



**Identificación de caracteres en placas de carros colombianos utilizando diferentes técnicas  
para su estudio comparativo**

Christian Daniel España Chamorro

Trabajo de investigación para optar al título de Ingeniero de Telecomunicaciones

Tutor

David Stephen Fernández Mc Cann

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Telecomunicaciones

Medellín, Antioquia, Colombia

2021

<b>Cita</b>	(España Chamorro, 2021)
<b>Referencia</b>	España Chamorro C. D. (2021). <i>Identificación de caracteres en placas de carros colombianos utilizando diferentes técnicas para su estudio comparativo</i> [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia.
<b>Estilo APA 7 (2020)</b>	



Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## DEDICATORIA

Dedico este trabajo a mi madre, Lucia, quien ha sido el faro de luz en mi vida y me ha apoyado en todos los aspectos y decisiones que he tomado.

A mi padre, Guillermo, quien ha sido mi guía y me ha mantenido en el mejor rumbo posible.

A mi hermana, Andrea, quien ha sido que una hermana, es mi gran amiga, que además me ha orientado y ha cultivado el anhelo de superación y progreso.

Finalmente, a todas esas personas, familia y amigos que directa e indirectamente me han ayudado durante mi vida universitaria y fuera de ella, pues ocupan un lugar especial en mi vida.

## AGRADECIMIENTO

Quiero mostrar mis más sinceros agradecimientos a mi tutor de trabajo de grado, el profesor David Fernandez Mc cann, por su paciencia, apoyo y dirección. Gracias a David he podido finalizar este proyecto, fue un camino lleno de retos, mucho aprendizaje y emocionante.

A mi familia, por estar a mi lado cuando todo ha ido bien, y más que todo ayudarme a levantar cuando todo parece ir por mal camino, este proyecto de grado les pertenece en la mayor medida.

Gracias a todos lo que me han ayudado y me han mostrado el camino para terminar este trabajo.

Gracias a la universidad de Antioquia y a todos los profesores que hacen parte del programa de ingeniería de telecomunicaciones por sus valiosos conocimientos durante el programa académico.

## Tabla de contenido

<b>Resumen</b> .....	XVI
<b>Abstract</b> .....	XVII
1 INTRODUCCIÓN .....	7
1.1 Oportunidades que existen de resolver la situación desde la tecnología.....	8
1.1.1 Tecnología RFID en un microchip .....	8
.....	9
1.1.2 Inteligencia artificial .....	10
1.2 Procedimiento seleccionado.....	11
2 Planteamiento del problema.....	14
2.1 Pregunta de investigación.....	14
2.2 Hipótesis.....	14
2.3 Contexto actual y futuro después de la implementación .....	14
2.3.1 Contexto actual .....	14
2.3.2 Contexto futuro .....	15
2.4 Formulación del problema tecnológico.....	16
2.4.1 Antecedentes de investigación que han buscado solución.....	16

2.4.2	Antecedentes tecnológicos o procedimentales que lo han resuelto con herramientas ya construidas.....	17
2.4.3	Sistemas comerciales para resolver el problema .....	17
2.4.4	Procedimientos tecnológicos que podrían utilizarse.....	18
2.4.5	Procedimiento tecnológico por realizarse para obtener la solución.....	19
	.....	20
3	Justificación .....	21
4	Objetivos.....	22
4.1	Objetivo general.....	22
4.2	Objetivos específicos.....	22
5	Marco teórico.....	24
5.1	Adquisición de la imagen.....	25
5.1.1	Modelo de color RGB.....	25
5.1.2	Modelo de color HSV .....	25
5.2	Procesamiento de la imagen.....	26
5.2.1	Filtrado de la imagen .....	26
5.2.2	Escala de grises.....	26
5.2.3	Binarización .....	27
5.2.4	Detección de contornos.....	28

5.2.5	Transformación morfológica: .....	29
5.3	Técnicas de machine learning .....	30
5.3.1	Descriptores de características.....	30
5.3.2	Algoritmos de Clasificación .....	34
5.4	Técnicas de Deep learning .....	38
5.4.1	Neurona modelo biológico.....	39
5.4.2	Neurona modelo artificial .....	40
5.4.3	Redes neuronales Feed-forward.....	43
5.4.4	Función de activación .....	45
5.4.5	Entrenamiento de redes neuronales .....	48
5.4.6	Redes neuronales convolucionales .....	54
5.4.7	Arquitectura básica red neuronal convolucional.....	55
5.5	Arquitecturas convolucionales para la clasificación de imágenes .....	59
5.5.1	VGGNet .....	59
5.5.2	GoogleNet o Inception.....	60
5.5.3	ResNet.....	61
5.6	Deep learning para la detección de objetos .....	62
5.7	Detección de objetos .....	63
5.8	Algoritmo YOLO .....	65

6	Base tecnológica a implementar en el procedimiento.....	75
6.1	Equipo de computo.....	75
6.2	Tecnologías empleadas software y librerías.....	75
6.2.1	Python .....	75
6.2.2	Scikit-learn.....	76
6.2.3	OpenCV .....	76
6.3	Darknet.....	76
6.4	Servicios a utilizar.....	77
6.5	Google Colaboratory .....	77
7	Metodología.....	78
7.1	Adquisición de la imagen.....	78
7.2	Preprocesado de las imágenes.....	79
7.3	Extracción placa vehicular .....	80
7.4	Segmentación de los caracteres.....	81
7.5	Data augmentation.....	84
7.6	Extracción de características .....	85
7.6.1	Histograma de gradientes orientados (HOG).....	85
7.6.2	HAAR .....	86
7.6.3	Local Binary Pattern (LBP) .....	87

7.7	Entrenamiento y creación de modelos .....	89
	n_estimators.....	90
7.8	Entrenamiento YOLO .....	91
7.8.1	Etiquetado de imágenes .....	91
7.8.2	Configuración de archivos .....	92
7.8.3	Creación de archivos.....	93
7.8.4	Entrenamiento YOLO.....	95
8	Resultados.....	96
9	Comparación de los resultados obtenidos con trabajos similares .....	97
10	Conclusiones.....	98
11	Líneas futuras.....	99
12	Referencias.....	100
<b>Anexos</b>	.....	<b>109</b>

## INDICE DE FIGURAS

<b>Figura 1</b> .....	7
Parque automotor registrado en el RUNT .....	7
<b>Figura 2</b> .....	9
Componentes tecnología RFID.....	9
<b>Figura 3</b> .....	10
Sistema de control de acceso vehicular. ....	10
<b>Figura 4</b> .....	11
Sistema ALPR basado en visión artificial. ....	11
<b>Figura 5</b> .....	20
Etapas por seguir para obtener los modelos.....	20
<b>Figura 6</b> .....	20
Etapas por seguir para usar los modelos. ....	20
<b>Figura 7</b> .....	31
Método HOG. ....	31
<b>Figura 8</b> .....	32
Filtros Haar. ....	32
<b>Figura 9</b> .....	33

Método LBP.....	33
<b>Figura 10</b> .....	35
Hiperplano de separación SVM.....	35
<b>Figura 11</b> .....	37
Algoritmo KNN.....	37
<b>Figura 12</b> .....	38
Algoritmo Random Forest.....	38
<b>Figura 13</b> .....	40
Estructura básica de una neurona biológica.....	40
<b>Figura 14</b> .....	41
Modelo de neurona artificial.....	41
<b>Figura 15</b> .....	44
Ejemplo de red neuronal feed-forward, la capa de entrada consta de tres entradas, dos capas ocultas cada una con cuatro neuronas y una capa de salida con dos neuronas.....	44
<b>Figura 16</b> .....	46
Función ReLU.....	46
<b>Figura 17</b> .....	47
Función sigmoid.....	47
<b>Figura 18</b> .....	52

Overfitting visto de forma gráfica..... 52

**Figura 19**..... 53

Red original (izquierda) y red a la que se le ha aplicado Dropout (derecha)..... 53

**Figura 20**..... 54

Diagrama básico del funcionamiento de los campos receptivos cerebrales ..... 54

**Figura 21** ..... 56

Representación visual de una CNN con dos capas de convolución y pooling ..... 56

**Figura 22**..... 58

Max pooling con un filtro 2x2 y un paso de 2 ..... 58

**Figura 23**..... 60

módulo en Inception V1. Los bloques amarillos de convolución 1x1 reducen el tamaño de entrada antes de alimentarlos a convoluciones 3x3 y 5x5 ..... 60

**Figura 24**..... 61

Conexión residual, donde se utiliza una conexión de omisión para omitir la entrada a las siguientes capas..... 61

**Figura 25** ..... 63

Tabla comparativa de detección sobre el dataset Pascal Voc ..... 63

**Figura 26**..... 64

Analogía de una CNN para la detección de objetos ..... 64

<b>Figura 27</b> .....	66
Imagen dividida en una cuadrícula 3x3 .....	66
<b>Figura 28</b> .....	67
Cuadrícula de la esquina superior izquierda de la figura 27 .....	67
<b>Figura 29</b> .....	68
Cuadrícula central izquierda de la figura 27 .....	68
<b>Figura 30</b> .....	69
Dos objetos en la misma celda .....	69
<b>Figura 31</b> .....	70
Tipo de cajas de anclaje .....	70
<b>Figura 32</b> .....	71
Múltiple detección sobre los objetos de interés .....	71
<b>Figura 33</b> .....	72
Intersección sobre unión .....	72
<b>Figura 34</b> .....	74
Técnica de supresión no máxima .....	74
<b>Figura 35</b> .....	78
Conjunto de imágenes de automóviles colombianos .....	78
<b>Figura 36</b> .....	79

Procedimiento para resaltar placa amarilla sobre el fondo .....	79
<b>Figura 37</b> .....	80
Procedimiento para resaltar placa blanca sobre el fondo .....	80
<b>Figura 38</b> .....	81
Extracción placa vehicular .....	81
<b>Figura 39</b> .....	82
Binarización adaptativa sobre la placa vehicular .....	82
<b>Figura 40</b> .....	83
Segmentación caracteres placa vehicular .....	83
<b>Figura 41</b> .....	84
Conjunto de datos de letras y números .....	84
<b>Figura 42</b> .....	86
Extracción de características mediante HOG .....	86
<b>Figura 43</b> .....	88
Función para calcular el vector de características con LBP .....	88
<b>Figura 44</b> .....	88
Imagen original, en escala de grises y siendo afectada por códigos LBP .....	88
<b>Figura 45</b> .....	92
Etiquetado de imágenes a través de labelImag .....	92

<b>Figura 46</b> .....	94
Contenido del archivo .data. ....	94

## INDICE DE TABLAS

Tabla 1 .....	96
<i>Porcentaje de predicción sobre los caracteres vehiculares implementando modelos de machine learning</i> .....	96
Tabla 2 .....	96
<i>Porcentaje de predicción sobre los caracteres vehiculares implementando modelos de deep learning</i> .....	96

## Resumen

El presente trabajo de grado es un estudio, implementación y posterior comparación de técnicas de visión artificial que están comprendidas bajo el enfoque de Machine Learning y Deep Learning mediante el desarrollo de un software de reconocimiento de placas vehiculares colombianas.

Este tipo de software son llamados ALPR (Automatic License Plate Recognition), los cuales son capaces de extraer e identificar los caracteres que componen las matrículas vehiculares a partir de imágenes capturadas por una cámara fotográfica.

En este proyecto se manejan técnicas básicas de procesamiento y segmentación de imágenes, como la binarización y la detección de contornos, ya que son pieza fundamental para extraer los números y letras de las matrículas. Adicionalmente, para el enfoque de Machine Learning se entrenan nueve modelos de clasificación, en los cuales se combinan tres descriptores de características, entre ellos están el Histograma de Gradientes, Local Binary Patterns y características HAAR; con tres algoritmos de clasificación tales como Support Vector Machine, Random Forest y KNN.

Por último, se entrena el algoritmo de detección y clasificación de objetos YOLO, con la ayuda del framework Darknet. Todo esto con el objetivo de encontrar y determinar cuál conjunto de técnicas ofrece mayor probabilidad de éxito al momento de reconocer los caracteres de la placa vehicular.

## **Abstract**

This degree work is a study, implementation and subsequent comparison of artificial vision techniques that are included under the Machine Learning and Deep Learning approach through the development of a Colombian license plate recognition software. This type of software is called ALPR (Automatic License Plate Recognition), which are capable of extracting and identifying the characters that make up vehicle license plates from images captured by a photographic camera. In this project, basic image processing and segmentation techniques are handled, such as binarization and contour detection, since they are a fundamental part of extracting the numbers and letters of the license plates. Additionally, for the Machine Learning approach, nine classification models are trained, in which three characteristic descriptors are combined, among them are the Gradient Histogram, Local Binary Patterns and HAAR characteristics; with three classification algorithms such as Support Vector Machine, Random Forest and KNN. Finally, the YOLO object detection and classification algorithm is trained, with the help of the Darknet framework. All this with the aim of finding and determining which set of techniques offers the greatest probability of success when recognizing the characters on the license plate.

## 1 INTRODUCCIÓN

El aumento creciente de las redes viales urbanas y nacionales contemporáneas en las últimas décadas, ha traído consigo que el número de vehículos este aumentando cada día. Según las estadísticas presentadas en el RUNT en el año 2019 el total de licencias de conducción activas aumento en un 18.1% [1]. Además, para el año 2021 el parque automotor registrado ha venido en aumento, tal como se observa en la figura 1.

**Figura 1**

*Parque automotor registrado en el RUNT*



*Nota.* Fuente: [1]

Este aumento del parque automotor ha generado que la administración del tráfico se haya vuelto más compleja en múltiples escenarios en donde es necesario conocer la información del vehículo a través de la matrícula del mismo.

Entre estos escenarios tenemos el exceso de velocidad en las carreteras, ya que tal hecho es una infracción a la ley en cualquier parte del mundo y debe ser penalizado; otro escenario donde

se presentan complicaciones es en las ciudades, propiamente en el control de estacionamientos para empresas o establecimientos que prestan servicios de aparcamiento y que desean administrar el ingreso y salida de vehículos con el fin de conocer el flujo vehicular y garantizar la seguridad a los usuarios como al establecimiento.

Además, a los sucesos anteriormente mencionados se han sumado el aumento del comportamiento criminal, como el robo de automóviles y otro amplio espectro de violaciones que se están volviendo cotidianas en la sociedad colombiana.

Este aumento significativo de alteraciones en el orden ha traído consigo la necesidad de contar con sistemas computacionales eficientes para ser utilizados como una herramienta de apoyo a la seguridad, el control de acceso y/o flujo vehicular en distintos escenarios.

## **1.1 Oportunidades que existen de resolver la situación desde la tecnología**

Existen diferentes tecnologías de las que se pueden hacer uso para resolver la problemática planteada a la hora de hacer gestión al tráfico vehicular. Entre estas soluciones tenemos:

### ***1.1.1 Tecnología RFID en un microchip***

RFID es la Identificación por Radiofrecuencia, en inglés Radio Frequency IDentification. Es un sistema enfocado en almacenamiento y recuperación de información remota, los cuales son usados por medio de transmisores o antenas.

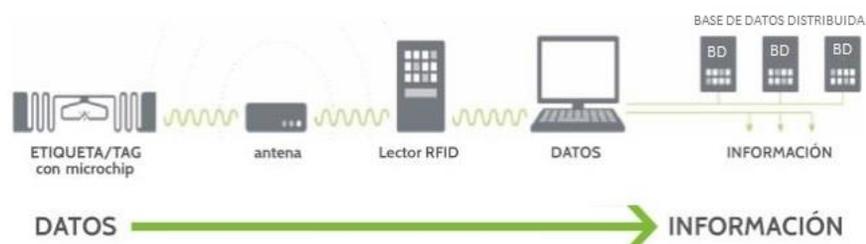
La tecnología RFID fue diseñada para identificar objetos a distancia e interconectarlos dentro de una configuración existente, sin la necesidad de tener contacto físico directo entre estos.

El RFID tiene como pieza fundamental el tag, también llamada etiqueta inteligente, la cual está compuesta por un microchip. Por lo tanto, esta solución pretende instalar tag's en los vehículos, los cuales generan códigos de acceso a la información mediante ondas de radiofrecuencia. Se busca que dichas ondas sean leídas mediante receptores especializados, para así obtener la información del vehículo que se encuentra alojada en bases de datos distribuidas programadas para tal fin, con el objetivo de validar y determinar el estado actual del vehículo y su

propietario frente a las autoridades y normas que regulan su desplazamiento por el territorio nacional. Los componentes principales de la tecnología RFID se puede observar en la figura 2.

**Figura 2**

*Componentes tecnología RFID*



*Nota.* Fuente Autor

Un escenario que puede hacer uso de esta tecnología es un control de acceso vehicular como se observa en la figura 3. Al acercarse el vehículo al acceso, el microchip o tag emite la información a la lectora de larga distancia. La lectora verifica la información con el sistema gestor de bases de datos en búsqueda de permisos para ingresar. Si la respuesta es satisfactoria entonces se procede a la apertura de la barrera.

### Figura 3

*Sistema de control de acceso vehicular.*



*Nota.* Fuente <https://www.ipsolutions.com.pe/control-de-acceso-vehicular.html>

#### 1.1.2 Inteligencia artificial

Inteligencia artificial es una rama de la informática que apunta a crear máquinas inteligentes que funcionan y reaccionan como los humanos.

Una de las tecnologías innovadoras que está surgiendo actualmente es la visión artificial, ya que por medio de algoritmos de aprendizaje automático permite a las computadoras ver imágenes de la misma manera que lo hace la visión humana.

El objetivo de la visión artificial no es solo ver, sino también procesar y analizar la información de la escena para proporcionar resultados útiles con los cuales tomar una decisión adecuada.

El campo de la visión artificial está ampliamente extendido en sistemas de control de tráfico vehicular, existe una tecnología llamada ALPR (Automatic license plate recognition), la cual se basa en la identificación del vehículo a partir de su matrícula vehicular, mediante el reconocimiento de los caracteres presentes en la placa vehicular en una transmisión de video o una imagen sin interacción humana.

Los sistemas ALPR normalmente poseen de dos componentes: una cámara que se utiliza para capturar imágenes de vehículos y un software que extrae las placas vehiculares de las imágenes capturadas mediante una herramienta de reconocimiento de caracteres y/o algoritmos de

aprendizaje que permiten que los píxeles se traduzcan en caracteres numéricos legibles, haciendo posible el cruce de información con una base de datos externa para validar y determinar el estado actual del vehículo y su propietario

Comúnmente los sistemas ALPR funciona en cinco etapas, incluyendo la adquisición de la imagen, preprocesamiento de la imagen, extracción placa vehicular, segmentación de los caracteres de la placa vehicular y finalmente reconocimiento óptico de caracteres para predecir individualmente lo caracteres.

En estos sistemas el reconocimiento de las imágenes de los números y letras de las matrículas se puede implementar mediante distintas técnicas de aprendizaje de máquina, siendo las más habituales las redes neuronales de y las máquinas de soporte de vectores (SVM). Incluso algunos sistemas optan por utilizar herramientas comerciales de reconocimiento óptico de caracteres (OCR) para este fin.

#### Figura 4

*Sistema ALPR basado en visión artificial.*



### 1.2 Procedimiento seleccionado

El presente trabajo es un estudio y posterior comparación de técnicas de Machine y Deep learning a través del desarrollo de un sistema de reconocimiento automático de matrículas de vehículos. Por lo tanto, se espera mostrar en un cuadro comparativo en donde se indique los diferentes porcentajes de acierto que tienen las técnicas de visión por computadora sobre los

caracteres que componen la matricula vehicular. Además, que el sistema ALPR que se desarrollará sea de gran ayuda, pues intentará automatizar el proceso de control de tráfico en diferentes escenarios como el pago automático de peajes, sistemas de gestión estacionamientos, búsqueda y seguimiento de vehículos, como también apoyar a la seguridad en los diferentes aspectos criminales que se presentan con los vehículos en la sociedad colombiana.

La propuesta se seleccionó debido a tres factores decisivos:

- Facilidad de implementación.

En la tecnología RFID es necesario instalar el tag sobre el vehículo en un lugar donde sea fácil su detección con el fin de garantizar una correcta comunicación entre el tag y la lectora, el punto negativo es que se debe reforzar que el microchip se instale en un lugar donde no pueda ser retirado por personas no autorizadas, y además que no esté expuesto a daño o deterioro en caso de accidentes. Por otro lado, en los sistemas ALPR este tipo de problema no tiene gran relevancia, pues la placa vehicular estará en su gran mayoría en una misma posición, y además la pieza fundamental del sistema que en este caso es la matricula vehicular no se convierte en un objeto de posible hurto como sucede con el tag en la solución de la tecnología RFID.

- Seguridad de la información.

Teniendo en cuenta que la tecnología RFID hace uso de la radiofrecuencia para el intercambio de información, y a pesar de que se tiene en cuenta protocolos de comunicación y seguridad, ha sido un sistema que ha tenido fallas en el blindaje y protección en el intercambio de datos, y no se ha logrado mejorar la privacidad en el envío de información entre estos sistemas, por lo que existe el riesgo de que la información que llega a los lectores RFID pueda ser interceptada por terceros [2].

Esta posible fuga de información no existe en los sistemas ALPR, ya que estos sistemas utilizan un software que por medio de técnicas de visión artificial y algoritmos de aprendizaje extraen información relevante de la imagen para proporcionar datos legibles.

- Costos.

Evaluando el costo de las soluciones, para dar inicio al sistema basado en RFID, se necesita una lectora y que todos los usuarios adquieran el tag inteligente. Caso contrario sucede en el sistema ALPR, donde solo se necesita una cámara para la tomar de imágenes de la placa vehicular y el software de detección. Adicionalmente, si se considera en aumento el número de usuarios, el sistema basado el RFID genera un costo adicional cada vez que se agregue un nuevo usuario al sistema ya que se necesita de un tag por cada auto, mientras que en un sistema de reconocimiento óptico no presenta gastos adicionales de ese tipo.

## **2 Planteamiento del problema**

### **2.1 Pregunta de investigación.**

¿cuál es el conjunto de técnicas que se utilizan para el reconocimiento de imágenes expuestas en el campo de Machine y Deep learning que ofrezcan mayor porcentaje de predicción sobre los caracteres que componen la placa vehicular?

### **2.2 Hipótesis**

El enfoque de deep learning ofrece algoritmos basados en técnicas de redes neuronales convolucionales las cuales se asemejan al comportamiento del cuerpo humano, junto a estos algoritmos como YOLO se obtendrá un mejor resultado principalmente en la predicción de los caracteres de la placa vehicular.

### **2.3 Contexto actual y futuro después de la implementación**

#### **2.3.1 Contexto actual**

En el proyecto de grado presentado en la universidad del cauca en el año 2018 [3]. Para la etapa de extracción de características se utiliza la Transformada Wavelet Discreta (DWT, Discrete Wavelet Transform), la cual otorga un análisis de señales transitorias y variantes en el tiempo, con el fin de identificar tendencias, cambios bruscos, inicio o finalización de eventos que ocurren en una determinada señal. Posteriormente para la etapa de identificación de los caracteres que componen la matricula vehicular, el proyecto hace uso de la técnica de correlación digital de imágenes; tal técnica es utilizada en el reconocimiento de patrones, ya que permite identificar la mayor correspondencia entre una imagen adquirida y las imágenes de una base de datos

previamente establecida, de tal manera que al calcular el coeficiente de correlación entre una pareja de píxeles, el valor más alto que se obtenga corresponderá a la imagen buscada.

En esta investigación presentada en el año 2019 [4], hacen uso del Análisis de componentes principales (Principal Component Analysis, PCA) para la etapa de extracción de características. PCA ayuda a identificar patrones en los datos basados en la correlación entre características. PCA tiene como objetivo encontrar las direcciones de variación máxima en datos de alta dimensión y proyectarlos en un nuevo subespacio con dimensiones iguales o menores que el original. Prácticamente, PCA convierte una matriz de  $n$  características en un nuevo conjunto de datos de menos que  $n$  características. Es decir, reduce la cantidad de características mediante la construcción de nuevas variables numéricas más pequeñas que capturan una porción significativa de la información encontrada en las características originales. En seguida, para la etapa de identificación de caracteres, se ha usado el algoritmo de clasificación K-Nearest Neighbor (KNN).

En este estudio presentado en el año 2018 [5]. En la etapa de extracción de características se utiliza el descriptor Haar-Like Features, el cual crea un vector de características que más adelante es usado para predecir los caracteres con la ayuda del clasificador Support Vector Machine.

Para este trabajo de investigación [6], luego de la segmentación de los caracteres de la placa vehicular, se ha implementado una red neuronal convolucional, la cual tiene como objetivo encontrar las características más relevantes presentes en los caracteres como también la identificación o predicción de los mismo.

### **2.3.2 Contexto futuro**

Tal como se puede evidenciar, existen múltiples estudios que pretenden dar solución a sistemas ALPR, pero todos ellos toman diversos caminos para alcanzar el objetivo. Adicionalmente, se puede notar que dichos estudios solo han tomado una técnica para cada etapa de su desarrollo, ya sea en la etapa de extracción de características como la etapa de identificación o predicción de los caracteres presentes en la placa vehicular.

El presente trabajo de grado desea implementar múltiples descriptores en la etapa de extracción de características, como los son: HAAR, HOG y LBP. Como también hacer uso de varios algoritmos de clasificación tales a: SVM, K-MEANS y RANDOMFOREST. Todo esto con el fin de realizar una comparación entre los diferentes descriptores y clasificadores, en aras de encontrar el conjunto de técnicas que brinden el mayor porcentaje de acierto sobre los dígitos presentes en una matrícula vehicular.

Adicionalmente, se desea ir un paso más allá y adentrarse en el campo del deep learning, para así implementar un algoritmo de detección de objetos tal como YOLO, el cual será usado para la segmentación y predicción de los caracteres presentes en la placa vehicular. Las anteriores actividades tienen como finalidad mostrar algunas técnicas que existen entre Machine Learning y Deep learning a la hora de predecir los caracteres.

Por lo tanto, el trabajo de grado ofrece una herramienta que ayudará a diseñar más fácilmente sistemas ALPR, o sistemas de cualquier índole que involucre reconocimiento de imágenes en el mundo de la visión artificial.

## **2.4 Formulación del problema tecnológico**

### ***2.4.1 Antecedentes de investigación que han buscado solución***

Muchas de las investigaciones han utilizado para la etapa de extracción de la placa vehicular métodos de detección de borde [7], color [8] y textura [9]. De los anteriores, se ha notado que el método de detección de borde en conjunto con operadores morfológicos es eficiente a la hora de localizar la matrícula vehicular. Para la etapa de segmentación de los caracteres se han encontrado métodos como la binarización [10] y la optimización global [11], de los cuales, el procedimiento de binarización ha tenido mejores resultados para esta tarea. Finalmente, para la etapa de reconocimiento de caracteres, las técnicas utilizadas en los sistemas ALPR, se pueden clasificar en técnicas basadas en el aprendizaje y técnicas de comparación de plantillas o matching. Para las técnicas fundamentadas en aprendizaje, las investigaciones han usado Support Vector

Machine (SVM) [12], Redes neuronales artificiales (ANN) [13]. Además, la comparación de plantillas o matching es una técnica usual dado que su implementación es fácil y rápida [14].

#### ***2.4.2 Antecedentes tecnológicos o procedimentales que lo han resuelto con herramientas ya construidas***

Existen algunos métodos disponibles para reconocer la placa vehicular, tal como se evidencia en [15], el autor del artículo utiliza OpenALPR [16], el cual es una biblioteca de código abierto diseñada específicamente para reconocer matrículas de todo el mundo. OpenALPR funciona en dos etapas principales: la primera de ellas es la detección de la matrícula y la segunda el reconocimiento de los caracteres presentes en la placa vehicular. Para la primera etapa que es la detección de la matrícula vehicular, la librería ofrece dos opciones para la extracción de características, tales a: características HAAR y características LBP. Para la segunda etapa, la cual es el reconocimiento de los caracteres, OpenALPR aplica Tesseract [17], que es un motor de OCR de código abierto, el cual debe ser entrenado con los caracteres de las matrículas que se desea reconocer.

Se encontró una herramienta similar a OpenALPR y que se hace llamar a sí misma como la competencia directa, la cual es: PLATE RECOGNIZER [18], con sede en Silicon Valley y Europa, el software ofrece una API en la nube, y se sabe que está desarrollada bajo el enfoque de redes neuronales profundas (Deep Neural Network, DNN).

#### ***2.4.3 Sistemas comerciales para resolver el problema***

Vigilant Solutions [19] es una gran empresa de videovigilancia y gestión. La compañía comprende completamente las complejidades de cómo usar la tecnología para la vigilancia, la detección del delito y la disuasión. Tiene un conjunto completo de productos que incluyen reconocimiento facial, reconocimiento automático de matrículas, servicios de investigación de

delitos y cámaras personales. Cuando están completamente integrados, los productos proporcionan la mejor solución para sus clientes principales.

Genetec [20] es una empresa que ofrece tanto reconocimiento automático de matrículas, como también videovigilancia IP y control de acceso. Security Center AutoVu es la solución automática de reconocimiento automático de matrículas (ALPR) de Genetec. AutoVu para estacionamiento ayuda a las instituciones educativas, ciudades y empresas a beneficiarse de un mayor cumplimiento de los usuarios y una mayor eficiencia operativa. Las soluciones AutoVu consisten en una gama de potentes dispositivos ALPR móviles y fijos, además de software y servicios en la nube. Todos los anteriores diseñados para facilitar y mejorar la gestión y aplicación del estacionamiento para proporcionar una mejor experiencia para los conductores.

#### ***2.4.4 Procedimientos tecnológicos que podrían utilizarse***

En el proyecto [21], para la detección de la placa vehicular se siguieron las siguientes etapas: Conversión de la imagen a escala de grises, luego se eliminó el ruido en la imagen mediante la función blur, en seguida se aplicó filtro Sobel para resaltar los bordes verticales. A continuación, se procedió a binarizar la imagen mediante el método de Otsu. Luego se utilizó el operador morfológico igual a cierre, con el objetivo de tener posibles regiones de la imagen que puedan contener la placa vehicular. A continuación, se aplicó el método de contornos para filtrar las posibles regiones que contengan la placa, en función de su área y relación de aspecto. Posteriormente se entrenó el clasificador SVM para predecir si las regiones de la imagen destacadas con la función de contornos contienen la placa vehicular. Una vez se obtuvo la placa, se procede a la segmentación de los caracteres, usando nuevamente la función de contornos en función de relación de aspecto y altura. Finalmente, para la etapa de predicción de los caracteres se entrena el algoritmo de clasificación K-Nearest Neighbor (KNN).

