

**ESTUDIO COMPARATIVO DE TÉCNICAS DE *MACHINE LEARNING*
BASADO EN VISIÓN ARTIFICIAL PARA CONTROL Y MONITOREO DE
ACCESO VEHICULAR**



**JESUS LEONARDO HORMAZA CHAMORRO
OSCAR DAVID NOGUERA SANCHEZ**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2019**

**ESTUDIO COMPARATIVO DE TÉCNICAS DE *MACHINE LEARNING*
BASADO EN VISIÓN ARTIFICIAL PARA CONTROL Y MONITOREO DE
ACCESO VEHICULAR**

**JESUS LEONARDO HORMAZA CHAMORRO
OSCAR DAVID NOGUERA SANCHEZ**

**Trabajo de Grado en la modalidad investigación para optar por el título de
Ingenieros Electrónicos**

**DIRECTOR
PhD. DIEGO HERNÁN PELUFFO ORDÓÑEZ
INGENIERO ELECTRÓNICO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2019**

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.”

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

San Juan de Pasto, Octubre de 2019

AGRADECIMIENTOS

“En primer lugar damos gracias a la Universidad de Nariño, de la misma manera al Departamento de Ingeniería Electrónica. Un agradecimiento especial a nuestro asesor Diego Hernán Peluffo, por todo el tiempo brindado, su apoyo, su asesoría, su paciencia, infinitas gracias. Así mismo se da las gracias a cada una de las personas que durante todo el proceso de formación profesional estuvieron presentes con su ayuda para lograr nuestras metas.

Jesús Leonardo Hormaza Chamorro
Oscar David Noguera Sanchez

DEDICATORIA

*“A Dios que es el portador de todo conocimiento.
A mis padres Miguel Hormaza,
Alicia Chamorro y hermanos, quienes han
Sido mi pilar y como muestra de mi
Agradecimiento por todo el apoyo
Recibido durante esta etapa de mi vida.”*

Jesús Leonardo Hormaza Chamorro.

*“A Dios porque todo se hace según su voluntad,
A mis padres Manuel Arturo Noguera,
Sandra Elci Sanchez, a mi hermano mayor
Adolfo Arturo Noguera quien me protege
Desde el cielo. El esfuerzo ha sido de todos.”*

Oscar David Noguera Sanchez

RESUMEN

La visión artificial o visión por computadora es uno de los sub-campos de la inteligencia artificial que consiste en el uso de imágenes o video para detectar, clasificar y rastrear objetos o eventos con el fin de realizar el análisis de una escena del mundo real. En los últimos años, esta disciplina ha mostrado tener una gran variedad de aplicaciones en ambientes científicos, industriales y de seguridad, entre otros. Particularmente, la aplicación en seguridad ha sido la motivación de esta investigación para desarrollar un sistema de reconocimiento automático de matrículas, enfocado al control y monitoreo vehicular, este tipo de sistemas se conoce en el mercado como ANPR (*Automatic Number Plate Recognition*) y son relativamente eficientes. No obstante, la localización óptima de la matrícula vehicular en una escena de video bajo condiciones externas no controladas, así como el diseño de sistemas eficientes de reconocimiento de caracteres es aún un tema abierto de investigación.

En este trabajo de tesis se presenta una metodología de comparación con el fin de localizar una matrícula vehicular en una imagen o video y clasificar los caracteres que la componen. Para este fin, se desarrolla un sistema de comparación de técnicas de procesamiento de imágenes y métodos de *machine learning*, que permita determinar aquellas que presenten mejor desempeño en términos de costo computacional y precisión.

ABSTRACT

Artificial vision or computer-based vision is one of the sub-fields of artificial intelligence that consists of the use of images or video to detect, classify and track objects or events in order to perform the analysis of a real-world scene. In recent years, this discipline has shown a wide variety of applications in scientific, industrial and security environments, among others. Particularly, the application in security has been the motivation of this research to develop an automatic license plate recognition system, focused on vehicle monitoring and control. Commercially speaking, such systems are known as ANPR (Automatic Number Plate Recognition) and show a relatively high efficiency. However, the optimal location of vehicle registration in an image or video scene under uncontrolled external conditions, as well as the design of efficient character recognition systems, is still an open research topic.

In this thesis, a comparison methodology is presented in order to locate a vehicular license plate in an image or video and classify the characters that compose it. To this purpose, a system for comparing image processing techniques and machine learning methods is developed, which will allow for determining those reaching a suitable performance in terms of computational cost and accuracy.

CONTENIDO

	Pág
INTRODUCCIÓN	13
Planteamiento del problema	14
Justificación	14
Contribuciones de este trabajo de grado.	15
Objetivo general	15
Objetivos específicos	15
Organización del documento	16
Marco teórico	16
Imágenes digitales	17
Conceptos básicos.	17
Relaciones básicas entre pixeles.	18
Procesamiento de imágenes digitales	19
Visión artificial para el control y monitoreo vehicular	20
Etapas de la visión artificial	21
Ventajas de la visión artificial	21
Aplicaciones de la visión artificial en el control vehicular	22
Sistemas ANPR	22
Historia de los Sistemas ANPR	23
Aplicaciones	23
Algoritmos 24	
1. DESARROLLO DEL PROYECTO	25
1.1. METODOLOGÍA	25
1.2. BASE DE DATOS	25
1.3. POSICIONAMIENTO DE LA PLACA	26
1.3.1. Descriptor haar	26
1.3.2. Patrones Binarios Locales.	27
1.3.3. Histograma de Gradientes Orientados	27

1.3.4.	Clasificadores en Cascada tipo Adaboost	28
1.4.	BINARIZACIÓN Y PRE-PROCESAMIENTO	29
1.5.	SEGMENTACIÓN	30
1.6.	CARACTERIZACIÓN	31
1.7.	CLASIFICACIÓN DE CARACTERES	32
1.7.1.	Máquina de soporte vectorial.....	32
1.7.2.	K-vecinos más cercanos.	33
1.7.3.	Árbol de Decisiones.....	34
2.	MARCO EXPERIMENTAL	36
2.1.	PREPARACIÓN DE LA BASE DE DATOS	36
2.2.	VERDADEROS POSITIVOS, VERDADEROS NEGATIVOS, FALSO POSITIVO Y FALSO NEGATIVO.....	36
2.3.	VARIABLES DE EFECTIVIDAD	37
2.4.	MATRIZ DE CONFUSIÓN	38
2.5.	PRUEBAS APLICADAS SOBRE LA BASE DE DATOS	39
3.	RESULTADOS.....	40
3.1.	ELECCION DE PARÁMETROS.....	40
3.2.	RESULTADOS DE LOS DETECTORES DE PLACA	42
3.3.	COMPARACIÓN DE LOS MODELOS PARA LA CLASIFICACIÓN DE CARACTERES.....	45
3.4.	RESULTADOS OBTENIDOS EN LA CLASIFICACIÓN DE CARACTERES.....	47
3.5.	RESULTADOS OBTENIDOS EN LA IDENTIFICACION Y RECONOCIMINETO DE PLACAS SOBRE UN SET DIFERENTE DE IMÁGENES.	48
3.6.	DESARROLLO EN PYTHON	49
3.7.	INTERFAZ DE USUARIO.....	50
4.	CONCLUSIONES.....	55
5.	RECOMENDACIONES	56
	BIBLIOGRAFÍA	57
	ANEXOS.....	61

LISTA DE FIGURAS

pág

Figura 1. Automatic Number Plate Recognition.	23
Figura 2. Metodología para detección y clasificación de placas vehiculares..	25
Figura 3. Filtros digitales para calcular el descriptor Haar.	27
Figura 4. Descriptores HOG sobrepuestos sobre una imagen.....	28
Figura 5. Detección de una placa vehicular sobre una imagen.....	29
Figura 6. Proceso de binarización de imagen.	30
Figura 7. Caracteres de la placa segmentada.....	31
Figura 8. Descriptor HOG.....	31
Figura 9. Frontera de decisión para los datos de entrada.	32
Figura 10. Ejemplo de K-NN con k=5.....	34
Figura 11. Ejemplo de una estructura de árbol de decisión.	35
Figura 12. Detección de placas.....	38
Figura 13. Relación del modelo SVM utilizando diferentes kernels.....	41
Figura 14. Desempeño del modelo KNN utilizando diferentes valores de k.....	41
Figura 15. Modelo decision tree con diferente numero de splits.	42
Figura 16. Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores HOG.....	43
Figura 17. Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores LBP.	43
Figura 18. Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores Haar.....	44
Figura 19. Resultados de P, R y F1 para cada clase. La grafica muestra los valores de R, P y F1 para clase diferente..	45
Figura 20 Resultados de R, P y medida F1 para cada clase utilizando el clasificador de árbol de decisiones.	46
Figura 21. Resultados de R, P y medida F1 para cada clase utilizando el clasificador KNN.	46
Figura 22. Interfaz de usuario.	51
Figura 23. función de cargar una nueva imagen.	
Figura 24. Detectores de placa en funcionamiento.	52
Figura 25. Botón Binarizar en funcionamiento.	52
Figura 26. Botón segmentar en funcionamiento.....	53
Figura 27. Extracción de características HOG de los caracteres.....	53
Figura 28. Métodos que se desarrollaron SVM, K-NN y decisión Tree.....	54

LISTA DE TABLA

TABLA 1. FUNCIONES DE DISTANCIA UTILIZADOS PARA EL PROCESAMIENTO DE IMÁGENES.	18
TABLA 2. FUNCIONES KERNEL EMPLEADAS EN CLASIFICADORES SVM.	33
TABLA 3. PRUEBAS OBTENDAS DE R, P y F1 con el descriptor HOG. El modelo entrenado con 8 etapas presento un mejor desempeño	40
TABLA 4. PRUEBAS OBTENDAS DE R, P y F1 con el descriptor LBP. El modelo entrenado con 7 etapas presento un mejor desempeño	40
TABLA 5. PRUEBAS OBTENDAS DE R, P y F1 con el descriptor Haar. El modelo entrenado con 7 etapas presento un mejor desempeño	40
TABLA 6. RESULTADOS PRUEBAS 4,5 y 6.	47
TABLA 7. RESULTADOS DE PRUEBAS SOBRE UN NUEVO CONJUNTO DE IMÁGENES.	48
TABLA 8:: RESULTADOS DEL PROTOTIPO REALIZADO EN PYTHON.	50

INTRODUCCIÓN

La evolución de la tecnología ha logrado dar solución a una gran cantidad de problemáticas que emergen a la par con el desarrollo científico, algunas actividades que comúnmente eran desarrolladas por humanos hoy en día pueden ser ejecutadas por un ordenador o un robot liberando así a las personas de tareas tediosas, repetitivas y/o peligrosas. En efecto, los sentidos como el olfato, gusto, tacto, audición y visión en muchas ocasiones han sido emulados a tal punto de satisfacer las necesidades del ser humano. El sentido de la visión es el sentido más desarrollado y el menos conocido debido a su complejidad convirtiéndose así en el de mayor importancia para obtener información del mundo físico, se calcula que más del 70% de las actividades que realiza el cerebro humano son utilizadas en el análisis de información adquirida a través de la visión.

La inteligencia artificial se define como el proceso de adquisición, caracterización e interpretación de información obtenida del mundo real a través de imágenes y/o video. Gracias a esta línea de investigación se ha podido dar solución a una gran diversidad de problemáticas que enfrenta la comunidad científica contemporánea en ámbitos de control de calidad, navegación autónoma, video vigilancia, entre otros. Desarrollando así sistemas más autónomos capaces de extraer información del mundo físico, procesarla, aprender y tomar decisiones basadas en lo visual para obtener una representación de la realidad.

El reconocimiento automático de matrículas (de sus siglas en inglés ANPR - *Automatic number plate recognition*) es un sistema de vigilancia en masa, que usa visión artificial para el reconocimiento óptico de caracteres en imágenes y obtener la lectura de matrículas vehiculares, para este fin se puede utilizar un circuito cerrado de televisión, radares o redes de cámaras diseñadas específicamente para esta función. Los sistemas de reconocimiento de placas son utilizados en diferentes aplicaciones como son: recaudación electrónica de peajes, control de acceso vehicular, rastreo y monitoreo vehicular, entre otros.

No obstante, la localización óptima de la matrícula vehicular en una escena bajo condiciones externas no controladas, así como el diseño de sistemas eficientes de reconocimiento de caracteres es aún un tema abierto de investigación. Al trabajar en ambientes no controlados existen muchos factores que afectan el buen desempeño de estos sistemas, iniciando desde el lugar de trabajo dado que en cada región debe existir su tecnología específica debido a la variación en el modelo de las placas. De igual manera, se pueden presentar condiciones como resolución pobre de la imagen, imagen desenfocada, y poca iluminación, entre otras. Estas situaciones evidencian un amplio campo de investigación.

En este trabajo de grado en la línea de investigación de procesamiento de imágenes, se presentó una metodología de comparación con el fin de identificar la posición de una matrícula vehicular y clasificar los caracteres que la componen, para ello se desarrolló un sistema de comparación de métodos de *machine learning*

para reconocimiento de patrones con el fin de determinar el método que presente mejor eficiencia en cuanto a velocidad de procesamiento y aciertos usando imágenes, con el objetivo de desarrollar un sistema eficiente para la detección y clasificación de placas vehiculares en un entorno no controlado.

Planteamiento del problema

Típicamente, un sistema ANPR es capaz de realizar vigilancia en masa y seguimiento de tráfico, algunas de las aplicaciones más comunes son sistema automatizado de vigilancia, sistema autónomo de cobro de peajes y estacionamientos, realizar un control de acceso vehicular a zonas restringidas, entre otras, estos están basados principalmente en el uso de visión artificial, procesamiento digital de imágenes y video, sistemas de reconocimientos de patrones y técnicas *machine learning* para el reconocimiento de caracteres.

Este tipo de sistemas son propensos al cambio de luminosidad, posición de la cámara, espaciamiento entre los caracteres de la matrícula y estado de la misma, de acuerdo con esto, para obtener un buen desempeño se deben manejar bajo un entorno controlado, esto hace referencia a mantener una posición fija de la cámara y tener control con las variaciones de luminosidad, por tanto, pese a la gran cantidad de investigaciones realizadas, no se ha logrado una solución definitiva. Con lo anterior, el procesamiento de imágenes e inteligencia artificial aplicados a sistemas ANPR sigue siendo un campo abierto de investigación.

Justificación

Una de las aplicaciones de la visión artificial en el entorno industrial son los sistemas ANPR. A diferencia de otras aplicaciones, estas presentan una dificultad añadida en el momento de procesar imágenes debido a que se trabaja con ENTORNOS NO CONTROLADOS, lo cual dificulta una buena adquisición. La iluminación que varía a cada hora del día, la inclinación y/o estado de las placas vehiculares, la posición de la cámara; son factores que dificultan en gran medida la adquisición de imágenes del entorno real, en la mayoría de los casos es necesario el uso de iluminación infrarroja para que las cámaras puedan operar a cualquier hora del día. Dado a que existen diferentes métodos ANPR estos no cuentan con buena eficiencia debido a diversos factores que genera el entorno no controlado, en condiciones nocturnas es donde hay mayor porcentaje de error debido a la influencia de las condiciones del ambiente llegando a errores del diez por ciento al momento de adquirir la imagen.

La motivante de este trabajo se desarrolló con el objetivo de obtener resultados que representen un aporte hacia la comunidad científica y académica, además que puedan servir como base para obtener mayor conocimiento en relación a la eficiencia de sistemas ANPR como también para futuras investigaciones en el entorno de visión artificial y control vehicular.

