

# **Análisis e implementación de detección y reconocimiento de placas vehiculares**

**Resumen:** La detección y reconocimiento de placas vehiculares es un proceso realizado en múltiples etapas, el pre-procesamiento de imágenes, ubicación de placas, localización y reconocimiento de caracteres. Analizamos las diversas metodologías empleadas durante este proceso, sus características y resultados e implementamos un algoritmo que lo ejecuta.

**Palabras clave:** Imagen digital, Detección de placas, Aprendizaje de Máquinas, Ingeniería de Software, Reconocimiento de Patrones.

**Abstract:** Vehicle licence plate detection and recognition it's a multi stage process, image pre-processing, licence plate location, character location and recognition. We analyzed the multiple methodology used during this process, their features and results and implemented an algorithm that executes it.

**Keywords:** Digital image, Licence plate detection, Machine learning, Software engineering, Pattern recognition.

## **1. Introducción**

De acuerdo a las cifras del INEGI desde 1980 al 2021 en México el parque vehicular se ha incrementado desde poco más de 5 millones de

vehículos a 53 115 396 vehículos [9], con el incremento del parque vehicular surge la necesidad de adoptar sistemas de detección automática de placas confiables bajo todo tipo de condiciones que permitan la recopilación y análisis de dicha infor-

mación, los cuales pueden asistir desde control de acceso, seguridad y análisis de información para fines de estimación de tendencias.

En este artículo se analizan los diversos factores que intervienen en una detección certera de las placas vehiculares, los diferentes tipos de ruido encontrados como son el gaussiano o sal y pimienta y sus métodos de atenuación, problemas causados por la baja iluminación o sobre exposición de luz, así como el algoritmo empleado durante la localización y reconocimiento para crear un sistema resiliente de detección automática de placas vehiculares.

## 2. Trabajo relacionado

Los algoritmos LPR se pueden descomponer de manera más fina en 3 etapas, ubicación de la placa (LPL), segmentación de los caracteres y reconocimiento óptico de los mismos (OCR), dentro de las cuales se clasifican en 3 familias, basados en algoritmo matemático, basados en características de la placa y basados en aprendizaje profundo [4].

El procesamiento matemático de imágenes consiste en transformar una imagen a otra que extrae información útil para la aplicación, los principales algoritmos matemáticos se dividen en 5 clases principales, extracción de bordes, operaciones morfológicas, métodos híbridos, transformaciones y ventanas concéntricas desplazables (SCW).

1. Extracción de bordes. Aplican la convolución del filtro pasa altas para resaltar altas frecuencias y atenuar el resto, consisten esencialmente en 4 pasos.

- Aplicación del filtro para extraer bordes.
- Binarización de contornos.
- Análisis de resultados para extraer resultados.

- Validación por análisis de propiedades geométricas, de textura o densidad.

Son sencillos y rápidos pero susceptibles al ruido, cambios de iluminación y distancia.

2. Operaciones morfológicas. Es un conjunto de operadores no lineales para buscar pequeñas geometrías en la imagen llamadas elementos estructurales (SE), es robusto al ruido, son lentos, no adaptables a sistemas de tiempo real, pobres resultados en imágenes de bajo contraste, la elección de los SE es un factor importante influenciado y limitado por la distancia de sensores fijos.
3. Algoritmos híbridos. Combinan la detección de bordes con operaciones morfológicas, simplifican la imagen manteniendo solo los bordes, completan las formas aplicando operaciones morfológicas y validan usando criterios geométricos o de textura. son una mejora considerable, pero mantienen los problemas de la identificación de bordes bajo diferentes condiciones de luz y el tiempo de procesamiento de las operaciones morfológicas.
4. Transformaciones. Analiza las subclases basadas en transformadas de HOUGH (HT), de ondulaciones (WT) y de geometría generalizada (GST).
  - HT: es una técnica de reconocimiento de patrones simples, tiene malos resultados con imágenes con ruido y un alto tiempo de procesamiento, no es aplicable a tiempo real y cuando las placas no son rectangulares (ya sea por daños, oclusión o iluminación), pero sin dichas condiciones tiene resultados certeros y puede localizar múltiples placas en una sola imagen
  - WT: tomando la diferencia entre la placa y sus caracteres, no depende de la

distancia y pueden detectar múltiples placas al mismo tiempo, puede dar falsos resultados cuando el contraste entre los caracteres y su fondo no es muy claro por lo que son susceptibles al ruido.