En el artículo [22], para la detección de la placa vehicular se utilizó ordenadamente los siguientes pasos: Conversión de la imagen a escala de grises, se aplicó filtro Sobel para encontrar los bordes verticales, luego binarización de la imagen con el método de Otsu, a continuación, se utilizó operadores morfológicos tales a erosión y dilatación, en aras de tener posibles regiones de

la imagen que puedan contener la placa vehicular. Luego se aplicó el método de contornos para filtrar las posibles regiones que contengan la placa, en función de su área y relación de aspecto. Después, para predecir si cada región es o no una placa vehicular se entrenó el algoritmo de clasificación Support Vector Machine (SVM). Luego de obtener la placa, se segmentan los caracteres usando un histograma ecualizado. Por último, para la predicción de los caracteres se entrena una red neuronal artificial (ANN), aplicando Perceptrón Multicapa (MLP).

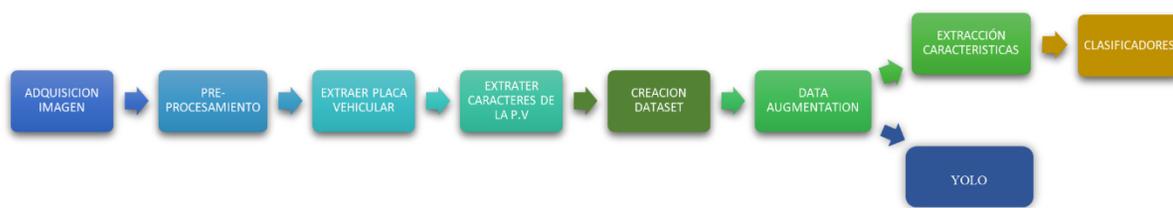
#### ***2.4.5 Procedimiento tecnológico por realizarse para obtener la solución.***

Inicialmente se debe crear la base de datos de imágenes que contienen placas de vehículos colombianos, la cual servirá como base fundamental para el desarrollo del trabajo de grado. La segunda etapa llamada preprocesamiento, se enfoca en eliminar información no necesaria obtenida por la adquisición de la imagen, y además intentar destacar la placa vehicular sobre el fondo que la contiene, con el objetivo de que al momento de aplicar la etapa de extracción de la placa vehicular se tenga más éxito en extraerla. Logrado con éxito la extracción de la matrícula vehicular, se procede a segmentar los caracteres que la componen utilizando técnicas de procesamiento digital de imágenes. Con los caracteres extraídos y utilizando técnicas de *data augmentation* [22] para aumentar la cantidad de muestras se crea el *dataset* de entrenamiento. Luego, se utilizará histogram of gradient, local binary pattern y haar-like features para extraer las características de las imágenes de los caracteres almacenados en el *dataset* de entrenamiento. Finalmente, las características extraídas por los anteriores algoritmos servirán de entrada para los clasificadores a utilizarán como: support vector machine, k-nearest neighbour y random forest, los cuales serán de gran ayuda para el entrenamiento de los modelos de clasificación. Además, como se ha mencionado anteriormente se utilizará una técnica de deep learning para la identificación de los caracteres basado en YOLO.

Finalmente, los diez modelos obtenidos a partir de los enfoques de machine y deep learning servirán para la clasificación de los caracteres de las placas y con estos evaluar las diferentes rutas y rendimientos que pueden ser utilizadas para esta tarea.

**Figura 5**

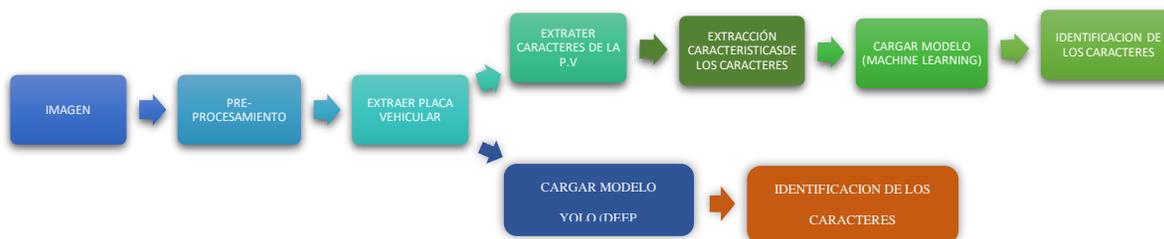
*Etapas por seguir para obtener los modelos.*



Una vez se ha logrado con éxito el entrenamiento de los modelos, es posible hacer uso de ellos para realizar las predicciones sobre los caracteres que componen la placa vehicular y la figura 6 muestra los pasos a seguir, como se observa en ella los enfoques de machine learning y deep learning se dividen a partir de la etapa de extracción de la placa vehicular, pues como se ha mencionado anteriormente YOLO reemplaza las fases de segmentación de los caracteres, extracción de características y uso clasificadores, para lograr la identificación de los caracteres.

**Figura 6**

*Etapas por seguir para usar los modelos.*



### **3 Justificación**

El proyecto de grado se basa en un estudio e implementación de diferentes técnicas que ofrecen los enfoques de machine learning y deep learning, para encontrar cuales técnicas ofrecen mejor desempeño al momento de reconocer y predecir imágenes. Se diseñan diez tipos software para el reconocimiento de caracteres en placas de carros colombianos y estos pueden ser utilizados como complemento principal para diseñar una herramienta comercial, ya sea para centros comerciales que desean gestionar el servicio de parqueadero, sistema de foto multas, una herramienta de domótica para controlar automáticamente el parqueadero de un hogar o un conjunto de apartamentos, entre otras.

## 4 Objetivos

### 4.1 Objetivo general

Comparar el desempeño de diferentes técnicas de reconocimiento de imágenes que utilizan machine learning y deep learning para clasificar los caracteres de una placa vehicular colombiana.

### 4.2 Objetivos específicos

Para desarrollar el sistema que sea capaz de realizar el objetivo general se requieren llevar a cabo los siguientes objetivos específicos:

1. Crear la base de datos para el entrenamiento de los modelos, a partir de fotos que contengan placas vehiculares.
2. Elegir tres esquemas de extracción de características y tres clasificadores con los cuales se trabajará la creación y uso de los modelos.
3. Implementar los sistemas de extracción de características seleccionados y los clasificadores, así como la etapa de preprocesado, para ser aplicado a toda la base de datos del entrenamiento.
4. Entrenar los modelos de clasificación cruzando extractores de características y clasificadores, para evaluar en conjunto sus desempeños y determinar cuál es la ruta optima extractor de características-clasificador (machine learning).

5. Entrenar el algoritmo de clasificación de objetos YOLO (deep learning), para evaluar su desempeño al momento de identificar los caracteres de la placa vehicular.

## 5 Marco teórico

Machine learning se fundamenta en el desarrollo de programas informáticos con inteligencia artificial, los cuales ofrecen a los sistemas la capacidad de aprender y mejorar a través de la experiencia. Para lograr que estos algoritmos aprendan, existen dos métodos: aprendizaje supervisado y aprendizaje no supervisado.

- Aprendizaje supervisado

En el aprendizaje supervisado antes de realizar una clasificación de imágenes se debe primeramente realiza una preparación en las imágenes. Esto consiste en preparar un conjunto de datos perfectamente etiquetado. Por lo tanto, se entrena al algoritmo otorgándole las preguntas, denominadas características, y las respuestas, denominadas etiquetas. Esto se hace con la finalidad de que el algoritmo las combine y pueda hacer predicciones [23].

- Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, en el no supervisado solo se le otorgan las características, sin proporcionarle al algoritmo ninguna etiqueta. Su función es la agrupación, por lo que el algoritmo es el encargado de realizar la clasificación a partir de las características que reciba en la entrada [23].

Para el desarrollo exitoso de este trabajo de grado se estudiarán métodos y técnicas de aprendizaje supervisado como lo que se presentan a continuación.

## **5.1 Adquisición de la imagen**

La adquisición de las imágenes se realiza a través de cámaras digitales profesionales, cámaras para videovigilancia o las cámaras integradas en los celulares, obteniendo como resultado fotografías en formato RGB, comúnmente conocidas como imágenes a color.

### **5.1.1 Modelo de color RGB**

Una imagen RGB está compuesta por la unión de tres planos de colores, cada plano representa la intensidad de los colores: rojo, verde y azul. Una imagen a color está representada por una cantidad de bits, que varían entre 8 y 24 o más bits. En una imagen de 24 bits, los bits se dividen en tres grupos: 8 para el rojo, 8 para el verde y 8 para el azul. Para representar los demás colores se usan combinaciones de esos bits. Una imagen de 24 bits ofrece 16.7 millones de colores.[24]

### **5.1.2 Modelo de color HSV**

El modelo de color HSV (matiz-saturación-valor) es una transformación no lineal del modelo RGB en coordenadas cilíndricas. Matiz se refiere al ángulo que representa el tipo de color, normalmente varía de 0 a 360 grados. La saturación se refiere a la profundidad de color, que se mide en porcentaje, que varía de 0 a 100%, y significa la cantidad de luz blanca mezclada con el matiz. El valor es el brillo del color, y también se indica en porcentaje, que varía de 0 a 100%. [24]

## 5.2 Procesamiento de la imagen

### 5.2.1 Filtrado de la imagen

La presencia de ruido es inevitable, a veces el ruido es inherente al proceso de adquisición de imágenes y puede afectar el proceso de clasificación. Por lo tanto, es necesario deshacerse del mayor ruido posible, reduciendo la amplitud de las variaciones de la imagen mientras se pierde un mínimo de información. Para realizarlo, se reemplaza cada pixel por la medida del valor de los pixeles de alrededor, de esta forma las variaciones rápidas de intensidad pueden ser suavizadas y reemplazadas por una transición más gradual. Una función que realiza tal suavizado es el filtrado gaussiano, pues se realiza multiplicando cada punto de la matriz de entrada con un kernel gaussiano y luego sumándolos a todos para producir la matriz de salida.[25]

$$G(x, y) = \frac{1}{2 * \pi * \sigma^2} * e^{\frac{-x^2+y^2}{2*\sigma^2}} \quad (1)$$

Donde  $\mu$  es la media y  $\sigma$  es la desviación estándar para cada una de las variables X, Y.

### 5.2.2 Escala de grises

Las imágenes en escala de grises consisten solo en tonos grises. Además, solo hay 256 colores grises, ya que estas imágenes están representadas en una profundidad entre 2 a 8 bits. lo que significa que al cero se le asigna el color blanco y a los valores mayores conforme vayan aumentando se les asigna diferentes tonos de gris y por último al 255 se le asignará el color negro. Se utilizan varios métodos diferentes en la visión por computadora para realizar la conversión de color a escala de grises. Las más comunes son las técnicas basadas en promedios ponderados de los canales rojo, verde y azul. [26]. Los valores utilizados por OpenCV son:

$$Gris = 0.299 \cdot Rojo + 0.587 \cdot Verde + 0.114 \cdot Azul \quad (2)$$

### 5.2.3 Binarización

La binarización es un proceso que transforma una imagen que está en escala de grises a una imagen en dos colores: blanco y negro. También podría entenderse como un proceso de segmentación de dos grupos de valores: cero como el fondo (información no relevante) y uno como primer plano (información relevante). De esta forma, al convertir las imágenes en blanco y negro se pierde muy poca información relevante y se simplifica la representación de cada pixel, pasando de 256 posibles valores a tan solo 2 [27]. Para binarizar la imagen se selecciona un valor umbral  $T(x, y)$  para cada pixel  $(x, y)$ , y la función transforma la imagen de escala de grises a una imagen binaria dependiendo de la palabra clave y de acuerdo a las fórmulas:

- **THRESH\_BINARY:**

$$dst(x, y) = \begin{cases} 0 & \text{si } I(x, y) < T(x, y) \\ 255 & \text{en otro caso} \end{cases} \quad (3)$$

- **THRESH\_BINARY\_INV:**

$$dst(x, y) = \begin{cases} 255 & \text{si } I(x, y) < T(x, y) \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

Donde  $I(x, y)$  es la imagen original,  $dst(x, y)$  es la imagen binarizada y  $T(x, y)$  es el umbral calculado para cada pixel. [28]

En los métodos de binarización mencionados anteriormente, se utiliza un valor global como umbral. Pero esto no puede ser bueno en todos los casos, ya que en las imágenes algunas zonas permanecen bajo más sombra y tienen condiciones de iluminación diferentes. En este caso, se

utiliza la binarización adaptativa, ya que calcula el umbral para pequeñas regiones de la imagen. Por lo tanto, se obtienen diferentes valores de umbrales para distintas regiones de la misma imagen y se obtiene mejores resultados en comparación a la técnica de umbral global para imágenes con iluminación variante. [29]

La técnica de binarización adaptativa decide como calcular el valor umbral dependiendo del método que se le asigne:

- **ADAPTIVE\_THRESH\_MEAN\_C**: el valor umbral es equivalente al valor del área vecina menos la constante  $c$ .
- **ADAPTIVE\_THRESH\_GAUSSIAN\_C**: El valor umbral es una suma gaussiana-ponderada de los valores de la vecindad menos la constante  $C$ .

La constante  $C$  es una resta de la suma media de los pixeles de la vecindad, además se debe asignar un `BlockSize` o kernel que determina el tamaño del área vecina. [30]

#### 5.2.4 *Detección de contornos*

La detección de contorno es un proceso en el análisis digital de imágenes que detecta los cambios en el nivel de intensidad de los pixeles contiguos y que ocurre en una ubicación específica. Estos cambios se pueden usar para determinar la profundidad, tamaño, orientación y propiedades de la superficie dentro de una muestra. Por lo tanto, un contorno es un conjunto de puntos que conectados unos con otros forman una curva, la cual une todos los puntos continuos en una imagen, que presentan el mismo color o intensidad para formar una figura que rodea a un objeto determinado. Cuanto mayor es el cambio de nivel de intensidad, más fácil resulta detectar el contorno. La detección de contornos es un paso importante en una amplia gama de métodos de procesamiento de imágenes, desde diversas tareas de segmentación hasta el reconocimiento de formas de objetos. A menudo, la detección de contorno es el primer paso en los métodos de reconocimiento de forma de objetos. [31]

### 5.2.5 *Transformación morfológica:*

Las transformaciones morfológicas son técnicas que se encargan de modificar la imagen, con el objetivo de lograr destacar un objeto de interés sobre el fondo. Para lo anterior, utilizan un kernel el cual se desliza sobre una imagen binaria. [32]

- Erosión

En el proceso de erosión el kernel decide si un pixel es uno, si todos los pixeles que se encuentran dentro del kernel son igual a uno. Si lo anterior no se cumple el pixel se hace cero y, por lo tanto, el grosor de los objetos disminuye.

- Dilatación

El proceso de dilatación es opuesto al de erosión. El kernel decide si un pixel es uno, si al menos un pixel que se encuentra dentro del kernel es igual a uno. Como consecuencia, el tamaño de los objetos aumenta.

- Cierre

El proceso de cierre, es el uso consecutivo de las dos técnicas mencionadas anteriormente. Es decir, para conseguir el operador morfológico cierre se aplica sobre la imagen el proceso de dilatación seguida de erosión.

- Black hat

El operador morfológico black hat se genera por la diferencia entre el operador morfológico cierre y la imagen de entrada.

### 5.3 Técnicas de machine learning

Machine learning tiene como objetivo principal lograr que un sistema o maquina sea capaz de aprender y analizar información previamente suministrada con el objetivo de predecir o generalizar ejemplos futuros. Para lograr este aprendizaje ML se basa en las siguientes características principales: extracción o descripción de características y clasificadores, las cuales se utilizan en la etapa de entrenamiento, para lograr crear el modelo que servirá para generar predicciones futuras.

#### 5.3.1 *Descriptores de características.*

Un descriptor de características es un algoritmo el cual genera una representación de una imagen o un parche de imagen que simplifica dicha imagen extrayendo información útil y desechando información extraña. Los descriptores de características codifican información interesante y genera *ubicaciones* (es decir, coordenadas de píxeles) de áreas significativas de la imagen, las cuales actúan como una especie de "huella digital" numérica que se puede utilizar para diferenciar una imagen de otra.

Normalmente, un descriptor de características convierte una imagen la cual posee un ancho, alto y está dotada de tres canales; en un vector de características de longitud  $n$ .

##### 5.3.1.1 **Histograma de gradientes orientados (HOG).**

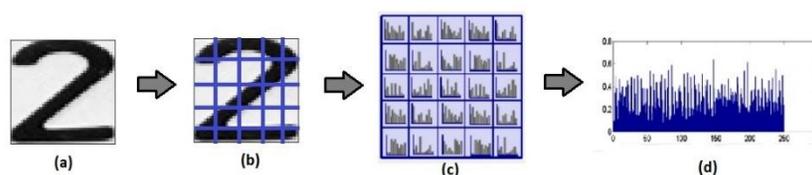
El método de HOG fue propuesto en 2005 por Dalal y Triggs [33] y demostró que el método en combinación con Support Vector Machine podía utilizarse para entrenar clasificadores altamente precisos. El uso de HOG como descriptor fue introducido para el reconocimiento humano, aunque está tomando gran auge para el reconocimiento de texto y palabras escritas a mano.

La idea de HOG es que la forma de los objetos en una imagen puede caracterizarse por la distribución de gradientes de intensidad locales que representan las direcciones de píxeles dominantes [34].

La figura 7 muestra la formación del vector HOG para una imagen de escala de grises.

**Figura 7**

*Método HOG.*



Nota. Fuente Autor

La imagen de un carácter en la figura (7a) es la imagen original en escala de grises. Primero se calculan las orientaciones de gradiente en cada pixel. El segundo paso es dividir la imagen en varios bloques como se muestra en la figura (7b) para calcular un histograma de cada orientación en la pequeña región rectangular como se observa en (7c), Finalmente, los histogramas de las celdas individuales se normalizan y concatenan para formar el vector HOG. El resultado del algoritmo HOG es una cantidad discreta de características que describen la imagen de entrada. El número de características depende del número de celdas y orientación como también el tamaño de la imagen.

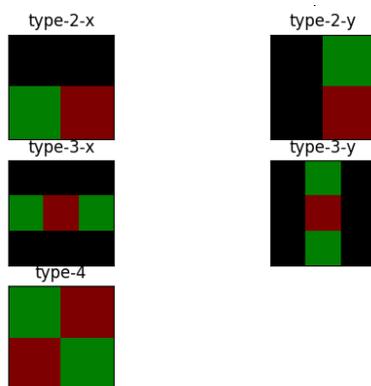
### 5.3.1.2 Haar-like features

Paul Viola y Michael Jones propusieron un método para la detección de objetos utilizando clasificadores en cascada basados en características Haar. [35]

Las características de HAAR se obtienen como resultado de aplicar por toda la imagen un conjunto de filtros Haar.

### Figura 8

*Filtros Haar.*



*Nota.* Fuente [47]

En la figura 8 presenta lo que se conoce como filtros Haar básicos, tal como se puede observar un filtro Haar es la combinación de entre dos y cuatro rectángulos, que son adyacentes en la dirección horizontal o vertical. Estos filtros capturan los cambios de intensidad en direcciones horizontales, verticales y diagonales.

Básicamente un filtro HAAR se puede definir como la diferencia de intensidad de zonas adyacentes de la imagen con diferentes configuraciones que permiten detectar cambios de intensidad en diferentes orientaciones, lo anterior se calculan restando de los valores de intensidad en el rectángulo verde y el rojo. El área roja se resta a la suma de las intensidades de píxeles del verde. Los filtros se aplican a múltiples escalas y posiciones en la imagen, de manera que se acaba obteniendo un número muy elevado de características. [36]

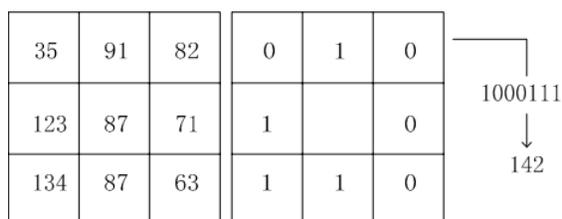
### 5.3.1.3 Local binary pattern (LBP)

LBP fue originalmente propuesto por Timo Ojalá [37] con el objetivo de clasificar texturas, y luego se extendió a varios campos, incluidos el reconocimiento facial y expresiones faciales.

LBP es un operador de textura simple pero muy eficiente que etiqueta los píxeles de una imagen al limitar la vecindad de cada píxel y considera el resultado como un número binario. El operador LBP original reemplaza el valor de los píxeles de una imagen con números decimales, que se llaman LBP o códigos LBP que codifican la estructura local alrededor de cada píxel. Cada píxel central se compara con sus ocho vecinos; los vecinos que tengan un valor menor que el del píxel central tendrán el bit 0, y los otros vecinos que tengan un valor igual o mayor que el del píxel central tendrán el bit 1, tal como se puede observar en la figura 9.

**Figura 9**

*Método LBP*



*Nota.* Fuente [38]

Se necesita calcular el valor de LBP para el píxel central. Se puede comenzar desde cualquier píxel vecino y trabajar en sentido horario o antihorario, pero el recorrido debe mantenerse constante para todos los píxeles de la imagen y todas las imágenes del conjunto de datos. El último paso es calcular un histograma sobre la matriz LBP de salida. Dado que un vecindario  $3 \times 3$  tiene  $2^8 = 256$  posibles patrones, nuestra matriz 2D LBP tiene un valor mínimo de 0 y un valor máximo de 255, lo que nos permite construir un histograma de 256 códigos LBP como nuestro vector de característica final. [38]

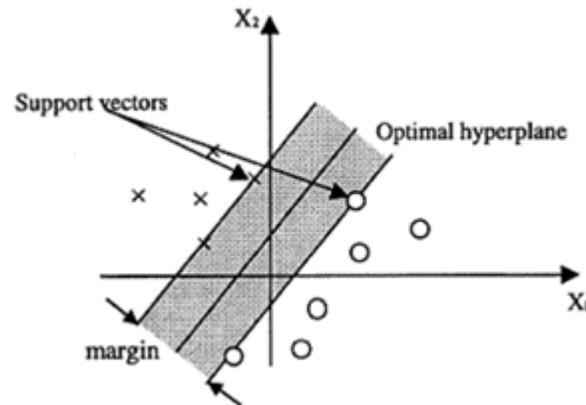
### **5.3.2 Algoritmos de Clasificación**

La clasificación es una de las técnicas importantes en la inteligencia artificial, pues es utilizada para clasificar los datos en clases específicas. En esta técnica, se utiliza un conjunto de datos de entrenamiento en el que todos los criterios o atributos ya están asociados con etiquetas de clase conocidas.

Con los descriptores de características mencionados en el apartado anterior, se describe el proceso de como convertir una imagen en un vector de características. Ahora, los algoritmos de aprendizaje automático o clasificadores son los encargados de tomar como entrada ese vector de características para aprender e intentar predecir con éxito una salida con una etiqueta de clase.

#### **5.3.2.1 Support Vector Machine**

Support vector machine es una técnica eficiente para resolver problemas de reconocimiento de patrones debido a su alto rendimiento. En general, Support Vector Machine se considera una técnica de clasificación, sin embargo, puede ser usada en problema de clasificación y regresión. SVM consiste en proyectar los vectores de entrada en un espacio de características de alta dimensión y luego buscar un hiperplano que divida y maximice la distancia mínima entre los datos de cada clase. Esta técnica fue diseñada principalmente para problemas de clasificación de dos clases. Para las clasificaciones multiclase se han propuesto varios métodos, los más populares son: one-vs-one y one-vs-all. La estrategia básica para aplicar SVM a la clasificación multiclase es descomponer el problema en varios problemas de clase binaria.

**Figura 10***Hiperplano de separación SVM**Nota.* Fuente [39]

La figura 10 muestra como el algoritmo computacional construye un hiperplano. Las dos clases se indican mediante puntos marcados con X y los círculos blancos. Los vectores de soporte son los puntos de datos, que están más cerca al hiperplano. Estos puntos definirán mejor la línea de separación calculando los márgenes. El margen es un espacio entre las dos líneas en los puntos de clase más cercanos. Esto se calcula como la distancia perpendicular desde la línea hasta vectores de soporte o puntos más cercanos. Si el margen es mayor entre las clases, entonces se considera un buen margen, un margen más pequeño es un margen malo. [39]

### 5.3.2.2 K-Nearest Neighbor

K-Nearest Neighbours es uno de los algoritmos de clasificación más antiguos y simples pero esencial en machine learning. Además, encuentra una aplicación intensa en reconocimiento de patrones y escenas de imágenes satelitales.

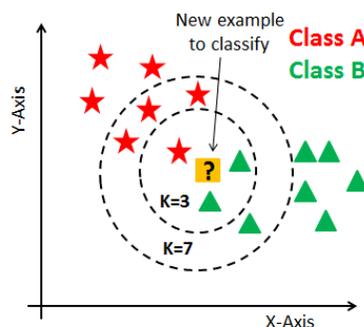
El algoritmo K-Nearest Neighbour se basa en un proceso de votación mayoritaria y clasifica cada ejemplo sin etiqueta, por la etiqueta mayoritaria entre sus K vecinos más cercanos en el conjunto de entrenamiento. Los k vecinos más cercanos están determinados por la distancia mínima entre el punto de consulta y las muestras de entrenamiento, tal distancia es calculada bajo una de las siguientes formulas: distancia euclidiana, distancia manhattan y la distancia de Minkowsky.[42][43]

- Distancia euclidiana:

$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (5)$$

Donde  $x_i$  es la muestra de entrenamiento,  $y_i$  es ejemplo de consulta.

En la figura 11 se supondrá que el cuadrado amarillo es un objeto para el cual el algoritmo necesita predecir la etiqueta. Primero, encuentra los K vecinos más cercanos al objeto y luego clasifica los puntos por mayoría de votos de sus k vecinos. Cada objeto vota por su clase y la clase con más votos se toma como la predicción. Con esto en mente, el objetivo de KNN Simplemente se basa en la idea de que los objetos que están cerca entre sí también tendrán características similares.

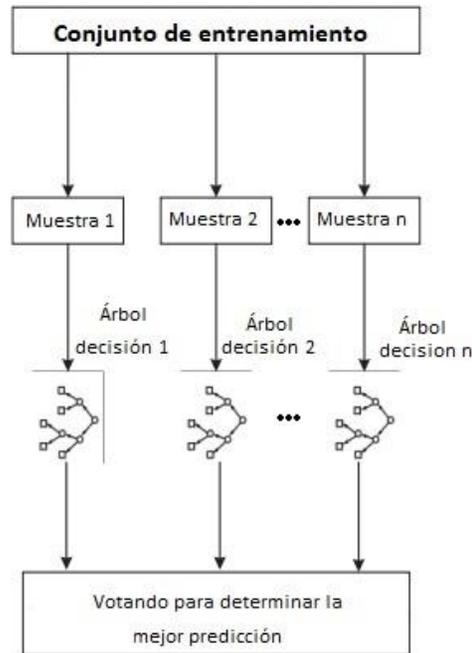
**Figura 11***Algoritmo KNN**Nota. Fuente [44]*

### 5.3.2.3 Random Forest

Random forest es un algoritmo basado en aprendizaje conjunto. El aprendizaje conjunto es un tipo de aprendizaje en el que se unen diferentes tipos de algoritmos o el mismo algoritmo varias veces para formar un modelo de predicción más potente. El algoritmo random forest combina múltiples algoritmos del mismo tipo, es decir, múltiples árboles de decisión, lo que da como resultado un bosque de árboles, de ahí el nombre random forest.[45]

Random forest posee dos etapas principales: primero crea un árbol de decisión por cada muestra escogida aleatoriamente del conjunto de entrenamiento. Segundo, cada árbol de decisión vota y se elige la clase más popular como la predicción final.

La figura 12 muestra el funcionamiento del algoritmo. En primer lugar, se selecciona muestras aleatorias del conjunto de datos. En segundo lugar, se construye un árbol de decisión para cada muestra y se obtiene una predicción de cada árbol. En tercer lugar, se realiza una votación de cada predicción y finalmente se selecciona la predicción con más votos como la predicción final.[46]

**Figura 12***Algoritmo Random Forest**Nota.* Fuente Autor

## 5.4 Técnicas de Deep learning

Deep Learning es un conjunto de técnicas y procedimientos algorítmicos basados en Machine Learning para lograr que una máquina aprenda de la misma forma que lo hace el ser humano. Tal aprendizaje se fundamenta en el comportamiento que lleva a cabo nuestro cerebro con la ayuda de las neuronas con el fin de reconocer imágenes, palabras y sonidos.

Deep Learning se divide en diferentes ramas como redes neuronales, redes neuronales convolucionales y redes neuronales recurrentes. Estas arquitecturas son utilizadas en campos como la visión por computador, reconocimiento del habla o reconocimiento de dígitos y caracteres.

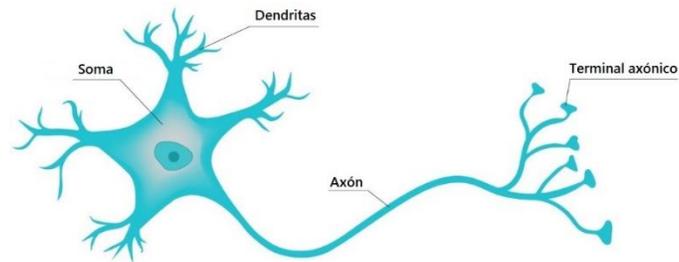
### **5.4.1 *Neurona modelo biológico***

El cerebro humano está compuesto por aproximadamente 10 mil millones de neuronas, cada una conectada a unas 10,000 otras neuronas. El cuerpo celular de la neurona se llama soma, donde las entradas (dendritas) y las salidas (axones) conectan el soma con otro (figura 13).

Cada neurona recibe entradas electroquímicas de otras neuronas en sus dendritas. Si estas entradas eléctricas son lo suficientemente potentes como para activar la neurona, entonces la neurona activada transmite la señal a lo largo de su axón, transmitiéndola a las dendritas de otras neuronas. Estas neuronas unidas también pueden activarse, continuando así el proceso de transmitir el mensaje.

La conclusión clave aquí es que el disparo de una neurona es una operación binaria: la neurona se dispara o no se dispara. No hay diferentes "grados" de disparo. En pocas palabras, una neurona solo se disparará si la señal total recibida en el soma excede un umbral dado.