Contribuciones de este trabajo de grado.

La detección de objetos es un campo muy amplio en la visión artificial. Aquí se realizó un estudio de los descriptores de imagen para definir cual se acomodó mejor al conjunto de datos que se empleó y se decidió cual es mejor para la detección de matrículas vehiculares en imágenes capturadas bajo entornos no controlados de iluminación y posición de vehículo.

Por otra parte, este trabajo de grado representa un aporte para el procesamiento, caracterización y clasificación de caracteres dado que se ha realizado un estudio para comparar algunos de los modelos clásicos de machine learning permitiendo así definir cual se acomoda mejor a los datos suministrados.

Específicamente, las contribuciones técnicas y metodológicas se listan a continuación:

Base de datos: Se construyó una base de datos con imágenes que fueron capturadas de manera manual en esta investigación y que se puede utilizar para el estudio de otros métodos de detección y comparación.

La metodología de comparación: Se desarrolló una metodología con el fin de comparar diferentes descriptores para evaluar el desempeño de un detector y diferentes técnicas de clasificación para la identificación de caracteres.

Interfaz gráfica: Se desarrolló un GUI de MatLab donde el usuario puede interactuar y poner en marcha el proceso de detección de placas vehiculares.

Objetivo general

Realizar un estudio comparativo de métodos de *machine learning* para control y monitoreo de acceso vehicular con el fin de determinar las técnicas que tienen mejor desempeño en ambientes reales no controlados usando visión artificial.

Objetivos específicos

- Implementar diferentes algoritmos de técnicas de *machine learning* para la detección y reconocimiento de caracteres en placas vehiculares usando visión artificial.
- Proponer una metodología de comparación para determinar qué técnica de *machine learning* presenta mejor desempeño en la detección de placas vehiculares.

- Desarrollar una interfaz de usuario interactiva que permita realizar la puesta en funcionamiento de las técnicas *machine learning* aplicadas.

Organización del documento

Este trabajo está dividido en 8 secciones principales, nombradas de la siguiente manera: Introducción, Descripción del problema, Objetivos, Marco teórico, Metodología, Marco experimental, Resultados, Conclusiones y Trabajo futuro.

En el capítulo 1, se presenta el planteamiento del problema, la justificación de este trabajo y las contribuciones científicas de esta investigación.

En el capítulo 2, se presenta el objetivo general y los objetivos específicos, que fueron planteados al inicio de este trabajo de grado.

En el capítulo 3, se presenta un breve marco teórico para el tema desarrollado en esta investigación.

En el capítulo 4, se describe la metodología empleada en este trabajo.

En el capítulo 5, se presenta el marco experimental, donde se da a conocer el tipo de pruebas realizadas sobre la base de datos, las medidas de desempeño y la base de datos utilizada.

En el capítulo 6, se da a conocer los resultados obtenidos de la investigación realizada.

Finalmente, en el capítulo 7 se presenta las conclusiones que se obtuvieron a partir de este trabajo, además se menciona el posible trabajo futuro, a partir del cual se puede mejorar los resultados obtenidos para definir nuevas investigaciones sobre el tema.

Marco teórico

En las últimas décadas, debido al notable incremento de vehículos activos que circulan por las principales ciudades, se ha originado la necesidad de desarrollar sistemas que permitan la gestión y el análisis del flujo vehicular a través de técnicas de visión artificial. Dentro de estas aplicaciones se encuentra el reconocimiento automático de una placa vehicular-ANPR (*Automatic Number Plate Recognition*) sobre una imagen.

Los sistemas para el análisis de flujo vehicular e identificación de placas son cada vez más robustos, debido al uso de diferentes técnicas en la adquisición y procesamiento de imágenes, además de sistemas computacionales capaces de

aprender e identificar objetos en una escena, así como también el aumento en la velocidad de procesamiento computacional.

En este apartado se definen conceptos fundamentales para el procesamiento de imágenes, segmentación de objetos, técnicas de *machine learning* y diferentes aplicaciones que requieren del procesamiento de imágenes. Todos estos conceptos teóricos se abarcan con el objetivo de tener mayor conocimiento sobre el tema en el que se va a desarrollar este trabajo de grado.

Imágenes digitales

Una imagen es una representación que contiene información descriptiva acerca de la escena que representa, por una parte, la palabra digital se relaciona al cálculo por métodos numéricos, así entonces, una imagen digital se podría definir como la representación numérica de una escena u objeto. No obstante, estas imágenes por sí mismas son incapaces de proveer información a una computadora. para esto, la ciencia ha desarrollado el procesamiento digital de imágenes, que manipula la información numérica en una serie de operaciones que lleven a un resultado deseado.

Conceptos básicos.

Pixel: es la mínima unidad de representación de una imagen digital en forma de cuadrícula y cada uno puede tener valores de cada canal de color para el caso de una imagen RGB.

Profundidad de color: hace referencia al número de bits que se necesita para codificar la información de color en una imagen.

Resolución de imagen: es el grado de detalle de una imagen, este valor se expresa en ppp (píxeles por pulgada), cuanto mayor sea el número de ppp, mayor calidad tendrá la imagen.

Modos de color: es el sistema de coordenadas que nos permiten describir el color de cada pixel utilizando valores numéricos, los modos de color más comunes son: modo escala de grises, modo monocromático, modo RGB, modo HSB y modo CMYK.

Formatos de imagen: las imágenes digitales se pueden almacenar en distintos formatos, cada uno se corresponde con una extensión específica del archivo que lo contiene. Los más utilizados en la actualidad son: BMP, GIF, JPG, TIF y PNG.

Relaciones básicas entre pixeles.

Vecinos de un pixel: un pixel ρ de coordenadas (x, y) presenta un total de cuatro vecinos en plano vertical y horizontal cuyas coordenadas vienen dadas por la ecuación 3.1.

$$(x + 1, y); (x - 1, y); (x, y + 1); (x, y - 1). \quad (3.1)$$

Este grupo de pixeles se denomina 4-vecinos de ρ y se representan por $N_4(\rho)$. Adicional a esto existen otros cuatro vecinos asociados a las diagonales, cuyas coordenadas están dadas por la ecuación 3.1.

$$(x + 1, y + 1); (x + 1, y - 1); (x - 1, y + 1); (x - 1, y - 1). \quad (3.2)$$

Y se representa por $N_8(\rho)$.

Conectividad: la conectividad entre pixeles es un concepto utilizado para definir los límites de los objetos y los componentes de áreas en una imagen. Para establecer si dos pixeles están conectados entre sí, se debe determinar si son adyacentes en un sentido (como ser 4-vecinos). Existen dos tipos de conectividad. 4-conectividad: se da cuando dos pixeles ρ y q están 4-conectados si q pertenece a $N_4(\rho)$. 8-conectividad: dos pixeles ρ y q están 8-conectados si q pertenece a $N_8(\rho)$.

Distancia: Dados tres pixeles ρ , q y z , de coordenadas (x, y) , (s, t) y (u, v) respectivamente, D es una función de distancia si cumple lo siguiente:

- $D(\rho, q) \geq 0$, ($D(\rho, q) = 0$), $\rightarrow \rho = q$,
- $D(\rho, q) = D(\rho, p)$,
- $D(\rho, z) \leq D(\rho, q) + D(q, z)$.

Las funciones de distancia usadas comúnmente se muestran en la tabla 1:

TABLA 1. Funciones de distancia utilizados para el procesamiento de imágenes

Tipo de Distancia	Ecuación Matemática
Distancia euclidiana	$D_E(\rho, q) = \sqrt{(x - s)^2 + (y - t)^2}$
Distancia D_4	$D_4(\rho, q) = (x - s) + (y - t) $
Distancia D_8	$D_8(\rho, q) = \max(x - s , y - t)$

Fuente: Automatic license plate recognition (ALPR): A state-of-the-art review. En: IEEE Transactions on circuits and systems for video technology, Vol. 23, no 2, 2013 p. 311-325.

Procesamiento de imágenes digitales

Binarización: es un proceso que consiste en la realización de un barrido por toda la imagen, con el fin de obtener la reducción de escala de grises de la imagen a dos únicos valores, negro (0) y blanco (255). La comparación de cada pixel de la imagen viene determinada por el umbral de sensibilidad (*Threshold*) que se puede determinar por los métodos de:

La intensidad local en el barrido: este proceso se lo conoce también como método de Bradley.

La Mediana local en el barrido: el cálculo de esta estadística puede ser lento, considere usar un tamaño de vecindario más pequeño para obtener resultados en menos tiempo.

La media ponderada gaussiana: se calcula la ponderada de gauss en el vecindario de cada pixel.

Dilatación: Se conoce como la realización que tiene como objetivo final un conjunto de puntos. Está constituido por todas las sumas de pares de puntos. De los cuales el primero corresponde al conjunto X y el segundo al elemento estructurante desplazado. La operación anterior es de crecimiento progresivo; para cada uno de los puntos de la imagen se cambia el origen del EE, de tal manera que si la intersección del conjunto X con sus pixeles a 1 del EE cambiado sobre el punto es no vacía entonces el punto analizado pertenece a la dilatación de X . Como se muestra en la ecuación 3.2:

$$X \oplus B = \{c | c = x + b, \forall x \in X \forall b \in B\}. \quad (3.3)$$

De igual manera, con el elemento estructurador se puede expresar mediante el reflejo de B que es obtenido por simetría respecto al punto definido como origen.

Erosión: Se conoce como una acción que realiza todo lo contrario a la dilatación, de esta manera se obtiene la reducción de los objetos mediante una operación semejante a la anterior. Se desplaza el elemento definido como estructurador por toda la imagen con la diferencia que en este caso el resultado será 1 para el punto de interés si todos los demás puntos del EE están contenidos dentro del conjunto del objeto. Lo anterior se puede escribir en función del complementario de la imagen así: "un punto pertenecerá a la imagen erosionada si el elemento estructurador desplazado para cada X tiene una intersección no vacía con el conjunto complementario del objeto." Así como se muestra en la ecuación 3.4:

$$X \ominus B = \{X | B_x \subseteq X\} = \{X | B_x \cap X^c \neq \emptyset\}. \quad (3.4)$$

Dependiendo del tamaño y la forma particular del elemento estructurador la erosión podrá reducir la dimensión del objeto, con esto se dice que los intersticios se hacen más grandes de igual manera que la distancia entre cada objeto crece y cada

elemento que tiene un tamaño pequeño en la imagen tiende a desaparecer por lo cual ayuda en gran proporción a eliminar el conocido ruido impulsional o de mínimo tamaño. A pesar que este proceso es contrario al realizado en la dilatación en ningún momento se debe considerar que mediante la erosión de una imagen en estado dilatado se podrá recuperar la imagen original. Este es un análisis erróneo ya que nunca sucede así.

Etiquetado de componentes conexas: Se define como componentes conexas a todas las partes de la imagen donde existe un camino, de manera que se pueda trasladar de uno a otro pixel sin abandonar la zona que contiene a los mismos. Cuando se trabaja con una imagen de tipo binaria es posible obtener la homogeneidad por el valor de sus pixeles. De esta manera se inicia con base a una imagen binarizada, es decir, en blanco y negro (ceros y unos). Todas las regiones o zonas donde exista presencia de pixeles con valor de 1 y cumplan con una conexión entre sí, por un camino o conjunto de pixeles del mismo valor se denominarán componentes conexas. A estos pixeles se los identifica con la misma etiqueta que debe ser única para construir de esta manera su identificador.

Visión artificial para el control y monitoreo vehicular

La visión artificial mecaniza el trabajo que se realiza en la obtención y análisis de la información adquirida del medio exterior mediante cámaras, videos, entre otros. Esta disciplina simula de manera directa la visión humana a través de sensores encargados de capturar la información existente en el mundo físico en tiempo real, que es procesada por un ordenador, para la toma de decisiones de acuerdo a las necesidades existente.

Todo este proceso general se divide en diferentes etapas: captación, pre-procesamiento, segmentación, descripción y reconocimiento de imágenes. Dentro de la primera etapa se desarrolla el proceso de adquisición de imágenes digitales; el procesamiento se enfoca en objetivos como la reducción de ruidos y el realce sobre las zonas de interés, la etapa de segmentación consiste en dividir la imagen en diferentes partes; la obtención de características se realiza en la etapa de descripción que junto con el reconocimiento permiten identificar los objetos y asignarle una etiqueta a cada imagen. Dentro de las aplicaciones de la visión por computador están: el reconocimiento de objetos, detección de eventos, reconstrucción de escenas (*mapping*) y restauración de imágenes.

Etapas de la visión artificial

Los sistemas basados en visión artificial tienen en común las etapas de captura, pre-procesamiento, segmentación y reconocimiento. Entre algunas aplicaciones y otras, varían las técnicas y métodos como se ejecuta cada etapa.

Captura: adquisición de imágenes mediante una cámara u otro tipo de sensor.

Pre-Procesamiento: tratamiento de imágenes para destacar posibles regiones de interés y eliminar información no relevante.

Segmentación: aislamiento de las regiones de interés para su posterior análisis e interpretación.

Reconocimiento: identificación y clasificación de objetos y estructuras en las regiones de interés.

Ventajas de la visión artificial

La visión artificial resalta su mayor ventaja en la medida cuantitativa debido a su velocidad, precisión y rentabilidad en comparación con la visión humana que se destaca en la interpretación cualitativa.

Aumento de la productividad: Se basa en la supervisión de producción mediante controles de calidad, se elimina el índice de producción defectuosa mejorando la productividad de forma positiva.

Disminución de la pérdida de materiales: Asimismo, debido al control que se lleva a cabo, se eliminan las unidades que están en mal estado, reduciendo el uso de los materiales en unidades defectuosas antes de que lleguen al final de la cadena de montaje.

Reducción de costes: al tener mayor cantidad de productividad y menor pérdida en materiales se obtiene una repercusión positiva en los costes económicos finales.

Producto final de mejor calidad: al poder garantizar productos libres de defectos y elementos en mal estado físico es posible ofrecer productos de mayor calidad para mejor satisfacción.

Aplicaciones de la visión artificial en el control vehicular

La visión artificial tiene un gran número de aplicaciones en diferentes campos de la industria, medicina y seguridad, adicionalmente en el ámbito de control y monitoreo vehicular existen varias aplicaciones como conteo de vehículos, semáforos inteligentes, sistemas de reconocimiento de placas entre otros:

Conteo de vehículos: existen algoritmos capaces identificar los vehículos que circulan, permitiendo realizar un conteo de estos y calcular diferentes variables como el flujo vehicular, la velocidad y densidad vehicular.

Sistema de semáforos inteligentes: es un sistema inteligente que optimiza los cambios de luces de los semáforos dependiendo de la densidad vehicular, tomando decisiones en tiempo real.

ANPR. Sistemas capaces de identificar y reconocer las placas vehiculares sobre las imágenes.

