebidas entrega un desempeño robusto y, en este caso, más eficiente que otras estructuras de *deep learning*. Acerca del funcionamiento de la configuración de red neuronal utilizada, basada en la arquitectura MobileNet, es posible afirmar que, a pesar de existir algunos errores específicos, es robusto ya que ante cortes movidos el desempeño por carácter está por sobre el 90% en aciertos; y en términos generales la red reconoce más del 97% de las muestras.

El entrenamiento específico de la red implicó la transformación de la base de datos existente, a la cual se le debió realizar una segmentación de caracteres para adecuar la etapa de clasificación al reconocimiento de cada una de las 32 clases presentes en las patentes. Por otro lado, fue necesario agregar patentes con la nueva tipografía utilizada con el propósito de incluirlas dentro de las posibilidades de decisión involucrando las distintas fuentes de letras.

Es necesario notar dentro de las conclusiones la aparición de complicaciones dentro del trabajo, entre las que destaca en primer lugar la implementación de los recursos computacionales, de software, desde un entorno a otro (Windows a Debian). Portar configuraciones de un sistema operativo a otro se torna un proceso lento y complicado, cualquier error puede implicar retrasos en los trabajos.

Respecto al funcionamiento del sistema se encontraron algunas deficiencias en la detección de patentes, pero en aspectos generales los resultados son buenos. Las tasas de acierto presentadas

en el capítulo de experimentos muestran un sistema que en conjunto es robusto. Por otra parte, aspectos como la iluminación, la frecuencia de aparición de vehículos frente a la cámara y el tiempo de procesamiento de la Raspberry son factores fundamentales en el desempeño del sistema.

En cuanto al experimento número 2 en laboratorio el funcionamiento del sistema para tomas donde la matrícula se encuentra bien iluminada, entrega detecciones y lecturas correctas reflejadas en sus tasas de acierto. De todas formas, en las grabaciones existen situaciones en donde no se lleva a cabo la lectura de las patentes de la manera esperada, de lo que se concluye que aspectos como la distancia entre la cámara y el vehículo, además de la iluminación son influyentes en los resultados. Como es lógico, una gran cantidad de fotogramas por segundo implica el procesamiento de mucha información de manera continua y la relectura de patentes por parte del sistema.

Con el tercer experimento se concluye que debe existir un tiempo específico entre vehículo y vehículo, debido a lo que demora el dispositivo en procesar una imagen, una vez la patente ha sido detectada. Producto de una frecuencia de aparición de vehículos mayor, en los vídeos, a la programación del sistema y a la capacidad de procesamiento del dispositivo la tasa de detección en este caso es muy baja (36%). Además, de este experimento es posible rescatar el hecho de que con la cámara web utilizada no existen problemas de funcionamiento en condiciones de iluminación controlada. Con respecto a las tasas de reconocimiento de patentes se puede decir que el sistema es consistente en las pruebas en laboratorio, dado que los porcentajes entre ambas pruebas son muy cercanos, 90.75% con computador y 88.9% con la Raspberry.

Del experimento realizado las conclusiones extraídas con respecto al funcionamiento del sistema son que debe asegurarse una iluminación que mantenga visible la patente en todo momento; por otra parte, el posicionamiento de la cámara debe asegurar una toma frontal del vehículo con lo cual el sistema pueda lograr una detección sin problemas de la patente. De este mismo experimento se extrae información acerca del funcionamiento de la cámara con luz natural, el cual es satisfactorio dados los resultados obtenidos.

Las conclusiones generales acerca del sistema en su conjunto de software y hardware son con respecto a situaciones que suceden o pueden suceder. El primer punto es con respecto al funcionamiento de la etapa de detección, ya que mejorar esta parte del sistema implicaría una mayor tasa de detecciones y más aún detecciones correctas de las patentes; el punto anterior involucraría una disminución en la cantidad de imágenes segmentadas por *sliding window* y menos procesamiento de la red neuronal, por ende un funcionamiento más fluido. El siguiente punto es la iluminación de la zona de la patente que no está incluida en el formato actual del sistema y que sería necesaria dependiendo de la ubicación de la cámara y la cantidad de luz natural, para esto es posible la implementación de un sistema infrarrojo que ayude a captar la forma de la patente por medio de su relieve.