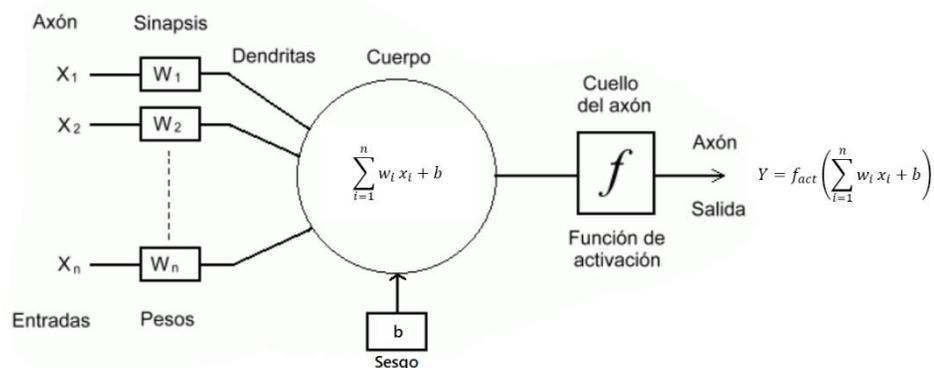
Sin embargo, se debe tener en cuenta que las redes neuronales artificiales simplemente se inspiran en lo que se sabe sobre el cerebro y cómo funciona. El objetivo del aprendizaje profundo no es imitar cómo funcionan el cerebro, sino tomar las piezas que se entienden y permitir trazar paralelos similares como una herramienta para resolver problemas de clasificación supervisada en la visión artificial.[48]

**Figura 13***Estructura básica de una neurona biológica*

*Nota.* Fuente Autor

**5.4.2 Neurona modelo artificial**

Las neuronas artificiales son los elementos que se encargan de llevar a cabo el cálculo de funciones matemáticas dentro de una red. El modelo matemático que generaliza una neurona biológica es el perceptrón de capa única propuesto por Rosenblatt, el cual es una función lineal que combina unos pesos dados con los datos de entrada, luego aplica una función de activación para obtener una salida. La figura 14 muestra el modelo de neurona artificial o perceptrón.

**Figura 14***Modelo de neurona artificial*

*Nota.* Fuente Autor

A continuación, se describen los componentes que presentan las redes neuronales artificiales [48]:

- Variables de entrada, debe contener al menos una. Generalmente se representa en forma de vector.

$$x_j = [x_1, x_2, x_3, \dots, x_n]$$

- Pesos sinápticos por cada variable de entrada que determinará la importancia de cada conexión. Es decir, representan la intensidad de interacción entre cada neurona presináptica  $j$  y la neurona postsináptica  $i$ . Se presenta en forma vectorial.

$$w_{ij} = [w_{i1}, w_{i2}, w_{i3}, \dots, w_{ij}]$$

- Se adiciona un parámetro llamado sesgo,  $b$ . El sesgo indicará la facilidad que tiene la neurona de activarse. Este proceso está basado en las neuronas biológicas,

las cuales envían un impulso eléctrico solo si a su entrada existe un potencial eléctrico mayor a un potencial umbral. El sesgo sería la inversa a ese potencial umbral.

- Una función de propagación que actúa sobre las entradas y los pesos antes de acceder la función de activación. La función de propagación comúnmente utilizada consiste en combinar linealmente las entradas y los pesos, obteniéndose:

$$h_i(x_1, x_2, x_3, \dots, x_n, w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}) = \sum_{j=1}^n w_{ij}x_j + b$$

- Una función de activación, la cual representa simultáneamente la salida de la neurona y se estado de activación.

$$y_i = f_{act}(h_i) = f_{act}\left(\sum_{j=1}^n w_{ij}x_j + b\right)$$

Entre las funciones de activación más comúnmente utilizadas se encuentra la tangente hiperbólica, la cual ofrece una salida acotada entre -1 y 1, la función sigmoide, que es una versión escalada de la tangente hiperbólica y ofrece una salida acotada entre 0 y 1, por último, la unidad lineal rectificadora (ReLU), la cual lleva a cabo la siguiente operación:  $f(y) = \max(0, y)$

- Finalmente, la salida de una neurona cuya entrada es un vector de variables  $x = [x_1, \dots, x_n]$  tendrá la siguiente forma:

$$y = f_{act}\left(\sum_{j=1}^n w_{ij}x_j + b\right) \quad (6)$$

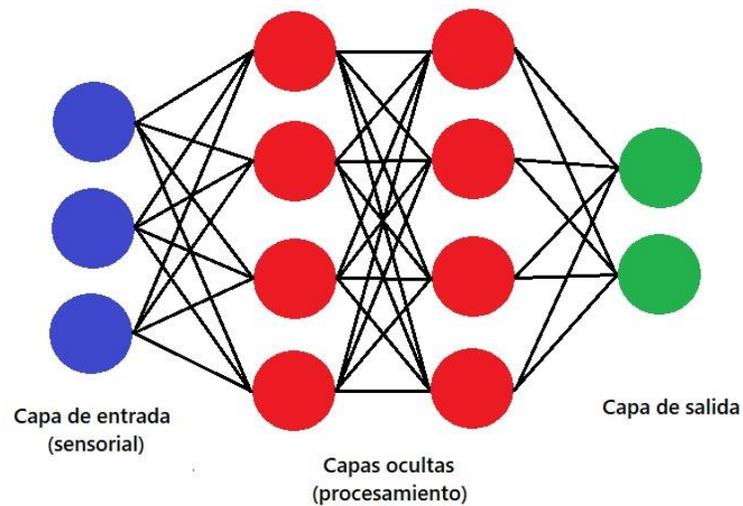
### 5.4.3 *Redes neuronales Feed-forward*

En este tipo de arquitecturas las neuronas generalmente están agrupadas en unidades estructurales denominadas capas. Se distinguen tres tipos de capas: capa de entrada, capas ocultas, y capa de salida. Una capa de entrada, también denominada sensorial, que en realidad no está compuesta por neuronas, sino que es una herramienta visual que recibe datos o señales procedentes del entorno. La capa oculta es la primera que consta de neuronas que se encargan de realizar cálculos gracias a los cuales la red neuronal es capaz de representar más fehacientemente determinadas características del entorno que se trata de modelar. Finalmente, una capa de salida se compone de neuronas que proporcionan la respuesta de la red neuronal. La capa de salida generalmente tiene tantas neuronas como etiquetas de clase; una neurona para cada salida potencial. Por ejemplo, si una red neuronal está construida para clasificar los dígitos, la capa de salida consistiría de 10 neuronas, una para cada dígito del 0-9. Este tipo de arquitecturas se denominan feedforward porque la información fluye desde la capa de entrada hacia la capa de salida, por lo tanto, solo permite conexiones desde las neuronas de la capa  $i$  hacia las neuronas de la capa  $i + 1$ .

Las redes neuronales feedforward son la piedra angular del Deep Learning moderno aplicado a la visión por computadora. Como se verá más adelante, las redes neuronales convolucionales son simplemente un caso especial de red neuronal feedforward.

**Figura 15**

*Ejemplo de red neuronal feed-forward, la capa de entrada consta de tres entradas, dos capas ocultas cada una con cuatro neuronas y una capa de salida con dos neuronas.*



*Nota.* Fuente Autor

Para obtener la salida de la red feedforward es necesario conocer los pesos, los sesgos y las funciones de activación de la red. Como consecuencia de que la entrada a una capa  $l$  es la salida de una capa  $l - 1$ , la salida de la capa  $l$  depende exclusivamente de la salida de la capa  $l - 1$ . Por lo tanto, la salida de una neurona  $i$  de la capa  $l$  se define por la siguiente expresión:

$$z_i^l = f_{act} \left( \sum_{j=1}^J z_j^{l-1} w_{ij}^l + b_i^l \right) \quad (7)$$

Donde  $z_j^l$  es la salida de una neurona  $j$  de la capa  $l - 1$  en la que hay un total de  $J$  neuronas,  $w_{ij}^l$  es el peso que conecta la neurona  $i$  de la capa  $l$  con la neurona  $j$  de la capa  $l - 1$  y  $b_i^l$  es el sesgo de la neurona  $i$  de la capa  $l$ .

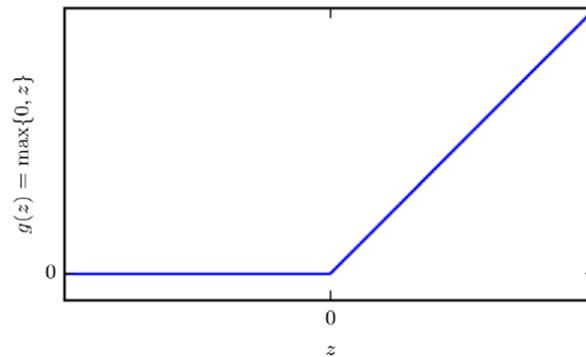
#### 5.4.4 Función de activación

Las funciones de activación son ecuaciones matemáticas que determinan la salida de una red neuronal. La función está unida a cada neurona en la red y determina si debe activarse o no, en función de si la entrada de cada neurona es relevante para la predicción del modelo. Las funciones de activación también ayudan a normalizar la salida de cada neurona en un rango entre 1 y 0 o entre -1 y 1. Existen diferentes tipos de función de activación [50], las más relevantes para este trabajo de grado son las siguiente:

##### 5.4.4.1 Unidad lineal rectificada (ReLU)

La opción predeterminada para una función de activación en redes neuronales modernas es ReLU. se ha vuelto muy popular en los últimos años, porque se descubrió que aumenta en gran medida la convergencia de las redes neuronales ya que acelera la dirección del gradiente. La función ReLU transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran. Normalmente es aplicada en las capas ocultas con la finalidad de introducir la no linealidad en un sistema que ha calculado operaciones lineales durante sus anteriores capas. Debido a sus propiedades, se destaca su buen comportamiento con imágenes y obtiene gran desempeño en redes convolucionales.

$$f(z) = \max(0, z) = \begin{cases} 0 & \text{para } x \leq 0 \\ z & \text{para } x \geq 0 \end{cases} \quad (8)$$

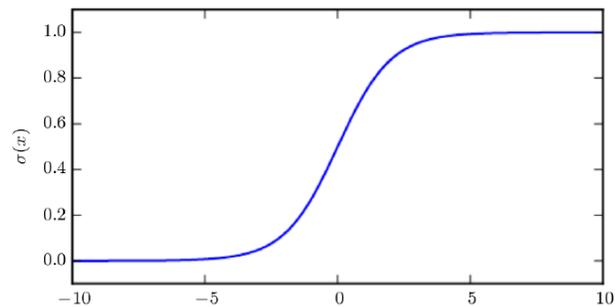
**Figura 16***Función ReLU*

*Nota.* Fuente [50]

#### 5.4.4.2 Función Sigmoid

La función sigmoide se usa para representar una distribución de probabilidad sobre una variable binaria. La función transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0. Sigmoid se usa en la capa de salida, para redes neuronales que necesitan una clasificación binaria.

$$\sigma(z) = \frac{1}{1 + e^{(-z)}} \quad (9)$$

**Figura 17***Función sigmoid*

Nota: Fuente [50]

#### 5.4.4.3 Función softmax

La función softmax es una generalización de la función sigmoid, ya que softmax acota las salidas de cada unidad para que estén entre 0 y 1, al igual que una función sigmoid. Pero también divide cada salida de tal manera que la suma total de las salidas es igual a 1. La salida de la función softmax es equivalente a una distribución de probabilidad categórica, indica la probabilidad de que alguna de las clases sea verdadera. Se usa en la capa de salida de redes neuronales que necesitan una clasificación multiclase.

$$f(z_j) = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad (10)$$

### 5.4.5 Entrenamiento de redes neuronales

El entrenamiento de redes neuronales puede entenderse como un problema de optimización, en donde los diferentes parámetros presentes en la red neuronal deben ser ajustados para aproximar con mayor precisión el entorno que se desea modelar. Por consiguiente, es necesario definir una función, la cual se desea optimizar a través de la variación de los parámetros de la red como lo son: los pesos y sesgos, aplicando diferentes algoritmos que ofrecen una determinada estrategia. El proceso de aprendizaje es un proceso iterativo, en el cual se va refinando la solución hasta alcanzar un nivel de operación suficientemente bueno.

#### 5.4.5.1 Función de coste

La función de coste, también conocida como función objetivo o función de pérdidas, es una medida de que tan bien la salida de un sistema de aprendizaje automático concuerda con la clase deseada de los datos de entrenamiento. El objetivo de los algoritmos de entrenamiento es encontrar el conjunto de pesos y sesgos que minimicen la función de error.

La función de coste más comúnmente usada en problemas de clasificación multiclase es cross entropy [51], se rige por la siguiente fórmula:

$$E(W) = - \sum_{c=1}^M y_{o,c} \cdot \log(p_{o,c}) \quad (11)$$

Donde  $M$  es el número de clases,  $y_{o,c}$  es una variable binaria que indica 1 si y solo si la muestra  $o$  pertenece a la clase  $c$ ,  $p_{o,c}$  es la probabilidad de salida que indica que la muestra  $o$  pertenece a la clase  $c$ .

### 5.4.5.2 Descenso del gradiente

El objetivo del descenso del gradiente será encontrar la configuración de pesos y sesgos que corresponda al mínimo global de la función de error.

Dado un conjunto de peso  $W(0)$  para el instante de tiempo  $t = 0$ , se calcula la dirección de máxima variación del error. La dirección de máximo crecimiento de la función  $E(W)$  en  $W(0)$  viene dada por el gradiente  $\nabla E(W)$ . Luego, se actualizan los pesos siguiendo el sentido contrario al indicado por el gradiente  $\nabla E(W)$ , dirección que indica el sentido de máximo decrecimiento, De este modo, se va produciendo un descenso por la superficie de error hasta alcanzar un mínimo local [52].

$$w_{i_{t+1}} = w_{i_t} - \eta \nabla E(W) \quad (12)$$

La variable  $\eta$  es una constante positiva llamada velocidad de aprendizaje o learning rate. La ecuación (12) se puede repetir iterativamente hasta alcanzar un criterio de parada o esta alcanzar un máximo de iteraciones. Cada iteración del algoritmo hasta finalizar se conoce como época de aprendizaje. Las expresiones correspondientes a la actualización de los pesos y sesgos de la red vienen dadas por las siguientes formulas:

$$w_{i_{t+1}} = w_{i_t} - \eta \frac{\partial E}{\partial w_{i_t}} \quad (13)$$

$$b_{i_{t+1}} = b_{i_t} - \eta \frac{\partial E}{\partial b_{i_t}} \quad (14)$$

El optimizador Descenso de Gradiente mencionado no es el único utilizado. De hecho, existe todo un conjunto de optimizadores que, basados en Descenso de Gradiente, intentan mejorar el rendimiento de éste. Entre los cuales se tiene [53] [54]:

- Descenso de gradiente estocástico (SGD)
- RMSprop
- Adam
- Adadelata
- Adragram
- Adamax
- Nadam
- Ftrl

### **5.4.5.3 Backpropagation**

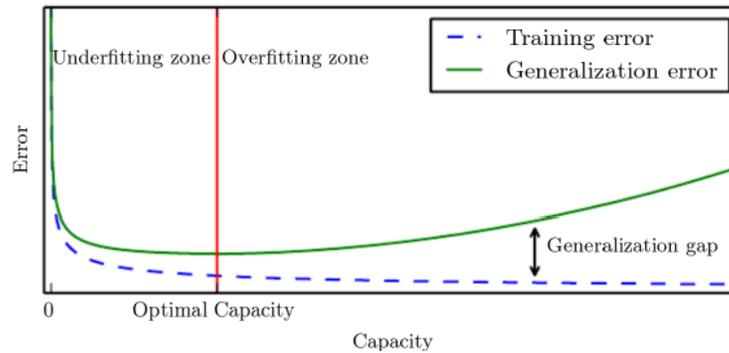
La idea principal de la backpropagation es calcular el gradiente para una red que consta de más de dos capas ocultas. Si solo hay dos capas, es decir, la entrada y la salida, el cálculo del error es simplemente la diferencia entre el valor de salida y el valor objetivo y, dependiendo del error, los pesos y sesgos se pueden cambiar. Sin embargo, si una red tiene más de dos capas, el cálculo del gradiente de error no es para nada trivial para las capas ocultas, debido a la gran cantidad de parámetros, pero se puede hacer utilizando el algoritmo de backpropagation.

Dicho algoritmo consiste en comenzar calculando las derivadas parciales de la función de coste a la salida de la red con respecto únicamente a los parámetros de la última capa. Una vez obtenidas estas derivadas, se pasa a la capa anterior, y se calcula nuevamente las derivadas parciales de la función de coste, pero ahora con respecto a los parámetros de esta capa. Y sucesivamente progresando hacia atrás, hasta llegar al inicio de la red.

Con lo anterior, el objetivo de backpropagation es calcular las derivadas parciales  $\frac{\partial E}{\partial w_{it}}$  y  $\frac{\partial E}{\partial b_{it}}$  que aparecen en las ecuaciones 13 y 14. Para ello, se asume que las capas ocultas de la red contienen en su función de activación, la función unidad lineal rectificadora (ReLU), en la capa de salida está presente la función Softmax y la función de coste para medir la pérdida de entrenamiento es Cross-entropy.

#### 5.4.5.4 Overfitting

un modelo de aprendizaje automático exitoso debería funcionar bien ante conjuntos de datos de entrada desconocidos, que son del mismo tipo que los conjuntos de datos de entrenamiento y validación. Por lo tanto, el rendimiento del algoritmo no solo debe evaluarse por su capacidad de minimizar el error de entrenamiento, sino también por su capacidad de minimizar la diferencia entre el error de entrenamiento y el error de validación. Esta diferencia se denomina error de generalización. Durante el entrenamiento de cualquiera de estos algoritmos, y en especial en las redes neuronales, se debe tener cuidado para evitar esta situación. Esta situación es detectable cuando el algoritmo tiene la capacidad de lograr un pequeño error de entrenamiento, pero no puede hacer que el error de generalización sea pequeño, este problema se llama overfitting.

**Figura 18***Overfitting visto de forma gráfica**Nota.* Fuente [50]

#### 5.4.5.5 Data Augmentation

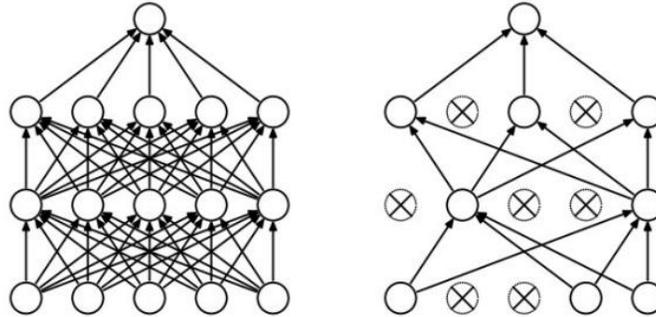
Es un método para aumentar el tamaño del conjunto de entrenamiento para que el modelo no pueda memorizarlo todo. El método crea artificialmente imágenes de entrenamiento a través de diferentes transformaciones, como rotar las imágenes a un ángulo dado, escalar las imágenes, traslación de la imagen a lo largo del eje  $x$  o  $y$ , etc.

#### 5.4.5.6 Dropout

El funcionamiento de dropout es muy simple pero eficaz, consiste en desconectar aleatoriamente neuronas (junto con sus conexiones) de la red neuronal durante el entrenamiento. Esto da como resultado, que su contribución a la activación de las neuronas se anula temporalmente durante la propagación hacia adelante y las actualizaciones de los parámetros no se apliquen a la neurona en la propagación hacia atrás. [55]

**Figura 19**

*Red original (izquierda) y red a la que se le ha aplicado Dropout*



*Nota.* Fuente [55]

#### 5.4.5.7 Batch normalization

Esta capa rápidamente se hizo muy popular principalmente porque ayuda a converger más rápido las redes neuronales [56]. Agrega un paso de normalización, lo que se refiere a reescalar los datos para tener una media de cero y una desviación estándar de uno. La normalización de las entradas a la capa tiene un efecto en el entrenamiento del modelo, reduciendo drásticamente la cantidad de épocas requeridas. También puede tener un efecto de regularización, reduciendo el error de generalización y acelera el proceso de entrenamiento de las redes neuronales profundas.

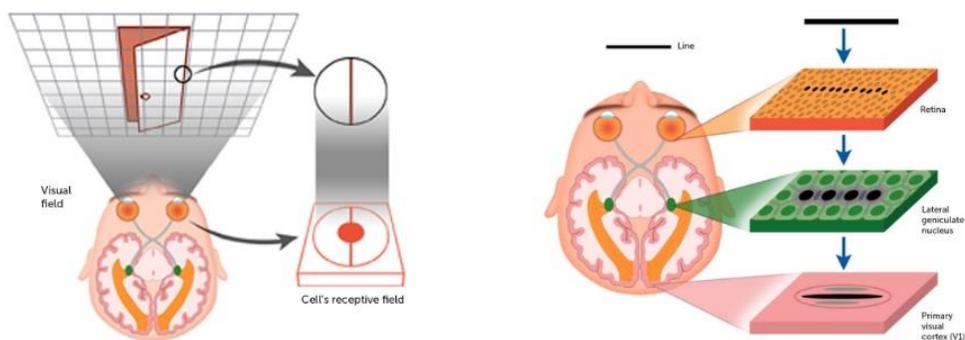
### 5.4.6 Redes neuronales convolucionales

Las Redes neuronales convolucionales (CNN) son redes neuronales inspiradas en la biología cerebral, propuestas por primera vez por LeCun [56] en 1988, basado en los primeros trabajos de Hubel y Wiesel en 1962 [57] sobre la corteza visual del gato, donde demostraron que algunas neuronas del cerebro solamente se activaban ante la presencia de bordes con una orientación específica. Gracias a estos estudios se sabe que la corteza visual tiene células con pequeñas subregiones sensibles, llamadas campos receptivos locales. Las células funcionan como filtros locales sobre el espacio de entrada y responden de manera diferente según el tipo de célula.

En la figura 26 los campos receptivos pueden detectar bordes a medida que la mirada se desplaza sobre una escena, los bordes que caen sobre la retina activan un grupo de fotorreceptores agrupados en línea. Esa actividad se transmite al núcleo geniculado lateral (LGN), en el cual se activa un grupo de células también agrupadas en línea. Finalmente, en la corteza visual, una célula cortical combina la entrada de las células en LGN, que da como resultado un campo receptivo lineal, específico de la orientación.

**Figura 20**

*Diagrama básico del funcionamiento de los campos receptivos cerebrales*



*Nota.* Fuente <https://brainconnection.brainhq.com/2004/03/06/overview-of-receptive-fields/>

El conjunto de patrones de actividad descritos ayuda al cerebro a identificar bordes y la imagen en su totalidad. Por lo tanto, esta idea de tener pequeños componentes especializados en una tarea concreta dentro de un sistema mayor es una de las bases fundamentales de una red neuronal convolucional.

Las CNN a menudo se usan para procesar imágenes 2D a fin de aprender características complejas no lineales de alta dimensión a partir de un gran conjunto de ejemplos, convirtiéndolas en una de las arquitecturas de aprendizaje automático más exitosas en visión artificial y ha logrado resultados de vanguardia en tareas como reconocimiento de caracteres, reconocimiento de objetos, reconocimiento de voz, detección de rostros y como se descubrirá en ese trabajo de grado es excelente también para reconocimiento de matrículas vehiculares.

#### ***5.4.7 Arquitectura básica red neuronal convolucional***

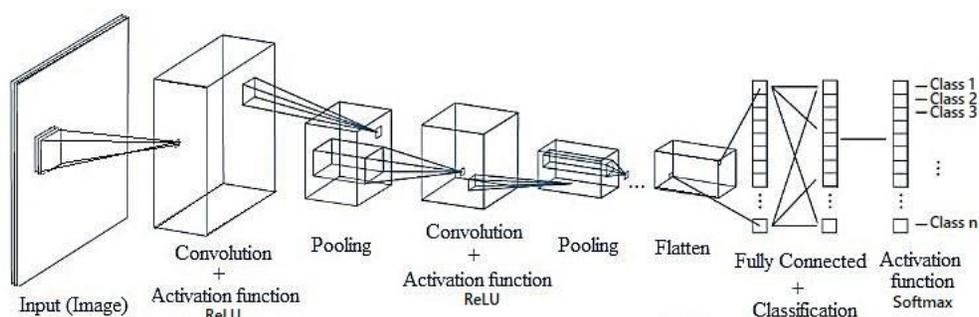
La mayoría de las redes neuronales convolucionales que se emplean para procesar imágenes ejecutan dos operaciones principales, la primera es la *convolución* y la segunda es *pooling* o agrupación. En términos generales, la convolución aplica diferentes filtros a la imagen, y se obtiene como salida múltiples imágenes transformadas y de dimensión reducidas, la cantidad de imágenes transformadas depende del número de filtros que se aplican. Estas imágenes transformadas se conocen comúnmente como mapas de características. Después, se le aplica la función de activación, entre las cuales destaca ReLU. Finalmente, se vuelve a reducir todos los mapas de características extrayendo los valores más significativos con la operación *pooling*. Los procesos de convolución y pooling se apilan juntos, y se puede considerar como un bloque, el cual se puede repetir varias veces en una red convolucional.

Después de aplicar un número de convoluciones y pooling adecuado, la salida de la última capa de pooling, se le aplica un método llamado *Flatten* o aplanamiento, que tiene como objetivo convertir el mapa de características en un vector de características. Finalmente, el vector se conecta

a una red neuronal convencional para realizar la clasificación de la imagen. La figura 21 muestra una estructura básica de una red neuronal convolucional.

**Figura 21**

*Representación visual de una CNN con dos capas de convolución y pooling*



*Nota.* Fuente [57]

### 5.4.7.1 Capa convolucional

La convolución es una operación matemática de dos funciones. La convolución entre las funciones  $I$  y  $K$  normalmente se denota  $I * K$ . La convolución produce una tercera función dada por la integral de la multiplicación puntual de estas dos funciones. Formalmente escrito, se denota como:

$$(I * K)(t) = \int_0^t I(\tau) * K(t - \tau) d\tau \quad (18)$$

Donde  $t$  puede representar el tiempo,  $I$  y  $K$  pueden representar dos señales.

Cuando se habla de convolución en términos de aprendizaje automático, visión por computadora o redes neuronales, hay cuatro conceptos principales a tener en cuenta: una de las dos funciones es la entrada, habitualmente una imagen la cual se denotara como  $I$ , la segunda función será el filtro o kernel, el cual se denotara con la letra  $K$ , la salida será el mapa de características, y por último y mas importante, cuando se trabaja con imágenes no se puede usar la

formula 18 directamente, ya que las imágenes digitales están basada en pixeles y no son continuas. Por lo tanto, se debe usar la forma discreta de convolución para  $I$  y  $K$  dada por:

$$(I * K)[i] = \sum_m I[m] * K(i - m) \quad (19)$$

La anterior formula da la convolución funciona para una sola dimensión. La fórmula extendida para convolución en dos dimensiones de un kernel  $K$  de tamaño  $m \times n$  colocado en el pixel  $(i, j)$  de una imagen  $I$  viene dada por:

$$(I * K)[i, j] = \sum_m \sum_n I[i - m, j - n] * K(m, n) \quad (20)$$

#### 5.4.7.2 Capa de pooling

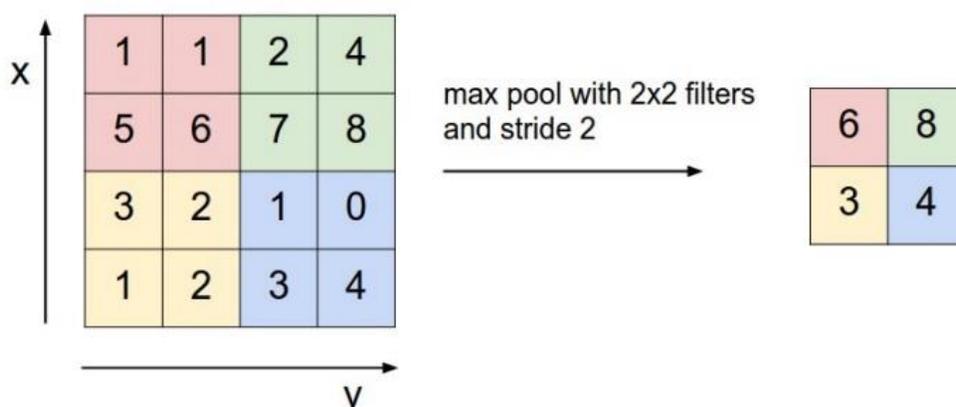
Después de una capa de convolución, es normal usar una capa de pooling sobre el mapa de características resultante. El objetivo de esta capa es la de simplificar los mapas de características generados por las capas previas.

El concepto de pooling implica que cuando una característica ha sido encontrada por una de las capas anteriores, su ubicación exacta no es tan importante. Solo las características relativas a las características del vecindario se vuelven importantes. Usando este razonamiento, las capas de pooling van reduciendo la dimensión espacial y la sensibilidad a los cambios y distorsiones de su entrada. La principal ventaja de esto es que, al reducir la dimensión, también se reduce el número de parámetros que deben aprenderse por lo que disminuimos el coste computacional global. La segunda ventaja es la ya comentada previamente, que las CNN cumpla con la característica de traslación invariante. Es decir, la respuesta será la misma sin importar donde este la característica detectada. [50][62]

La operación de pooling más conocida es la de *max pooling*. Esta operación simplemente devuelve el valor máximo en cada subregión del mapa de características. En la Figura 22, se muestra un ejemplo de Max pooling con un kernel de  $2 \times 2$  y un paso de 2.

**Figura 22**

*Max pooling con un filtro 2x2 y un paso de 2*



*Nota.* Fuente [63]

~~Otra operación de pooling de uso común es *average pooling*. Funciona de la misma manera que *max pooling*, excepto que devuelve el valor promedio de cada subregión en lugar del valor máximo.~~

### 5.4.7.3 Capa flaten

Es necesario convertir los mapas de características en un vector de características 1D. Esta operación se llama flattening. Se selecciona la salida de la última capa de pooling y se transforma en un vector único.