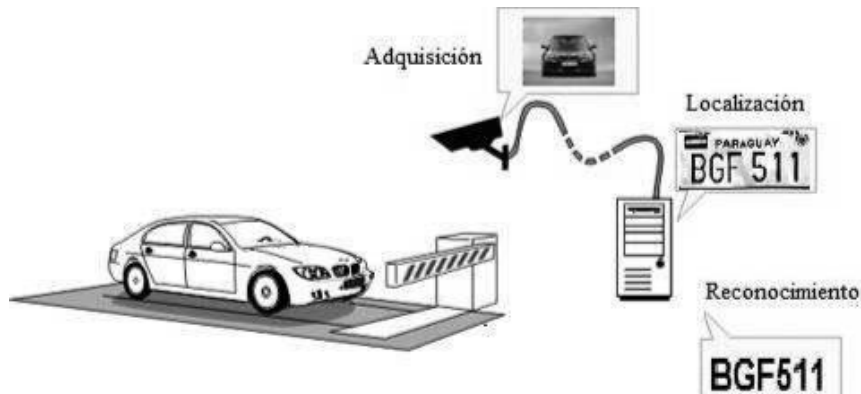
Sistemas ANPR

Un sistema ANPR, es una aplicación de la visión artificial que tiene como principal objetivo la identificación automática de caracteres en imágenes de placas vehiculares, posee un grado de dificultad en su entorno de desarrollo debido a que es totalmente incontrolable. Existen diferentes aspectos que influyen dentro de su funcionamiento como lo son: variación de iluminación, ángulo de la cámara, tamaño de la imagen, entre otros. Para contrarrestar las dificultades hoy en día se opta por utilizar iluminación infrarroja con el fin de eliminar la variación de luz, así como también diferentes técnicas de procesamiento de imágenes para un mejor reconocimiento de la placa vehicular.

Estos sistemas de reconocimiento automático de placas vehiculares se deben desarrollar de forma particular para cada región o país debido a la variación en el modelo de placas.

Como se observa en la **Figura 1**, el desarrollo de un ANPR se resume en utilizar diferentes herramientas de procesamiento de imágenes para llevar a cabo cada etapa que componen el mismo. Entre ellas se tiene: Detección de la imagen, normalización y realce, reconocimiento óptico de caracteres.

Figura 1. Etapas de reconocimiento de placas.



Fuente: Sistema de reconocimiento automático de placa automotriz paraguaya. 2016

Historia de los Sistemas ANPR

Los primeros ensayos de este tipo de sistemas (ANPR) se desarrollaron en los años 70 con el objetivo de detectar carros robados mediante el sistema EMI Electronics Computer Recognition System (CRS), obteniendo su primer resultado hacia el año 1981.

Con el avance de la tecnología los ANPR fueron de igual manera mejorando su eficiencia de tal manera que para el año 2005 se logra desarrollar sistemas capaces de obtener imágenes de placas para vehículos con velocidad de 160 km/h, aumentando de esta manera la posibilidad de recuperar automóviles hurtados luego de confrontar con bases de datos. De la misma manera las empresas aseguradoras han optado por utilizar estos sistemas para brindar mayor seguridad.

Aplicaciones

Dentro de las aplicaciones de un sistema ANPR se tiene:

Gestión de Aparcamiento, se usa la matrícula a forma de llave o comando para permitir estacionamiento o ingreso.

Control de robo en autopistas, ayuda a señalar si un vehículo fue robado o no tiene permiso de circulación, también se usa para búsqueda de automóviles.

Control de velocidad, con el uso de cámaras se puede vigilar que no haya infracciones por exceso de velocidad.

Algoritmos

Dentro del proceso necesario para el reconocimiento de placas vehiculares existen los siguientes pasos:

Localización de la matrícula: este procedimiento tiene como objetivo principal determinar la posición y aislar la placa dentro de la fotografía.

Normalización: mediante este algoritmo se ajusta el contraste y el brillo de una fotografía.

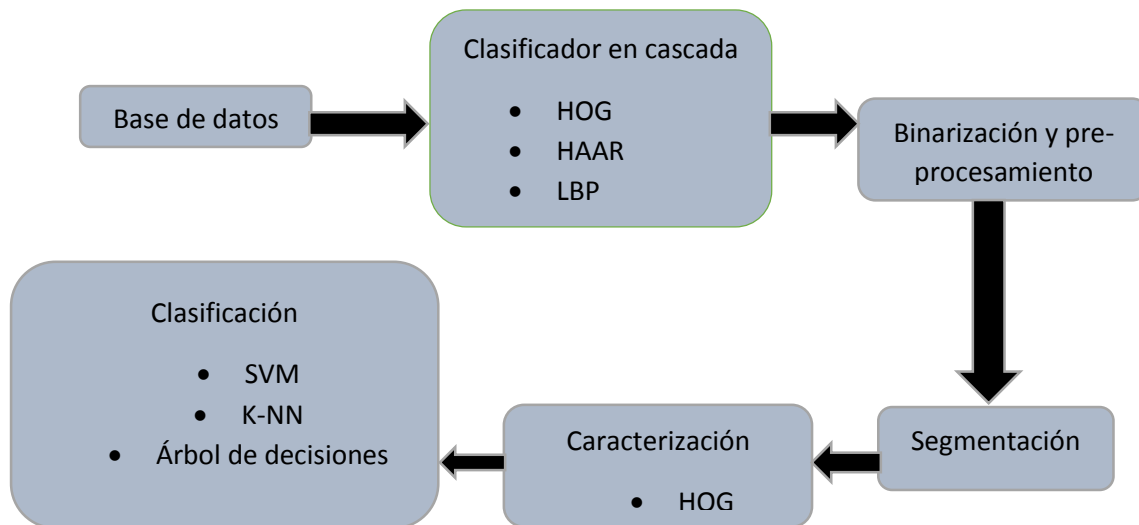
Reconocimiento óptico de Caracteres: se conoce como el proceso de reconocimiento de cada letra dentro del rectángulo de la placa.

1. DESARROLLO DEL PROYECTO

1.1. METODOLOGÍA

La metodología que fue propuesta para el estudio comparativo enfocado en la detección de placas vehiculares y reconocimiento óptico de caracteres se dividió en las etapas de: detección de la placa, binarización y pre-procesamiento, segmentación, caracterización y clasificación de caracteres. En la **Figura 2** se muestra el diagrama de la metodología utilizada en esta investigación. (Sección 4.1), posicionamiento de la placa vehicular (sección 4.2), pre procesamiento y binarización (sección 4.3), segmentación (sección 4.4), caracterización (sección 4.5) y por ultimo clasificación (sección 4.5)

Figura 2. Metodología para detección y clasificación de placas vehiculares.



Fuente: Esta investigación.

1.2. BASE DE DATOS

La base de datos está conformada por imágenes que se capturaron de forma manual durante el desarrollo de esta investigación. Para ello se utilizó la cámara digital de una Tablet marca Lenovo, con una resolución de 8 megapíxeles. Las imágenes están almacenadas en un formato .bmp con tres bandas de color tipo RGB.

Adquisición: Las imágenes positivas y negativas se adquirieron bajo condiciones de iluminación natural y en diferentes horarios. Se hicieron tomas de vehículos tanto de la parte posterior como de la parte trasera. Se realizó tomas de diferentes ángulos, alturas y distancias, esto con el fin de crear una base de datos lo suficientemente completa al momento de entrenar el detector.

Etiquetado: Las imágenes positivas deben ser etiquetadas, resaltando las regiones de interés o ROIs (*regions of interest*). Para este fin se utilizó la herramienta (*trainingImageLabeler*) de matlab2015b, que permite generar una estructura con la dirección de las imágenes y las coordenadas de las ROIs. Esto con el fin, de evitar procesamiento de regiones de la imagen que no tienen información útil para el proceso.

Organización de archivos: las imágenes se almacenaron dentro de subcarpetas dependiendo de la clase a la que pertenecen, al igual que las imágenes de los caracteres, una por cada tipo carácter diferente.

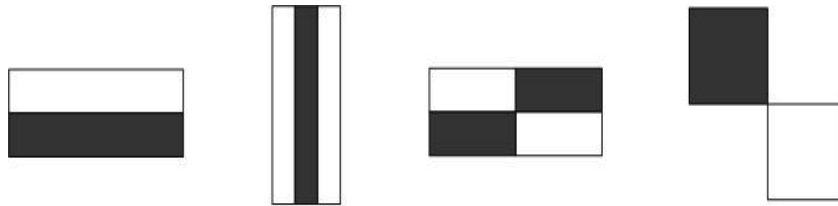
1.3. POSICIONAMIENTO DE LA PLACA

Los descriptores de imagen son una herramienta muy importante en la visión artificial, tienen el fin de representar la información necesaria de para obtener un buen aprendizaje de los clasificadores, en este apartado se describen los descriptores que se utilizaron para entrenar el detector de placas, así como también, el funcionamiento del clasificador en cascada.

1.3.1. Descriptor haar

Los descriptores de imágenes tipo haar toman principalmente tres tipos de filtros digitales, con el fin de detectar orillas, líneas y diagonales, en la **Figura 3** se puede apreciar los filtros que se aplican. El valor de una característica de dos regiones rectangulares es la diferencia de las intensidades promedio de los pixeles dentro de las dos regiones, estas regiones tienen el mismo tamaño y la misma forma además de ser adyacentes en posición vertical u horizontal. Una característica de tres rectángulos calcula la suma dentro de los rectángulos de las orillas y lo resta de la suma del rectángulo del centro. Finalmente, una característica basada en cuatro rectángulos calcula la diferencia entre los pares diagonales de los rectángulos.

Figura 3. Filtros digitales para calcular el descriptor Haar.



Fuente: Face Detection Based on Multi-Block LBP Representation, en Advances in Biometrics. 2017

1.3.2. Patrones Binarios Locales.

Los patrones binarios locales, LBP por sus siglas en inglés (*Logicial binary pattern*), se conoce como un operador de textura simple y con una gran eficiencia donde realiza la acción de etiquetar cada pixel de una imagen específica analizando de esta manera su vecindad, de esta forma analiza el nivel de gris de cada uno de los pixeles estudiando si supera un determinado umbral y codificando cada comparación con un respectivo número binario. Una de sus principales características es el bajo costo computacional y su gran capacidad discriminativa que ha comprobado poseer por lo cual, LBP en los últimos tiempos se ha convertido en una de las soluciones más utilizadas en relación a trabajos o aplicaciones donde se utilice textura. Este operador es muy conocido por poseer una gran robustez que suministra en cambios de la intensidad de la escala de grises que en gran parte es causado por las diferencias en iluminación, cuando se trabaja aplicaciones reales. Para determinar el cálculo de LBP en una determinada imagen en escala de grises se procede con la ecuación 4.2.1:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad S(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (4.2.1)$$

donde p es el número de vecinos a tener en cuenta, R es el tamaño del vecindario y g_c con g_p corresponderán a los valores de gris del pixel central y cada uno de los p pixeles de todo el vecindario respectivo.

1.3.3. Histograma de Gradientes Orientados

Histograma de gradientes orientados, HOG (*histogram of oriented gradients*) están basados en la orientación del gradiente en áreas locales de una imagen. Se obtiene dividiendo la imagen en celdas de $N \times N$ pixeles, calculando el histograma de la dirección del gradiente en cada una de las celdas. El descriptor convierte la información local de los gradientes para cada celda, en una representación de la imagen que captura la forma global del objeto, el descriptor HOG es formado por el

conjunto de histogramas concatenados. En la **Figura 4** se puede observar el descriptor sobrepuesto sobre la imagen original de una placa.

Figura 4. Descriptores HOG sobrepuestos sobre una imagen.



Fuente: esta investigación

1.3.4. Clasificadores en Cascada tipo Adaboost

El clasificador Adaboost funciona bajo la arquitectura de clasificadores en cascada que es organizada en etapas. Para cada etapa se preparan un conjunto de muestras positivas y muestras negativas, con las que se seleccionan un número de clasificadores débiles que en su conjunto forman un clasificador fuerte. Los clasificadores débiles son escogidos dentro del espacio de características de representación (Haar, LBP o HOG). Esta selección se realiza para garantizar una determinada tasa de clasificación exitosa de las muestras positivas y negativas

El análisis en cascada se efectúa con ventanas sobre la imagen a fin de determinar si corresponde a la clase de muestras positivas o negativas. Si la ventana no corresponde a las muestras positivas se descarta, de lo contrario pasa a la siguiente etapa. En cada etapa se aumenta la complejidad de los clasificadores que describen las muestras positivas y las muestras negativas. Para LBP o HOG se necesita ponderar los clasificadores obtenidos para lograr eliminar patrones de histogramas repetitivos y redundantes.

Figura 5. Detección de una placa vehicular sobre una imagen, la placa localiza y se marca con un recuadro azul.



Fuente: registro fotográfico de esta investigación.

1.4. BINARIZACIÓN Y PRE-PROCESAMIENTO

Una vez identificada la posición de la placa, en la etapa de binarización y pre-procesamiento, se realizó el recorte de la región donde se ubicada la placa y se normalizo el tamaño 240×320 pixeles, esto con el fin de facilitar el manejo de la imagen. Se desarrolló una conversión a escala de grises con la ecuación 4.3.1, para reducir el tamaño de los datos y facilitar el pre-procesamiento de la imagen:

$$ImGray = 0.299 * R + 0.587 * G + 0.114 * B. \quad (4.3.1)$$

Esta conversión permitió reducir el número de datos que se tuvieron que computar, el valor de intensidad de color de una imagen en escala de grises tiene un valor entre 0 y 255, para binarizar se debe elegir un umbral de intensidad para separar los objetos de interés del fondo con la ecuación 4.3.2. Se optó por usar umbrales locales debido a que la iluminación no es uniforme sobre algunas imágenes. Consiste en calcular un umbral para cada pixel de la imagen $T_{p(x,y)}$. Existen diferentes técnicas de umbralización local:

La intensidad local en el barrido, este proceso se lo conoce también como método de Bradley.

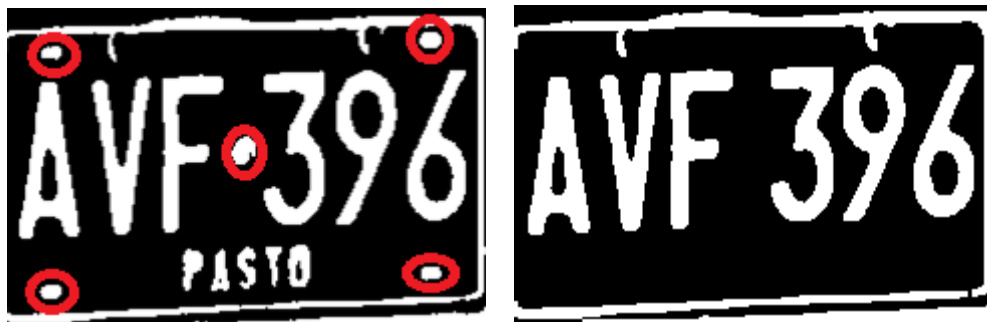
La Mediana local en el barrido, el cálculo de esta estadística puede ser lento, considere usar un tamaño de vecindario más pequeños para obtener resultados en menos tiempo.

La media ponderada gaussiana en el vecindario, basándose en la media ponderada de Gauss de los pixeles vecinos, así el resultado de la binarización se define como:

$$\rho(x, y) = \begin{cases} 1 & \text{si } \rho(x, y) > T_{\rho(x,y)} \\ 0 & \text{si } \rho(x, y) \leq T_{\rho(x,y)} \end{cases} \quad (4.3.2)$$

Adicionalmente se elimina regiones menores a 400 píxeles, esto no representó pérdida de información y para resaltar los caracteres de la placa se realizaron las operaciones morfológicas de erosión y dilatación con una máscara de unos con tamaño de 2x2 y 3x3 respectivamente, en la **Figura 6** se puede observar los resultados de binarización y pre-procesamiento sobre una imagen.

Figura 6. Proceso de binarización de imagen, a la izquierda imagen binarizada, a la derecha imagen binarizada después de haber sido pre-procesada.