La Raspberry como opción, de hardware de procesamiento a utilizar, presenta algunos problemas por la cantidad de memoria de procesamiento disponible, de 1 GB. Además de no contar con una unidad de GPU con la capacidad de colaborar en el procesamiento del sistema. Dado que la etapa

de identificación es muy demandante en relación a la cantidad de datos a procesar, son 24 imágenes por cada patente leída, el tiempo entre detección y detección es mayor al del procesamiento de la información. Por otra parte, al no contar con un sistema de refrigeración aparece el problema del sobrecalentamiento producido por las exigencias de procesamiento, lo que hace necesaria una unidad de enfriamiento para el computador o cualquier placa de desarrollo que se utilice. De todas maneras, bajo condiciones de uso controladas, el sistema de la Raspberry logra el objetivo de funcionamiento y desempeño de los experimentos realizados.

Una propuesta para desarrollar en trabajos futuros es el uso de los distintos tipos de redes neuronales artificiales enfocadas a diversas aplicaciones y áreas. Tal como se desarrolló en este trabajo el uso de las redes convolucionales por ser las más adecuadas para el trabajo con imágenes, dado que su funcionamiento está orientado a matrices bidimensionales [40].

También como trabajo futuro, pero esta vez al desarrollo del sistema, se propone la utilización de otro método en la etapa de detección, una que implique una mayor robustez ante la variación de ángulos, los cambios de color en las patentes y el fondo. El mejoramiento en el desempeño de la primera tarea deberá reflejarse en una mejora al funcionamiento del sistema de identificación de patentes.

Bibliografía

- [1] E. Alegre, G. P. Martinsanz y A. d. l. E. Hueso, Conceptos y métodos en Visión por Computador (Editado), España: Grupo de Visión del Comité Español de Automática, 2016.
- [2] Y.-N. Chen, C.-C. Han, C.-T. Wang, B.-S. Jeng y K.-C. Fan, «The Application of a Concolution Neural Network on Face and License Plate Detection,» de *The 18th International Conference on Pattern Recognition*, 2006.
- [3] J. Sridevi, P. D. Anuhya, D. Ravali, C. M. Swamy y P. Sandeep, «Toll Collection Using Raspberry Pi,» *International Journal of Applied Sciences, Engineering and Management*, vol. 06, nº 02, pp. 115-117, 2017.
- [4] L. Dlagnekov, «License plate detection using Adaboost,» *Computer Science and Engineering Department*, vol. 2, 2004.
- [5] G. Li, R. Zeng y L. Lin, «Research on Vehicle License Plate Location Based on Neural Networks,» *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, vol. 3, pp. 174-177, 2006.
- [6] S. Yoshimori, Y. Mitsukura y M. Fukumi, «License Plate Detection,» de *Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, Oxford, 2003.
- [7] G. Li, R. Zeng y L. Lin, «Research on Vehicle License Plate Location Based on Neural Networks,» de *Proceedings of the First Internetonal Conference on Innovative Computing, Information and Control*, 2006.
- [8] G. Cao, J. Chen y J. Jiang, «An adaptive approach to vehicle license plate localization,» de *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No.03CH37468)*, 2003.

- [9] W. Jia, H. Zhang, X. He y M. Piccardi, «Mean Shift for Accurate License Plate Localization,» de *8th International IEEE Conference on Intelligent Transportation Systems*, Viena, Austria, 2005.
- [10] F. Martín, M. García y J. Alba, «New methods for automatic reading of VLP's (VEHICLE LICENSE PLATES),» *Signal Processing Patten Recognition and application*, 2002.
- [11] S.-Z. Wang y H.-J. Lee, «Detection and recognition of license plate characters with different appearances,» de *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Shanghai, China, 2003.
- [12] W. C. Man, L. C. On y C. H. Weng, «A Macao License Plate Recognition System,» *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pp. 18-21, 2005.
- [13] S. Draghici, «A neural network based artificial vision system for licence plate recognition,» *Int. J. Neural Syst.*, vol. 8, nº 1, pp. 113-126, 1997.
- [14] K. Jun, L. Xinyue, L. Yinghua y Z. Xiaofeng, «A novel license plate localization method based on textural feature analysis,» *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, vol. 2005, pp. 275-279, 2005.
- [15] D.-S. Kim y S.-I. Chien, «Automatic car license plate extraction using modified generalized symmetry transform and image warping,» *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570)*, vol. 3, pp. 2022-2027, 2001.
- [16] H. Li y C. Shen, «Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs,» *arXiv*, vol. 1601.05610v1, 2016.
- [17] R. Sharma, K. Jasani, A. Jaiswal y P. Bais, «Automated Parking and Security System,» *International Journal for Advance Research, Ideas and Innovations in Technology*, vol. 3, nº 2, pp. 709-715, 2017.
- [18] F. E. Pacheco, «Detección de objetos y su aplicación a patentes de vehículos,» Valparaiso, Chile, 2014.
- [19] K. V. Mardia y T. J. Hainswoth, «A Spatial Thresholding Method for Image Segmentation,» *IEEE Transaction on patern analysis and machine intelligence*, vol. 10, nº 6, pp. 919-927, 1988.
- [20] V. Franc y V. Hlaváč, «License Plate Character Segmentation Using Hidden Markov Chains,» Berlin, Alemania, 2005.