#### **5.4.7.4 Capa fully connected**

Después haber transformado la salida mediante flatten, el razonamiento de alto nivel de la red es manejado por una serie de capas fully connected. Sus capas son idénticas a las descritas en las redes feedforward. Como norma general, la última de sus capas será unidimensional y tendrá tantas neuronas como clases que se estén clasificando.

### **5.5 Arquitecturas convolucionales para la clasificación de imágenes**

Se han desarrollado diferentes arquitecturas convolucionales a partir de la década de 1990. Muchas de las arquitecturas revolucionarias fueron construidas y enviadas al concurso de reconocimiento visual a gran escala ImageNet (ILSVRC) [64]. Esta es una competencia anual donde el objetivo es obtener el error de clasificación más bajo en un conjunto de datos dado que contiene más de diez millones de imágenes y más de 1000 clases. Esta competencia ha sido en los últimos años el factor que impulsó el desarrollo posterior de las CNN. Todas las arquitecturas presentadas resuelven las tareas generales de clasificar objetos dentro de una imagen.

#### **5.5.1 VGGNet**

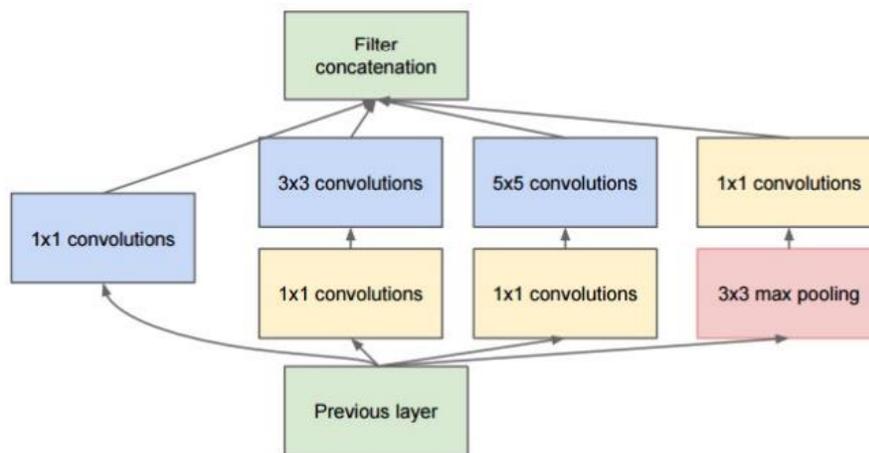
Fue la arquitectura finalista de ILSVRC-2014 [65] con casi 140 millones de parámetros. Sus principales contribuciones fueron mostrar que la profundidad es un componente crítico para un buen rendimiento al usar filtros de 3 x 3 mucho más pequeños en cada capa convolucional. La gran ventaja de VggNet fue la idea de que la convolución múltiple 3 x 3 en secuencia puede emular el efecto de campos receptivos más grandes, por ejemplo, 5 x 5 y 7 x 7. Estas ideas también se utilizarán en arquitecturas de red más recientes como Inception y ResNet. Existen diferentes versiones de VGG, las cuales contienen entre 11, 13, 16 y 19 capas.

### 5.5.2 GoogleNet o Inception

Inception [66], ganó el ILSVRC-2014. Inspirado por VggNet, usó filtros 3 por 3 mucho más pequeños en cada capa convolucional en lugar de filtros 5 por 5 o 7 por 7. Esta idea abre la posibilidad de aumentar el número de capas y construir redes más profundas. Su propia contribución fue reducir drásticamente el número de parámetros utilizados a 40 millones. Esto se logró mediante el uso de filtros baratos de 1 por 1 antes de costosos bloques paralelos, conocidos como cuello de botella. La Figura 23 muestra un módulo en Inception versión 1 usando esta técnica. Para reducir aún más el número de parámetros, se utilizó la average pooling como clasificadores en lugar de capas completamente conectadas. El desarrollo de Inception ha continuado y la versión más reciente es InceptionV4 [67], especialmente inspirada por ResNet. [68]

**Figura 23**

*módulo en Inception V1. Los bloques amarillos de convolución 1x1 reducen el tamaño de entrada antes de alimentarlos a convoluciones 3x3 y 5x5*



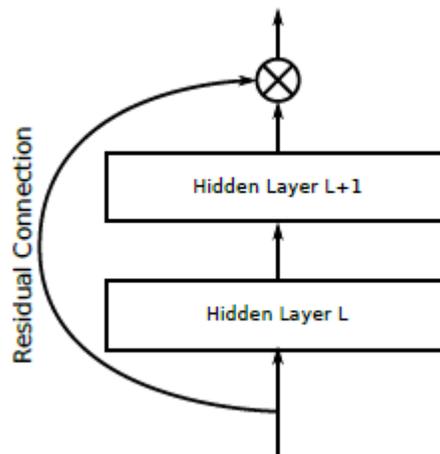
*Nota.* Fuente [66]

### 5.5.3 ResNet

ResNet [68] ganó el ILSVRC 2015 y representa una nueva forma revolucionaria de construir CNN, llamadas CNN residuales. Consiste en 152 capas, un aumento extremo en comparación con arquitecturas anteriores. Esto se logró utilizando conexiones de omisión como se muestra en la Figura 3.4. Una conexión de omisión es una conexión utilizada por la señal de entrada para omitir varias capas. Con esta técnica, se pueden entrenar CNN con más de 1000 capas. Existen diferentes versiones de ResNet con 18, 34, 50 y 152 capas.

**Figura 24**

*Conexión residual, donde se utiliza una conexión de omisión para omitir la entrada a las siguientes capas*



*Nota.* Fuente [68]

## 5.6 Deep learning para la detección de objetos

El acelerado avance en la visión artificial para escenarios en donde no solo es necesario reconocer y clasificar una imagen, ha empezado a utilizar las redes convolucionales para la detección de objetos, las cuales intenta localizar dibujando un cuadro delimitador alrededor del objeto en el interior de una imagen, el cual que se desea detectar. Estas técnicas se encuentran divididas en dos categorías:

- Algoritmos de dos fases: este tipo de algoritmos tienen como objetivo en su primera fase generar un conjunto de posibles rectángulos delimitadores o propuesta de región de interés, esto quiere decir que devuelven regiones que contienen posibles candidatos a ser un objeto que se desea detectar; la segunda fase consiste en utilizar redes neuronales convolucionales para asignar las clases pertinentes a cada objeto que aparece en la imagen. Ejemplos de algoritmos de dos fases son: R-CNN, Fast R-CNN y Faster R-CNN.
- Algoritmos de una fase: Este tipo de algoritmos divide la imagen en una cuadrícula, posteriormente cada celda de la cuadrícula pasan a una red convolucional, si el centro de un objeto cae en una celda de cuadrícula, esa celda de cuadrícula es responsable de detectar ese objeto. Ejemplos de algoritmos de dos fases: SSD y YOLO.

Los algoritmos de dos fases y debido a estas, son más precisos, pero más lentos, y generan problemas para procesar imágenes en tiempo real. Los algoritmos de una fase son más rápidos, no obstante, menos precisos debido a su única fase.

Actualmente el algoritmo YOLO, ha alcanzado una precisión comparable con los algoritmos de dos fases. Razón por la cual, se ha seleccionado este algoritmo para la fase experimental de este trabajo.

### Figura 25

*Tabla comparativa de detección sobre el dataset*

Detection Framework	mAP	FPS
Faster RCNN - VGG16	73,2	7
Faster RCNN - ResNet	76,4	5
YOLO v1	63,4	45
SSD 500	76,8	19
YOLO (416×416 image size)	76,8	67
YOLO (480×480 image size)	77,8	59

*Nota.* Fuente [69]

## 5.7 Detección de objetos

Las CNN utilizadas para la clasificación de imágenes presentadas anteriormente en la figura 21, son debido a que se está trabajando en aprendizaje supervisado alimentadas con imágenes y su respectiva etiqueta. Por ejemplo, si se estuviera intentando clasificar tres tipos de imágenes: carro, motocicleta y bicicleta, la última capa de la CNN tuviera tan solo tres neuronas, una para cada clase y el vector de salida en formato one-hot encoding sería el siguiente:

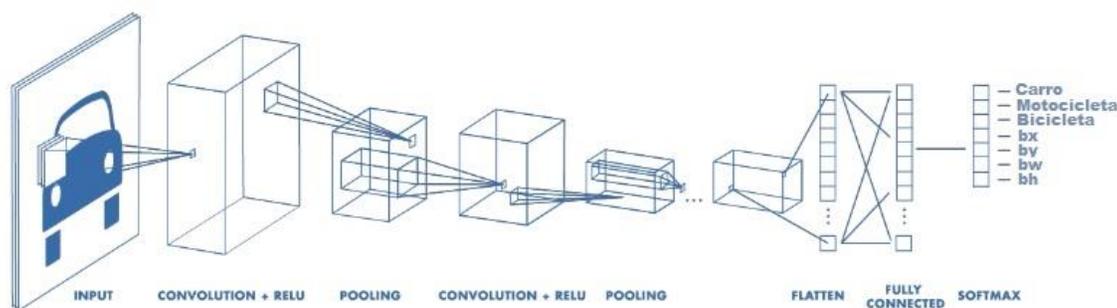
- Si la CNN predice un carro, el vector de salida sería:  $y = [1,0,0]$
- Si la CNN predice una motocicleta, el vector de salida sería:  $y = [0,1,0]$

- Si la CNN predice una bicicleta, el vector de salida sería:  $y = [0,0,1]$

Teniendo lo anterior en mente, si se desea utilizar una CNN para la detección de objetos, es necesario utilizar imágenes que contengan el objeto de interés, la etiqueta del objeto, y finalmente cuatro coordenadas las cuales especifican la posición del objeto dentro de la imagen. Por consiguiente, la capa final de la CNN debe ser modificada añadiendo cuatro neuronas las cuales mostrarán las predicciones para las coordenadas del objeto. La figura 35 muestra una analogía gráfica de cómo sería una CNN entrenada para la detección de objetos.

**Figura 26**

*Analogía de una CNN para la detección de objetos*



*Nota.* Fuente autor

La detección de objetos consiste en la habilidad de un modelo para encontrar la ubicación del objeto de interés dentro de una imagen situando un cuadro delimitador a su alrededor. La posición del objeto dentro de una imagen se define mediante el par  $(b_x, b_y)$ , los cuales representan el punto medio del objeto, y los valores  $b_w$  y  $b_h$  que representan el ancho y alto del cuadro

delimitador. Además de localizar el objeto es necesario clasificar dicho objeto de interés entre  $n$  posibles categorías o clases.

Por lo tanto, dada una imagen de entrada, el modelo construirá un vector de salida de la siguiente forma:

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Donde  $p_c$  indica la probabilidad sobre si se encuentra el objeto deseado en la imagen, si  $p = 1$  indica que la imagen contiene con seguridad al objeto, de lo contrario no lo contiene. Tal como se mencionó anteriormente los valores  $b_x, b_y, b_w, b_h$  forman el rectángulo delimitador alrededor del objeto, y los valores  $c_1, \dots, c_n$ , representa la clase a la que pertenece el objeto.

## 5.8 Algoritmo YOLO

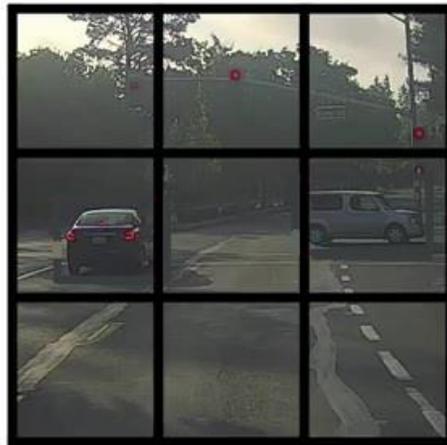
YOLO [70] procesa (mira) la imagen solo una vez de forma inteligente, de ahí parte su nombre You Only Look Once. El modelo YOLO está basado en una red convolucional inspirada en GoogleNet para la clasificación de imágenes. Dicha red fue entrenada inicialmente para resolver problemas de clasificación de ImageNet y luego fue adaptada para abordar problemas de detección. EL objetivo final de YOLO es crea un vector, el cual simplifica los objetos detectados y las coordenadas en donde se encuentra el objeto.

El algoritmo YOLO funciona de la siguiente manera: comienza colocando una cuadrícula o grid sobre la imagen. El grid divide la imagen de entrada en una cuadrícula  $S \times S$ . Para explicar el funcionamiento de YOLO se supondrá que la imagen se dividió con una cuadrícula de  $3 \times 3$

como se observa en la figura 27. Además, se tienen tres clases que representan los objetos que se desean detectar: peatones, automóviles y motocicletas, respectivamente.

### Figura 27

*Imagen dividida en una cuadrícula 3x3*



*Nota.* Fuente [71]

Para cada celda de la cuadrícula, se ejecuta el algoritmo de detección de objetos que se describió en la sección anterior, con el cual se obtiene un vector o etiqueta de la siguiente forma:

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

- $p_c$  define la probabilidad si un objeto está presente en la cuadrícula
- $b_x, b_y, b_w, b_h$  especifican el cuadro delimitador si hay un objeto

- $c_1, c_2, c_3$  representan las clases. Entonces si hay un automóvil los valores serán los siguientes:  $c_1 = 0, c_2 = 1, c_3 = 0$ .

El vector para la primera cuadrícula de la esquina superior de la figura 27, será la siguiente:

### Figura 28

*Cuadrícula de la esquina superior izquierda de la figura 27*



*Nota.* Fuente [71]

$$\mathbf{y} = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

Ya que no existe ningún objeto en esta cuadrícula  $p_c = 0$ , y como consecuencia ninguno de los siguientes valores es relevante. La etiqueta anterior será la misma para todas las cuadrículas de la primera y tercera fila, como también la cuadrícula central. Ya que ninguna contiene un objeto de interés.

El vector para la cuadrícula central izquierda de la figura 27, será la siguiente:

## Figura 29

*Cuadrícula central izquierda de la figura 27*



Nota: Fuente [71]

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Dado que existe un objeto en esta cuadrícula  $p_c = 1$ . por lo tanto,  $b_x, b_y, b_w, b_h$  se calcularán en relación a la celda de la cuadrícula que se está tratando actualmente. La etiqueta anterior será la misma para la cuadrícula central derecha, puesto que existe el objeto de interés vehiculó.

Cabe resaltar que un objeto puede ocupar varias celdas. Por lo tanto, la celda responsable de detectar ese objeto será aquella en la que se encuentra el centro del objeto.

En consecuencia, con todo lo anterior, para cada una de las nueve cuadrículas, se tendrá un vector de salida con ocho dimensiones, que, al juntarlos, la salida tendrá un volumen de  $3 \times 3 \times 8$ . Por lo tanto, lo que YOLO desea construir es un modelo que produzca como salida un tensor (una generalización de la noción de matriz a dimensiones superiores) de tamaño  $S \times S \times (5 + n)$ . Para cada una de las  $S \times S$  celdas de la cuadrícula se obtiene un vector con la probabilidad del objeto,

su posición y la clase codificada usando one-hot encoding, donde  $n$  es el número de clases posibles.

Uno de los problemas de la detección de objetos usando la aproximación que se ha descrito anteriormente es el hecho de que cada celda de la cuadrícula solo puede detectar un objeto. Entonces si se tiene varios objetos en la misma celda, la técnica que se ha descrito hasta ahora no permite detectarlos. Para solucionar esto, YOLO hace uso de las cajas de anclaje.

En la figura 30 el punto medio de los objetos peatón y automóvil se encuentra en la misma cuadrícula (cuadrícula central inferior). Para predecir con éxito las dos clases se utilizan las cajas de anclaje, las cuales básicamente son rectángulos a diferentes escalas y rotaciones, como los que se muestra en la figura 31. Cabe mencionar que se puede aumentar el número de cajas de anclaje. Se han tomado dos para que el concepto sea más fácil de describir.

### **Figura 30**

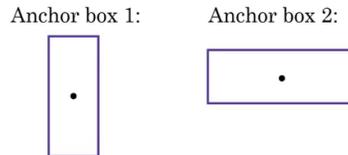
*Dos objetos en la misma celda*



*Nota.* Fuente [71]

### Figura 31

*Tipo de cajas de anclaje*



*Nota.* Fuente [71]

Al agregar las cajas de anclaje el vector de salida para todas las cuadrículas cambia, pues en lugar de tener una sola salida, se tendrán dos salidas debido a que se ha utilizado dos cajas de anclaje. Por lo tanto, el vector para la cuadrícula central inferior, será la siguiente:

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Los dos valores para  $p_c = 1$ , ya que la cuadrícula contiene los dos objetos de interés (peatón y automóvil), y estos objetos se diferencian en la codificación de clases. En donde la parte superior del vector  $c_1 = 1$  ya que es la clase que pertenece al objeto peatón mientras que las clases

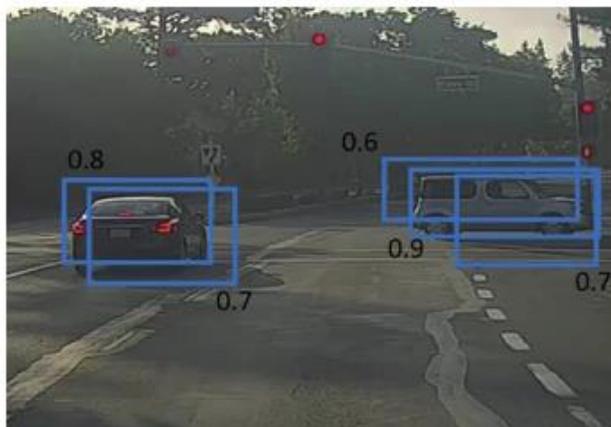
$c_2 = 0$  y  $c_3 = 0$ . Lo mismo sucede con la clase automóvil, en la parte inferior del vector, la clase que corresponde al objeto se puntúa como  $c_2 = 1$ , y las demás clases son cero.

Por lo tanto, con la adición de las cajas de anclaje, el modelo de YOLO intentara predecir un tensor de tamaño  $S \times S \times (k \times (5 + n))$ , donde  $k$  es el numero de cajas de anclaje y  $n$  es el número de clases posibles.

Otro de los problemas comunes con los algoritmos de detección de objetos es que, en lugar de detectar un objeto solo una vez, podría detectarlo varias veces, tal como se muestra en la figura 32.

### Figura 32

*Múltiple detección sobre los objetos de interés*



*Nota.* Fuente [71]

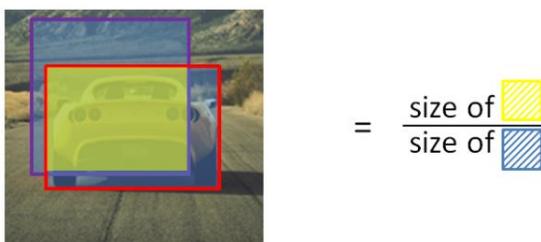
Para solucionar estos inconvenientes YOLO hace uso de la propiedad intersección sobre unión y supresión no máxima:

- Intersección sobre unión

Intersección sobre unión (intersection over union, IOU), es una métrica de evaluación que indica cuanto se parecen dos conjuntos. En detección de objetos, cuanto se acerca el área de una detección, a la etiqueta real en los datos. IOU calcula la intersección sobre la unión del cuadro delimitador real y el cuadro delimitador predicho, tal como se muestra en la figura 33, en donde el cuadro rojo es cuadro delimitador real y el cuadro morado es el predicho.

**Figura 33**

*Intersección sobre unión*



*Nota.* Fuente [72]

Se podría considerar que la predicción del cuadro delimitador es suficientemente buena cuando IOU es mayor que 0.5. El valor umbral de 0.5 es arbitrario y se puede cambiar de acuerdo al problema específico que se intenta resolver. Intuitivamente, cuanto más aumente el umbral, mejores serán las predicciones.

- Supresión no máxima

En la figura 34 los autos se detectan más de una vez, la técnica de supresión no máxima elimina varios delimitadores predichos para obtener una sola detección por objeto. La técnica funciona de la siguiente manera:

Inicialmente analiza las probabilidades asociadas a cada detección y escoge la predicción con mayor probabilidad. En la figura 34(a), 0.9 es la probabilidad más alta, por lo cual se selecciona este cuadro delimitador primero.

El paso siguiente es suprimir los cuadros que tienen un valor de IOU bajo con respecto al cuadro delimitador seleccionado en el anterior paso. Por lo que, los cuadro con probabilidades 0.6 y 0.7 serán eliminados, como observa en la Figura 34(b).

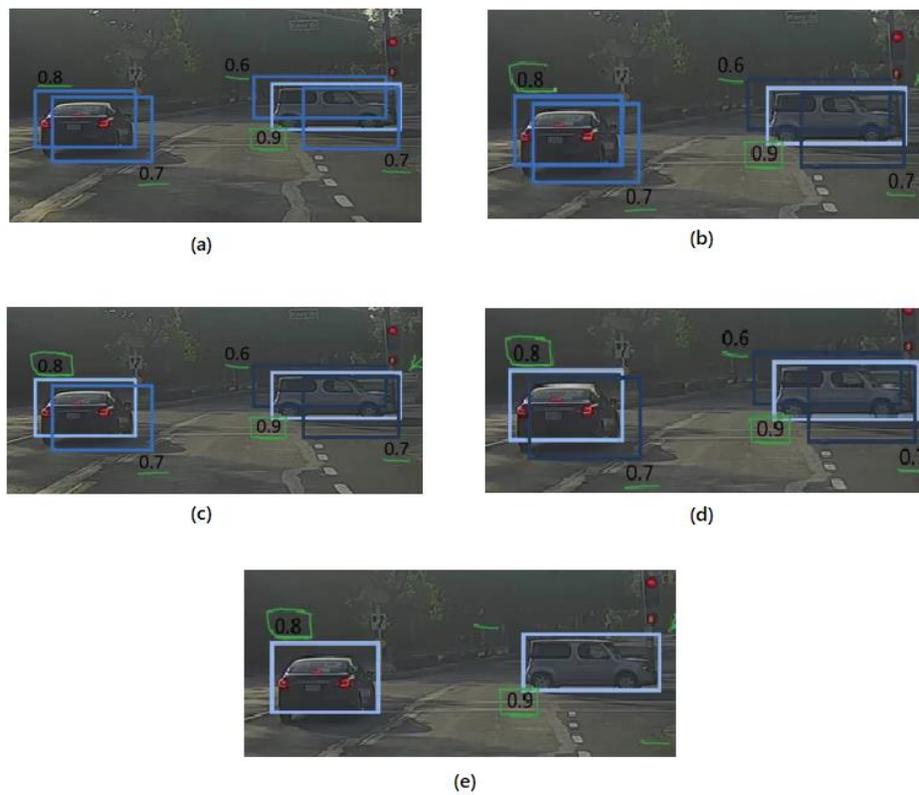
De los cuadros delimitadores faltantes, nuevamente se selecciona el siguiente cuadro con la segunda probabilidad más alta, en este caso sería el cuadro delimitador con probabilidad 0.8.

Nuevamente se suprime los cuadros que tienen un valor de IOU bajo con respecto al cuadro delimitador seleccionado en el anterior paso. Por lo tanto, el cuadro con probabilidad 0.7 será eliminado.

Los anteriores pasos se repiten las veces necesarias hasta obtener los cuadros delimitadores finales, como se observa en la figura 34(e).

**Figura 34**

*Técnica de supresión no máxima*



*Nota.* Fuente [71]

## **6 Base tecnológica a implementar en el procedimiento**

### **6.1 Equipo de computo**

Para el desarrollo de este trabajo de grado se dispone de un computador personal de marca Asus X441UV, el cual tiene las siguientes características:

- Windows 10
- Intel Core i5
- Tarjeta gráfica Nvidia 920MX
- 16 GB de RAM

### **6.2 Tecnologías empleadas software y librerías**

#### **6.2.1 Python**

Python versión 3 es un lenguaje de programación de alto nivel y como lenguaje de programación puede ser utilizado por cualquier público en dominios de tecnología, porque es general y tiene una comunidad inmensa que lo respalda y demuestra ser superior a java.

Este tipo de lenguaje de programación se puede ocupar de matrices, aritmética, objetos y variables; así como también de otros conceptos de la computación y hacer desarrollos muy completos y robustos.

Python tiene bibliotecas como Numpy y Scikit-learn, las cuales se utilizan para la computación científica, computación avanzada y el aprendizaje automático. Por las anteriores razones es que Python se convierte en uno de los mejores lenguajes para inteligencia artificial.

### **6.2.2 Scikit-learn**

Scikit-learn es probablemente la librería más útil para Machine Learning en Python, es de código abierto y es reutilizable en varios contextos, fomentando el uso académico y comercial. Cuenta con algoritmos de clasificación y extracción de características, los cuales serán la base fundamental del desarrollo de esta tesis de grado.

### **6.2.3 OpenCV**

OpenCV (Open Source Computer visión) es una librería opensource enfocada a la visión por computador desarrollada por Intel. Desde su aparición en 1999 ha sido utilizada en infinidad de aplicaciones las cuales se han beneficiado de sus potentes funciones. Entre ellas destacan el manejo de imágenes, el reconocimiento de objetos, movimiento, gestos y caras. Es multiplataforma, con versiones para distribuciones GNU/Linux, Mac OS X, Windows, Android e IOS.

### **6.3 Darknet**

Darknet es un framework de redes neuronales de código abierto, es un framework rápido y altamente preciso para la detección de objetos en tiempo real y también es usado para imágenes.

## **6.4 Servicios a utilizar**

## **6.5 Google Colaboratory**

Google Colaboratory es un servicio en la nube, y es un entorno de máquinas virtuales basado en Jupyter Notebooks. Google Colab ofrece GPU gratuita por alrededor de 12 horas para entrenar los modelos de deep learning.

Se hace uso de este servicio para entrenar el algoritmo de YOLO con el framework darknet, ya que ahorra tiempo de entrenamiento gracias a sus tarjetas gráficas que ofrece.

## 7 Metodología

Como se ha mencionado anteriormente el trabajo de grado utiliza dos enfoques: Machine learning y Deep learning. Por lo tanto, se iniciará con la descripción del método de Machine learning, continuando con el método de Deep learning. Cabe destacar que los procesos empleados para la adquisición de la imagen, la extracción de la placa vehicular y la segmentación de los caracteres serán los mismos en los dos enfoques utilizados.

### 7.1 Adquisición de la imagen

Se inicia con la creación de una base de datos de imágenes corresponden a vehículos colombianos. Las fotografías fueron tomadas en su gran mayoría por el lado posterior del vehículo alrededor de 1.5 a 2 metros de distancia. En total se obtuvieron 198 imágenes recolectadas con diferentes cámaras de celular. La figura 35, muestra algunas de las imágenes que componen el conjunto de datos.

**Figura 35**

*Conjunto de imágenes de automóviles colombianos*



*Nota.* Fuente autor

## 7.2 Preprocesado de las imágenes

Una vez introducida una imagen al sistema se intenta hacer sobresalir o destacar la placa vehicular sobre el fondo. Esto tiene dos alternativas ya que el conjunto de imágenes posee autos particulares y autos de servicio nacional. En los primeros la placa es de color amarillo, mientras en los segundos la placa es de color blanco.

Para las placas vehiculares de color amarillo, se separa la imagen en sus tres canales de color (rojo, verde y azul). A continuación, se resta los canales rojo y verde, el resultado se puede observar en la figura 36(a). Finalmente, se binariza la imagen con la ayuda del método: `cv2.threshold` y el argumento: `cv2.THRESH_BINARY`. El resultado se muestra en la figura 36(b). Con el anterior procedimiento se logra con éxito destacar la placa vehicular del fondo.

**Figura 36**

*Procedimiento para resaltar placa amarilla sobre el fondo*



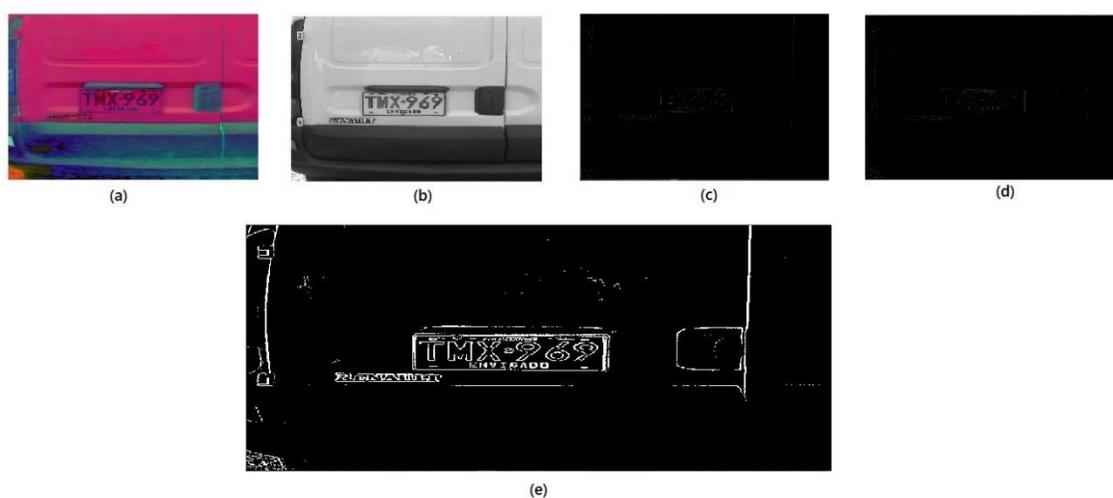
*Nota.* Fuente autor

El procedimiento para las placas de color blanco se muestra en la figura 37. La imagen se transforma al espacio de color HSV (37a). Luego se selecciona el canal *value* de la imagen con el

cual se trabajará (37b). Se aplica la transformación morfológica de BlackHat al canal *value*(37c), a continuación, se resta el canal *value* y la transformación BlackHat (37d). Finalmente, se binariza la imagen con la ayuda del método: *cv2.threshold* y el argumento *cv2.THRESH\_BINARY\_INV*, lo cual ayuda con éxito a destacar la placa blanca sobre el fondo (37e).

**Figura 37**

*Procedimiento para resaltar placa blanca sobre el fondo*



*Nota.* Fuente autor

### 7.3 Extracción placa vehicular

Para la etapa de extracción de la placa vehicular se utiliza la detección de contornos. Ya que como se ha mencionado anteriormente un contorno es un conjunto de puntos que conectados unos con otros de manera consecutiva forman una figura que rodea un objeto determinado, que en este caso es la placa vehicular.