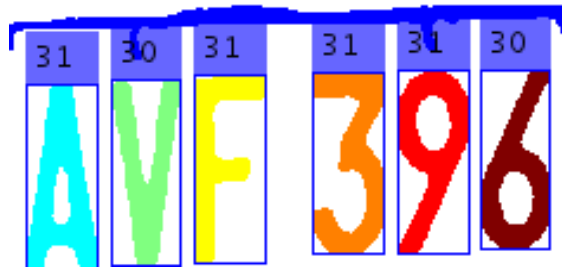
Fuente: esta investigación.

1.5. SEGMENTACIÓN

Para la segmentación de los objetos de la imagen binaria se utilizó un análisis de conectividad de píxeles, para establecer los límites entre los objetos y regiones dentro de una imagen, se hace un recorrido sobre la imagen con una conectividad 8, en donde dos píxeles, $p(x, y)$ y $q(x, y)$ con valores de 1 están conectados si $q(x, y)$ pertenece a $N_8(p(x, y))$, en tal caso se les asignó una misma etiqueta a estos píxeles asignándolos a un grupo, una vez etiquetados los objetos dentro de la imagen se procedió a realizar un análisis de los objetos para extraer diferentes propiedades como área, ancho, alto, orientación de los objetos. Se resalta con un recuadro sobre la imagen cada objeto que se encontró.

Como se muestra en la **Figura 7**, se hizo una discriminación del ancho, alto y área de los objetos para eliminar aquellos que no son de interés, y se obtiene las posiciones para recortar los caracteres de la imagen original y se escala a un tamaño de 80×40 píxeles.

Figura 7. Caracteres de la placa segmentada. Las regiones de la placa se segmentan en función de su ancho, area y tamaño.

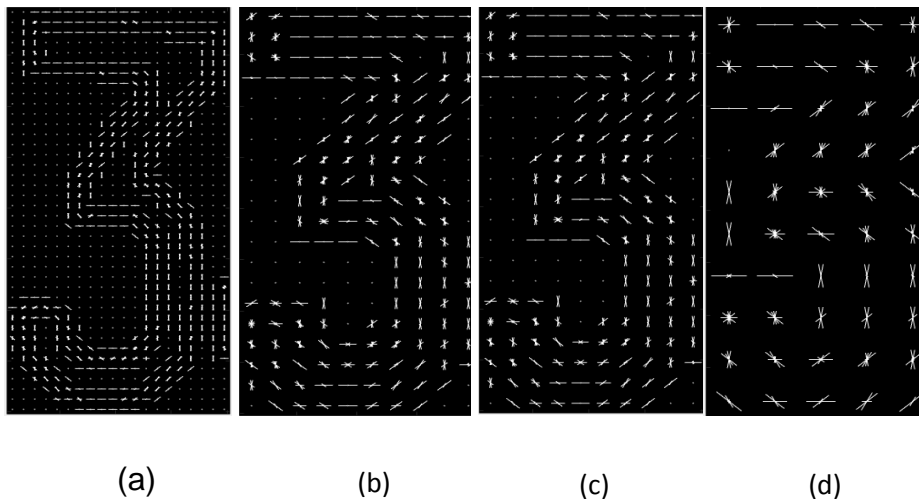


Fuente: esta Investigación.

1.6. CARACTERIZACIÓN

En esta etapa de la investigación se desarrolló el proceso de extraer los descriptores de cada carácter, el descriptor HOG describe la forma del objeto por lo que es ideal para la clasificación de caracteres con una fuente de letra definida. Se utilizó celdas de un tamaño de 6x6 píxeles sobre la imagen del carácter binarizada, un tamaño de celdas muy pequeño puede generar un número muy elevado de descriptores con información redundante, por otro lado, un tamaño de celda muy grande representa pérdida de información de la forma del objeto, en la **Figura 8** se analizó el descriptor HOG utilizando diferentes tamaños de celda. El resultado de esta configuración es un vector de características HOG de un tamaño de 972 para cada imagen analizada.

Figura 8. Descriptor HOG. (a) Tamaño de celda 2x2. (b) Tamaño de celda 4x4. (c) Tamaño de celda (6x6). (d) Tamaño de celda 8x8.



Fuente: Esta Investigación.

1.7. CLASIFICACIÓN DE CARACTERES

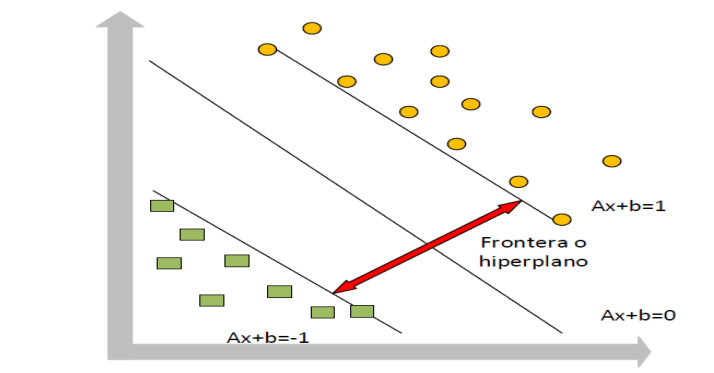
Para poder identificar a que clase pertenece cada objeto, es necesario definir unas fronteras de decisión entre las diferentes clases, normalmente se calculan mediante un entrenamiento, en el que se utiliza un conjunto de ejemplos de cada clase previamente etiquetados, conocido como: aprendizaje supervisado, en donde los clasificadores tienen el conocimiento de cada patrón y también el criterio o métrica para clasificar entre las clases de patrones. Los clasificadores son utilizados comúnmente en aplicaciones como: segmentación de imágenes, reconocimiento de objetos, detección de novedad, entre otras.

1.7.1. Máquina de soporte vectorial

Por su significado en inglés *Support Vector Machine* (SVM). Este método de clasificación supervisado puede ser empleado para clasificación binaria o regresión. Las SVM son muy aplicadas en temas como procesamiento del lenguaje natural, el habla, el reconocimiento de imágenes y la visión artificial [45]. Una máquina de vectores de soporte construye un hiperplano óptimo en forma de superficie de decisión, como se puede apreciar en la **Figura 9**, de modo que el margen de separación entre las dos clases se amplía al máximo. Los SVM son muy adecuados para problemas de clasificación con múltiples categorías y pueden procesar un gran número de entradas. Considere la siguiente notación: Se define el i -ésimo vector de datos como x_i , tal que $x_i \in \mathbb{R}^d$, donde d es la cantidad de variables o atributos. Así, la matriz de n datos puede escribirse como:

$$X = (x_1, \dots, x_n).$$

Figura 9. Frontera de decisión para los datos de entrada.



Fuente: Implementation of algorithms based on support vector machine (SVM) for electric systems: topic review. 2016

El entrenamiento de una máquina de vectores de soporte tiene dos fases:

1. Transformar los datos de entrada en un espacio de características altamente dimensional. En esta fase es suficiente con especificar el Kernel; los datos no se transforman explícitamente al espacio de características.
2. Resolver un problema de optimización cuadrática que se ajuste a un hiperplano óptimo para clasificar las características transformadas en dos clases.

Dentro de las funciones Kernel más importantes que se emplean con los clasificadores SVM se muestran en la **Tabla 2**.

TABLA 2. Funciones kernel empleadas en clasificadores svm.

Tipo de Kernel	Ecuación
Gaussiano	$K(x_i, x_j) = \exp(-\frac{\ x_i - x_j\ ^2}{2\sigma^2})$, σ es la anchura del Kernel
Lineal	$K(x_i, x_j) = x_i^T x_j$
Polinómica	$K(x_i, x_j) = (x_i^T x_j + 1)^\rho$, ρ es el orden del polinomio

Fuente: Implementation of algorithms based on support vector machine (SVM) for electric systems: topic review. 2016

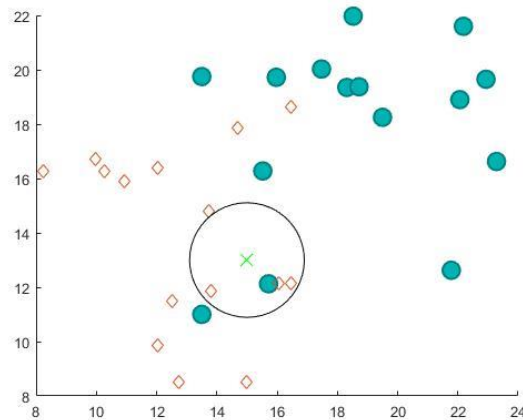
1.7.2. K-vecinos más cercanos.

El clasificador K-NN (*K-Nearest Neighbor*) es un método que pertenece a la familia de clasificadores supervisados, basado en el entrenamiento mediante ejemplos cercanos en el espacio de los objetos. El K-NN a pesar de ser un clasificador con décadas desde su aparición, sigue siendo uno de los más eficaces.

El principio de este modelo es considerar que los objetos que pertenecen a una clase deben estar cercanos entre sí en el espacio de características. Se utiliza el concepto de distancia para determinar la cercanía entre los objetos. Así, en la **Figura 10** se puede ver un ejemplo de un clasificador K-NN con un valor K=4, el nuevo objeto (x) será clasificado como objeto de la clase rombo, dado que en esa región existen tres objetos de clase rombo y uno de clase círculo.

Se describe el funcionamiento del algoritmo K-NN, tanto para K=1 como para K>1.

Figura 10. Ejemplo de K-NN con $k=5$. Muestra un ejemplo de clasificación de un nuevo objeto x .



Fuente: esta investigación.

Algoritmo de K-NN para $K=1$

Seleccionar la métrica a utilizar.

1. Calcular la distancia de un patrón x por clasificar, a cada conjunto fundamental.
2. Obtener la distancia mínima.
3. Asignar a x la clase del patrón con la mínima distancia.

Algoritmo de K-NN para $K>1$:

1. Seleccionar la métrica a utilizar.
2. Calcular la distancia de un patrón x por clasificar, a cada conjunto fundamental.
3. Obtener los K menores valores de distancia. Usar la regla de majority para asignar la clase al patrón x .

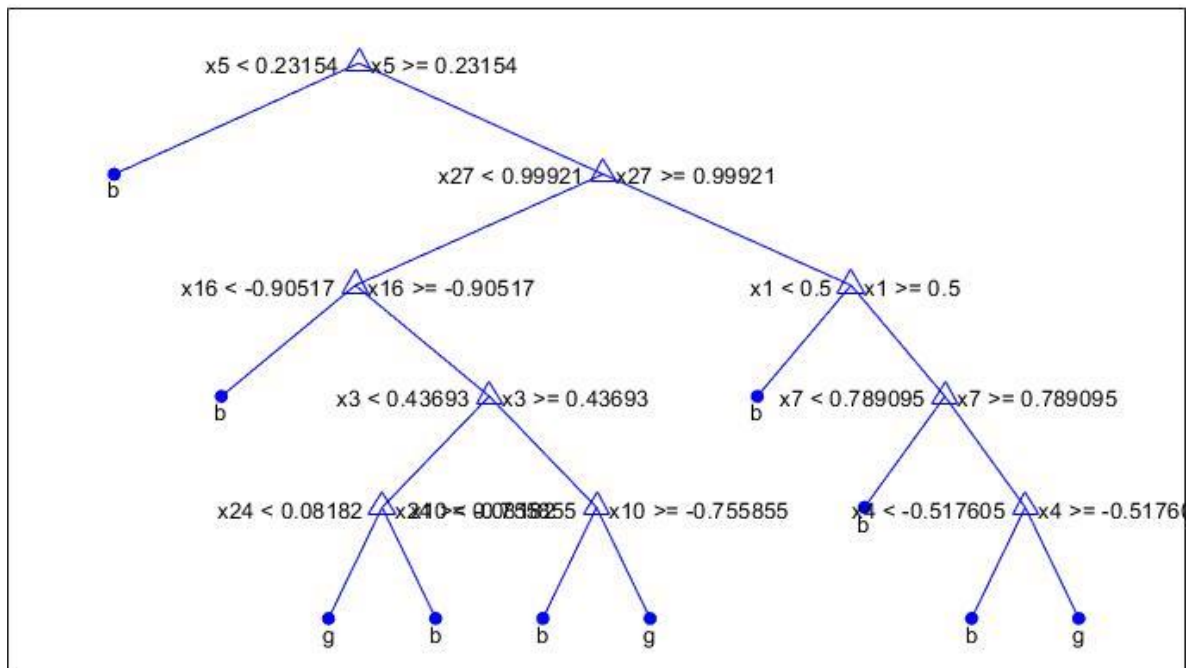
1.7.3. Árbol de Decisiones

Los árboles de decisiones (*decision tree*) son conocidos como clasificadores potentes, se debe emplear una estructura jerárquica para poder modelar las relaciones entre las características y sus probables resultados. Estos árboles de decisiones se fabrican realizando particiones; la totalidad de los datos se dividen inicialmente en un subconjunto para posteriormente dividirlos en conjuntos más pequeños hasta que el algoritmo establezca que los datos son homogéneos.

Para cada partición o nodo el árbol calcula la impureza, luego ordena la observación para determinar qué características pueden usarse como criterios de división. Para

todos los puntos de división, el árbol busca maximizar la disminución de la impureza, dividiendo las observaciones en el nodo en dos nodos secundarios, y luego midiendo las impurezas para cada nodo con diferentes puntos de corte. Una vez que se elige una característica como el mejor candidato de división entonces la característica se divide utilizando el mejor punto de corte, y el proceso se repite hasta que se minimiza la impureza total y se encuentran los nodos finales (hojas), en la **Figura 11** se puede observar un ejemplo de la estructura de un árbol de decisión con un total de nueve spllets (divisiones) y 10 nodos finales.

Figura 11. Ejemplo de una estructura de árbol de decisión. Tiene un total de 9 divisiones o spllets.



Fuente: esta investigación.

2. MARCO EXPERIMENTAL

En este apartado se describe las métricas que se utilizó para la evaluación de los modelos, así como también las pruebas que se desarrollaron durante el desarrollo de la investigación.

2.1. PREPARACIÓN DE LA BASE DE DATOS

La base de datos se elaboró durante el desarrollo de la investigación está conformada por imágenes organizadas de la siguiente manera:

- 885 imágenes positivas, en donde existe el objeto de interés dentro de la imagen.
- 50 imágenes positivas capturadas en otro ambiente, para realizar pruebas de evaluación del detector y clasificadores.
- 40 imágenes de prueba que se utilizaran para evaluar los mejores detectores y clasificadores
- 6000 imágenes negativas (no está presente el objeto de interés)
- 110 muestras de cada carácter alfanumérico utilizado para representar una placa, para un total de 4320 imágenes
- 20 muestras de cada carácter alfanumérico para realizar una validación de los clasificadores.

Los archivos de imágenes positivas fueron almacenados en la carpeta “positivas”, las imágenes negativas almacenadas dentro de la carpeta “negativas”. Los caracteres almacenados en la carpeta “caracteres” que contiene 2 subcarpetas, “TRAINING_SET” Y “TEST_SET” cada una contiene 36 subcarpetas con 110 y 20 imágenes respectivamente de cada letra y número, así mismo, las imágenes positivas de prueba fueron almacenadas en la carpeta TEST.