- [21] X. Shi, W. Zhao y Y. Shen, «Automatic License Plate Recognition System Based on Color Image Processing,» *Lncs*, vol. 3483, pp. 1159-1168, 2005.
- [22] C. Oz y F. Ercal, «A practical license plate recognition system for real-time environments,» *Computational Intelligence and Bioinspired Systems*, pp. 881-888, 2005.
- [23] B. Lee, K. Park, H. C. Kang, H. Kim y C. Kim, «Adaptive local binarization method for recognition of vehicle license plates,» *Combinatorial Image Analysis, Proceedings, Lecture Notes in Computer Science*, pp. 646-655, 2004.
- [24] S. Pereira y T. Pun, «Robust Template Matching for Affine Resistant Image Watermarks,» *IEEE Transaction on image processing*, vol. 9, nº 6, pp. 1123-1129, 2000.
- [25] C. Anagnostopoulos, E. Kayafas y V. Loumos, «Digital image iprocessing and neural networks for vehicle license plate identification,» *Journal Of Electrical Engineering*, vol. 1, 2001.
- [26] K. K. Kim, K. I. Kim, J. B. Kim y H. J. Kim, «Learning-based approach for license plate recognition,» de *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, Sydney, NSW, Australia, 2000.
- [27] P. Comelli, P. Ferragina, M. Granieri y F. Stabile, «Optical Recognition,» *IEEE Transactions on vehicular technology*, vol. 44, nº 4, pp. 790-799, 1995.
- [28] Y. Huang y C. Suen, «Experts for the Recognition of Unconstrained Handwritten Numerals,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, nº 1, pp. 90-94, 1995.
- [29] L. Xu, A. Krzyzak y C. Suen, «Classifiers and Their Applications to Handwriting Recognition,» *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, nº 3, pp. 418-435, 1992.
- [30] J. Cao, M. Ahmadi y M. Shridhar, «Recognition of handwritten numerals with multiple feature and multistage classifier,» *Pattern Recognition*, vol. 28, nº 2, pp. 153-160, 1995.
- [31] C. Y. Suen, C. Nadal, T. A. Mai, R. MAi, R. Legault y L. Lam, «Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts,» de *Workshop on Frontiers in Handwriting Recognition*, Montreal, Canada, 1990.
- [32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo: Morgan Kaufmann Publishers, 1988.
- [33] Y. Freund y R. Schapire, «A decision-theoric generalization of on-line learning and an application to boosting,» *Journal of Computer and System Sciences*, vol. 55, nº 1, pp. 119-139, 1997.