Se procede a encontrar los contornos con la ayuda de la función: *cv2.findContours()*, cabe mencionar que los contornos tienen propiedades equivalentes a área, perímetro, relación de

aspecto, extensión, solidez, entre otras. Por lo tanto, es necesario escoger valores adecuados para estas propiedades, con el fin de que la función retorne correctamente la placa vehicular.

La figura 38(a) muestra los contornos encontrados sobre la imagen al aplicar la función `cv2.findContours()`. Una vez se logra seleccionar la placa vehicular con los contornos, el paso siguiente es seleccionar las coordenadas en la imagen en donde se encuentra la matrícula para extraerla en su totalidad, como se observa en la figura 38(b).

**Figura 38**

*Extracción placa vehicular*



(a)



(b)

*Nota.* Fuente autor

#### 7.4 Segmentación de los caracteres

Para esta etapa es necesario binarizar la placa vehicular de forma adaptativa, debido a que las placas difieren en cuanto a condiciones de luz en distintas áreas, ya que por lo general se encuentran incrustadas en cavidades sobre los vehículos, lo cual genera una sombra que cubre algunos caracteres. Por lo tanto, usar una binarización con un valor global, como el que se utilizó para extracción de placa vehicular no siempre tendrá buenos resultados.

Con lo anterior en mente, se procede a binarizar la imagen con el método: *cv2.adaptiveThreshold*, y el argumento: *cv2.THRESH\_BINARY\_INV*. El resultado después de aplicar esta función se puede observar en la figura 39.

### Figura 39

*Binarización adaptativa sobre la placa vehicular*



*Nota.* Fuente autor

A continuación, para la segmentación de los caracteres nuevamente se utilizará la detección de contornos, aunque se hace necesario hacer un ajuste en el los parámetros de los contornos con el fin de seleccionar solo los dígitos de la placa y así evitar los caracteres que componen el nombre del municipio donde fue registrado el vehículo. La figura 40 muestra con éxito la segmentación de los caracteres presentes en la placa vehicular.

**Figura 40**

*Segmentación caracteres placa vehicular*



*Nota.* Fuente autor

El procedimiento anterior se aplica sobre todo el conjunto inicial de fotografías de autos colombianos, para así extraer todos los caracteres posibles, con el objetivo de crear un nuevo conjunto de datos que servirá como base fundamental para la extracción de características y el entrenamiento de los clasificadores.

La figura 41 muestra dos conjuntos de datos, uno corresponde para las letras y el otro para los números que componen la matrícula vehicular. Cada carpeta como su nombre lo indica tiene almacenada cierta cantidad de imágenes correspondiente a cada carácter.

**Figura 41**

*Conjunto de datos de letras y números*



*Nota.* Fuente autor

## 7.5 Data augmentation

Esta etapa se hace sumamente necesaria pues el objetivo es aumentar el tamaño y la diversidad del conjunto de datos de entrenamiento, esto contribuye a la generalización de los algoritmos de clasificación y aumentan la precisión al momento de predecir los caracteres. Esta técnica se realiza tanto en el conjunto de caracteres que son usados para machine learning, como también en el conjunto de datos de placas vehiculares, el cual es usado para el entrenamiento de YOLO.

## 7.6 Extracción de características

Una vez se ha logrado segmentar cada carácter de la placa vehicular por separado, se procede a la extracción de características que posee cada uno de ellos y que lo diferencian de los demás, utilizando los siguientes algoritmos:

### 7.6.1 *Histograma de gradientes orientados (HOG)*

Para esta etapa del proyecto se utilizan las funciones que ofrece la biblioteca de aprendizaje de máquinas Scikit-learn. Para calcular las características que presentan de los dígitos aplicando HOG se hace uso de la siguiente función:

```
hog(image, orientations=12, pixels_per_cell=(8, 8), cells_per_block=(2, 2), block_norm='L2Hys', visualize=True, transform_sqrt=False, feature_vector=True, multichannel=None )
```

la anterior función, retorna dos parámetros: el primero es el vector de características, que contiene la información relevante de la imagen, y que se utiliza en los algoritmos de clasificación. El segundo parámetro que retorna es la imagen siendo afectada por la técnica HOG.

La figura 42 muestra imagen de entrada y la imagen se salida siendo afectada por HOG.

## Figura 42

*Extracción de características mediante HOG*



*Nota.* Fuente autor

### 7.6.2 HAAR

Para calcular las características que presentan los dígitos por medio de filtros HAAR, se utiliza la siguiente función que ofrece la biblioteca Sckit-Learn:

```
haar_like_feature(int_image, r, c, width, height, feature_type)
```

En la anterior función: *int\_image* corresponde a la imagen transformada en escala de grises, *r* corresponde a la coordenada de fila de la esquina superior izquierda de la ventana de detección, *c* corresponde a la coordenada de columna de la esquina superior izquierda de la ventana de detección, *width* corresponde al ancho de la ventana de detección, *height* corresponde a la altura de la ventana de detección, el parámetro más relevante y que afecta directamente la cantidad de características extraídas como el tiempo de ejecución, es *feature\_type*, ya que es el parámetro que selecciona el tipo de característica a considerar. *feature\_type* tiene los filtros que se presentaron en la figura 8.

Es posible seleccionar todos los diferentes filtros que proporciona HAAR, pero el tiempo de ejecución en la extracción de características es bastante elevado y se requiere de un computador

con mejor hardware. Por lo tanto, se escogió tan solo el descriptor type-4, el cual varía al largo de los ejes X, Y sobre la imagen para encontrar características.

La función de *haar* retorna el vector de características, que contiene la información relevante de la imagen. El número de características retornadas con este descriptor son iguales a 160000 por cada dígito.

### 7.6.3 Local Binary Pattern (LBP)

Para calcular los códigos LBP se hace uso de la siguiente función:

```
local_binary_pattern(image, P, R, method='uniform')
```

Sin embargo, la variable *lbp* devuelta por la anterior función no se puede usar directamente como un vector de características. Debido a que *lbp* es una matriz de dos dimensiones con el mismo ancho y alto que la imagen de entrada.

Por lo tanto, para construir el vector de característica real, se necesita hacer una llamada a `np.histogram` con el objetivo de aplanar la matriz y convertirla a un vector. Luego el histograma se normaliza de modo que sume a 1.

Con todo lo anterior en mente, se desarrolla la siguiente función (ver figura 43), que retorna tanto el histograma, como el vector de características, que se usará en predicciones futuras.

**Figura 43**

*Función para calcular el vector de características con LBP*

```
def descriptor(image, numPoints, radius, eps=1e-7):

    lbp = feature.local_binary_pattern(image, numPoints, radius, method="uniform")
    (hist, _) = np.histogram(lbp.ravel(), bins=np.arange(0, numPoints + 3),
                             range=(0, numPoints + 2))

    hist = hist.astype("float")
    hist /= (hist.sum() + eps)

    return lbp, hist
```

*Nota.* Fuente autor

La figura 44 muestra la imagen de entrada, la imagen en escala de grises, la imagen siendo afectada por los códigos LBP.

**Figura 44**

*Imagen original, en escala de grises y siendo afectada por códigos LBP*



*Nota.* Fuente autor

## 7.7 Entrenamiento y creación de modelos

Después de aplicar los extractores de características mencionados anteriormente para todas las imágenes de los caracteres que componen el conjunto de datos, se obtendrá una lista de vectores de características y una lista de etiquetas. Entonces, es necesario dividir las listas de vectores y etiquetas, en un conjunto de entrenamiento y un conjunto de prueba, las cuales servirán de entrenamiento para los clasificadores a utilizar. Para lograr lo anterior, se hace uso de la función `train_test_split()`. La función recibe tres parámetros: la lista de las características calculadas, las etiquetas de imágenes y el tamaño del `test_size`, que en este caso será de 0.3, lo que significa que 70% de la información será usada como entrenamiento y 30% como prueba. La función retorna cuatro parámetros: `X_train` que son las características extraídas que se usaran para el entrenamiento, `X_test` son las características extraídas que se usaran como prueba, `y_train` son las etiquetas de las imágenes que se usaran para el entrenamiento, `y_test`, son las etiquetas de las imágenes que se usaran como prueba.

Posteriormente se importan los clasificadores a usar y entrenar:

- KNN

`KNeighborsClassifier (n_neighbors=5, metric= 'Euclidean distance')`

`n_neighbors` es la cantidad de vecinos cercanos a considerar, `metric` es la distancia que se usara para calcular los vecinos cercanos

- SVM

`svm.SVC (C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=0.01, kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)`

Los parámetros modificados de la anterior ecuación fueron: `decision_function_shape` para seleccionar el enfoque one vs rest y `kernel = 'linear'`. Los demás parámetros se tomaron por defecto.

- Random Forest

```
RandomForestClassifier(n_estimators=100)
```

`n_estimators` son la cantidad de árboles a crear, los demás parámetros se tomaron por defecto.

A continuación, se entrena cada clasificador con el método:

```
fit(X_train, y_train)
```

Finalmente, se crea el modelo para ser cargado en futuras predicciones:

```
joblib.dump()
```

## 7.8 Entrenamiento YOLO

Existen dos formas para el entrenamiento de YOLO: la primera se puede entrenar YOLO desde cero, la segunda se puede utilizar unos pesos previamente entrenados (proviene del modelo darknet53 [73]) para las capas convolucionales. Es recomendable usar la segunda opción para el entrenamiento sobre la primera, debido a que se parte de una red que ya ha sido entrenada previamente y no costará demasiado encontrar la configuración óptima de pesos, ya que si se parte de cero el entrenamiento puede tomar mucho tiempo.

Para iniciar con la configuración y entrenamiento, se usa Darknet el cual es un framework para entrenar redes neuronales, es de código abierto y está escrito en C/CUDA y sirve como base para YOLO. Por lo tanto, se debe clonar el repositorio de GitHub [75] que contiene toda la estructura de YOLO.

### 7.8.1 *Etiquetado de imágenes*

Para el entrenamiento es necesario pasarle como entrada a YOLO las imágenes etiquetadas. Existe un programa llamado labelImg, el cual facilita esta tarea. La figura 45 muestra el funcionamiento de esta herramienta, la cual permite seleccionar el objeto de interés. El programa retorna un archivo .txt que contiene las coordenadas del objeto dentro de la imagen. Dichas coordenadas serán usadas por YOLO para predecir el rectángulo delimitador.

**Figura 45***Etiquetado de imágenes a través de labelImag**Nota.* Fuente autor

### 7.8.2 Configuración de archivos

Para configurar el entrenamiento de YOLO, se utiliza como referencia el archivo *yolo – obj.cfg* el cual contiene la estructura de la red YOLO, además de una serie de parámetros que se deben modificar. Las modificaciones se han basado en lo explicado en la documentación de YOLO [73][74].

- Batch

es la cantidad de muestras que se procesarán en un lote. Se utiliza para que, en el entrenamiento, las imágenes se procesen en conjuntos a la red, en lugar de una en una, ya que supondría realizar muchos cambios en los pesos de la red. Ni tampoco todas las imágenes a la vez, ya que no es factible en términos de memoria. Para realizar el entrenamiento el batch debe tener como valor 64, ya que por defecto viene igual a 1.

- Subdivision

es la cantidad de mini batches que divide el batch para acelerar el proceso de entrenamiento y mejorar la generalización. El valor de este parametro sera igual a 16.

- Classes

Corresponde al número de clases u objetos que se quieren clasificar. Para esta etapa se desea clasificar los caracteres de la placa vehicular, entonces el valor del parámetro será igual a 36.

- Filters

En las capas convolucionales se modifica el número de filtros. La cantidad de filtros dependen del número de clases que se desea detectar. Por lo tanto, se sigue la siguiente ecuación:  
 $filters = (classes + 5) * 3.$

### 7.8.3 Creación de archivos

Dentro de la carpeta Darkent\data\ se deben crear un conjunto de archivos:

- Traint.txt

este archivo contiene las rutas de las imágenes que se utilizarán para el entrenamiento. La ruta de cada imagen ira en una sola línea.

- Test.txt

este archivo contiene las rutas de las imágenes que se utilizarán en el test.

- `.name`

en este archivo se deben almacenar los nombres de los objetos que se desea que YOLO detecte, cada nombre ira en una línea.

- `.data`

este archivo contiene una serie de parámetros y rutas que se deben adjuntar: como el número de clases, la ruta del archivo con la lista de imágenes para entrenar, la ruta del archivo con la lista de imágenes para realizar el test, la ruta del archivo que contiene el nombre de los objetos, y, por último, la ruta donde se desea que YOLO guarde los pesos que ira entrenando cada cierta época. La figura 61, muestra el contenido del archivo `.data`.

#### **Figura 46**

*Contenido del archivo `.data`.*

```
classes = 36
train = /content/gdrive/MyDrive/placas_yolo/train.txt
valid = /content/gdrive/MyDrive/placas_yolo/test.txt
names = /content/gdrive/MyDrive/placas_yolo/classes.names
backup = /content/gdrive/MyDrive/models
```

*Nota.* Fuente autor

#### 7.8.4 Entrenamiento YOLO

Una vez se ha configurado y creado los diferentes archivos necesarios, YOLO está listo para ser entrenado utilizando el siguiente comando.

```
!./darknet detector train cfg/placas.data cfg/yolov3_placa_train.cfg  
darknet53.conv.74 -dont_show
```

`darknet detector train` sirve para iniciar el entrenamiento, `cfg/placas.data` es el archivo que contiene las rutas de las carpetas donde están los datos, y el número de clases, `cfg/yolov3_placa_train.cfg` es el archivo que se configuró previamente y que contiene la arquitectura de la red YOLO, y `darknet53.conv.74` son los pesos iniciales.

## 8 Resultados

La tabla 1 muestra el porcentaje de aciertos que tienen los diferentes algoritmos de clasificación de machine learning. Se puede observar que mejor ruta extractor de características-clasificador que tiene mayor porcentaje de predicción es HOG en compañía de KNN con el 99.2%

Por otro lado, la tabla 2 muestra el porcentaje de acierto que tiene YOLO frente a la clasificación de los caracteres de la placa vehicular, comprobando que este tipo de algoritmos al asemejarse al comportamiento humano logra mayor predicción que los algoritmos de machine learning.

**Tabla 1**

*Porcentaje de predicción sobre los caracteres vehiculares implementando modelos de machine learning*

	SVM	KNN	RANDOMFOREST
HOG	98.86%	99.2 %	98.6 %
LBP	54.06 %	54.66 %	52 %
HAAR	98.33 %	98.53 %	98.2 %

*Nota.* Fuente autor

**Tabla 2**

*Porcentaje de predicción sobre los caracteres vehiculares implementando modelos de deep learning*

YOLO	100%
------	------

*Nota.* Fuente autor

## 9 Comparación de los resultados obtenidos con trabajos similares

El trabajo realizado por Fernández Sánchez [21], en el cual para la etapa de extracción de la placa vehicular como de los caracteres que la componen se utilizaron técnicas de visión artificial, y posteriormente para la etapa de reconocimiento de los caracteres vehiculares implementaron el algoritmo de KNN. Ejecutaron pruebas sobre imágenes por el lado posterior del vehículo y lograron un porcentaje de predicción final del sistema del 85%.

El estudio realizado por Durán [76], en donde se extrajo la placa vehicular utilizando técnicas de visión artificial, para luego utilizar la herramienta comercial de Tesseract, en donde no fue necesario la segmentación de los caracteres. Lograron un porcentaje de predicción para los caracteres vehiculares del 80.1%.

Mientras que en el trabajo realizado por Wibirama, S., & Nugroho, H. A. [77], utilizaron como clasificador a support vector machine, con el cual lograron un porcentaje de predicción del 89.77%

Por otro lado, para los desarrollos utilizando deep learning, Angara [78], utilizo nuevamente técnicas de visión artificial para la extracción de la placa vehicular como también para la segmentación de los caracteres, para finalmente utilizar redes neuronales convolucionales para su reconocimiento, alcanzando un porcentaje de predicción del 91.12%.

Además, el desarrollo implementado por Jørgensen [80], el cual se basa en utilizar el algoritmo de detección de objetos YOLO en su versión 2, para las etapas de extracción de la placa vehicular, como también para la segmentación y reconocimiento de los caracteres consiguió un porcentaje de predicción del 97.8%.

## 10 Conclusiones

Este trabajo de grado ha desarrollado un sistema de reconocimiento automático de matrículas vehiculares, logrando su objetivo principal el cual era utilizar diferentes técnicas que ofrece el amplio campo de la visión por computadora para mostrar y comparar el rendimiento que tienen frente a problemas del mundo real. Lo anterior se logró gracias a la investigación e implementación de técnicas de visión artificial, técnicas de machine learning y técnicas de deep learning. Además, al buen diseño y desarrollo de cada una de las diferentes etapas que conforman el experimento, las cuales son la extracción de la placa vehicular, la segmentación de los caracteres y el reconocimiento de estos.

La experimentación realizada demuestra que el sistema utilizado para la etapa del reconocimiento de caracteres usando técnicas de machine learning como lo son: los descriptores HOG y HAAR, trabajando en conjunto con los clasificadores SVM, KNN y RANDOM FOREST, superan algunos estudios anteriores, en los cuales usan KNN y SVM. Adicionalmente, se logró mejores resultados que investigaciones donde usan herramientas comerciales como tesseract.

Por otro parte, se pudo concluir que el uso del algoritmo de detección de objetos YOLO que se implementó en este trabajo de grado, supera el rendimiento a los modelos de machine learning, como también a otros estudios mencionados.

El uso de redes neuronales convolucionales avanzadas que utiliza YOLO permite crear un sistema realmente capaz de “aprender” las características de los caracteres de la placa vehicular, sin necesidad de usar técnicas de extracción de características, lo cual hace de esta técnica una solución viable y confiable en sistemas ALPR comerciales.

## 11 Líneas futuras

Se han encontrado algunas limitaciones en el sistema desarrollado. El principal es el rendimiento relativamente débil de la etapa de segmentación de los caracteres. Debido a que el ambiente genera sombras muy fuertes sobre las placas vehiculares y cubren parcialmente o totalmente los caracteres, y como se ha demostrado al utilizar técnicas de visión artificial estas son susceptibles a extraer erróneamente los caracteres. Este inconveniente podría resolverse sustituyendo el enfoque actual con enfoques más robustos, como los que se utilizaron en esta tesis a la hora de usar YOLO para extraer la placa vehicular. Un algoritmo para solucionar este inconveniente y además profundizar en el campo del deep learning y la detección de objetos podría ser Mask R-CNN ya que es un algoritmo de dos fases y se podría esperar mejores resultados.

## 12 Referencias

- [1] RUNT. (2021). RUNT EN CIFRAS. Obtenido de RUNT: <https://www.runt.com.co/runt-en-cifras>
- [2] Aguilar Rodriguez, W. L., Aguilar Rodriguez, W. G., & Leguizamón Páez, M. A. (Junio de 2021). Tecnología Microchip Para Acceder a Información Vehicular Como Apoyo a Procesos de Control y Seguridad. *Revistas Universidad Tecnológica de Pereira*, 24(02). Obtenido de <https://revistas.utp.edu.co/index.php/revistaciencia/article/download/20241/1399>
- [3] Nuñez Bedoya, A. M., & Maya Perfetti, N. I. (2018). ANÁLISIS DE DESEMPEÑO DE ALGORITMOS DE RECONOCIMIENTO DE PLACAS VEHICULARES. Popayan, Cauca, Colombia. Obtenido de <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1630>
- [4] D Gunawan et al 2019 IOP Conf. Ser.: Mater. Sci. Eng. 648 012011
- [5] M. Satsangi, M. Yadav and P. S. Sudhish, "License Plate Recognition: A Comparative Study on Thresholding, OCR and Machine Learning Approaches," 2018 International Conference on Bioinformatics and Systems Biology (BSB), Allahabad, India, 2018, pp. 1-6.
- [6] Núñez Sánchez-Agustino, F. J. (01 de junio de 2016). Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional. España. Obtenido de <http://hdl.handle.net/10609/52222>
- [7] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proceedings of the 17th International Conference on Pattern Recognition, (ICPR)*, 2004.
- [8] X. Shi, W. Zhao, and Y. Shen, "Automatic license plate recognition system based on color image processing," in *Computational Science and Its Applications-ICCSA 2005*, 2005, pp. 1159–1168
- [9] F. Kahraman, B. Kurt, and M. Gökmen, "License plate character segmentation based on the gabor transform and vector quantization," in *Computer and Information Sciences- ISCIS*, 2003, pp. 381–388.

- [10] B. R. Lee, K. Park, H. Kang, H. Kim, and C. Kim, "Adaptive local binarization method for recognition of vehicle license plates," in *Lecture Notes in Computer Science*, 2004, pp. 646–655.
- [11] V. Franc and V. Hlaváč, "License plate character segmentation using hidden markov chains," in *Lecture Notes in Computer Science*, 2005, pp. 385–392.
- [12] K. I. K. J. H. Kim, "Color texture-based object detection: An application to license plate localization," in *Lecture Notes in Computer Science*, 2002, pp. 293–309.
- [13] H. E. Kocer and K. K. Cevik, "Artificial neural networks based vehicle license plate recognition," *Procedia Computer Science*, vol. 3, pp. 1033– 1037, 2011.
- [14] Y.-P. Huang, S.-Y. Lai, and W.-P. Chuang, "A template-based model for license plate recognition," in *Proceedings of the International Conference on Networking, Sensing and Control*, 2004, pp. 737–742.
- [15] G. G. Desai and P. P. Bartakke, "Real-Time Implementation Of Indian License Plate Recognition System," 2018 IEEE Punecon, Pune, India, 2018, pp. 1-5.
- [16] Open Automatic License Plate Recognition (OpenALPR) - Matthew Hill. *openalpr* GitHub repository, 2015, [online] Available: <https://github.com/openalpr/openalpr>.
- [17] Ray Smith, An Overview of the Tesseract OCR Engine, Google Inc. [online] Available: <https://github.com/tesseract-ocr/tesseract>
- [18] <https://platerecognizer.com/> Available: <https://github.com/marcbelmont/deep-license-plate-recognition>
- [19] <https://www.vigilantsolutions.com/>
- [20] <https://www.genetec.com/>

- [21] Fernandez Sanchez, L. (26 de 11 de 2019). AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM USING MACHINE LEARNING TECHNIQUES. England. Obtenido de <http://hdl.handle.net/10810/29790>.
- [22] A. Menon and B. Omman, "Detection and Recognition of Multiple License Plate From Still Images," 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 2018, pp. 1-5.
- [23] Gall, Richard. (2018). Different types of Machine Learning. <https://jaxenter.com/basic-introduction-machine-learning-145140.html>
- [24] P. Ganesan, V. Rajini, B. S. Sathish and K. B. Shaik, "HSV color space based segmentation of region of interest in satellite images," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, 2014, pp. 101-105.
- [25] [https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html)
- [26] S. Macêdo, G. Melo and J. Kelner, "A Comparative Study of Grayscale Conversion Techniques Applied to Descriptor Based Tracking," 2015 XVII Symposium on Virtual and Augmented Reality, Sao Paulo, 2015, pp. 1-6.
- [27] P. I. Quy and M. Polasek, "Using thresholding techniques for object detection in infrared images," Proceedings of the 16th International Conference on Mechatronics - Mechatronika 2014, Brno, 2014, pp. 530-537.
- [28] <https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>

- [29] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty and R. Ray, "Adaptive thresholding: A comparative study," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, 2014, pp. 1182-1186.
- [30] [https://docs.opencv.org/master/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html)
- [31] S. D. Lokmanwar and A. S. Bhalchandra, "Contour detection based on gaussian filter," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 722-725.
- [32] [https://docs.opencv.org/trunk/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html)
- [33] Dalal, N. y Triggs, B. (2005). Histograms of oriented gradients for human detection. 10.1109/CVPR.2005.177
- [34] M. R. Guedira, A. E. Qadi, M. R. Lrit and M. E. Hassouni, "A novel method for image categorization based on histogram oriented gradient and support vector machine," 2017 *International Conference on Electrical and Information Technologies (ICEIT)*, Rabat, 2017, pp. 1-5.
- [35] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, pp. I-I.
- [36] S. JABRI, M. SAIDALLAH, A. EL BELRHITI EL ALAOUI and A. EL FERGOUGUI, "Moving Vehicle Detection Using Haar-like, LBP and a Machine Learning Adaboost Algorithm," 2018 *IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, Sophia Antipolis, France, 2018, pp. 121-124.

- [37] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002.
- [38] K. Meena and A. Suruliandi, "Local binary patterns and its variants for face recognition," *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, Tamil Nadu, 2011, pp. 782-786, doi: 10.1109/ICRTIT.2011.5972286.
- [39] Y. Liu, R. Wang and Y. Zeng, "An Improvement of One-Against-One Method for Multi-Class Support Vector Machine," *2007 International Conference on Machine Learning and Cybernetics*, Hong Kong, 2007, pp. 2915-2920.
- [40] Wani, M. & Bhat, Heena. (2018). One-against-all and one-against-one multiclass Support Vector Machine algorithms for wind speed prediction. *International Journal of Renewable Energy Research*. 8. 909-915.
- [41] Jonathan Milgram, Mohamed Cheriet, Robert Sabourin. "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?. Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes 1, Oct 2006, La Baule (France). (inria-00103955)
- [42] H. b. Jaafar, N. b. Mukahar and D. A. Binti Ramli, "A methodology of nearest neighbor: Design and comparison of biometric image database," *2016 IEEE Student Conference on Research and Development (SCORED)*, Kuala Lumpur, 2016, pp. 1-6, doi: 10.1109/SCORED.2016.7810073.
- [43] K. Fathoni, M. Zikky, A. S. Nurhayati and I. Prasetyaningrum, "Application of K-Nearest Neighbor Algorithm For Puzzle Game of Human Body's System Learning on Virtual

Mannequin," *2018 International Conference on Applied Science and Technology (iCAST)*,

Manado, Indonesia, 2018, pp. 530-535, doi: 10.1109/iCAST1.2018.8751571.

[44] <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>

[45] L. Breiman, "Random forests", *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, 2001.

[46]H. Lan and Y. Pan, "A Crowdsourcing Quality Prediction Model Based on Random Forests," 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), Beijing, China, 2019, pp. 315-319, doi: 10.1109/ICIS46139.2019.8940306.

[47] [https://scikit-image.org/docs/dev/auto\\_examples/features\\_detection/plot\\_haar.html](https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_haar.html)

[48] Larranaga, Pedro & Inza, Iñaki & Moujahid, Abdelmalik. (2020). Tema 8. Redes

Neuronales - Scientific Figure on ResearchGate. Available from:

[https://www.researchgate.net/figure/Figura-2-Modelo-de-neurona-artificial-standard\\_fig2\\_268291232](https://www.researchgate.net/figure/Figura-2-Modelo-de-neurona-artificial-standard_fig2_268291232)

[49]<https://medium.com/maule-devs/aprendiendo-acerca-de-redes-neuronales-artificiales-5c81adbbe7ce>

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>

[51] G. E. Nasr, E. Badr, and C. Joun, "Cross entropy error function in neural networks: Forecasting gasoline demand." in *FLAIRS Conference*, 2002, pp. 381– 384.

[52] Michael A Nielsen. *Neural networks and deep learning*. URL: <http://neuralnetworksanddeeplearning.com/>

[53] <https://keras.io/api/optimizers/>

[54] Diederik P. Kingma, & Jimmy Ba. (2014). Adam: A Method for Stochastic Optimization.

[55] Nitish Srivastava, Geo\_rey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov.

Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[56] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[57] Hacibeyoglu, Mehmet. (2018). Human Gender Prediction on Facial Mobile Images using Convolutional Neural Networks. *International Journal of Intelligent Systems and Applications in Engineering*. 3. 203-208. 10.18201/ijisae.2018644778.

[58] Y. LeCun, “Convolutional neural networks local receptive fields weight sharing pooling,” 1989.

[59] D. Hubel and T. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *Journal of Physiology*, pp. 215–243, 1968

[60] <https://brainconnection.brainhq.com/2004/03/06/overview-of-receptive-fields/>

[61] <https://www.science.co.il/moshe/documents/deep-learning/CNN/>

[62] Y-T Zhou, Rama Chellappa, Aseem Vaid, and B Keith Jenkins. Image restoration using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1141–1151, 1988.

[63] A. Karpathy, “Cs231n convolutional neural networks for visual recognition.” (Stanford University)

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large Scale Hierarchical Image Database,” in *CVPR09*, 2009.

[65] Karen Simonyan, & Andrew Zisserman. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.

- [66] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, & Andrew Rabinovich. (2014). Going Deeper with Convolutions.
- [67] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, & Alex Alemi. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.
- [68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770{778, 2016.
- [69] The PASCAL Visual Object Classes Home page. <http://host.robots.ox.ac.uk/pascal/VOC/>
- [70] Joseph Redmon, Santosh Divvala, Ross Girshick, & Ali Farhadi. (2015). You Only Look Once: Unified, Real-Time Object Detection.
- [71] [https://www.coursera.org/learn/convolutional-neural-networks?=-](https://www.coursera.org/learn/convolutional-neural-networks?=)
- [72] <http://datahacker.rs/deep-learning-intersection-over-union/>
- [73] Redmon J. ImageNet Classification Available: <https://pjreddie.com/darknet/imagenet/#darknet53>
- [74] Redmon J. YOLO: Real-Time Object Detection. Available: <https://pjreddie.com/darknet/yolo/>
- [75] AlexeyAB. AlexeyAB/darknet. In: GitHub [Internet] Available: <https://github.com/AlexeyAB/darknet>
- [76] Durán, M. A. (2014). ANALISIS, DISEÑO E IMPLEMENTACION DE UN SISTEMA DE CONTROL DE INGRESO DE VEHICULOS BASADO EN VISION ARTIFICIAL Y RECONOCIMIENTO DE PLACAS EN EL PARQUEADERO DE LA UNIVERSIDAD POLITECNICA SELESIANA-SEDE CUENCA. Cuenca, Ecuador.
- [77] Wibirama, S., & Nugroho, H. A. (2017, July). Long distance Automatic Number Plate Recognition under perspective distortion using zonal density and Support Vector Machine. In Science and Technology-Computer (ICST), 2017 3rd International Conference on (pp. 159-164). IEEE.

- [78] Angara, N. S. (2015). AUTOMATIC LICENSE PLATE RECOGNITION USING DEEP LEARNING TECHNIQUES. Tyler, Texas, Search Results.
- [79] Jørgensen, H. (2017). Automatic License Plate Recognition using Deep Learning Techniques. Trondheim, Noruega.