2.2. VERDADEROS POSITIVOS, VERDADEROS NEGATIVOS, FALSO POSITIVO Y FALSO NEGATIVO.

Un verdadero positivo (TP) ocurre cuando el modelo predice de manera correcta la clase positiva, en este caso cuando detecta correctamente la sección de la imagen como placa y es placa o cuando se clasifica un carácter a la clase que realmente pertenece, así mismo, un verdadero negativo (TN) ocurre cuando el modelo predice de manera correcta la clase negativa. Un falso positivo (FP) se da cuando el modelo predice incorrectamente la clase positiva y un falso negativo (FN) resulta cuando el modelo predice erróneamente la clase negativa

2.3. VARIABLES DE EFECTIVIDAD

Sensibilidad (Recall): hace referencia al número de elementos identificado correctamente como positivo frente al total de las identificaciones positivas. La sensibilidad se calcula como la proporción entre los verdaderos positivos y la suma entre los verdaderos positivos y falsos negativos. La ecuación 5.3.1 muestra cómo se realiza el cálculo de esta variable.

$$R = \frac{TP}{TP+FN} * 100. \quad (5.3.1)$$

Precisión: hace referencia al número de elementos identificados correctamente como positivos y el total de observaciones positivas. La ecuación 5.3.2 define el cálculo de esta métrica.

$$P = \frac{TP}{TP+FP} * 100. \quad (5.3.2)$$

Medida f1: en puntaje f1 es el promedio ponderado de precisión y sensibilidad por lo tanto esta puntuación tiene en cuenta tanto los falsos positivos como los falsos negativos, se calcula con la ecuación 5.2.3.

$$f1 = 2 \frac{R*P}{R+P}. \quad (5.2.3)$$

Porcentaje de clasificación (CP): es la tasa total de aciertos y se calcula con la ecuación 5.2.4

$$CP = \frac{VN+VP}{VN+VP+FN+FP} * 100. \quad (5.2.4)$$

Para la detección de la posición de la placa vehicular cada imagen puede ser etiquetada en una de dos clases: Una conformada por las imágenes que pertenecen a la clase de placas vehiculares (SP) y otra conformada por imágenes diferentes a de la clase de interés (NP), se utilizaron las mismas medidas de desempeño mencionadas anteriormente, sin embargo, se deben calcular de manera diferente si se observa la **Figura 11**, se ha detectado una placa sin embargo la detección no se ha realizado de forma completa, no obstante, no se puede catalogar la detección como un falso positivo debido a que la detección es correcta.

De esta se definió las siguientes medidas de desempeño: precisión (**P**), Recall (**R**) y medida F1 (**F1**) como esta en las ecuaciones 5.2.5, 5.2.6 y 5.2.7 respectivamente:

$$R(G, D) = \frac{\sum_{i=1}^{|G|} BestMatch(G_i)}{|G|}, \quad (5.2.5)$$

$$BestMatch(G_i) = \max_j \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)}$$

$$P(G, D) = \frac{\sum_{j=1}^{|D|} BestMatch(D_j)}{|D|}, \quad (5.2.6)$$

$$BestMatch(D_j) = \max_i \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)}$$

$$F1 = 2 \frac{R(G,D)*P(G,D)}{R(G,D)+P(G,D)}. \quad (5.2.7)$$

donde:

- D_j es la lista de rectángulos de las detecciones.
- G_i es la lista de rectángulos de las imágenes positivas.
- $BestMatch$, cociente entre el área de la intersección y la unión de G_i y D_j .
- P, precisión: mide la probabilidad de que una detección sea verdadera.
- R, Recall (sensibilidad): mide la probabilidad de que un objeto que este

Figura 12. Detección de placas. Se muestra la placa detectada sobre una imagen.



Fuente: esta investigación.

2.4. MATRIZ DE CONFUSIÓN

Es una herramienta que permite la visualización del desempeño de un modelo de clasificación supervisada. Las filas representan la clase real de cada objeto, mientras que las columnas representan las predicciones realizadas por el modelo, permite ver si el sistema está confundiendo las clases, facilita el cálculo de P y R para sistemas multi-clase.

2.5. PRUEBAS APLICADAS SOBRE LA BASE DE DATOS

En la metodología propuesta, para identificar la posición de la placa se comparó diferentes clasificadores en cascada adaboost que se configuraron de diferentes maneras, además se utilizó diferentes descriptores de imagen.

- **Prueba 1:** Se realizó el entrenamiento del clasificador en cascada variando el número de capas y características HOG.
- **Prueba 2:** Se realizó el entrenamiento del clasificador en cascada utilizando diferente número de capas y características LBP.
- **Prueba 3:** Se realizó el entrenamiento del clasificador en cascada utilizando diferente número de capas y características HAAR.

Para el reconocimiento de caracteres se desarrolló las siguientes pruebas:

- **Prueba 4:** Se realizó la clasificación de los caracteres utilizando **SVM** como modelo para clasificar los caracteres de la matricula vehicular.
- **Prueba 5:** Se desarrolló la clasificación utilizando **K-NN** como modelo para clasificar los caracteres de la matricula vehicular.
- **Prueba 6:** Se realizó la clasificación utilizando **decisión tree** como modelo para clasificar los caracteres de la matricula vehicular.
- **Prueba 7:** Se eligió al mejor detector que presente el menor índice de falsos negativos y se enlaza con cada uno de los mejores modelos de clasificadores de cada tipo para evaluar el resultado de detecciones sobre un conjunto nuevo de imágenes.

3. RESULTADOS

3.1. ELECCION DE PARÁMETROS

Para el detector de placas se realizó el entrenamiento de clasificadores en cascada con diferentes descriptores: HOG, LBP y HAAR, se hizo pruebas variando el número de etapas. Los resultados obtenidos de sensibilidad o recall (R), precisión (P) y medida de desempeño f1 (F1) se muestran en las tablas 3, 4 y 5.

TABLA 3. Pruebas obtenidas de r, p y f1 con el descriptor hog. El modelo entrenado con 8 etapas presento un mejor desempeño

Nº de etapas	R	P	F1
3	0.6063	0.0336	0.0637
4	0.6611	0.1168	0.1985
5	0.7064	0.3879	0.5008
6	0.7034	0.7096	0.7095
7	0.7072	0.7491	0.7275
8	0.7129	0.7836	0.7466

TABLA 4. Pruebas obtenidas de R, P y F1 con el descriptor LBP. El modelo entrenado con 7 etapas presento un mejor desempeño

Nº de etapas	R	P	F1
3	0.4979	0.0131	0.0255
4	0.7369	0.0245	0.0474
5	0.7688	0.0589	0.1095
6	0.7787	0.2695	0.40025
7	0.7684	0.7174	0.7420
8	0.7332	0.7259	0.7294

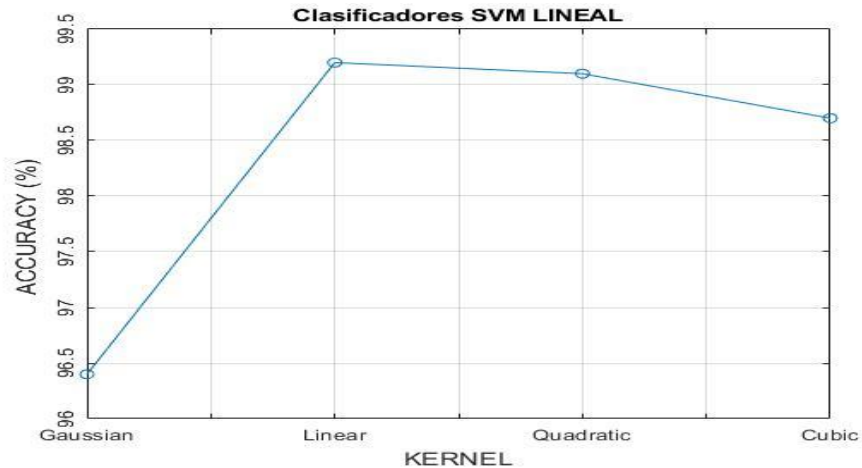
TABLA 5. Pruebas obtenidas de R, P y F1 con el descriptor Haar. El modelo entrenado con 7 etapas presento un mejor desempeño

Nº de etapas	R	P	F1
3	0.5579	0.0231	0.04436
4	0.6469	0.0345	0.0673
5	0.6898	0.0589	0.1095
6	0.7397	0.7480	0.7438
7	0.7496	0.5347	0.6242

Para el entrenamiento del modelo SVM se probó diferentes Kernels con el fin de determinar cuál presenta un mejor ajuste a los descriptores de la imagen y eficiencia

a la hora de identificar los caracteres. Se probaron diferentes tipos: gaussian Kernel, linear Kernel, quadratic Kernel y cubic Kernel, en la **Figura 13** se realiza una relación entre el porcentaje de clasificación y los Kernels aplicados. El modelo que mostro mejor resultado fue el entrenamiento con linear Kernel.

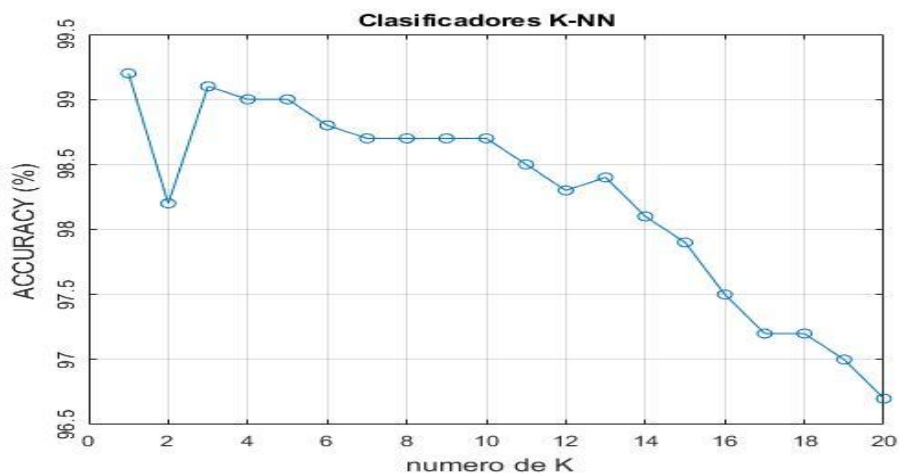
Figura 13. Relación del modelo SVM utilizando diferentes Kernels.



Fuente: esta Investigación.

Para el clasificador K-NN se crearon diferentes modelos con variaciones en el valor de k entre 1 y 20, en la **Figura 14** se muestra la relación entre los valores de k y el porcentaje de clasificación. Para valores de k por encima de 7 la tasa de clasificación del modelo empieza a decrecer. El modelo que mostro el mejor resultado fue con k=1.

Figura 14. Desempeño del modelo K-NN utilizando diferentes valores de K.

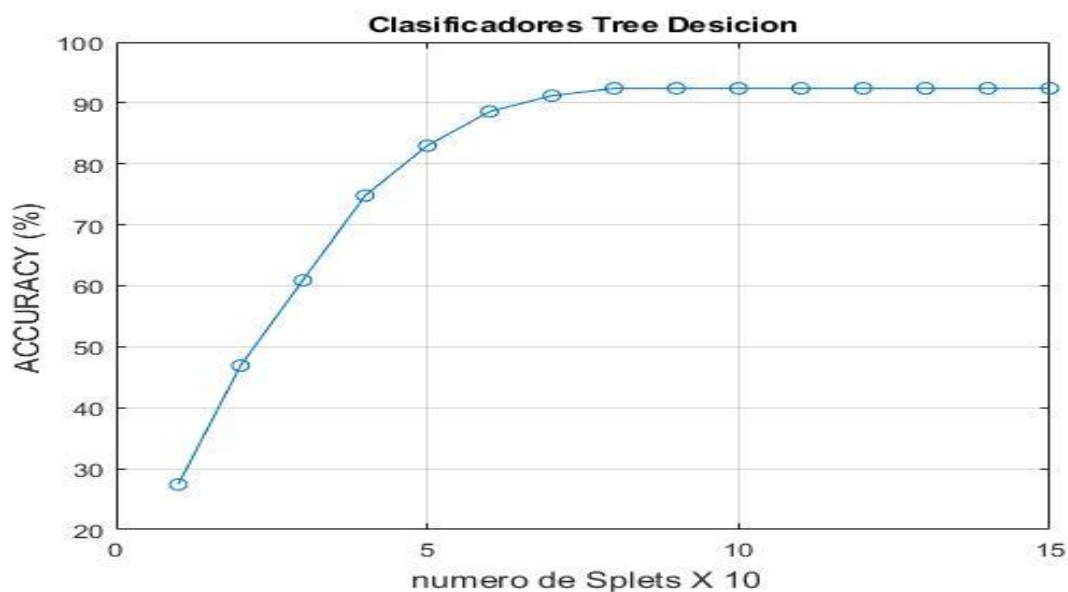


Fuente: esta investigación.

Para el clasificador de árbol de decisiones se realizó el entrenamiento variando el número de splets entre valores de 10 a 150 con intervalos de 10, en la **Figura 15**

se observa la relación ente porcentaje de clasificación y numero de splets. Para splets superiores 80 el modelo no mostro mejoras los resultados.

Figura 15. Modelo decision tree con diferente numero de splets.

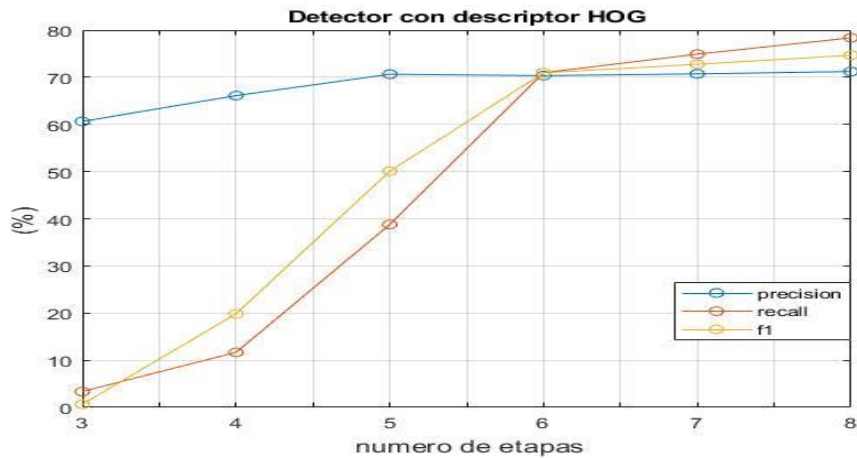


Fuente: esta investigación.

3.2. RESULTADOS DE LOS DETECTORES DE PLACA

Para la prueba 1 consistió en entrenar el detector de placas con descriptores HOG se realizó el entrenamiento de los modelos variando el número de etapas, como resultado el modelo con 8 etapas presento el mejor desempeño con relación a las métricas establecidas. Se obtuvieron valores de recall de 71.29%, precisión de 78.36% y una medida F1 de 75.66%. En la **Figura 16** se aprecia la relación de precisión, recall y medida F1 con el número de etapas.

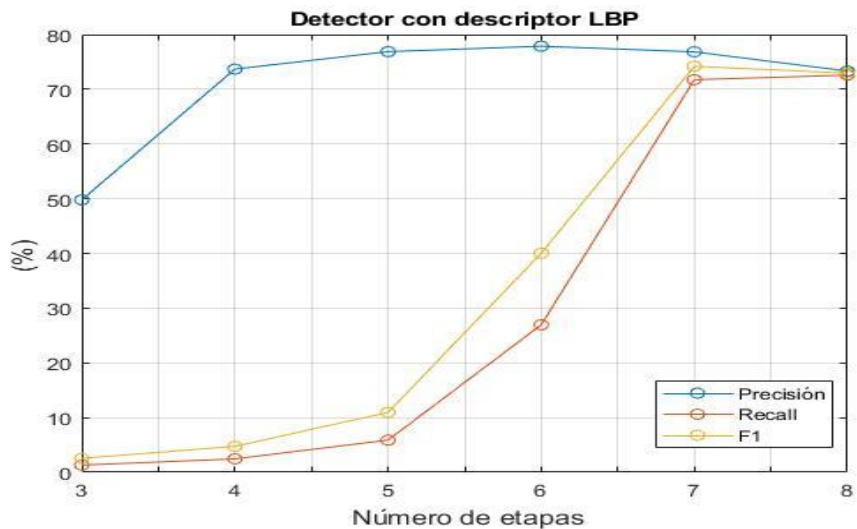
Figura 16. Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores HOG.