- [34] P. Viola y M. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features,» de *Computer Vision and Pattern Recognition*, Kauai, HI, USA, 2001.
- [35] M. A. Valencia, C. Y. Márquez y L. P. Sánchez, «Algoritmo Backpropagation para Redes Neuronales: conceptos y aplicaciones,» *Repositorio digital Instituto Politécnico Nacional Centro de Investigación en Computación*, nº 125, 2006.
- [36] P. Méndez y J. Ibarra, «Implementación de una red neuronal de convolución para el reconocimiento de poses en,» *Avances en Ciencias e Ingenierías*, vol. 6, nº 2, 2014.
- [37] R. P. Foundation, «Raspberry pi,» Raspberry Pi Foundation, [En línea]. Available: <https://www.raspberrypi.org/forums/viewtopic.php?t=107606>. [Último acceso: 12 01 2018].
- [38] S. Lee, H. Kim, K. Son y J. Park, «Car Plate Recognition Based on CNN Using Embedded System with GPU,» de *10th International Conference on Human System Interactions (HSI)*, Ulsan, Republic of Korea, 2017.
- [39] D. Ciresan, U. Meier, J. Masci, L. Gambardella y J. Schmidhuber, «Flexible, High Performance Convolutional Neural Networks for Image Classification,» de *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, 2011.
- [40] S. Kim, D. Kim y H. Kim, «A Recognition of Vehicle License Plate Using A Genetic Algorithm Based Segmentation,» de *3rd IEEE International Conference on Image Processing*, 1996.
- [41] C. Anagnostopoulos, I. Anagnostopoulos, G. Tsekouras, G. Kouzas, V. Loumos y E. Kayafas, «Using sliding concentric windows for license plate segmentation and processing,» *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, vol. 2005, pp. 337-342, 2005.
- [42] M. Shell, «Preparation of papers for IEEE Transaction and Journals,» May 2007.

A Códigos de funcionamiento del sistema

Listado A-1 Cuerpo del programa

```
1 import numpy as np
2 import cv2
3 from time import time
4 from cortes2MN3 import corte
5 #import matplotlib.pyplot as plt
6 #from data_image import data_augmentation
7 #from skimage.transform import resize
8 from keras.models import load_model
9 #import tensorflow as tf
10 import keras
11 #from keras.utils.generic_utils import CustomObjectScope
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
266
```

```

A = []
B = []
flag = 0
while(True):
    T_I = time()
    # Capture frame-by-frame
    _, frame = cap.read()

    # to detect faces in video
    #Existe aquí la variante a tomar la imagen de la placa en RGB o GRAY
    #rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    plates = plate_cascade.detectMultiScale(gray, 1.3, 5)
#    plt.imshow(frame)
    #plates = plate_cascade.detectMultiScale(rgb, 1.3, 5)

    for (x,y,w,h) in plates:
        if flag == 1:
            flag = 0
            break
        flag = 1
        #Esto podría usarse para obtener aquí la imagen en 3 canales
        #roi_gray = rgb[y+2:y+(h-15), x:x+(w-5)]
        roi_gray = cv2.cvtColor(frame[y+2:y+(h-15), x:x+(w-5)],
cv2.COLOR_BGR2RGB)
        cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0),2)
        roi_gray2=cv2.resize(roi_gray, (350, 100))
#        cv2.imwrite('patente.png',roi_gray)
        cv2.imwrite('resize.png',roi_gray2)

        clase = clasificador(corte(roi_gray2))

        #time.sleep(1)

        clase_end = ''
        for i,n in enumerate(n_im):
            a = []
            for j in range(n):
                a.append(clase[np.sum(n_im[:,i])+j])
            for k in range(4):
                if a.count(a[k]) >= 2:
                    clase_end += '{}'.format(a[k]).upper()
                    break
        """Estas tres lineas se usan para hacer pruebas libres"""
        print('',clase_end)
        #cv2.putText(frame, clase_end, (100,150), cv2.FONT_HERSHEY_TRIPLEX, 4,
(255,0,0), 2, cv2.LINE_AA)
        T_F = time()
        Tiempo = T_F - T_I
        print('',Tiempo)
        B.append(Tiempo)
        A.append(clase_end)
        archivo = open("Patentes.txt","w")
        archivo.write(str(A))
        archivo.write(str(B))
        archivo.close()

# Display the resulting frame
#cv2.imshow('frame',frame)

#time.sleep(0.5)
# press the letter "q" to save the picture
if cv2.waitKey(delay=10) & 0xFF == ord('q'):
    break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