## Anexos

### Anexo 1. Predicciones de los caracteres utilizando el descriptor HOG y el clasificador SVM

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KUM 876
AVH 247	AVH 247
GGP 719	GGP 719
FUY 368	FUY 368
IMW 908	IMW 908
AVF 687	AVF 687
AVJ 529	AVJ 529
CMP 655	CMP 655
CMD 846	CMD 846
ENX 310	ENX 310
DLU 976	DLU 976
AUR 541	AUR 541
GCZ 106	GCZ 106
GDQ 411	GDQ 411
IHZ 896	IHZ 896
BFK 596	BFK 596
AUY 075	AUW 075
MSW 538	MSW 538
DTP 518	DTP 518
BBO 289	BBO 289
AUY 298	AUY 298
IHZ 132	IHZ 132
ENX 497	ENX 497
FJK 598	FJK 598
AVK 755	AVK 755
CFT 762	CFT 762
GDQ 411	GD O 411
BKG 926	BKG 926
DLV 485	DLV 485
GDQ 244	GDQ 244
IAM 555	IAM 555
MFT 694	MFT 694

<i>BXW 351</i>	BXW 351
<i>LUA 808</i>	LUM 808
<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUW 900
<i>DTK 396</i>	DTK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 741
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HHM 205
<i>UDQ 619</i>	UDQ 619
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GDQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	EGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	ORW 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVW025
<i>HZL 298</i>	HZL 298
<i>IIK 396</i>	IIK 396
<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IHX 269
<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897

<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327
<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	EPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IIK 396</i>	IIK 396
<i>AUR 919</i>	AUR 919
<i>IIK 066</i>	IIK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871
<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311

<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLE 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MIX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKI 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AWD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140
<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	IVZ 391
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IOL 488
<i>IDL 488</i>	IDL 488

<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FGY 852
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	ISV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	DTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541
<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVD 092
<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915

<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>DTK 343</i>	DTK 343
<i>CDX 857</i>	CDX 857
<i>CQG 666</i>	<b>E</b> QG 666
<i>AVE 928</i>	AVE 928
<i>IIL 737</i>	IIL 737
<i>AUX 086</i>	AUX 086
<i>KDU 868</i>	KDU 868
<i>KLV 110</i>	KLV 110
<i>RKM 195</i>	RKM 195
<i>RKM 195</i>	RKM 195
<i>BYO 101</i>	BYO 101
<i>IIN 804</i>	IIN 804
<i>CVQ 795</i>	CVQ 795
<i>AVD 423</i>	AVD 423
<i>HFN 803</i>	HFN 803
<i>CQG 666</i>	<b>G</b> QG 666
<i>AVK 544</i>	AVK 544
<i>DLV 101</i>	DLV 101
<i>GRU 302</i>	GRU 302
<i>BBQ 727</i>	BBQ 727
<i>BPJ 644</i>	BPJ 644
<i>AVC 227</i>	AVC 227
<i>AUT 673</i>	AUT 673
<i>AVF 870</i>	AVF 870
<i>BYE 791</i>	BYE 791
<i>AVG 965</i>	AVG 965
<i>AUW 723</i>	AUW 723
<i>GRU 254</i>	GRU 254
<i>DHV 151</i>	DHV 151
<i>UBU 549</i>	UBU 549
<i>IHW 983</i>	<b>IHM</b> 983
<i>DQK 705</i>	DQK 705
<i>AUP 821</i>	AUP 821
<i>IIK 439</i>	IIK 439
<i>AUS 121</i>	<b>AOG</b> 121
<i>AUX 044</i>	AUX 044
<i>AUY 677</i>	AUY 677
<i>NEN 129</i>	NEN 129

<i>GTY 980</i>	GTY 980
<i>AVH 247</i>	AVH 247
<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106

**Anexo 2. Matriz de confusión para las letras utilizando el descriptor HOG y el clasificador**

**SVM**





#### Anexo 4. Predicción de los caracteres utilizando el descriptor HOG y el clasificador KNN

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
<i>KUM 876</i>	KUM 876
<i>AVH 247</i>	AVH 247
<i>GGP 719</i>	GGP 719
<i>FUY 368</i>	FUY 368
<i>IMW 908</i>	IMW 908
<i>AVF 687</i>	AVF 687
<i>AVJ 529</i>	AVJ 529
<i>CMP 655</i>	CMP 655
<i>CMD 846</i>	CMD 846
<i>ENX 310</i>	ENX 310
<i>DLU 976</i>	DLU 976
<i>AUR 541</i>	AUR 541
<i>GCZ 106</i>	GCZ 106
<i>GDQ 411</i>	GDQ 411
<i>IHZ 896</i>	IHZ 896
<i>BFK 596</i>	BFK 596
<i>AUY 075</i>	AUY 075
<i>MSW 538</i>	MSW 538
<i>DTP 518</i>	DTP 518
<i>BBO 289</i>	BBO 289
<i>AUY 298</i>	AUY 298
<i>IHZ 132</i>	IHZ 132
<i>ENX 497</i>	ENX 497
<i>FJK 598</i>	FJK 598
<i>AVK 755</i>	AVK 755
<i>CFT 762</i>	CFT 762
<i>GDQ 411</i>	GDQ 411
<i>BKG 926</i>	BKG 926
<i>DLV 485</i>	DLV 485
<i>GDQ 244</i>	GDQ 244
<i>IAM 555</i>	IAM 555
<i>MFT 694</i>	MFT 694

<i>BXW 351</i>	BXW 351
<i>LUA 808</i>	LUM 808
<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUW 900
<i>DTK 396</i>	DTK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 741
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HHM 205
<i>UDQ 619</i>	UDQ 619
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GDQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	EGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	URW 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVW 025
<i>HZL 298</i>	HZL 298
<i>IIK 396</i>	IIK 396
<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IHX 269
<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897

<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327
<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	LPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IIK 396</i>	IIK 396
<i>AUR 919</i>	AUR 919
<i>IIK 066</i>	IIK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871
<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311

<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLE 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MIX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AWD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	ITL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140
<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	IVZ 391
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IDL 488
<i>IDL 488</i>	IDL 488

<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FGY 852
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	ISV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	DTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541
<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVD 092
<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915

PEG 915	PEG 915
PEG 915	PEG 915
DTK 343	DTK 343
CDX 857	CDX 857
CQG 666	DQG 666
AVE 928	AVE 928
IIL 737	IIL 737
AUX 086	AUX 086
KDU 868	KDU 868
KLV 110	KLV 110
RKM 195	RKM 195
RKM 195	RKM 195
BYO 101	BYO 101
IIN 804	IIN 804
CVQ 795	CVQ 795
AVD 423	AVD 423
HFN 803	HFN 803
CQG 666	DQG 666
AVK 544	AVK 544
DLV 101	DLV 101
GRU 302	GRU 302
BBQ 727	BBQ 727
BPJ 644	BPJ 644
AVC 227	AVC 227
AUT 673	AUT 673
AVF 870	AVF 870
BYE 791	BYE 791
AVG 965	AVG 965
AUW 723	AUW 723
GRU 254	GRU 254
DHV 151	DHV 151
UBU 549	UBU 549
IHW 983	IHM 983
DQK 705	DQK 705
AUP 821	AUF 821
IIK 439	IIK 439
AUS 121	AUS 121
AUX 044	AUX 044
AUY 677	AUY 677
NEN 129	NEN 129

<i>GTY 980</i>	GTY 980
<i>AVH 247</i>	AVH 247
<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106





## Anexo 7. Predicción de los caracteres utilizando el descriptor HOG y el clasificador

### Random forest

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KUM 876
AVH 247	AVH 247
GGP 719	GGP 719
FUY 368	FUY 368
IMW 908	IMW 908
AVF 687	AVF 687
AVJ 529	AVJ 529
CMP 655	CMP 655
CMD 846	CMD 846
ENX 310	ENX 310
DLU 976	DLU 976
AUR 541	AGR 541
GCZ 106	GCZ 106
GDQ 411	GDQ 411
IHZ 896	IHZ 896
BFK 596	BFK 596
AUY 075	AUM 075
MSW 538	MSW 538
DTP 518	DTP 518
BBO 289	BBO 289
AUY 298	AUY 298
IHZ 132	IHZ 132
ENX 497	ENX 497
FJK 598	FJK 598
AVK 755	AVK 755
CFT 762	CFT 762

<i>GDQ 411</i>	GD <del>O</del> 411
<i>BKG 926</i>	BKG 926
<i>DLV 485</i>	DLV 485
<i>GDQ 244</i>	GDQ 244
<i>IAM 555</i>	IAM 555
<i>MFT 694</i>	MFT 694
<i>BXW 351</i>	BXW 351
<i>LUA 808</i>	LUM 808
<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUY 900
<i>DTK 396</i>	<del>O</del> TK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 741
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HHM 205
<i>UDQ 619</i>	DD <del>D</del> 619
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GDQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	EGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	DRW 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVS 025
<i>HZL 298</i>	HZL 298
<i>IHK 396</i>	IHK 396

<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IHX 269
<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897
<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327
<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	LPP 844
<i>DLV 101</i>	DL0 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IIK 396</i>	IIK 396
<i>AUR 919</i>	AUR 919
<i>IIK 066</i>	IIK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871

<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311
<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLE 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MIX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AWD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140

<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	IVI 391
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IOL 488
<i>IDL 488</i>	IDL 488
<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FGY 852
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	ISV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	OTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541

<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVD 092
<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>DTK 343</i>	DTK 343
<i>CDX 857</i>	CDX 857
<i>CQG 666</i>	EQG 666
<i>AVE 928</i>	AVE 928
<i>IIL 737</i>	IIL 737
<i>AUX 086</i>	AUX 086
<i>KDU 868</i>	KDU 868
<i>KLV 110</i>	KLV 110
<i>RKM 195</i>	RKM 195
<i>RKM 195</i>	RKM 195
<i>BYO 101</i>	BYO 101
<i>IIN 804</i>	IIN 804
<i>CVQ 795</i>	CVQ 795
<i>AVD 423</i>	AVD 423
<i>HFN 803</i>	HFN 803
<i>CQG 666</i>	EQG 666
<i>AVK 544</i>	AVK 544
<i>DLV 101</i>	DLV 101
<i>GRU 302</i>	GRU 302
<i>BBQ 727</i>	BBQ 727
<i>BPJ 644</i>	BPJ 644
<i>AVC 227</i>	AVC 227
<i>AUT 673</i>	AUT 673
<i>AVF 870</i>	AVF 870
<i>BYE 791</i>	BYE 791
<i>AVG 965</i>	AVG 965
<i>AUW 723</i>	AUW 723
<i>GRU 254</i>	GRU 254
<i>DHV 151</i>	DHV 151
<i>UBU 549</i>	UBU 549
<i>IHW 983</i>	IHM 983
<i>DQK 705</i>	DQK 705

<i>AUP 821</i>	AUF 021
<i>IIK 439</i>	IIK 439
<i>AUS 121</i>	AOS 121
<i>AUX 044</i>	AUX 044
<i>AUY 677</i>	AUY 677
<i>NEN 129</i>	NEN 129
<i>GTY 980</i>	GTY 980
<i>AVH 247</i>	AVH 247
<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106





## Anexo 10. Predicción de los caracteres utilizando el descriptor HAAR y el clasificador

### SVM

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
<i>KUM 876</i>	KUM 876
<i>AVH 247</i>	AVH 247
<i>GGP 719</i>	GGP 719
<i>FUY 368</i>	FUY 368
<i>IMW 908</i>	IMW 908
<i>AVF 687</i>	AVF 687
<i>AVJ 529</i>	AVJ 529
<i>CMP 655</i>	CMP 655
<i>CMD 846</i>	CMD 846
<i>ENX 310</i>	ENX 310
<i>DLU 976</i>	DLU 976
<i>AUR 541</i>	AUR 541
<i>GCZ 106</i>	GCZ 106
<i>GDQ 411</i>	GDQ 411
<i>IHZ 896</i>	IHZ 896
<i>BFK 596</i>	BFK 596
<i>AUY 075</i>	AUY 075
<i>MSW 538</i>	MSW 538
<i>DTP 518</i>	DTP 518
<i>BBO 289</i>	BBO 289
<i>AUY 298</i>	AUY 298
<i>IHZ 132</i>	IHZ 132
<i>ENX 497</i>	ENX 497
<i>FJK 598</i>	FJK 598
<i>AVK 755</i>	AVK 755
<i>CFT 762</i>	CFT 762

<i>GDQ 411</i>	GDQ 411
<i>BKG 926</i>	BKG 926
<i>DLV 485</i>	DLV 485
<i>GDQ 244</i>	GDQ 244
<i>IAM 555</i>	IAM 555
<i>MFT 694</i>	MFT 694
<i>BXW 351</i>	BXW 351
<i>LUA 808</i>	LUZ 808
<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUY 900
<i>DTK 396</i>	DTK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 347
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HRM 205
<i>UDQ 619</i>	UDQ 119
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GDQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	HGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	UXJ 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVS 025
<i>HZL 298</i>	HZL 298
<i>IHK 396</i>	IHK 396

<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IHX 269
<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897
<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327
<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	CPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IIK 396</i>	IIK 396
<i>AUR 919</i>	AUR 019
<i>IIK 066</i>	IIK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871

<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311
<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLV 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	<b>MF</b> X 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AVD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140

<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	FVI 391
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IDL 488
<i>IDL 488</i>	IDL 488
<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	FNX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FEY 857
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	TSV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	OTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541

<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVW 092
<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>DTK 343</i>	DTK 343
<i>CDX 857</i>	CDX 857
<i>CQG 666</i>	EQG 666
<i>AVE 928</i>	AVE 928
<i>IIL 737</i>	IIL 737
<i>AUX 086</i>	AUX 086
<i>KDU 868</i>	KDU 868
<i>KLV 110</i>	KLV 110
<i>RKM 195</i>	RKM 195
<i>RKM 195</i>	RKM 195
<i>BYO 101</i>	BYO 101
<i>IIN 804</i>	IIN 804
<i>CVQ 795</i>	CVQ 795
<i>AVD 423</i>	AVD 423
<i>HFN 803</i>	HFN 803
<i>CQG 666</i>	EQG 666
<i>AVK 544</i>	AVK 544
<i>DLV 101</i>	DLV 101
<i>GRU 302</i>	GRU 302
<i>BBQ 727</i>	BBQ 727
<i>BPJ 644</i>	BPJ 644
<i>AVC 227</i>	AVC 227
<i>AUT 673</i>	AUT 673
<i>AVF 870</i>	AVF 870
<i>BYE 791</i>	BYE 791
<i>AVG 965</i>	AVG 965
<i>AUW 723</i>	AUW 723
<i>GRU 254</i>	GRU 254
<i>DHV 151</i>	DHV 151
<i>UBU 549</i>	UBU 549
<i>IHW 983</i>	IHW 983
<i>DQK 705</i>	DQK 705

<i>AUP 821</i>	AUF 861
<i>IIK 439</i>	IIK 439
<i>AUS 121</i>	AOS 121
<i>AUX 044</i>	AUX 044
<i>AUY 677</i>	AUY 677
<i>NEN 129</i>	NEN 129
<i>GTY 980</i>	GTY 980
<i>AVH 247</i>	AVH 247
<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 373
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106







### Anexo 13. Predicción de los caracteres utilizando el descriptor HAAR y el clasificador

#### KNN

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KUM 876
AVH 247	AVH 247
GGP 719	GGP 719
FUY 368	FUY 368
IMW 908	IMW 908
AVF 687	AVF 687
AVJ 529	AVJ 529
CMP 655	CMP 655
CMD 846	CMD 846
ENX 310	ENX 310
DLU 976	DLU 976
AUR 541	AUR 541
GCZ 106	GCZ 106
GDQ 411	GDQ 411
IHZ 896	IHZ 896
BFK 596	BFK 596
AUY 075	AUY 075
MSW 538	MSW 538
DTP 518	DTP 518
BBO 289	BBO 289
AUY 298	AUY 298
IHZ 132	IHZ 132
ENX 497	ENX 497
FJK 598	FJK 598
AVK 755	AVK 755
CFT 762	CFT 762
GDQ 411	GDQ 411
BKG 926	BKG 926
DLV 485	DLV 485
GDQ 244	GDQ 244

<i>IAM 555</i>	IAM 555
<i>MFT 694</i>	MFT 694
<i>BXW 351</i>	BXW 351
<i>LUA 808</i>	LUA 808
<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUY 900
<i>DTK 396</i>	DTK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 147
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HHM 205
<i>UDQ 619</i>	UDQ 619
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GDQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	EGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	URP 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVS 025
<i>HZL 298</i>	HZL 298
<i>IIK 396</i>	IIK 396
<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IHX 269

<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897
<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327
<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	CPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IJK 396</i>	IJK 396
<i>AUR 919</i>	AUR 019
<i>IJK 066</i>	IJK 046
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871
<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575

<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311
<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLV 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MIX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AVD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140
<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	TVI 391
<i>EMR 275</i>	EMR 275

<i>IDL 488</i>	IDL 488
<i>IDL 488</i>	IDL 488
<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FGY 851
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	ISV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	OTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541
<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVL 092

<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>DTK 343</i>	DTK 343
<i>CDX 857</i>	CDX 857
<i>CQG 666</i>	EQG 666
<i>AVE 928</i>	AVE 928
<i>IIL 737</i>	IIL 737
<i>AUX 086</i>	AUX 086
<i>KDU 868</i>	KDU 868
<i>KLV 110</i>	KLV 110
<i>RKM 195</i>	RKM 195
<i>RKM 195</i>	RKM 195
<i>BYO 101</i>	BYO 101
<i>IIN 804</i>	IIN 804
<i>CVQ 795</i>	CVQ 795
<i>AVD 423</i>	AVD 423
<i>HFN 803</i>	HFN 803
<i>CQG 666</i>	EQG 666
<i>AVK 544</i>	AVK 544
<i>DLV 101</i>	DLV 101
<i>GRU 302</i>	GRU 302
<i>BBQ 727</i>	BBQ 727
<i>BPJ 644</i>	BPJ 644
<i>AVC 227</i>	AVC 227
<i>AUT 673</i>	AUT 673
<i>AVF 870</i>	AVF 870
<i>BYE 791</i>	BYE 791
<i>AVG 965</i>	AVG 965
<i>AUW 723</i>	AUW 723
<i>GRU 254</i>	GRU 254
<i>DHV 151</i>	DHV 151
<i>UBU 549</i>	UBU 549
<i>IHW 983</i>	IHH 983
<i>DQK 705</i>	DQK 705
<i>AUP 821</i>	AUI 861
<i>IIK 439</i>	IIK 439
<i>AUS 121</i>	AOS 121
<i>AUX 044</i>	AUX 044

<i>AUY 677</i>	AUY 677
<i>NEN 129</i>	NEN 129
<i>GTY 980</i>	GTY 980
<i>AVH 247</i>	AVH 247
<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	OFZ 711
<i>IIN 094</i>	ITN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 104







## Anexo 16. Predicción de los caracteres utilizando el descriptor HAAR y el clasificador

### Random forest

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KUM 876
AVH 247	AVH 247
GGP 719	GGP 719
FUY 368	FUY 368
IMW 908	IMW 908
AVF 687	AVF 687
AVJ 529	AVJ 529
CMP 655	CMP 655
CMD 846	CMD 846
ENX 310	ENX 310
DLU 976	DLU 976
AUR 541	AUR 541
GCZ 106	GCZ 106
GDQ 411	GDQ 411
IHZ 896	IHZ 896
BFK 596	BFK 596
AUY 075	AUY 075
MSW 538	MSW 538
DTP 518	DTP 518
BBO 289	BBO 289
AUY 298	AUY 298
IHZ 132	IHZ 132
ENX 497	ENX 497
FJK 598	FJK 598
AVK 755	AVK 755
CFT 762	CFT 762
GDQ 411	GDQ 411
BKG 926	BKG 926
DLV 485	DLV 485
GDQ 244	GDQ 244
IAM 555	IAM 555
MFT 694	MFT 694
BXW 351	BXW 351
LUA 808	LUZ 808

<i>CPZ 236</i>	CPZ 236
<i>AUS 753</i>	AUS 753
<i>AVI 111</i>	AVI 111
<i>KIN 864</i>	KIN 864
<i>FGR 368</i>	FGR 368
<i>AUV 230</i>	AUV 230
<i>AUY 900</i>	AUY 900
<i>DTK 396</i>	DTK 396
<i>RKM 195</i>	RKM 195
<i>CWB 710</i>	CWB 710
<i>NEG 741</i>	NEG 347
<i>MFT 694</i>	MFT 694
<i>GUL 746</i>	GUL 746
<i>HHM 205</i>	HRM 205
<i>UDQ 619</i>	UDQ 219
<i>CAP 875</i>	CAP 875
<i>GUL 328</i>	GUL 328
<i>INP 888</i>	INP 888
<i>AVH 903</i>	AVH 903
<i>BSQ 392</i>	BSQ 392
<i>CFT 762</i>	CFT 762
<i>GDQ 254</i>	GUQ 254
<i>UDN 328</i>	UDN 328
<i>AUW 327</i>	AUW 327
<i>EGX 687</i>	HGX 687
<i>AVK 755</i>	AVK 755
<i>IHX 572</i>	IHX 572
<i>DRW 868</i>	UXM 868
<i>FOW 279</i>	FOW 279
<i>SVS 025</i>	SVS 025
<i>HZL 298</i>	HZL 298
<i>IIK 396</i>	IIK 396
<i>JIV 688</i>	JIV 688
<i>CPI 159</i>	CPI 159
<i>AVI 043</i>	AVI 043
<i>IHX 269</i>	IWX 269
<i>AVB 650</i>	AVB 650
<i>IHZ 897</i>	IHZ 897
<i>IIL 592</i>	IIL 592
<i>AUW 327</i>	AUW 327

<i>SVR 086</i>	SVR 086
<i>IHX 572</i>	IHX 572
<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	CPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IJK 396</i>	IJK 396
<i>AUR 919</i>	AUR 419
<i>IJK 066</i>	IJK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871
<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311
<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454

<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053
<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLV 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MWX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AVD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140
<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	IVI 371
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IDL 488
<i>IDL 488</i>	IDL 488
<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656

<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141
<i>HZV 807</i>	HZV 807
<i>GRN 103</i>	GRN 103
<i>RAV 850</i>	RAV 850
<i>COF 597</i>	COF 597
<i>COF 597</i>	COF 597
<i>HMR 910</i>	HMR 910
<i>IVZ 391</i>	IVZ 391
<i>HMR 910</i>	HMR 910
<i>RHN 653</i>	RHN 653
<i>AUX 577</i>	AUX 577
<i>RHN 653</i>	RHN 653
<i>AVK 790</i>	AVK 790
<i>AVG 144</i>	AVG 144
<i>AVA 495</i>	AVA 495
<i>FGY 852</i>	FEY 852
<i>DVY 988</i>	DVY 988
<i>MGX 359</i>	MGX 359
<i>DBL 326</i>	DBL 326
<i>IIL 726</i>	IIL 726
<i>FAQ 531</i>	FAQ 531
<i>NPD 472</i>	NPD 472
<i>AVA 426</i>	AVA 426
<i>HZV 807</i>	HZV 807
<i>HZV 631</i>	HZV 631
<i>AUP 979</i>	AUP 979
<i>ISV 689</i>	ISV 689
<i>AUX 215</i>	AUX 215
<i>DTK 343</i>	UTK 343
<i>AUZ 679</i>	AUZ 679
<i>QEP 541</i>	QEP 541
<i>MOO 907</i>	MOO 907
<i>BCK 002</i>	BCK 002
<i>FUM 378</i>	FUM 378
<i>AVD 092</i>	AVW 092
<i>AVH 811</i>	AVH 811
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915
<i>PEG 915</i>	PEG 915

<i>DTK 343</i>	<i>OTK 343</i>
<i>CDX 857</i>	<i>CDX 857</i>
<i>CQG 666</i>	<i>EQG 666</i>
<i>AVE 928</i>	<i>AVE 928</i>
<i>IIL 737</i>	<i>IIL 737</i>
<i>AUX 086</i>	<i>AUX 086</i>
<i>KDU 868</i>	<i>KDU 868</i>
<i>KLV 110</i>	<i>KLV 110</i>
<i>RKM 195</i>	<i>RKM 195</i>
<i>RKM 195</i>	<i>RKM 195</i>
<i>BYO 101</i>	<i>BYO 101</i>
<i>IIN 804</i>	<i>IIN 804</i>
<i>CVQ 795</i>	<i>CVQ 795</i>
<i>AVD 423</i>	<i>AVD 423</i>
<i>HFN 803</i>	<i>HFN 803</i>
<i>CQG 666</i>	<i>EQG 666</i>
<i>AVK 544</i>	<i>AVK 544</i>
<i>DLV 101</i>	<i>DLV 101</i>
<i>GRU 302</i>	<i>GRU 302</i>
<i>BBQ 727</i>	<i>BBQ 727</i>
<i>BPJ 644</i>	<i>BPJ 644</i>
<i>AVC 227</i>	<i>AVC 227</i>
<i>AUT 673</i>	<i>AUT 673</i>
<i>AVF 870</i>	<i>AVF 870</i>
<i>BYE 791</i>	<i>BYE 791</i>
<i>AVG 965</i>	<i>AVG 965</i>
<i>AUW 723</i>	<i>AUW 723</i>
<i>GRU 254</i>	<i>GRU 254</i>
<i>DHV 151</i>	<i>DHV 151</i>
<i>UBU 549</i>	<i>UBU 549</i>
<i>IHW 983</i>	<i>IHW 983</i>
<i>DQK 705</i>	<i>DQK 705</i>
<i>AUP 821</i>	<i>WUY 061</i>
<i>IIK 439</i>	<i>IIK 439</i>
<i>AUS 121</i>	<i>AOS 121</i>
<i>AUX 044</i>	<i>AUX 044</i>
<i>AUY 677</i>	<i>AUY 677</i>
<i>NEN 129</i>	<i>NEN 129</i>
<i>GTY 980</i>	<i>GTY 980</i>
<i>AVH 247</i>	<i>AVH 247</i>

<i>HRM 975</i>	HRM 975
<i>IIN 804</i>	IIN 804
<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106





### Anexo 19. Predicción de los caracteres utilizando el descriptor LBP y el clasificador SVM

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	NPW 472
AVH 247	GYH 647
GGP 719	GGP 719
FUY 368	LPT 094
IMW 908	IMW 608
AVF 687	AYZ 687
AVJ 529	AYJ 526
CMP 655	CHP 635
CMD 846	TXR 847
ENX 310	KWX 774
DLU 976	SLP 977
AUR 541	AQN 541
GCZ 106	GTZ 746
GDQ 411	YPI 777
IHZ 896	IHX 477
BFK 596	GJX 072
AUY 075	ADT 473
MSW 538	MCW 558
DTP 518	UTP 210
BBO 289	GBO 286
AUY 298	AUT 268
IHZ 132	IHZ 122
ENX 497	EHK 797
FJK 598	KJK 374
AVK 755	AYK 755
CFT 762	TLT 762
GDQ 411	CUQ 412
BKG 926	SXT 977
DLV 485	DLT 485
GDQ 244	GDQ 244
IAM 555	IVM 351
MFT 694	MFT 964
BXW 351	BXW 351
LUA 808	IEK 000

CPZ 236	TPZ 229
AUS 753	YUC 736
AVI 111	ATI 111
KIN 864	XIX 494
FGR 368	KGR 774
AUV 230	APY 658
AUY 900	AUT 600
DTK 396	DTK 299
RKM 195	WWM 165
CWB 710	TWB 764
NEG 741	MEG 701
MFT 694	MLT 664
GUL 746	GUL 746
HHM 205	HHW 385
UDQ 619	DDG 716
CAP 875	CAP 872
GUL 328	BDL 528
INP 888	IHP 444
AVH 903	AYH 683
BSQ 392	QCQ 263
CFT 762	CLT 762
GDQ 254	CDQ 254
UDN 328	UDH 224
AUW 327	AQB 017
EGX 687	ECX 607
AVK 755	AYK 755
IHX 572	XNK 572
DRW 868	DNH 060
FOW 279	ZOW 276
SVS 025	CYC 026
HZL 298	HKL 670
IJK 396	KKK 277
JIV 688	JIV 644
CPI 159	TYX 757
AVI 043	AAI 441
IHX 269	IHX 299
AVB 650	AYB 650
IHZ 897	IHZ 867
IIL 592	IIL 592
AUW 327	YXB 327

SVR 086	TTD 446
IHX 572	IHX 572
CCZ 048	TTK 474
MSM 839	WTM 427
IHX 267	IHX 297
FJQ 884	ZJQ 444
EIY 393	KXT 273
AUW 316	ARB 516
EPR 253	EPR 253
DOF 044	POI 444
AVH 818	AYH 414
AVG 842	AYG 842
NBX 683	HQK 282
IHX 296	IHX 269
CPP 844	JSJ 444
DLV 101	DJY 102
RMZ 081	RMZ 081
NMC 962	WWT 662
MAY 571	MVT 271
NMC 962	HMT 962
AVJ 917	AYJ 917
DLX 266	WLX 266
DLX 266	DLX 266
AUW 762	ADB 792
AUP 638	XSP 768
AUW 762	YDB 762
IJK 396	KKX 777
AUR 919	APX 719
IJK 066	ZZX 066
DLV 101	DLV 206
KHM 100	KHM100
KIK 871	KIK 871
ISV 609	IYY 946
QEQ 022	QHC 722
JCO 304	JTO 347
MTX 575	MTX 575
KDS 131	ZDC 712
CEW 311	TEB 677
CEW 311	TRB 217
AUY 454	ADT 494

PFG 656	PLG 656
RBL 053	RBL 859
AUZ 630	ADZ 628
AVF 585	AAP 583
DLV 293	ULV 263
GDP 273	GDP 273
MIX 908	WTX 944
BOE 646	QOK 646
AVF 585	AVL 585
HKT 997	KXT 977
AVG 045	AYG 442
HBZ 691	HWZ 961
IIL 416	ZKL 719
RCK 868	RYK 898
IIL 416	KKL 719
QEO 574	QEO 574
AVI 807	AVI 847
GDP 273	GDP 273
CZQ 846	TZG 449
HFO 614	HZO 914
AVD 423	AYR 423
AVD 411	ATD 401
HNS 084	HHS 084
DCB 797	DTB 797
IIL 856	KKL 466
IIL 856	KKL 439
AVJ 517	AYJ 517
USV 045	UAV 045
KLV 110	KLT 774
NVG 005	HWC 005
AVF 811	AAF 811
WNV 140	WHY 140
DLV 286	DEY 286
EMR 275	EMR 273
IVZ 391	LLL 777
EMR 275	EMR 273
IDL 488	IDL 488
IDL 488	IDL 488
IHZ 504	IHZ 204
PFG 656	CLY 792