Fuente: esta investigación.

Para la prueba 2 donde se desarrolló el entrenamiento del detector con descriptores LBP se realizó con diferentes etapas, donde se obtuvo mejor resultado con el modelo de siete etapas en relación a las métricas establecidas. se obtuvieron valores de recall de 76.84%, precisión de 71.74% y una medida F1 de 74.20%. En la **Figura 17** se puede apreciar la relación que presenta la precisión, recall y medida F1 con el número de etapas.

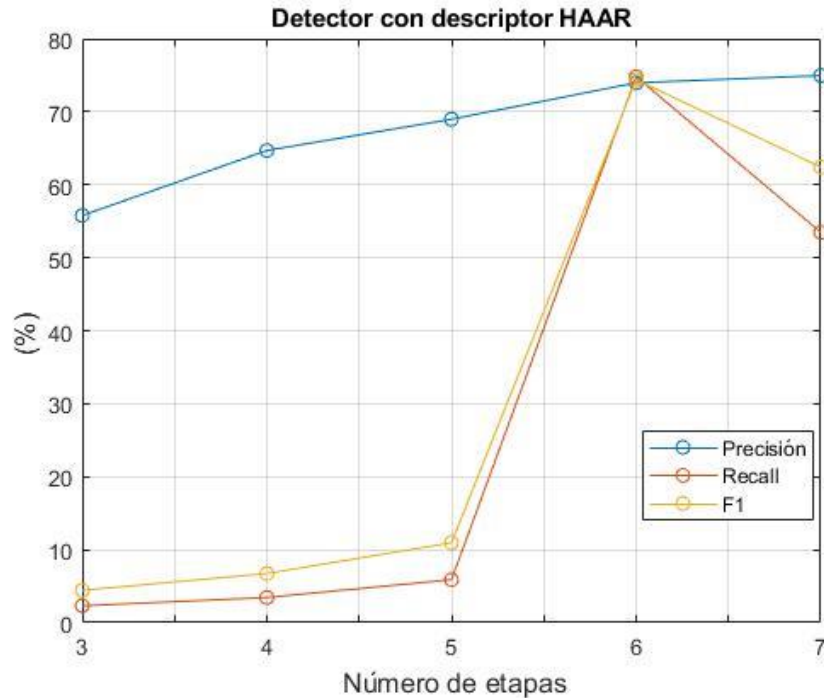
Figura 17: Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores LBP.



Fuente: esta investigación.

De igual forma la prueba 3 se desarrolló con descriptores Haar en donde el número de etapas se varia y como resultado el detector con un total de siete etapas presento un mayor rendimiento en relación a las métricas estipuladas con valores de recall 73.97%, precisión de 74.80% y una medida F1 de 74,38% como parámetros de evaluación propuestos. En la **Figura 18** se puede apreciar la relación que presenta la precisión, recall y medida F1 con el número de etapas.

Figura 18. Relación entre el número de etapas y valores de P, R y F1 para el modelo entrenado con descriptores Haar.



Fuente: esta investigación.

Se puede apreciar para los modelos de detección de placas que el número de etapas que se implementa es crucial para un buen detector. A mayor número de etapas el índice de precisión aumenta (menos falsos positivos), no obstante, si se implementa demasiadas etapas el recall aumenta, pero aumenta el riesgo de un falso negativo generando una no detección de la placa. El tiempo de entrenamiento aumenta considerablemente con relación al número de etapas, en promedio cada etapa demora alrededor de media hora de entrenamiento.

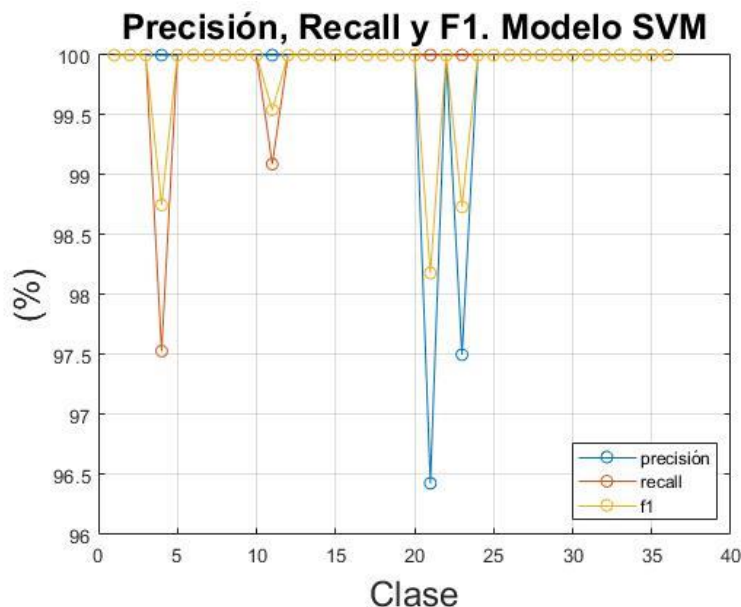
De los modelos entrenados el entrenamiento con descriptores HOG y 7 etapas presenta una mejor relación de F1 y R, lo que indica que tiene un menor riesgo de realizar una falsa detección negativa.

3.3. COMPARACIÓN DE LOS MODELOS PARA LA CLASIFICACIÓN DE CARACTERES.

En este apartado se da a conocer los resultados de los diferentes clasificadores, el entrenamiento se realizó con descriptores HOG que consta de un conjunto de 942 características por cada imagen para una matriz de tamaño 3960x972.

El modelo SVM: entrenado con un Kernel lineal, presento mejores resultados en comparación a los demás tipos de Kernel. Generando valores promedio de accuary de 99.3%, recall de 99.58%, precisión de 99.6% y un puntaje F1 de 99.59%. En la **Figura 19** se puede apreciar los resultados de recall, precisión y medida F1 para cada una de las clases, esta prueba se realiza con un conjunto de 20 muestras de cada clase un total de 720 imágenes.

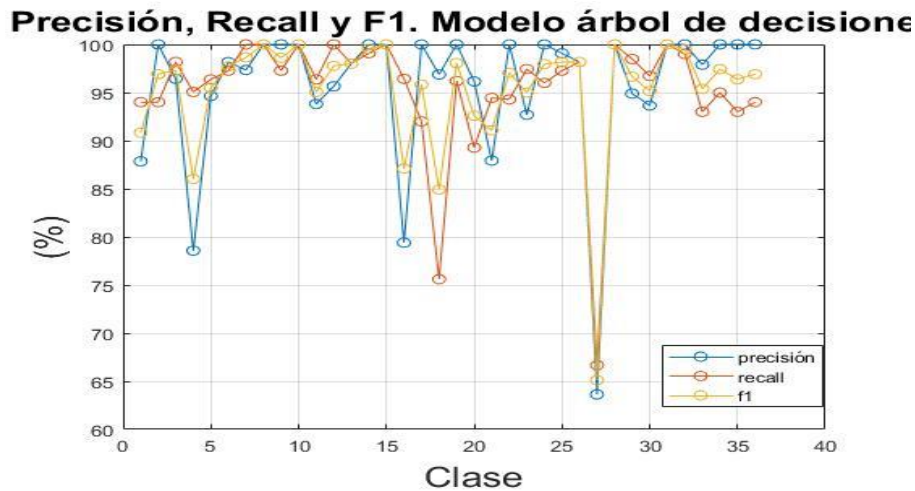
Figura 19. Resultados de P, R y F1 para cada clase. La grafica muestra los valores de R, P y F1 para clase diferente.



Fuente: esta Investigación.

El clasificador Decision Tree: fue entrenado con diferentes números de splets. El modelo entrenado con un número de 90 splets tiene un mejor desempeño en comparación a los demás, si se utiliza las métricas establecidas para su valoración. Presento unos valores de accuary de 94.4%, recall de 94.62%, precisión de 95.11% y un puntaje F1 de 94.86%, en la **Figura 20** se muestra el resultado de recall, precisión y medida F1 para cada clase.

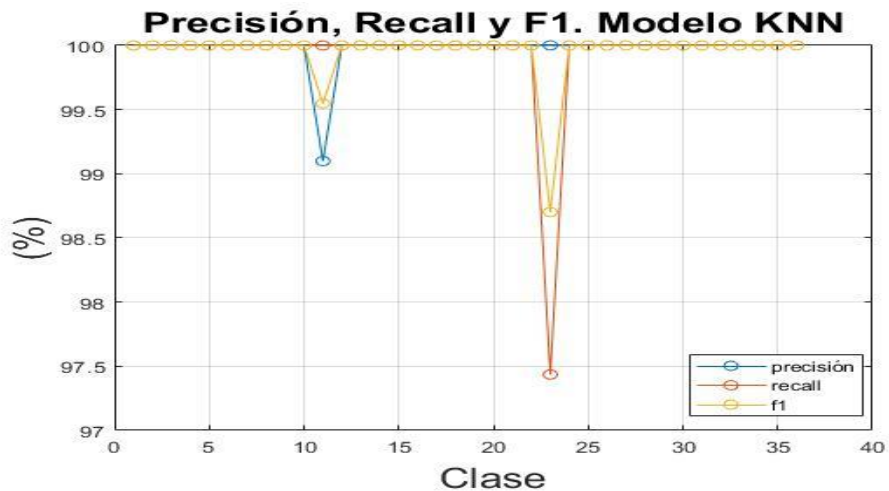
Figura 20. Resultados de R, P y medida F1 para cada clase utilizando el clasificador de árbol de decisiones.



Fuente: esta investigación

El modelo K-NN: se entrenó con diferentes valores del parámetro K. los mejores resultados se aprecian cuando se define K=1 obteniendo valores de accuary 99.2%, recall de 99.72%, precisión de 99.74% y un puntaje F1 de 99.73%, en la **Figura 21** se muestra el resultado de recall, precisión y medida f1 para cada clase.

Figura 21. Resultados de R, P y medida F1 para cada clase utilizando el clasificador KNN.



Fuente: esta investigación.

3.4. RESULTADOS OBTENIDOS EN LA CLASIFICACIÓN DE CARACTERES.

Los resultados de las pruebas 4,5 y 6 se presentan en la Tabla 6, se aprecia los valores de recall (R), precisión (P) y medida F1.

TABLA 6. RESULTADOS PRUEBAS 4,5 y 6.

Etiqueta	Clase	K-NN			SVM			TREE		
		R (%)	P (%)	F1 (%)	R (%)	P (%)	F1 (%)	R (%)	P (%)	F1 (%)
1	0	100	100	100	100	100	100	85	85	85
2	1	100	100	100	100	100	100	90	94,73	92,30
3	2	100	100	100	100	100	100	80	100	88,89
4	3	100	100	100	100	95,24	97,56	95	79,16	86,36
5	4	100	100	100	100	100	100	85	100	91,89
6	5	100	100	100	100	100	100	100	95,23	97,56
7	6	100	100	100	100	100	100	100	90,91	95,23
8	7	100	100	100	100	100	100	100	100	100
9	8	100	100	100	100	100	100	100	94,11	96,96
10	9	100	100	100	100	100	100	100	100	100
11	A	100	100	97,56	100	100	100	100	95,23	97,56
12	B	100	95,23	100	100	100	100	100	100	100
13	C	100	100	100	100	100	100	95	86,36	90,47
14	D	100	100	100	100	100	100	100	100	100
15	E	100	100	100	100	100	100	100	86,95	93,02
16	F	100	100	100	100	100	100	96,42	87,09	91,52
17	G	100	100	100	100	100	100	100	96,15	98,03
18	H	100	100	100	100	100	100	95	90,47	92,68
19	I	100	100	100	100	95,23	97,56	100	100	100
20	J	100	100	100	100	100	100	95	95	95
21	K	100	100	100	100	100	100	90	85,71	87,80
22	L	100	100	100	100	100	100	100	100	100
23	M	100	100	100	100	100	100	95	100	97,43
24	N	95	100	97,435	95	100	97,43	95	100	97,43
25	O	100	100	100	100	100	100	95	86,36	90,47
26	P	100	100	100	100	100	100	100	90,90	95,23
27	Q	100	100	100	100	100	100	80	94,11	86,48
28	R	100	100	100	100	100	100	100	100	100
29	S	100	100	100	100	100	100	95	100	97,43
30	T	100	100	100	100	100	100	95	95	95
31	U	100	100	100	100	100	100	100	95,24	97,56
32	V	100	100	100	100	100	100	90	100	94,74
33	W	100	100	100	100	100	100	100	100	100
34	X	95	100	97,43	95	100	97,435	85	100	91,89

35	Y	100	95,238	97,56	95	95	95	95	100	97,43
36	Z	100	100	100	100	100	100	75	100	85,71

Fuente: esta investigación.

3.5. RESULTADOS OBTENIDOS EN LA IDENTIFICACION Y RECONOCIMINETO DE PLACAS SOBRE UN SET DIFERENTE DE IMÁGENES.

Se realizaron pruebas sobre un nuevo conjunto de imágenes y así evaluar los detectores y clasificadores de caracteres con el fin de analizar el desempeño que presentan.

TABLA 7. Resultados de pruebas sobre un nuevo conjunto de imágenes.