```

```
if __name__ == "__main__":  
    TakeSnapshotAndSave()
```

Listado A-2 Pre-Procesamiento de imágenes

```
1 import numpy as np  
2 from skimage.transform import resize  
3 """  
4 Se importa la imagen de la placa completa desde la carpeta de trabajo  
5 Se le aplica un resize para estandarizar el tamaño, y poder utilizar  
6 Información a priori como lo es el tamaño y separación de las letras  
7 Deacuerdo a comparaciones de varias patentes se conoce que las letras  
8 Dentro de las imágenes obtenidas no superan los 50 pixeles de ancho.  
9 Por lo tanto se utiliza esta información para segmentarlas  
10 """  
11 cut_a = list(range(0,13,4))  
12 cut_b = list(range(45,56,3))  
13 cut_c = list(range(120,131,3))  
14 cut_d = list(range(165,176,3))  
15 cut_e = list(range(245,256,3))  
16 cut_f = list(range(290,301,3))  
cut = np.concatenate([cut_a, cut_b, cut_c, cut_d, cut_e, cut_f])  
name = ["a", "b", "c", "d", "e", "f"]  
name_im = list(range(24))  
def corte(im_complete, im_resize=[224,224,3]):  
    car = np.zeros([len(cut), 224, 224, 3])  
    for i, r in enumerate(cut):  
        car[i] = resize(im_complete[:, r:r+50], im_resize, mode='reflect')  
  
    return car
```

B Código de transfer learning para la red

Tabla 0-1 Entrenamiento para clasificadores de la red

```
1 from keras.applications import MobileNet
2 from keras.layers import Dense, Dropout, Input
3 from keras.models import Model
4 from keras.preprocessing.image import ImageDataGenerator
5 from data_image_new import data_im_bigd as data
6
7 cnn
8 MobileNet(input_shape=[224,224,3],include_top=False,pooling="avg") =
9 inputs = Input([None,None,3])
10 x = cnn(inputs)
11 x = Dropout(.2)(x)
12 out = Dense(32, activation="softmax")(x)
13 model = Model(inputs, out)
14 model.compile(optimizer="adam", loss="categorical_crossentropy")
15
16 """
label = [0,0,0,0,0,0,0..., 1, 0,0,0... ,0,0,0]
"""

d = data.data_dict("./224_04_12")
d.Data_to_dict(1, img_per_batch=5)

shift = 0.3
generator = ImageDataGenerator(rotation_range=10,
featurewise_center=True, width_shift_range=shift,
height_shift_range=shift)

#Cada epoca del modelo, es 10 epocas del data aug
d.Fit_generator(model, generator, 50, batch_size=8, verbose=2,
path_saver='./CNN/MobileNet_bigd_04_12.h5')

d.Confusion_matrix(model, confusion_matrix=[0,1,2,3,4])

#%% Save
model.save('./CNN/MobileNet_bigd_04_12.h5')
```

C Detalle del desempeño de la red por caracter

Tabla 0-2 Resultados de la red neuronal

Clase	Número total muestras	Número de aciertos	Tasa de acierto [%]	Errores comunes
0, 1, 2, 3, 5, D, H, P, S, Z	148	148	100	No hay
7	148	147	99.32	1
J	148	147	99.32	0
K	148	147	99.32	W
T	148	146	98.65	1
8	148	145	97.97	E
F	148	145	97.97	E
L	148	145	97.97	1
R	148	145	97.97	B
Y	148	145	97.97	7
6	148	144	97.3	E
W	148	144	97.3	F, H
X	148	144	97.3	W, 0, V

C Detalle del **desempeño de la red por caracter**

4	148	142	95.95	T
C	148	142	95.95	0
9	148	141	95.3	0, E, J, V, 7
V	148	141	95.3	0, T, F
B	148	140	94.6	E, 8, 2
G	148	136	91.9	0, C
N	124	115	92.74	V, T, H
U	20	20	100	No hay

D Tabla comparativa placas de desarrollo

Una alternativa a los problemas de funcionamiento presentados por la Raspberry Pi 3 podría ser analizar opciones a esta placa. La Tabla 0-3 presenta algunos mini computadores, el primero de bajo costo y el segundo con alto poder de procesamiento debido a su GPU integrada.

Tabla 0-3 Opciones a la Raspberry Pi 3

Placa	ODROID-XU4	Jetson TX1
Características		
Procesador	Samsung Exynos 2GHz octacore	Quad ARM® A57/2 MB L2
RAM	2 GB	4 GB
Wifi	Si	Si
USB	2 (3.0) 1(2.0)	1 (3.0) 1 (2.0)
HDMI	Si	HDMI
GPU	Mali-T628	CUDA NVIDIA Maxwell