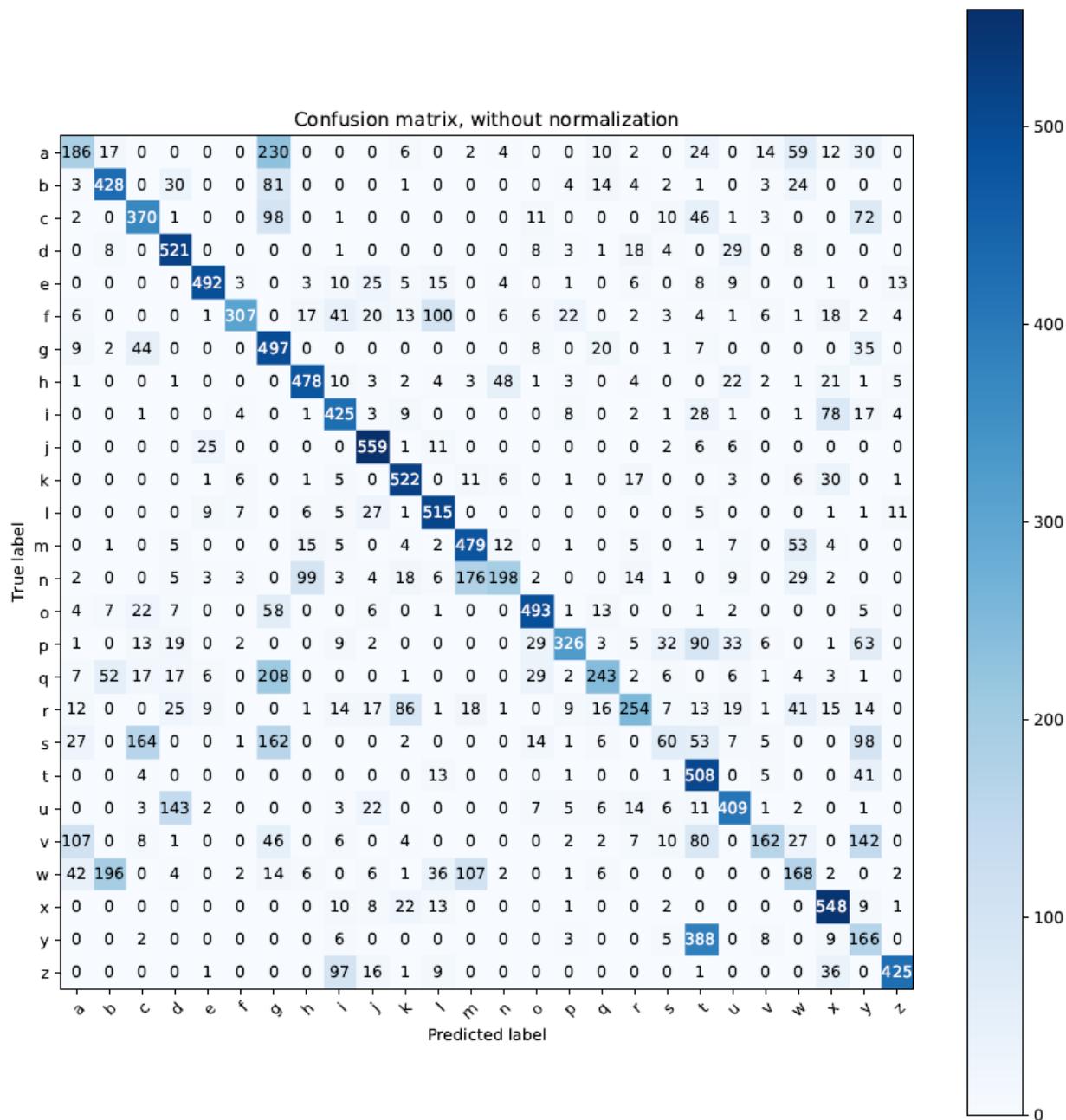
ENX 439	EHX 426
CLU 141	ALD 141
HZV 807	HZT 847
GRN 103	GDH 143
RAV 850	KYT 464
COF 597	TOZ 597
COF 597	TOF 567
HMR 910	HMW 694
IVZ 391	KTZ 267
HMR 910	HMR 620
RHN 653	UHH 653
AUX 577	ARX 577
RHN 653	RXX 622
AVK 790	AAK 760
AVG 144	ATG 144
AVA 495	AYG 496
FGY 852	LTT 777
DVY 988	DTT 600
MGX 359	MGX 236
DBL 326	DBL 226
IIL 726	KKL 726
FAQ 531	LVC 531
NPD 472	QPS 479
AVA 426	ATA 429
HZV 807	HZT 807
HZV 631	HZY 631
AUP 979	AUR 676
ISV 689	KYY 986
AUX 215	AUX 213
DTK 343	UJX 272
AUZ 679	ADZ 679
QEP 541	QEP 541
MOO 907	MOO 747
BCK 002	QTX 442
FUM 378	ZUM 274
AVD 092	AAP 892
AVH 811	ATH 422
PEG 915	IEC 913
PEG 915	CEY 772
PEG 915	PEG 916

<i>DTK 343</i>	<i>UTK 343</i>
<i>CDX 857</i>	<i>CDX 857</i>
<i>CQG 666</i>	<i>TSC 766</i>
<i>AVE 928</i>	<i>AYK 628</i>
<i>IIL 737</i>	<i>KIL 727</i>
<i>AUX 086</i>	<i>AUX 086</i>
<i>KDU 868</i>	<i>KUU 494</i>
<i>KLV 110</i>	<i>XLT 774</i>
<i>RKM 195</i>	<i>YKW 769</i>
<i>RKM 195</i>	<i>RXX 771</i>
<i>BYO 101</i>	<i>BTO 101</i>
<i>IIN 804</i>	<i>IHH 804</i>
<i>CVQ 795</i>	<i>CYQ 765</i>
<i>AVD 423</i>	<i>AVD 422</i>
<i>HFN 803</i>	<i>HLH 402</i>
<i>CQG 666</i>	<i>CCY 999</i>
<i>AVK 544</i>	<i>AAK 544</i>
<i>DLV 101</i>	<i>DLY 141</i>
<i>GRU 302</i>	<i>CRU 242</i>
<i>BBQ 727</i>	<i>BBG 727</i>
<i>BPJ 644</i>	<i>SCJ 777</i>
<i>AVC 227</i>	<i>YTC 227</i>
<i>AUT 673</i>	<i>ADT 972</i>
<i>AVF 870</i>	<i>AYL 878</i>
<i>BYE 791</i>	<i>BTZ 767</i>
<i>AVG 965</i>	<i>AYC 665</i>
<i>AUW 723</i>	<i>AUW 725</i>
<i>GRU 254</i>	<i>GRU 224</i>
<i>DHV 151</i>	<i>DHT 127</i>
<i>UBU 549</i>	<i>URU 342</i>
<i>IHW 983</i>	<i>HLM 681</i>
<i>DQK 705</i>	<i>RGK 743</i>
<i>AUP 821</i>	<i>XUL 511</i>
<i>IHK 439</i>	<i>KKK 427</i>
<i>AUS 121</i>	<i>ACT 722</i>
<i>AUX 044</i>	<i>AUX 044</i>
<i>AUY 677</i>	<i>TST 977</i>
<i>NEN 129</i>	<i>HEH 729</i>
<i>GTY 980</i>	<i>CTT 944</i>
<i>AVH 247</i>	<i>GYH 647</i>

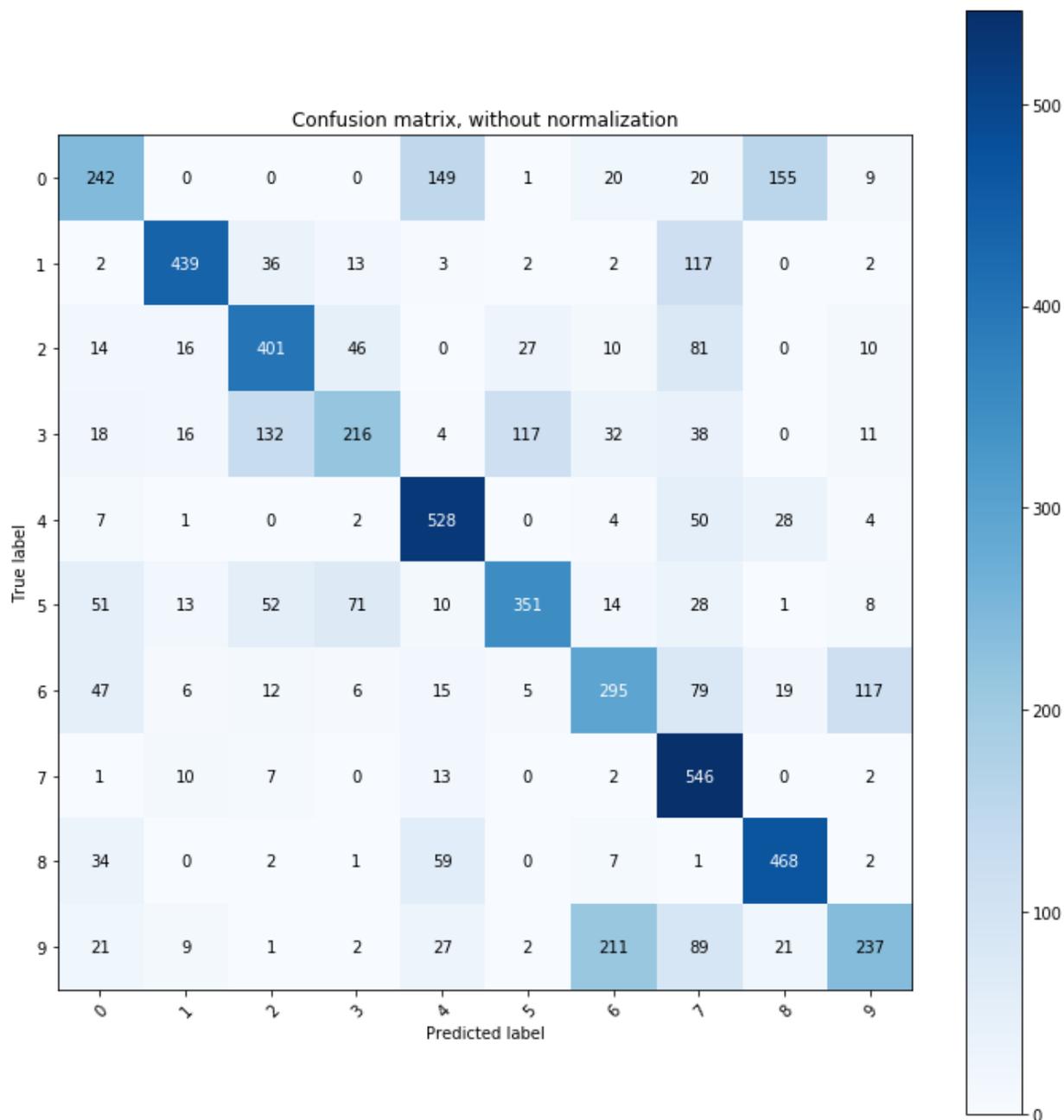
<i>HRM 975</i>	<i>PRN 675</i>
<i>IIN 804</i>	<i>IIN 844</i>
<i>DBU 436</i>	<i>QGP 729</i>
<i>AUS 801</i>	<i>ARY 887</i>
<i>GDP 664</i>	<i>CDJ 664</i>
<i>QFZ 711</i>	<i>QJZ 711</i>
<i>IIN 094</i>	<i>ITH 067</i>
<i>DBV 375</i>	<i>DBT 270</i>
<i>ASH 122</i>	<i>TTI 777</i>
<i>GRU 254</i>	<i>CNU 224</i>
<i>KGL 532</i>	<i>XYJ 367</i>
<i>GUN 608</i>	<i>CDH 604</i>
<i>EIM 428</i>	<i>EIM 424</i>
<i>MGX 359</i>	<i>MGX 556</i>
<i>AUS 801</i>	<i>ARY 887</i>
<i>AVJ 106</i>	<i>TYY 107</i>

### Anexo 20. Matriz de confusión para las letras utilizando el descriptor LBP y el clasificador

#### SVM



**Anexo 21. Matriz de confusión para los números utilizando el descriptor LBP y el clasificador SVM**



## Anexo 22. Predicción de los caracteres utilizando el descriptor LBP y el clasificador KNN

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KIM 472
AVH 247	GVE 147
GGP 719	SSP 716
FUY 368	FVY 594
IMW 908	ZNW 250
AVF 687	ACF 987
AVJ 529	ACJ 636
CMP 655	CHP 933
CMD 846	CWD 041
ENX 310	ZKF 174
DLU 976	PLP 979
AUR 541	ASH 692
GCZ 106	GVZ 146
GDQ 411	CPC 417
IHZ 896	FHZ 469
BFK 596	QJK 262
AUY 075	ART 472
MSW 538	NCW 228
DTP 518	UTI 110
BBO 289	GBC 289
AUY 298	VUY 264
IHZ 132	IHZ 122
ENX 497	ENX 467
FJK 598	LLE 260
AVK 755	VVK 725
CFT 762	TLU 662
GDQ 411	CPP 422
BKG 926	GJP 613
DLV 485	DLY 403
GDQ 244	SRC 244
IAM 555	ZVN 226
MFT 694	NFT 694

<i>BXW 351</i>	<i>WXW 227</i>
<i>LUA 808</i>	<i>FEH 030</i>
<i>CPZ 236</i>	<i>TIZ 229</i>
<i>AUS 753</i>	<i>CFC 723</i>
<i>AVI 111</i>	<i>VVI 111</i>
<i>KIN 864</i>	<i>XZR 464</i>
<i>FGR 368</i>	<i>ZAK 178</i>
<i>AUV 230</i>	<i>APV 360</i>
<i>AUY 900</i>	<i>RUT 206</i>
<i>DTK 396</i>	<i>UTK 266</i>
<i>RKM 195</i>	<i>RKM 153</i>
<i>CWB 710</i>	<i>CWA 714</i>
<i>NEG 741</i>	<i>NEG 166</i>
<i>MFT 694</i>	<i>MFT 664</i>
<i>GUL 746</i>	<i>GUJ 149</i>
<i>HHM 205</i>	<i>HEM 203</i>
<i>UDQ 619</i>	<i>DDA 719</i>
<i>CAP 875</i>	<i>SAP 873</i>
<i>GUL 328</i>	<i>GDL 328</i>
<i>INP 888</i>	<i>FHP 444</i>
<i>AVH 903</i>	<i>AVH 683</i>
<i>BSQ 392</i>	<i>DPO 163</i>
<i>CFT 762</i>	<i>TLT 762</i>
<i>GDQ 254</i>	<i>SHB 224</i>
<i>UDN 328</i>	<i>URN 324</i>
<i>AUW 327</i>	<i>VPB 627</i>
<i>EGX 687</i>	<i>ZCX 101</i>
<i>AVK 755</i>	<i>GVK 752</i>
<i>IHX 572</i>	<i>FKX 572</i>
<i>DRW 868</i>	<i>UEF 020</i>
<i>FOW 279</i>	<i>FOW 276</i>
<i>SVS 025</i>	<i>PVC 023</i>
<i>HZL 298</i>	<i>HZL 170</i>
<i>IJK 396</i>	<i>FZK 237</i>
<i>JIV 688</i>	<i>FIV 680</i>
<i>CPI 159</i>	<i>SSF 769</i>
<i>AVI 043</i>	<i>RVP 442</i>
<i>IHX 269</i>	<i>LEX 266</i>
<i>AVB 650</i>	<i>VSQ 660</i>
<i>IHZ 897</i>	<i>IHZ 867</i>

<i>IIL 592</i>	<i>FFL 392</i>
<i>AUW 327</i>	<i>SRB 226</i>
<i>SVR 086</i>	<i>CYK 486</i>
<i>IHX 572</i>	<i>ZHX 572</i>
<i>CCZ 048</i>	<i>TYL 444</i>
<i>MSM 839</i>	<i>MVN 427</i>
<i>IHX 267</i>	<i>IHX 297</i>
<i>FJQ 884</i>	<i>FJA 004</i>
<i>EIY 393</i>	<i>EZK 212</i>
<i>AUW 316</i>	<i>SDW 616</i>
<i>EPR 253</i>	<i>EPR 151</i>
<i>DOF 044</i>	<i>UPL 044</i>
<i>AVH 818</i>	<i>AYH 418</i>
<i>AVG 842</i>	<i>SCG 042</i>
<i>NBX 683</i>	<i>NAX 385</i>
<i>IHX 296</i>	<i>EHX 166</i>
<i>CPP 844</i>	<i>JTI 844</i>
<i>DLV 101</i>	<i>UEV 106</i>
<i>RMZ 081</i>	<i>RMZ 001</i>
<i>NMC 962</i>	<i>RMV 622</i>
<i>MAY 571</i>	<i>NVT 271</i>
<i>NMC 962</i>	<i>NMY 962</i>
<i>AVJ 917</i>	<i>AAJ 917</i>
<i>DLX 266</i>	<i>DJX 266</i>
<i>DLX 266</i>	<i>ULX 296</i>
<i>AUW 762</i>	<i>SDW 762</i>
<i>AUP 638</i>	<i>PPP 759</i>
<i>AUW 762</i>	<i>VDW 162</i>
<i>IHK 396</i>	<i>FZK 120</i>
<i>AUR 919</i>	<i>VPR 726</i>
<i>IHK 066</i>	<i>ZZH 563</i>
<i>DLV 101</i>	<i>DLC 206</i>
<i>KHM 100</i>	<i>KHM100</i>
<i>KIK 871</i>	<i>KIK 871</i>
<i>ISV 609</i>	<i>FSC 949</i>
<i>QEQ 022</i>	<i>SLY 412</i>
<i>JCO 304</i>	<i>JPO 304</i>
<i>MTX 575</i>	<i>MTX 613</i>
<i>KDS 131</i>	<i>EDC 721</i>
<i>CEW 311</i>	<i>CEW 277</i>

CEW 311	CEW 216
AUY 454	SUT 434
PFG 656	PFG 626
RBL 053	RBL 053
AUZ 630	AHZ 620
AVF 585	AVF 582
DLV 293	DLV 263
GDP 273	GDP 276
MIX 908	MIX 644
BOE 646	QOE 646
AVF 585	SCF 385
HKT 997	EKT 617
AVG 045	VCS 043
HBZ 691	HAZ 161
IIL 416	EEL 716
RCK 868	PCK 898
IIL 416	ZEL 769
QEO 574	BEO 574
AVI 807	AYI 847
GDP 273	GDP 273
CZQ 846	YZG 406
HFO 614	HFO 614
AVD 423	VYU 422
AVD 411	QVD 001
HNS 084	HNC 084
DCB 797	DVB 767
IIL 856	EZL 032
IIL 856	EZL 026
AVJ 517	AVJ 317
USV 045	RVR 045
KLV 110	KLA 124
NVG 005	HRS 633
AVF 811	VVF 811
WNV 140	WNV 140
DLV 286	DZV 289
EMR 275	ZNK 272
IVZ 391	JTL 076
EMR 275	ZNR 773
IDL 488	IDL 488
IDL 488	IUL 488

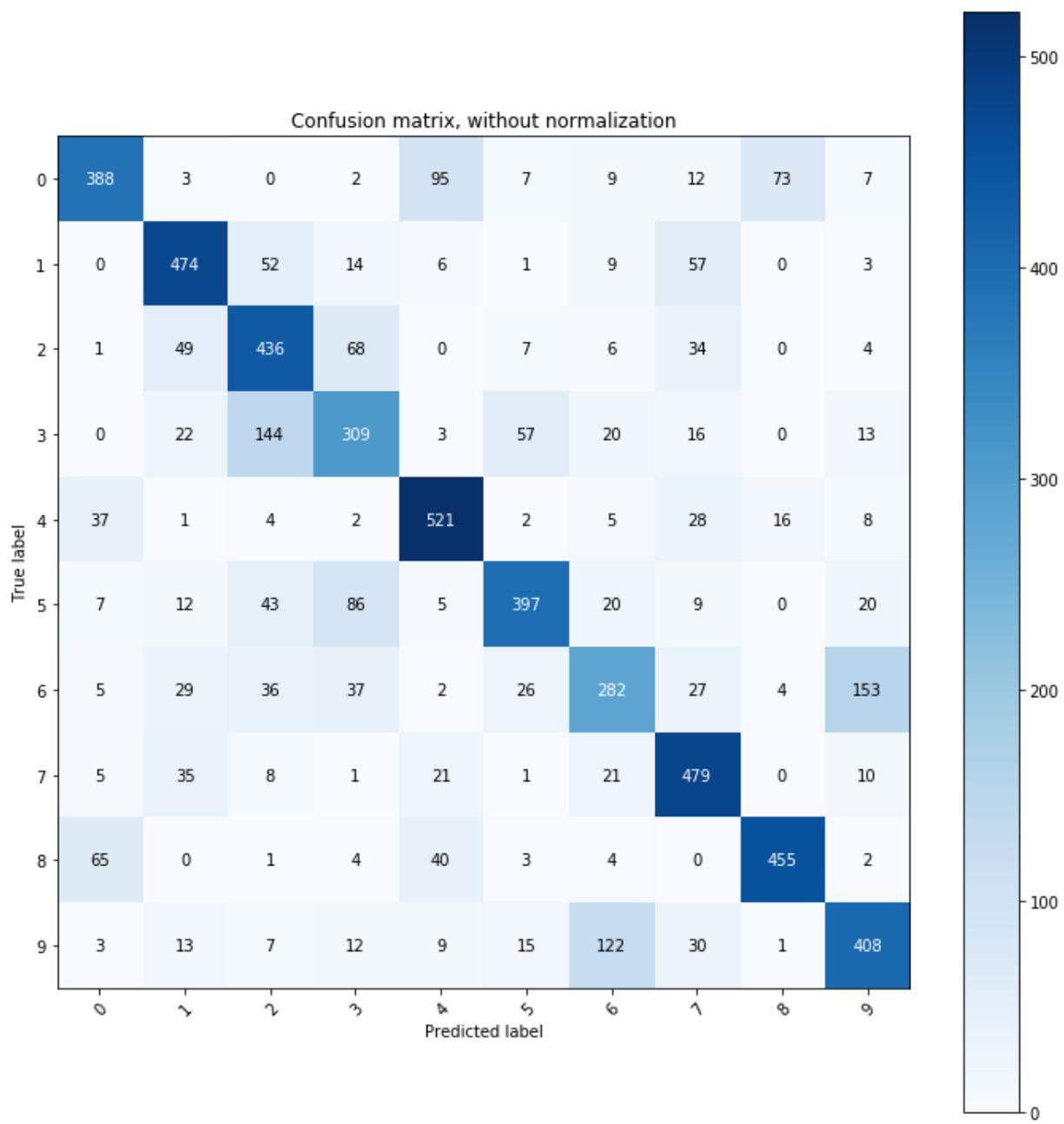
<i>IHZ 504</i>	<i>ZEL 204</i>
<i>PFG 656</i>	<i>PJS 771</i>
<i>ENX 439</i>	<i>WNX 422</i>
<i>CLU 141</i>	<i>ALD 901</i>
<i>HZV 807</i>	<i>HZV 847</i>
<i>GRN 103</i>	<i>GKH 142</i>
<i>RAV 850</i>	<i>RCY 024</i>
<i>COF 597</i>	<i>COZ 997</i>
<i>COF 597</i>	<i>GOF 567</i>
<i>HMR 910</i>	<i>HNK 610</i>
<i>IVZ 391</i>	<i>ZYL 723</i>
<i>HMR 910</i>	<i>HNK 610</i>
<i>RHN 653</i>	<i>UHN 322</i>
<i>AUX 577</i>	<i>VPX 671</i>
<i>RHN 653</i>	<i>REK 222</i>
<i>AVK 790</i>	<i>AVK 768</i>
<i>AVG 144</i>	<i>SCA 144</i>
<i>AVA 495</i>	<i>GCG 466</i>
<i>FGY 852</i>	<i>LTR 771</i>
<i>DVY 988</i>	<i>PPI 600</i>
<i>MGX 359</i>	<i>MGX 236</i>
<i>DBL 326</i>	<i>DBL 329</i>
<i>IIL 726</i>	<i>EEL 276</i>
<i>FAQ 531</i>	<i>LVQ 333</i>
<i>NPD 472</i>	<i>BPQ 479</i>
<i>AVA 426</i>	<i>VYQ 426</i>
<i>HZV 807</i>	<i>HZC 807</i>
<i>HZV 631</i>	<i>HJY 632</i>
<i>AUP 979</i>	<i>SUD 976</i>
<i>ISV 689</i>	<i>IVC 986</i>
<i>AUX 215</i>	<i>VUX 212</i>
<i>DTK 343</i>	<i>EEE 772</i>
<i>AUZ 679</i>	<i>SUZ 979</i>
<i>QEP 541</i>	<i>QEC 301</i>
<i>MOO 907</i>	<i>NUO 647</i>
<i>BCK 002</i>	<i>BYL 442</i>
<i>FUM 378</i>	<i>FUN 170</i>
<i>AVD 092</i>	<i>GVU 063</i>
<i>AVH 811</i>	<i>GCH 412</i>
<i>PEG 915</i>	<i>PZC 623</i>

PEG 915	PZC 371
PEG 915	PZA 611
DTK 343	UPE 641
CDX 857	PUX 057
CQG 666	TCP 666
AVE 928	AVE 928
IIL 737	FLL 725
AUX 086	SUX 006
KDU 868	KPP 060
KLV 110	XLS 220
RKM 195	TKR 769
RKM 195	KKN 761
BYO 101	QTO 107
IIN 804	IIN 804
CVQ 795	PVQ 795
AVD 423	SVD 423
HFN 803	ELH 402
CQG 666	YCV 666
AVK 544	VVK 544
DLV 101	PLY 241
GRU 302	CFJ 252
BBQ 727	BBA 737
BPJ 644	VPE 146
AVC 227	VVP 227
AUT 673	ART 973
AVF 870	VVF 870
BYE 791	QTZ 762
AVG 965	AYS 665
AUW 723	AKW 722
GRU 254	SKU 224
DHV 151	DHY 111
UBU 549	FPE 247
IHW 983	WEN 683
DQK 705	PGX 746
AUP 821	HUL 579
IIK 439	ZZK 422
AUS 121	SPC 121
AUX 044	SUX 000
AUY 677	TPY 777
NEN 129	HZH 126

<i>GTY 980</i>	<i>VTP 944</i>
<i>AVH 247</i>	<i>GVH 147</i>
<i>HRM 975</i>	<i>HRN 675</i>
<i>IIN 804</i>	<i>IIN 044</i>
<i>DBU 436</i>	<i>QAP 419</i>
<i>AUS 801</i>	<i>VDC 807</i>
<i>GDP 664</i>	<i>CRF 614</i>
<i>QFZ 711</i>	<i>SJZ 711</i>
<i>IIN 094</i>	<i>FFH 067</i>
<i>DBV 375</i>	<i>DAY 271</i>
<i>ASH 122</i>	<i>YVL 777</i>
<i>GRU 254</i>	<i>CKU 224</i>
<i>KGL 532</i>	<i>KVE 121</i>
<i>GUN 608</i>	<i>VUH 604</i>
<i>EIM 428</i>	<i>EIM 420</i>
<i>MGX 359</i>	<i>MGX 266</i>
<i>AUS 801</i>	<i>VUC 887</i>
<i>AVJ 106</i>	<i>SYP 207</i>



**Anexo 24. Matriz de confusión para los números utilizando el descriptor LBP y el clasificador KNN**



## Anexo 25. Predicción de los caracteres utilizando el descriptor LBP y el clasificador

### Random forest

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
KUM 876	KDN 476
AVH 247	GVH 277
GGP 719	SSP 729
FUY 368	FPY 594
IMW 908	ZMW 600
AVF 687	AVF 681
AVJ 529	SCJ 526
CMP 655	SHP 955
CMD 846	CWB 846
ENX 310	KKX 274
DLU 976	PLP 979
AUR 541	ADK 541
GCZ 106	QSZ 146
GDQ 411	VPQ 411
IHZ 896	IHZ 469
BFK 596	QXK 696
AUY 075	VDT 973
MSW 538	NCW 338
DTP 518	DTP 210
BBO 289	GBO 289
AUY 298	VUY 264
IHZ 132	IHZ 122
ENX 497	EHX 467
FJK 598	LJK 360
AVK 755	VVK 755
CFT 762	TIT 761
GDQ 411	CPQ 411
BKG 926	AXY 626
DLV 485	DLY 482
GDQ 244	GDQ 244

<i>IAM 555</i>	<i>IVM 321</i>
<i>MFT 694</i>	<i>MFT 664</i>
<i>BXW 351</i>	<i>BXW 351</i>
<i>LUA 808</i>	<i>FEK 000</i>
<i>CPZ 236</i>	<i>TPZ 226</i>
<i>AUS 753</i>	<i>YDC 732</i>
<i>AVI 111</i>	<i>VVI 111</i>
<i>KIN 864</i>	<i>III 494</i>
<i>FGR 368</i>	<i>LGR 298</i>
<i>AUV 230</i>	<i>VRV 650</i>
<i>AUY 900</i>	<i>RDT 600</i>
<i>DTK 396</i>	<i>DTK 266</i>
<i>RKM 195</i>	<i>RKM 163</i>
<i>CWB 710</i>	<i>VWB 414</i>
<i>NEG 741</i>	<i>NEO 291</i>
<i>MFT 694</i>	<i>MLT 664</i>
<i>GUL 746</i>	<i>QUL 749</i>
<i>HHM 205</i>	<i>HHM 105</i>
<i>UDQ 619</i>	<i>UDG 719</i>
<i>CAP 875</i>	<i>CCP 872</i>
<i>GUL 328</i>	<i>QUF 528</i>
<i>INP 888</i>	<i>IHP 444</i>
<i>AVH 903</i>	<i>AVH 603</i>
<i>BSQ 392</i>	<i>BCQ 363</i>
<i>CFT 762</i>	<i>CLT 762</i>
<i>GDQ 254</i>	<i>QDB 354</i>
<i>UDN 328</i>	<i>UDN 224</i>
<i>AUW 327</i>	<i>ARB 917</i>
<i>EGX 687</i>	<i>ESX 601</i>
<i>AVK 755</i>	<i>VVK 755</i>
<i>IHX 572</i>	<i>IHX 572</i>
<i>DRW 868</i>	<i>DEF 060</i>
<i>FOW 279</i>	<i>FOW 276</i>
<i>SVS 025</i>	<i>CVC 025</i>
<i>HZL 298</i>	<i>HZL 190</i>
<i>IIK 396</i>	<i>IIK 199</i>
<i>JIV 688</i>	<i>JIV 600</i>
<i>CPI 159</i>	<i>SPI 716</i>
<i>AVI 043</i>	<i>AVI 441</i>
<i>IHX 269</i>	<i>IHX 266</i>