IMAGEN	N° DE PLACA	SE IDENTIFICÓ PLACA SÍ/ NO			NÚMERO DE PLACA RECONOCIDO		
		HOG	LBP	HAAR	SVM	TREE	KNN
Im1.bmp	AUV 898	SÍ	SÍ	SÍ	AUV 898	ABV 898	AUV 898
Im2.bmp	CFT 029	SÍ	SÍ	SÍ	CFT 024	BFD 029	CFT 029
Im3.bmp	AVC 495	SÍ	SÍ	SÍ	AVC 095	AVV 495	AVC 495
Im4.bmp	GRD 959	SÍ	SÍ	SÍ	GRD 959	PRD 959	GRD 959
Im5.bmp	KAN 000	SÍ	SÍ	NO	KAN 000	KAN 000	KAN 000
Im6.bmp	AVE 678	SÍ	SÍ	SÍ	AUE 678	ARE 678	AVE 678
Im7.bmp	AVI 423	SÍ	SÍ	SÍ	AVU 423	AVT 483	AVI 423
Im8.bmp	AVE 540	SÍ	SÍ	SÍ	AVE 540	AVE 540	AVE 540
Im9.bmp	AUZ 042	SÍ	SÍ	SÍ	AUM 042	AUL 042	AUZ 042
Im10.bmp	AVJ 039	SÍ	SÍ	SÍ	AVW 039	AVJ 039	AVU 039
Im11.bmp	AVG 229	SÍ	SÍ	SÍ	AVG 229	AVF 889	AVG 229
Im12.bmp	RGM 489	SÍ	SÍ	SÍ	RGM489	JFT 489	RGM 489
Im13.bmp	AUT 380	SÍ	SÍ	SÍ	AUT 380	AUT 380	AUT 380
Im14.bmp	KHB 259	SÍ	SÍ	SÍ	KHB 259	KHB 959	KHB 259
Im15.bmp	IIN 363	SÍ	SÍ	SÍ	IIN 363	UTN 363	IIN 363
Im16.bmp	UGV 857	SÍ	SÍ	SÍ	UGV 857	UFV 857	UGV 857
Im17.bmp	AVF 396	SÍ	SÍ	SÍ	AVF 396	AVF 396	AVF 396
Im18.bmp	CNA 080	SÍ	SÍ	SÍ	CNN 080	VNA 080	CNA 080
Im19.bmp	DIS862	SÍ	SÍ	SÍ	DIS 862	DTS 862	DIS 862
Im20.bmp	AVJ 468	SÍ	SÍ	SÍ	AVW 468	AVJ 468	AVJ 468
Im21.bmp	UDQ 600	NO	SÍ	SÍ	UDO 600	UDB 600	UDU 600
Im22.bmp	FCU 691	SÍ	SÍ	NO	FCU 691	FCV 691	FCU 691

Im23.bmp	MMF 668	NO	NO	NO	--- ----	--- ----	--- ----
Im24.bmp	STT 670	SÍ	SÍ	SÍ	STT 670	SLL 670	STT 670
Im25.bmp	BMQ 113	SÍ	SÍ	SÍ	BUQ 113	BUO 113	BUQ 113
Im26.bmp	AUN 198	SÍ	SÍ	SÍ	AUN 198	AUV198	AUN 198
Im27.bmp	AUW 268	SÍ	SÍ	SÍ	AUW268	AUM 268	AUW 268
Im28.bmp	BEL 667	SÍ	SÍ	SÍ	BEL 667	BEL 667	BEL 667
Im29.bmp	KMN 450	SÍ	SÍ	SÍ	---- --	--- ---	--- ---
Im30.bmp	CUN 683	SÍ	SÍ	SÍ	CUN 688	CUN 683	CUN 680
Im31.bmp	CMB 405	SÍ	SÍ	SÍ	---- --	---- --	---- --
Im32.bmp	AUW 801	SÍ	SÍ	SÍ	AUW801	AUB 801	AUW 801
Im33.bmp	BDI 862	SÍ	SÍ	SÍ	BDO 862	BDI 862	BDI 862
Im34.bmp	BGB 787	SÍ	SÍ	SÍ	BGB 787	BFB 787	BGB 787
Im35.bmp	IDK 029	SÍ	SÍ	SÍ	IDK 029	FDK 089	IDK 029
Im36.bmp	NPE 198	SÍ	SÍ	SÍ	NPE 199	NPE 199	NPE 199
Im37.bmp	NWA 263	SÍ	SÍ	SÍ	NWA263	NWA 266	NWA 263
Im38.bmp	CBS 835	SÍ	SÍ	SÍ	CBB 835	CBB 835	CBS 835
Im39.bmp	AVK 395	SÍ	SÍ	SÍ	AVK 395	WVK 395	AVK 395
Im40.bmp	AUZ 330	SÍ	SÍ	SÍ	AUZ 330	AUZ 330	AUZ 330

Fuente: esta investigación.

En la tabla 7, se puede observar que el modelo SVM ha clasificado 12 caracteres erróneos, el modelo árbol de decisiones realizó 38 caracteres mal clasificados mientras que el modelo KNN presentó mejores resultados con solo tres caracteres mal clasificados.

De las 40 imágenes de prueba, el detector entrenado con descriptores HOG no detectó 2 placas, el modelo entrenado con descriptores HAAR no detectó tres placas, mientras que el modelo entrenado con descriptores LBP solo no detectó una placa. De las placas identificadas no se logró identificar los caracteres de la placa debido a una mala segmentación de los caracteres.

3.6. DESARROLLO EN PYTHON

Dentro del trabajo desarrollado se realizó la implementación de un prototipo sobre una tarjeta de desarrollo Raspberry Pi 3 para la detección y reconocimiento de placas vehiculares teniendo como base para prueba de funcionamiento una escena de video con un total de 20 vehículos, que se capturó con una cámara en una posición fija a una altura de un metro aproximadamente. Para este proceso se utilizó

un detector entrenado con descriptores LBP con 7 etapas junto a un clasificador SVM Como resultados en la etapa de detección el prototipo identificó la posición de las placas de manera correcta en 19 vehículos, para el proceso de reconocimiento se obtuvo dos errores al momento de la clasificación de caracteres en dos placas como se muestra en la Tabla 8.

TABLA 8: Resultados del prototipo realizado en python.

N° DE PLACA	SE IDENTIFICÓ PLACA SI/ NO (LBP)	NÚMERO DE PLACA RECONOCIDO (SVM)
SVP 894	SI	SVP 894
SVR 231	NO	--- ---
ITW 230	SÍ	ITW 230
IIM 831	SÍ	IIM 831
SOY 155	SÍ	SOY 155
CMI 348	SÍ	CMI 348
AUO 048	SÍ	AUO 048
SVP 590	SÍ	SVP 590
SDP 182	SÍ	SOP 182
BMN 296	SÍ	BMN 296
BYI 913	SÍ	BYI 913
CSH 372	SÍ	ISH 372
AUW 341	SÍ	AUW 341
CBV 913	SÍ	CBV 913
SEY 061	SÍ	SEY 061
AUL 239	SÍ	AUL 239
DVZ 502	SÍ	DVZ 502
FBK 842	SÍ	FBK 842
AUV 451	SÍ	AUV 451
FHV 070	SÍ	FHV 070

Fuente: esta investigación

3.7. INTERFAZ DE USUARIO.

Los resultados obtenidos pueden visualizarse sobre una interfaz que se muestra en la **Figura 22**, se ingresa una nueva imagen y probar sobre esta los modelos de detección de placas entrenados, realizar un breve pre-procesamiento, caracterizar las regiones de interés, clasificar los caracteres y por último la interfaz retorna un string con la placa identificada del Vehículo.

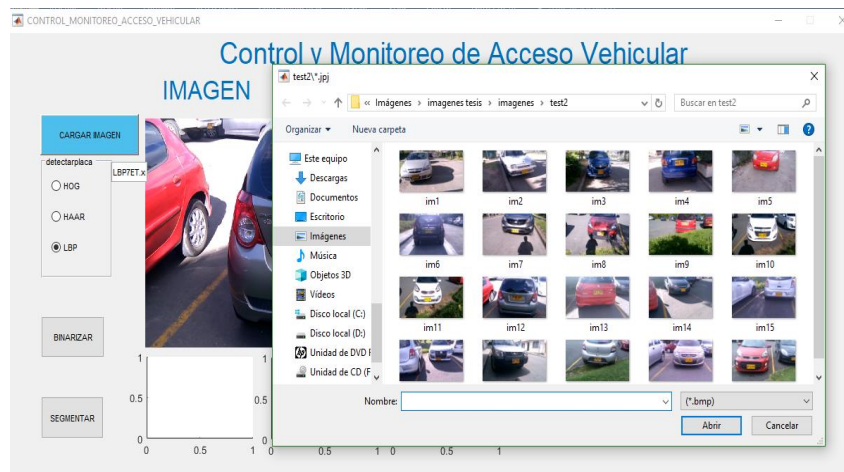
Figura 22. Interfaz de usuario. La figura muestra la interfaz de usuario para el reconocimiento de matrículas.



Fuente: esta investigación.

Como se muestra en la **Figura 23**, el botón cargar imagen permite adjuntar una nueva imagen a la interfaz desde la una carpeta con nombre **Test** que debe estar almacenada en la misma dirección de la interfaz, las imágenes deben estar guardadas en un formato .bmp o .jpg (no reconocerá imágenes de un formato diferente), dentro de la carpeta existen algunas imágenes para realizar pruebas de funcionamiento.

Figura 23. Función de cargar una nueva imagen.



Fuente: esta investigación.

La detección de placa tiene los modelos de detección entrenados guardados en una carpeta llamada detectores, en la interfaz se puede elegir cual modelo se desea utilizar para realizar la identificación de la matrícula vehicular. Los modelos cargados son: HOG con 8 etapas, LBP con 7 etapas y HAAR con 7 etapas, si se desea entrenar un nuevo modelo se puede utilizar la función train_model, también almacenada dentro de la carpeta. En la **figura 24** se puede observar cada tipo de detector en funcionamiento, generando un recuadro al contorno de la placa de un color diferente para cada detector; azul HOG, verde LBP y rojo HAAR.

Figura 24. Detectores de placa en funcionamiento. Cada detector se diferencia por el color del recuadro de detección, azul para HOG, rojo para HAAR y verde para LBP



Fuente: esta investigación.

El botón binarizar permitió convertir una imagen a tipo binario y realizar un procesamiento sobre la sección de la placa detectada. En la **Figura 25** se puede observar la puesta en marcha de esta función.

Figura 25. Botón Binarizar en funcionamiento.



Fuente: esta investigación.

Con el botón de segmentar se realizó el proceso de segmentación de los caracteres de la placa, En la **Figura 26** se presenta el resultado de binarizar seguido de la segmentación de los caracteres.

Figura 26. Botón segmentar en funcionamiento. Separación de caracteres de la placa.



Fuente: esta investigación.

El botón caracterizar cumple con dos funciones dentro de la interfaz, primero realiza el recorte de cada carácter y posteriormente extrae los descriptores HOG para cada uno de los mismos. El resultado de este proceso se muestra en la **Figura 27**, se utiliza un tamaño de celda y bloque de 6x6 para a extracción de los descriptores.

Figura 27. Extracción de características HOG de los caracteres



Fuente: esta investigación

En la parte que corresponde a la selección de los caracteres, se puede elegir entre los métodos que se desarrollaron SVM, K-NN y decisión Tree, estos modelos están almacenados dentro de la carpeta clasificadores, si se desea entrenar un nuevo modelo se puede realizar con función `train_clasificador` modificando los parámetros deseados para cada tipo, en la **Figura 28** se puede observar el resultado de la clasificación de los caracteres para cada modelo, nótese que el modelo SVM ha clasificado mal un carácter mientras que el modelo árbol de decisiones y K-NN han clasificado correctamente los caracteres de la placa.

Figura 28. Métodos que se desarrollaron SVM, K-NN y decisión Tree.

The figure displays three sequential screenshots of a classification interface. Each screenshot features a blue header bar with the text 'A V F 3 9 6'. Below the header, the word 'CLASIFICACION' is displayed. The interface contains three radio button options: SVM, DE-Tree, and K-NN. In the first screenshot, the 'K-NN' option is selected. In the second screenshot, the 'SVM' option is selected. In the third screenshot, the 'DE-Tree' option is selected.

Fuente: esta investigación.

4. CONCLUSIONES

Los descriptores LBP presentaron un mejor desempeño para la detección de placas vehiculares junto a un clasificador en cascada dado que estos pueden diferenciar la textura de los objetos. Esta combinación entre clasificadores en cascada y estos descriptores se puede utilizar en la detección de diferentes objetos, no solo placas.

Para la medida de desempeño de sensibilidad en la detección de placas, no presentó amplia variación de un modelo entrenado a otro, no obstante, la diferencia de la sensibilidad y medida F1 fueron las que permitieron establecer comparación entre métodos.

El pre-procesamiento y binarización de la imagen fue una etapa de alta importancia, dado que si se hace de manera correcta los caracteres se segmentarán adecuadamente, pero si este proceso presenta fallas los caracteres serán confundidos con el fondo de la placa. Por lo anterior, fue necesario utilizar métodos para calcular el umbral local, dado que las placas presentan variación en la intensidad de luz sobre ellas.

Los descriptores HOG fueron los más eficientes para la caracterización de los caracteres, puesto que están asociados a la forma del objeto y presentaron mejores resultados en comparación a otros descriptores, no obstante, son computacionalmente costosos de calcular

El modelo de clasificación K-NN y SVM fueron los modelos que mejor se acoplaron a los descriptores HOG al momento de clasificar los caracteres sobre las placas vehiculares, mientras que el modelo de árbol de decisiones presentó varias fallas al momento de clasificar.

5. RECOMENDACIONES

El tiempo de entrenamiento de los detectores es relativamente grande entre 6 a 10 horas, por tanto, se recomienda utilizar un equipo con buenas características de memoria y velocidad para disminuir el tiempo de entrenamiento, y realizar el entrenamiento directamente sobre OpenCV utilizando Linux como sistema operativo.

Como trabajo futuro, aparte de explorar otros métodos de clasificación y de detección, se trabajará con bases de datos de imágenes de menor resolución para incrementar el tiempo de ejecución de la aplicación, así como también poder realizar la identificación de placas vehiculares en tiempo real.

Los descriptores HOG son más recomendados para la clasificación de los caracteres, puesto que están asociados a la forma del objeto y presentaron mejores resultados a la hora de realizar el entrenamiento en comparación a otros descriptores, no obstante, son computacionalmente costosos de calcular.

BIBLIOGRAFÍA

ALVARES, Mayra. Análisis, diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la Universidad Politécnica. Cuenca, 2014, 144p. Proyecto de Grado para la obtención del título de Ingeniero de sistemas. Universidad politécnica salesiana. Ingeniería de sistemas.

BERNARDO, John y RABOY, Amor. Sistema de Monitoreo e Información de la Placa del Vehículo utilizando la técnica de reconocimiento óptico de caracteres (OC). 2015 disponible en: <http://dx.doi.org/10.2139/ssrn.2797304>

BARCIA, orlando. Reconocimiento automático a través de visión artificial, correlación estadística y Matlab aplicado a las matrículas de vehículos. Ecuador, 2014, 147p. Universidad politécnica salesiana. 2014. [Consultado 18 de noviembre de 2018]. Disponible en: repositorio espe <http://dspace.ups.edu.ec/handle/123456789/10981>.

CONSTANTE, Patricia, et al. Artificial Vision Techniques to Optimize Strawberry's Industrial Classification. En: IEEE Latin America Transactions. Vol. 14, No 6 (Jun, 2016); p. 2576-2581.

CHEN, Tao y LU, Shijian. Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression. En: IEEE Transactions on Vehicular Technology. Vol. 65, no 6 (jun, 2015); p. 4006-4015.

CRUZ, Angel y GONZÁLEZ, Fabio A. Aprendizaje Transductivo para la Clasificación de Imágenes Médicas de Radiología. 2008.

DIMIERI, Leonardo. Aspectos físicos de la visión humana. (2015).

DÉNIZ, Oscar, et al. Face recognition using histograms of oriented gradients. En: Pattern Recognition Letters. Vol. 32, No 12 (Sep, 2011); p. 1598-1603.

DU, Shan, et al. Automatic license plate recognition (ALPR): A state-of-the-art review. En: IEEE Transactions on circuits and systems for video technology, Vol. 23, no 2 (feb, 2013) p. 311-325.

ESPINOSA, Alejandro; ENRIQUE, Carlos y VENEGAS VILLACIS, Carlos Luis. Optimización e integración de una nariz electrónica autónoma embebida en un sistema robótico para la identificación de sustancias explosivas como TNT y pólvora base doble en ambientes controlados. Sangolqui, 2017, 128p. trabajo de grado de Ingeniería Electrónica Automatización y Control. Universidad de las fuerzas armadas, departamento de eléctrica y electrónica.

ESPINOZA VÁSQUEZ, Gerardo Alfonso Joel. Sistema de reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio. Lima, 2014, 67p. trabajo de grado de ingeniería. Pontificia universidad católica del Perú. Facultad de ciencias e ingeniería.

GARCÍA FUENTES, JORGE, et al. Visión artificial aplicada al control de la calidad. En: 3C Tecnología. Vol. 3, No 5 (Dic, 2014); p. 297-308.

GONZALO, Pajares Martinsanz, et al. Vision por computador. Imagenes digitales y aplicaciones. 2008.