AVB 650	VVB 650
IHZ 897	IHZ 897
IIL 592	IIL 391
AUW 327	VRW 326
SVR 086	CYR 406
IHX 572	IHX 572
CCZ 048	TYZ 460
MSM 839	MVM 436
IHX 267	IHX 297
FJQ 884	LJQ 404
EIY 393	EZX 291
AUW 316	VDW 516
EPR 253	EPR 233
DOF 044	BOI 044
AVH 818	VYH 410
AVG 842	VVG 042
NBX 683	MQX 902
IHX 296	IHX 296
CPP 844	PJP 844
DLV 101	ULV 101
RMZ 081	RMZ 001
NMC 962	KMV 662
MAY 571	NVT 271
NMC 962	NMY 962
AVJ 917	AVJ 917
DLX 266	DLX 266
DLX 266	DLX 266
AUW 762	VUW 762
AUP 638	UBP 629
AUW 762	VUW 762
IJK 396	IJK 199
AUR 919	CBR 719
IJK 066	IZX 066
DLV 101	DLV 101
KHM 100	KHM100
KIK 871	KIK 871
ISV 609	ISV 949
QEQ 022	SFP 496
JCO 304	PPO 304
MTX 575	MTX 575

<i>KDS 131</i>	<i>ZDC 711</i>
<i>CEW 311</i>	<i>YRW 217</i>
<i>CEW 311</i>	<i>CEW 211</i>
<i>AUY 454</i>	<i>VUT 434</i>
<i>PFG 656</i>	<i>PFG 626</i>
<i>RBL 053</i>	<i>RBL 033</i>
<i>AUZ 630</i>	<i>AUZ 910</i>
<i>AVF 585</i>	<i>SVF 585</i>
<i>DLV 293</i>	<i>DFV 261</i>
<i>GDP 273</i>	<i>GDP 276</i>
<i>MIX 908</i>	<i>MIX 644</i>
<i>BOE 646</i>	<i>AOE 646</i>
<i>AVF 585</i>	<i>VVF 585</i>
<i>HKT 997</i>	<i>KKT 667</i>
<i>AVG 045</i>	<i>VVG 042</i>
<i>HBZ 691</i>	<i>HBZ 661</i>
<i>IIL 416</i>	<i>IIL 716</i>
<i>RCK 868</i>	<i>RSK 898</i>
<i>IIL 416</i>	<i>IIL 716</i>
<i>QEO 574</i>	<i>BEO 578</i>
<i>AVI 807</i>	<i>VVI 807</i>
<i>GDP 273</i>	<i>GDP 273</i>
<i>CZQ 846</i>	<i>CZG 449</i>
<i>HFO 614</i>	<i>HFO 914</i>
<i>AVD 423</i>	<i>VVD 411</i>
<i>AVD 411</i>	<i>AVD 001</i>
<i>HNS 084</i>	<i>HNC 004</i>
<i>DCB 797</i>	<i>DVB 767</i>
<i>IIL 856</i>	<i>IIL 036</i>
<i>IIL 856</i>	<i>IIL 036</i>
<i>AVJ 517</i>	<i>AYJ 517</i>
<i>USV 045</i>	<i>DSV 055</i>
<i>KLV 110</i>	<i>KLV 114</i>
<i>NVG 005</i>	<i>NVS 003</i>
<i>AVF 811</i>	<i>VVF 811</i>
<i>WNV 140</i>	<i>WNY 140</i>
<i>DLV 286</i>	<i>DLV 289</i>
<i>EMR 275</i>	<i>EMR 273</i>
<i>IVZ 391</i>	<i>TTX 167</i>
<i>EMR 275</i>	<i>EMR 275</i>

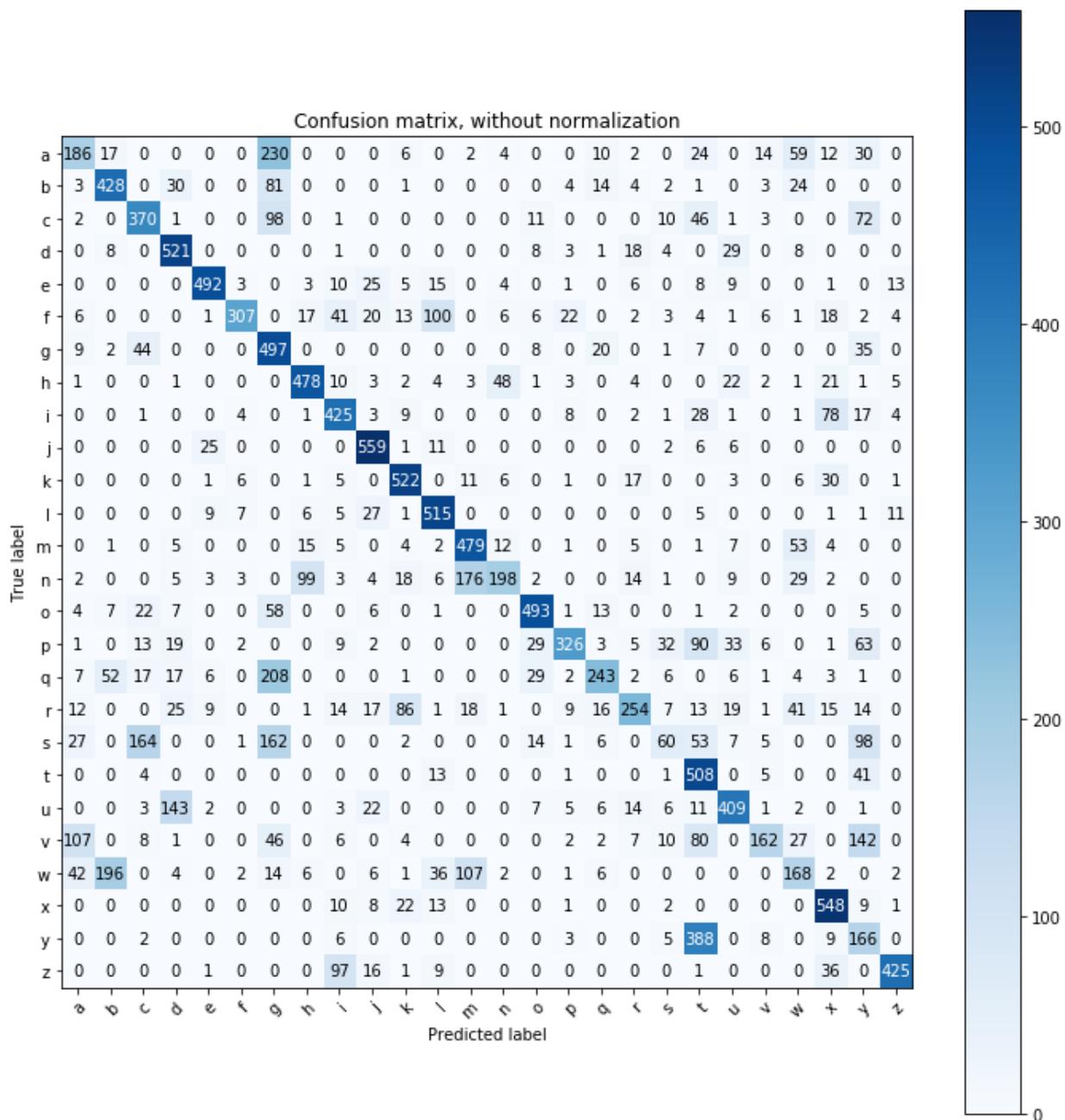
<i>IDL 488</i>	<i>IDF 488</i>
<i>IDL 488</i>	<i>IDL 488</i>
<i>IHZ 504</i>	<i>IHZ 304</i>
<i>PFG 656</i>	<i>PFV 936</i>
<i>ENX 439</i>	<i>EHX 426</i>
<i>CLU 141</i>	<i>SED 171</i>
<i>HZV 807</i>	<i>HZV 847</i>
<i>GRN 103</i>	<i>SRH 143</i>
<i>RAV 850</i>	<i>RVV 028</i>
<i>COF 597</i>	<i>COF 597</i>
<i>COF 597</i>	<i>COF 597</i>
<i>HMR 910</i>	<i>HMR 610</i>
<i>IVZ 391</i>	<i>ITZ 361</i>
<i>HMR 910</i>	<i>HMR 610</i>
<i>RHN 653</i>	<i>RHN 623</i>
<i>AUX 577</i>	<i>APX 571</i>
<i>RHN 653</i>	<i>RHH 622</i>
<i>AVK 790</i>	<i>AVK 790</i>
<i>AVG 144</i>	<i>AVA 149</i>
<i>AVA 495</i>	<i>VCG 499</i>
<i>FGY 852</i>	<i>XTU 792</i>
<i>DVY 988</i>	<i>UTP 600</i>
<i>MGX 359</i>	<i>MGX 256</i>
<i>DBL 326</i>	<i>DBL 329</i>
<i>IIL 726</i>	<i>IIL 726</i>
<i>FAQ 531</i>	<i>LVQ 531</i>
<i>NPD 472</i>	<i>BPQ 479</i>
<i>AVA 426</i>	<i>VYV 426</i>
<i>HZV 807</i>	<i>HZV 007</i>
<i>HZV 631</i>	<i>HZY 631</i>
<i>AUP 979</i>	<i>ADU 976</i>
<i>ISV 689</i>	<i>ICV 686</i>
<i>AUX 215</i>	<i>VUX 213</i>
<i>DTK 343</i>	<i>DYX 372</i>
<i>AUZ 679</i>	<i>VUZ 979</i>
<i>QEP 541</i>	<i>QEC 301</i>
<i>MOO 907</i>	<i>NOO 647</i>
<i>BCK 002</i>	<i>BYK 402</i>
<i>FUM 378</i>	<i>FUM 270</i>
<i>AVD 092</i>	<i>VVP 062</i>

AVH 811	VYH 411
PEG 915	PEC 913
PEG 915	PES 672
PEG 915	JES 612
DTK 343	DTK 343
CDX 857	CDX 037
CQG 666	PQV 666
AVE 928	AVE 628
IIL 737	IIL 727
AUX 086	AUX 006
KDU 868	KUU 460
KLV 110	KLV 110
RKM 195	YKK 166
RKM 195	RKM 261
BYO 101	BYO 101
IIN 804	IIN 804
CVQ 795	CVB 795
AVD 423	AVD 425
HFN 803	HLH 402
CQG 666	CQV 666
AVK 544	AAK 544
DLV 101	UFY 141
GRU 302	CRU 303
BBQ 727	BBA 722
BPJ 644	BPJ 649
AVC 227	VVC 227
AUT 673	AUT 973
AVF 870	VVF 870
BYE 791	BTZ 761
AVG 965	VYS 665
AUW 723	AUW 722
GRU 254	SRU 234
DHV 151	DHY 121
UBU 549	UDD 246
IHW 983	KHM 683
DQK 705	DOK 745
AUP 821	PUH 521
IHK 439	IHK 736
AUS 121	APV 731
AUX 044	VUX 040

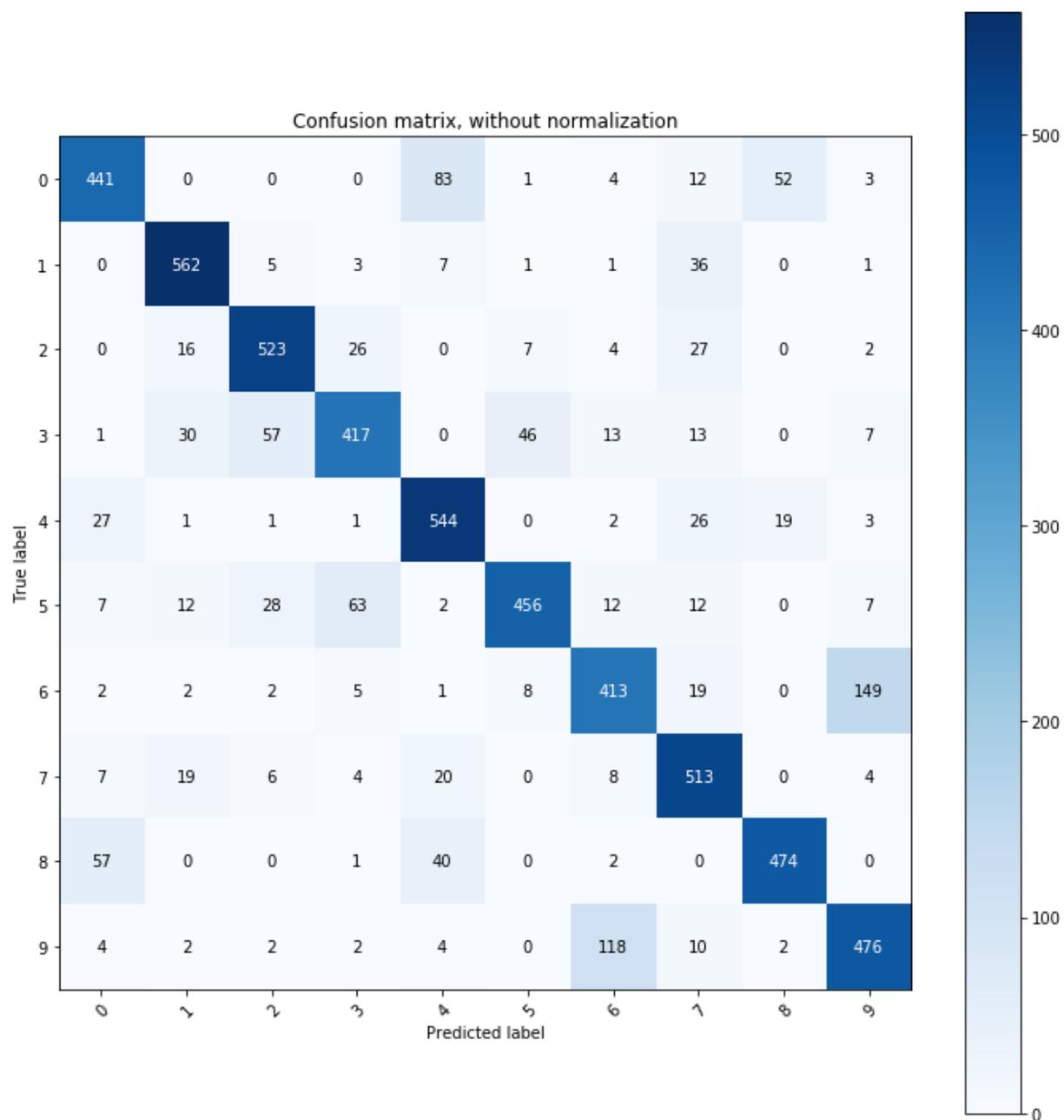
<i>AUY 677</i>	<i>YUY 927</i>
<i>NEN 129</i>	<i>HEH 126</i>
<i>GTY 980</i>	<i>CYY 940</i>
<i>AVH 247</i>	<i>GAH 277</i>
<i>HRM 975</i>	<i>HRN 675</i>
<i>IIN 804</i>	<i>IIM 004</i>
<i>DBU 436</i>	<i>BAP 419</i>
<i>AUS 801</i>	<i>ARC 801</i>
<i>GDP 664</i>	<i>CRF 614</i>
<i>QFZ 711</i>	<i>QJZ 711</i>
<i>IIN 094</i>	<i>IIH 067</i>
<i>DBV 375</i>	<i>DBT 270</i>
<i>ASH 122</i>	<i>TTZ 777</i>
<i>GRU 254</i>	<i>CRU 224</i>
<i>KGL 532</i>	<i>KVF 222</i>
<i>GUN 608</i>	<i>QUH 604</i>
<i>EIM 428</i>	<i>EIM 420</i>
<i>MGX 359</i>	<i>MGX 556</i>
<i>AUS 801</i>	<i>VDS 801</i>
<i>AVJ 106</i>	<i>GYP 102</i>

Anexo 26. Matriz de confusión para las letras utilizando el descriptor LBP y el clasificador

Random forest



**Anexo 27. Matriz de confusión para los números utilizando el descriptor LBP y el clasificador Random forest**



## Anexo 28. Predicción de los caracteres utilizando YOLO

<i>Matricula vehicular</i>	<i>Caracteres predichos</i>
<i>KUM 876</i>	<i>KUM 876</i>
<i>AVH 247</i>	<i>AVH 247</i>
<i>GGP 719</i>	<i>GGP 719</i>
<i>FUY 368</i>	<i>FUY 368</i>
<i>IMW 908</i>	<i>IMW 908</i>
<i>AVF 687</i>	<i>AVF 687</i>
<i>AVJ 529</i>	<i>AVJ 529</i>
<i>CMP 655</i>	<i>CMP 655</i>
<i>CMD 846</i>	<i>CMD 846</i>
<i>ENX 310</i>	<i>ENX 310</i>
<i>DLU 976</i>	<i>DLU 976</i>
<i>AUR 541</i>	<i>AUR 541</i>
<i>GCZ 106</i>	<i>GCZ 106</i>
<i>GDQ 411</i>	<i>GDQ 411</i>
<i>IHZ 896</i>	<i>IHZ 896</i>
<i>BFK 596</i>	<i>BFK 596</i>
<i>AUY 075</i>	<i>AUY 075</i>
<i>MSW 538</i>	<i>MSW 538</i>
<i>DTP 518</i>	<i>DTP 518</i>
<i>BBO 289</i>	<i>BBO 289</i>
<i>AUY 298</i>	<i>AUY 298</i>
<i>IHZ 132</i>	<i>IHZ 132</i>
<i>ENX 497</i>	<i>ENX 497</i>
<i>FJK 598</i>	<i>FJK 598</i>
<i>AVK 755</i>	<i>AVK 755</i>
<i>CFT 762</i>	<i>CFT 762</i>
<i>GDQ 411</i>	<i>GDQ 411</i>
<i>BKG 926</i>	<i>BKG 926</i>
<i>DLV 485</i>	<i>DLV 485</i>
<i>GDQ 244</i>	<i>GDQ 244</i>
<i>IAM 555</i>	<i>IAM 555</i>
<i>MFT 694</i>	<i>MFT 694</i>
<i>BXW 351</i>	<i>BXW 351</i>
<i>LUA 808</i>	<i>LUA 808</i>
<i>CPZ 236</i>	<i>CPZ 236</i>
<i>AUS 753</i>	<i>AUS 753</i>

<i>AVI 111</i>	<i>AVI 111</i>
<i>KIN 864</i>	<i>KIN 864</i>
<i>FGR 368</i>	<i>FGR 368</i>
<i>AUV 230</i>	<i>AUV 230</i>
<i>AUY 900</i>	<i>AUY 900</i>
<i>DTK 396</i>	<i>DTK 396</i>
<i>RKM 195</i>	<i>RKM 195</i>
<i>CWB 710</i>	<i>CWB 710</i>
<i>NEG 741</i>	<i>NEG 741</i>
<i>MFT 694</i>	<i>MFT 694</i>
<i>GUL 746</i>	<i>GUL 746</i>
<i>HHM 205</i>	<i>HHM 205</i>
<i>UDQ 619</i>	<i>UDQ 619</i>
<i>CAP 875</i>	<i>CAP 875</i>
<i>GUL 328</i>	<i>GUL 328</i>
<i>INP 888</i>	<i>INP 888</i>
<i>AVH 903</i>	<i>AVH 903</i>
<i>BSQ 392</i>	<i>BSQ 392</i>
<i>CFT 762</i>	<i>CFT 762</i>
<i>GDQ 254</i>	<i>GDQ 254</i>
<i>UDN 328</i>	<i>UDN 328</i>
<i>AUW 327</i>	<i>AUW 327</i>
<i>EGX 687</i>	<i>EGX 687</i>
<i>AVK 755</i>	<i>AVK 755</i>
<i>IHX 572</i>	<i>IHX 572</i>
<i>DRW 868</i>	<i>DRW 868</i>
<i>FOW 279</i>	<i>FOW 279</i>
<i>SVS 025</i>	<i>SVS 025</i>
<i>HZL 298</i>	<i>HZL 298</i>
<i>IIK 396</i>	<i>IIK 396</i>
<i>JIV 688</i>	<i>JIV 688</i>
<i>CPI 159</i>	<i>CPI 159</i>
<i>AVI 043</i>	<i>AVI 043</i>
<i>IHX 269</i>	<i>IHX 269</i>
<i>AVB 650</i>	<i>AVB 650</i>
<i>IHZ 897</i>	<i>IHZ 897</i>
<i>IIL 592</i>	<i>IIL 592</i>
<i>AUW 327</i>	<i>AUW 327</i>
<i>SVR 086</i>	<i>SVR 086</i>
<i>IHX 572</i>	<i>IHX 572</i>

<i>CCZ 048</i>	CCZ 048
<i>MSM 839</i>	MSM 839
<i>IHX 267</i>	IHX 267
<i>FJQ 884</i>	FJQ 884
<i>EIY 393</i>	EIY 393
<i>AUW 316</i>	AUW 316
<i>EPR 253</i>	EPR 253
<i>DOF 044</i>	DOF 044
<i>AVH 818</i>	AVH 818
<i>AVG 842</i>	AVG 842
<i>NBX 683</i>	NBX 683
<i>IHX 296</i>	IHX 296
<i>CPP 844</i>	CPP 844
<i>DLV 101</i>	DLV 101
<i>RMZ 081</i>	RMZ 081
<i>NMC 962</i>	NMC 962
<i>MAY 571</i>	MAY 571
<i>NMC 962</i>	NMC 962
<i>AVJ 917</i>	AVJ 917
<i>DLX 266</i>	DLX 266
<i>DLX 266</i>	DLX 266
<i>AUW 762</i>	AUW 762
<i>AUP 638</i>	AUP 638
<i>AUW 762</i>	AUW 762
<i>IJK 396</i>	IJK 396
<i>AUR 919</i>	AUR 919
<i>IJK 066</i>	IJK 066
<i>DLV 101</i>	DLV 101
<i>KHM 100</i>	KHM 100
<i>KIK 871</i>	KIK 871
<i>ISV 609</i>	ISV 609
<i>QEQ 022</i>	QEQ 022
<i>JCO 304</i>	JCO 304
<i>MTX 575</i>	MTX 575
<i>KDS 131</i>	KDS 131
<i>CEW 311</i>	CEW 311
<i>CEW 311</i>	CEW 311
<i>AUY 454</i>	AUY 454
<i>PFG 656</i>	PFG 656
<i>RBL 053</i>	RBL 053

<i>AUZ 630</i>	AUZ 630
<i>AVF 585</i>	AVF 585
<i>DLV 293</i>	DLV 293
<i>GDP 273</i>	GDP 273
<i>MIX 908</i>	MIX 908
<i>BOE 646</i>	BOE 646
<i>AVF 585</i>	AVF 585
<i>HKT 997</i>	HKT 997
<i>AVG 045</i>	AVG 045
<i>HBZ 691</i>	HBZ 691
<i>IIL 416</i>	IIL 416
<i>RCK 868</i>	RCK 868
<i>IIL 416</i>	IIL 416
<i>QEO 574</i>	QEO 574
<i>AVI 807</i>	AVI 807
<i>GDP 273</i>	GDP 273
<i>CZQ 846</i>	CZQ 846
<i>HFO 614</i>	HFO 614
<i>AVD 423</i>	AVD 423
<i>AVD 411</i>	AVD 411
<i>HNS 084</i>	HNS 084
<i>DCB 797</i>	DCB 797
<i>IIL 856</i>	IIL 856
<i>IIL 856</i>	IIL 856
<i>AVJ 517</i>	AVJ 517
<i>USV 045</i>	USV 045
<i>KLV 110</i>	KLV 110
<i>NVG 005</i>	NVG 005
<i>AVF 811</i>	AVF 811
<i>WNV 140</i>	WNV 140
<i>DLV 286</i>	DLV 286
<i>EMR 275</i>	EMR 275
<i>IVZ 391</i>	IVZ 391
<i>EMR 275</i>	EMR 275
<i>IDL 488</i>	IDL 488
<i>IDL 488</i>	IDL 488
<i>IHZ 504</i>	IHZ 504
<i>PFG 656</i>	PFG 656
<i>ENX 439</i>	ENX 439
<i>CLU 141</i>	CLU 141

<i>HZV 807</i>	<i>HZV 807</i>
<i>GRN 103</i>	<i>GRN 103</i>
<i>RAV 850</i>	<i>RAV 850</i>
<i>COF 597</i>	<i>COF 597</i>
<i>COF 597</i>	<i>COF 597</i>
<i>HMR 910</i>	<i>HMR 910</i>
<i>IVZ 391</i>	<i>IVZ 391</i>
<i>HMR 910</i>	<i>HMR 910</i>
<i>RHN 653</i>	<i>RHN 653</i>
<i>AUX 577</i>	<i>AUX 577</i>
<i>RHN 653</i>	<i>RHN 653</i>
<i>AVK 790</i>	<i>AVK 790</i>
<i>AVG 144</i>	<i>AVG 144</i>
<i>AVA 495</i>	<i>AVA 495</i>
<i>FGY 852</i>	<i>FGY 852</i>
<i>DVY 988</i>	<i>DVY 988</i>
<i>MGX 359</i>	<i>MGX 359</i>
<i>DBL 326</i>	<i>DBL 326</i>
<i>IIL 726</i>	<i>IIL 726</i>
<i>FAQ 531</i>	<i>FAQ 531</i>
<i>NPD 472</i>	<i>NPD 472</i>
<i>AVA 426</i>	<i>AVA 426</i>
<i>HZV 807</i>	<i>HZV 807</i>
<i>HZV 631</i>	<i>HZV 631</i>
<i>AUP 979</i>	<i>AUP 979</i>
<i>ISV 689</i>	<i>ISV 689</i>
<i>AUX 215</i>	<i>AUX 215</i>
<i>DTK 343</i>	<i>DTK 343</i>
<i>AUZ 679</i>	<i>AUZ 679</i>
<i>QEP 541</i>	<i>QEP 541</i>
<i>MOO 907</i>	<i>MOO 907</i>
<i>BCK 002</i>	<i>BCK 002</i>
<i>FUM 378</i>	<i>FUM 378</i>
<i>AVD 092</i>	<i>AVD 092</i>
<i>AVH 811</i>	<i>AVH 811</i>
<i>PEG 915</i>	<i>PEG 915</i>
<i>PEG 915</i>	<i>PEG 915</i>
<i>PEG 915</i>	<i>PEG 915</i>
<i>DTK 343</i>	<i>DTK 343</i>
<i>CDX 857</i>	<i>CDX 857</i>

<i>CQG 666</i>	<i>CQG 666</i>
<i>AVE 928</i>	<i>AVE 928</i>
<i>IIL 737</i>	<i>IIL 737</i>
<i>AUX 086</i>	<i>AUX 086</i>
<i>KDU 868</i>	<i>KDU 868</i>
<i>KLV 110</i>	<i>KLV 110</i>
<i>RKM 195</i>	<i>RKM 195</i>
<i>RKM 195</i>	<i>RKM 195</i>
<i>BYO 101</i>	<i>BYO 101</i>
<i>IIN 804</i>	<i>IIN 804</i>
<i>CVQ 795</i>	<i>CVQ 795</i>
<i>AVD 423</i>	<i>AVD 423</i>
<i>HFN 803</i>	<i>HFN 803</i>
<i>CQG 666</i>	<i>CQG 666</i>
<i>AVK 544</i>	<i>AVK 544</i>
<i>DLV 101</i>	<i>DLV 101</i>
<i>GRU 302</i>	<i>GRU 302</i>
<i>BBQ 727</i>	<i>BBQ 727</i>
<i>BPJ 644</i>	<i>BPJ 644</i>
<i>AVC 227</i>	<i>AVC 227</i>
<i>AUT 673</i>	<i>AUT 673</i>
<i>AVF 870</i>	<i>AVF 870</i>
<i>BYE 791</i>	<i>BYE 791</i>
<i>AVG 965</i>	<i>AVG 965</i>
<i>AUW 723</i>	<i>AUW 723</i>
<i>GRU 254</i>	<i>GRU 254</i>
<i>DHV 151</i>	<i>DHV 151</i>
<i>UBU 549</i>	<i>UBU 549</i>
<i>IHW 983</i>	<i>IHW 983</i>
<i>DQK 705</i>	<i>DQK 705</i>
<i>AUP 821</i>	<i>AUP 821</i>
<i>IIK 439</i>	<i>IIK 439</i>
<i>AUS 121</i>	<i>AUS 121</i>
<i>AUX 044</i>	<i>AUX 044</i>
<i>AUY 677</i>	<i>AUY 677</i>
<i>NEN 129</i>	<i>NEN 129</i>
<i>GTY 980</i>	<i>GTY 980</i>
<i>AVH 247</i>	<i>AVH 247</i>
<i>HRM 975</i>	<i>HRM 975</i>
<i>IIN 804</i>	<i>IIN 804</i>

<i>DBU 436</i>	DBU 436
<i>AUS 801</i>	AUS 801
<i>GDP 664</i>	GDP 664
<i>QFZ 711</i>	QFZ 711
<i>IIN 094</i>	IIN 094
<i>DBV 375</i>	DBV 375
<i>ASH 122</i>	ASH 122
<i>GRU 254</i>	GRU 254
<i>KGL 532</i>	KGL 532
<i>GUN 608</i>	GUN 608
<i>EIM 428</i>	EIM 428
<i>MGX 359</i>	MGX 359
<i>AUS 801</i>	AUS 801
<i>AVJ 106</i>	AVJ 106