HUÉRFANO, Oscar Mauricio y GONZÁLEZ HUERTAS, Oscar Eduardo. Análisis técnico de las soluciones disponibles en el mercado para el montaje de peajes electrónicos. Bogotá, 2015, 92p. Trabajo de grado de electrónica. Universidad Santo Thomas. Facultad de ingeniería. Departamento de electrónica,

JARA ESTUPIÑAN, Jefferson; GIRAL, Diego y MARTÍNEZ SANTA, Fernando. Implementation of algorithms based on support vector machine (SVM) for electric systems: En:Tecnura. Vol. 20, No 48 (Jun, 2016); p. 149-170.

KULKARNI, Prathamesh, et al. Automatic number plate recognition (ANPR) system for indian conditions. En: 19th International Conference Radioelektronika. IEEE. (Abril, 2009); p. 111-114.

LEÓN VÁSQUEZ, Iván Andrés y VÁSQUEZ TAPIA, Adrián Jonatan. Diseño e implementación de un Robot Móvil Autónomo y Teleoperado para labores agrícolas. Fase 1: fumigación para plantaciones de rosas. Sangolqui, 2013, 177p. Carrera de Ingeniería Electrónica Automatización y Control. Universidad de las fuerzas armadas, departamento de eléctrica y electrónica.

LÓPEZ MÁRQUEZ, Gabriel Andrés. Sistema inteligente de reconocimiento de patrones con visión artificial para la alerta automática de intrusos en las áreas de almacenamiento de las PYMES. Ambato, 2016, 95p. Ingeniería en Electrónica y Comunicaciones. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Departamento de Ingeniería Electrónica y Comunicaciones.

MARTÍNEZ, Claudia Cruz, et al. Sistema de Navegación Reactiva Difusa para Giros Suaves de Plataformas Móviles Empleando el Kinect. En: Revista electrónica de Computación, Informática, Biomédica y Electrónica. Vol. 5, No 3 (abril, 2016).

MAGRO, Rafael. Binarización de imágenes digitales y su algoritmia como herramienta aplicada a la ilustración entomológica. En: Boletín de la Sociedad Entomológica Aragonesa. Vol. 53, No 53 (Dic, 2013); p. 443-464.

MITA, Takeshi; KANEKO, Toshimitsu y HORI, Osamu. Joint haar-like features for face detection. En: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Vol 2. (2005); p. 1619-1626.

MORFIN-MONTES, Francisco. Sistema de reconocimiento de automóviles en centros comerciales SRACC. Mexico D.F., 2013, 94p. Proyecto de Grado para la obtención del título en Ingeniería en sistemas computacionales. Instituto politécnico nacional escuela superior de computo. Facultad de ingeniería.

OCAMPO, María Lourdes García y JARA, Eustaquio Alcides Martínez. Sistema de reconocimiento automático de placa automotriz paraguaya. En: FPUNE Scientific. No 10 (Ago,2016).

ORTIZ, Manuel Martín. Procesamiento digital de imágenes. Benemérita universidad Autónoma de Puebla, 2013, p 114.

SOBRADO MALPARTIDA, Eddie Ángel. Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. [En línea].[10 de enero del 2018] disponible en: <https://pdfs.semanticscholar.org/fdd7/15bb54a9eb4373f18390b8dbce37f9000941.pdf>

PATEL, Chirag; SHAH, Dipti y PATEL, Atul. Automatic number plate recognition system (anpr). En: A survey. International Journal of Computer Applications. Vol. 69, No 9 (May, 2013); p. 21-33.

PURANIC, Aniruddh; DEEPAK, K. T.; UMADEVI, V. Vehicle number plate recognition system: a literature review and implementation using template matching. En: International Journal of Computer Applications. Vol. 134, No 1 (Ene, 2016); p. 12-16.

ROJAS PASTRANA, Dennis Lida. Desarrollo de un sistema de reconocimiento de placas y su influencia en la detección de vehículos robados en la Municipalidad de San Isidro. Lima, 2017,86p. Tesis para optar el Título de Ingeniero de Sistemas y Cómputo. Universidad Inca Garcilaso de la Vega. Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones.

SANABRIA, John J. y ARCHILA, John F. Detección y análisis de movimiento usando visión artificial. En: Scientia et technica. Vol. 16, No 49 (Dic, 2011);p. 180-188.

SALAZAR, Rubén Darío Vásquez y MESA, Ahmed Alejandro Cardona; OSORIO, Leidy Yohana Ocampo. Propuesta de sistema de conteo de alevines de tilapia roja de bajo costo usando técnicas de visión artificial. En: Revista Politécnica. Vol. 11, No 21 (Dic, 2015); p. 85-95.

SANTAMARÍA, María Violeta Bances y MOSCOL, Mario Fernando Ramos. Semáforos inteligentes para la regulación del tráfico vehicular. En: Ingeniería: Ciencia, Tecnología e Innovación. Vol. 1, No 1 (Jun, 2015); p. 37-45.

TIAN, Jiangmin, et al. A two-stage character segmentation method for Chinese license plate. En: Computers & Electrical Engineering. Vol. 46, (Ago, 2015); p. 539-553.

URREGO, Germán Enrique, et al. Adquisición de variables de tráfico vehicular usando visión por computador. En: Revista de Ingeniería. No 30 (Nov, 2009); p. 7-15.

VIOLA, Paul, et al. Rapid object detection using a boosted cascade of simple features. En: CVPR (1). Vol. 1, (May, 2004); p. 511-518

VÁSQUEZ, Julian y MELO, Jersson. Sistema automático de reconocimiento de placas vehiculares. Bogota, 2018, 76p. Proyecto de Grado para la obtención del título en Ingeniería de Telecomunicaciones. Universidad cooperativa de Colombia. Facultad de ingeniería.

WATANABE, Tomoki; ITO, Satoshi y YOKOI, Kentaro. Co-occurrence histograms of oriented gradients for human detection. En: IPSJ Transactions on Computer Vision and Applications. Vol. 2, (Mar, 2010); p. 39-47.

ZHANG, Lun, et al. Face detection based on multi-block lbp representation. En: International conference on biometrics. Springer. (2007). p. 11-18.

ANEXOS

Esta sección ha sido destinada para resultados tangibles logrados con el trabajo realizado en esta tesis. Estos anexos contienen una descripción más ampliada de los resultados mencionados en la sección 6.

ANEXO 1. PSEUDOCODIGO RECONOCIMIENTO DE PLACAS VEHICULARES.

Algoritmo
<ol style="list-style-type: none">1. Cargar una nueva imagen2. Aplicar el detector sobre la imagen IF detectar placa=TRUE: Continuar con el proceso de extraer caracteres Pre-procesar la imagen Binarizar Segmentar Ordenar los caracteres de izquierda a derecha Extraer descriptor HOG Clasificar Concatenar los resultados del clasificador ELSE: Volver a cargar una imagen3. El proceso termina.

ANEXO 2. CÓDIGO DE EXTRACCIÓN DE PARAMETROS DE EVALUACIÓN PARA EL DETECTOR

Algoritmo para extraer los parámetros de evaluación

```
function [F1, P, R] = EvaluarDetector(G,D)
    %G Lista de las cajas donde estan las plac
    %D Lista de las cajas que se detectaron con los modelos
    SMTraslapeD = 0;
    Ndet = 0;
    SMTraslapeG = 0;
    Nobj = 0;
    for im=1:length(D)
        ITraslape=bboxOverlapRatio(G(im).objectBoundingBoxes,D(im).objectBoundingBoxes);
        ndet = size(D(im).objectBoundingBoxes,1);
        Ndet = Ndet + ndet;
        nobj = size(G(im).objectBoundingBoxes,1);
        Nobj = Nobj + nobj;
        if ndet>0 && nobj>0
            % Calculando Precisión
            SMTraslapeD=SMTraslapeD+sum(max(IndiceTraslape,[],1));
            % Calculando Sensibilidad R
            SMTraslapeG = SMTraslapeG +Sum(max(ITraslape,[],2));
        end
    end
    % Salidas F1 = Medida F1 P R
    P = sumMejorTraslapeD/Ndet;
    R = sumMejorTraslapeG/Nobj;
    F1 = 2*P*R/(P+R);
```

ANEXO 3. CÓDIGO PYTHON

Código realizado para el reconocimiento de placas en el lenguaje de programación Python.

Algoritmo para detección de características Python

```
import numpy as np
import cv2
import PIL
import os
from matplotlib import pyplot as plt
from PIL import Image
from scipy.misc import imread
import time
import matplotlib.patches as mpatches
from skimage import morphology
from skimage.color import rgb2gray
from skimage import io
from skimage.filters import threshold_otsu, threshold_local
from skimage.measure import label, regionprops
from skimage.color import label2rgb
import joblib
from HOG.HOGpatterns import HOGPatterns
from time import time
filename = '/home/pi/Documents/anpr/caracter_classifier.joblib.pkl'
clf = joblib.load(filename)
desc = HOGPatterns(6, 6)
dat=[]
predict=[]
placas_in=[]
def convert(s):
    new=""
    for x in s:
        new+=x
    return new

def detect_show(frame):

    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    gray = cv2.equalizeHist(gray)
    placas_can=placa_cascade.detectMultiScale(gray,minSize=(32,63),maxSize=(150,200))
    for(x,y,w,h) in placas_can:
        frame=cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,255),2)
        roi_gray=gray[y:y+h, x:x+w]
    return placas_can,frame
```

```

placa_cascade=cv2.CascadeClassifier('HAAR6ET.xml')
pl=cv2.CascadeClassifier.empty(placa_cascade)
rootDir = '/home/pi/Documents/anpr/imagenes/test/'
# por cada archivo en el folder hacer
for dirName, subDirList, fileList in os.walk(rootDir,topdown=True):
for fname in fileList:
im=cv2.imread(dirName+'/'+fname,1)
l=Image.open(dirName+'/'+fname)
print(fname)
#detectar placa
a,img=detect_show(im)
print(a)
if len(a)>0:
fig,ax=plt.subplots(1,3,figsize=(10,6))
ax[0].imshow(img)
box = (a[0][0], a[0][1], a[0][2]+a[0][0],a[0][3]+a[0][1])
placa_det = l.crop(box)
placa_det = placa_det.resize((320, 240), PIL.Image.ANTIALIAS)
ax[1].imshow(placa_det )
placa_det.save("placa.bmp","BMP")
l=Image.open('placa.bmp')
img=io.imread('placa.bmp')
#prep-rocesamiento y binarización
gray=rgb2gray(img)
thd=threshold_local(gray,49,'gaussian')
imgbin=gray<thd
im = morphology.remove_small_objects(imgbin,400)
h=morphology.rectangle(2,2)
h2=morphology.rectangle(3,3)
im_binarize= morphology.erosion(im,h)
im_binarize = morphology.binary_erosion(im_binarize,h)
im_binarize = morphology.binary_dilation(im_binarize,h2)
label_image = label(im_binarize )
image_label_overlay = label2rgb(label_image, image=im_binarize )
ax[2].imshow(image_label_overlay)

properties=regionprops(label_image)
properties = sorted(properties, key=lambda p: p.bbox[1], reverse=False)
i=0
c=0
for region in properties:
minr, minc, maxr, maxc = region.bbox
bbox = properties[i].bbox
area = properties[i].area
bbox = (bbox[1], bbox[0], bbox[3], bbox[2])
#predict=['A','B','C','0','1','2']
if area >= 800 and maxc - minc<50:
rect = mpatches.Rectangle((minc, minr), maxc - minc, maxr - minr,

```

```

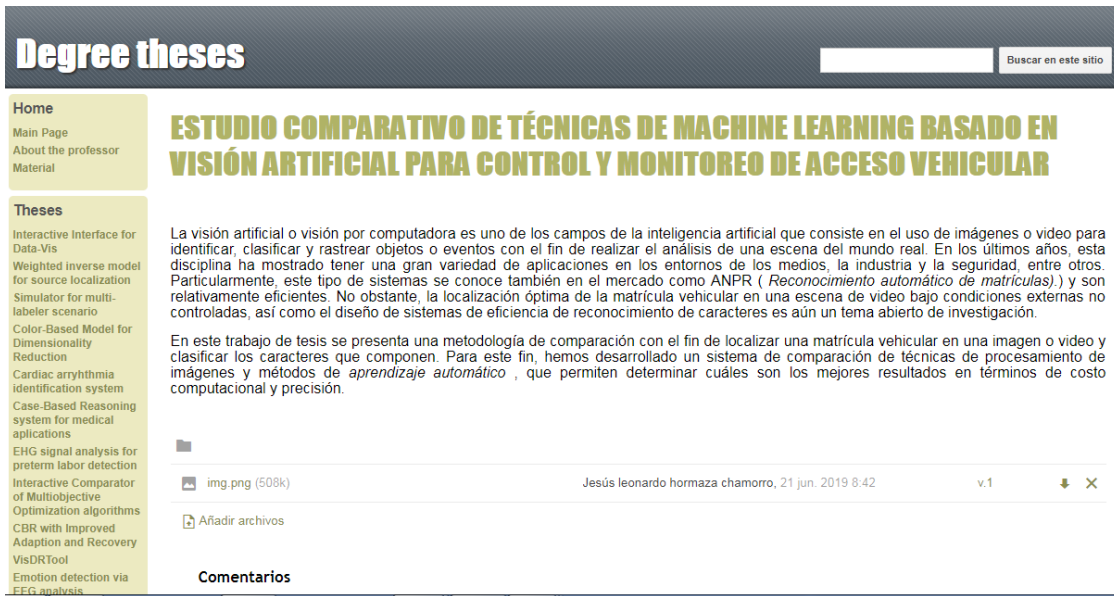
fill=False, edgecolor='blue', linewidth=2)
cr = l.crop(bbox)
cr = cr.resize((40, 80), PIL.Image.ANTIALIAS)

cr.save("l" + str(i) + ".bmp", "bmp")
image=cv2.imread("l" + str(i) + ".bmp")
#extraer caracteres HOG
fd, hog_image = desc.describe(image)
dat.append(fd)
#clasificar caracteres
predict.insert(c, clf.predict(dat))
c=c+1
i=i+1
k=len(predict)-1
print(k)
print(predict[k])
placa=convert(predict[k])
print(placa)
placas_in.append(placa)
predict.clear()
dat.clear()
#predict[0:5]=[predict[0],predict[1],predict[2],predict[3],predict[4],predict[5]]
else:
predict.clear()
dat.clear()
continue
plt.tight_layout()
plt.show()
cv2.waitKey()
cv2.destroyAllWindows()
print(placas_in)

```

ANEXO 4. PÁGINA WEB

Dentro del desarrollo de este proyecto, se contempla la creación de una página web en Google Sites, donde se puede encontrar información general acerca de la interfaz desarrollada, así como algunos scripts. Un manual de usuario explica el funcionamiento de la interfaz gráfica, son incluidos con el fin de dar un mejor entendimiento acerca del proyecto y fomentar la divulgación de los resultados obtenidos.



The screenshot shows a Google Sites page with a dark header containing the title "Degree theses" and a search bar. A left sidebar lists various thesis topics under "Home" and "Theses". The main content area features a large green heading: "ESTUDIO COMPARATIVO DE TÉCNICAS DE MACHINE LEARNING BASADO EN VISIÓN ARTIFICIAL PARA CONTROL Y MONITOREO DE ACCESO VEHICULAR". Below the heading is a paragraph of text in Spanish describing the field of artificial vision and the specific thesis work. At the bottom of the page, there is a file upload section showing a preview of an image file named "img.png (508k)" and a "Comentarios" (Comments) section.

Link: <https://sites.google.com/site/degreethesisdiegopeluffo/vehicle-monitoring-via-image-processing>