- GST: busca la geometría local buscando 4 esquinas usando una red neuronal para validar la placa. Para adaptar el sistema a tiempo real es necesario tomar un número fijo de escaneos, considerando condiciones de luz y ruido, siendo susceptible a placas dañadas o esquinas ocultas.

5. SCW. Se basa en intensidades de pixel para binarizar la imagen, desde donde se toman regiones de interés que después son validadas, imágenes de baja resolución o pobre iluminación no son procesables.

Los algoritmos basados en las características de la placa se dividen en 3 tipos, basados en textura, color y cuantización vectorial (VQ).

1. Textura: Asumen que la placa tiene determinadas características que las distingue del resto de los objetos, buscando las siguientes características.
  - Múltiples caracteres en una región pequeña.
  - Caracteres y fondo con alto contraste.
  - Tamaño relativamente constante.
  - Inclinación en un determinado rango.

Dichas características son explotadas por métodos, probabilísticos, determinísticos o combinados, se ve afectada por el ruido, distancia y resolución de la imagen.

2. Color: Supone que la placa cuenta con una combinación de colores única casi exclusiva de la misma, son altamente susceptibles

a imágenes de baja iluminación y/o contraste.

3. VQ: Separa la imagen en bloques de vectores de alto contraste, depende altamente de la distancia, no puede procesar imágenes con mucho ruido o localizar múltiples placas.

Existen muchas arquitecturas de aprendizaje profundo por citar algunos ellos son redes neuronales convolucionales profundas o redes de creencia profunda. Son algoritmos robustos que pueden ser rápidos y capaces de procesar imágenes con mucho ruido o problemas de iluminación [4].

Savneet Kaur [1] analizó el problema de la detección de caracteres de placa vehicular en condiciones poco favorables, enfrentado problemas como son ruido causado por colores difuminados, mal clima como puede ser lluvia o neblina, baja precisión de una imagen captada por una cámara fija e imágenes corruptas o borrosas.

Para realizar dicha detección se divide en 3 tareas, el preprocesamiento de la imagen, que incluye restauración de la imagen y mejoramiento de contraste, detectar la placa en ambientes poco favorables y detectar los caracteres dentro de la placa. El proceso consiste en realizar un análisis de multiresolución para descomponer las imágenes, se realiza restauración de imagen y mejora de contraste para detectar la placa, una vez extraída la placa se aplica un filtro de desenfoque para eliminar el ruido y posteriormente aplicar un algoritmo de detección de caracteres, logrando detectar placas y su código de manera eficiente tanto en climas de neblina (ruido gaussiano) y con lluvia (ruido sal y pimienta).

Noppakun Boonsim [2] considera el problema de la localización de la placa durante la noche, tomando como candidato a la región obtenida por la intersección de algoritmos basados en color de la placa, textura de los caracteres y regiones rectangulares, realizando algoritmos de mejoramiento

de contraste iterada mente hasta obtener un resultado favorable de dicha intersección.

El algoritmo de color se basa en filtrado de regiones de la imagen con altas intensidades de los colores dominantes en una placa vehicular, binarizando la imagen para obtener las regiones candidatas. Para la detección de bordes se realizan varios pasos que van de la conversión de la imagen a blanco y negro, aplicación de filtro Sobel, aplicación de operaciones morfológicas, binarización aplicando el método Otsu y finalmente se buscan características geométricas para obtener los candidatos. Para la detección de caracteres se convierte la imagen a blanco y negro, se binariza usando el método Otsu, se aplica un filtro vertical para separar líneas, se descartan grupos pequeños y cada grupo restante es contado como un carácter.

Para el mejoramiento de contraste se usó una técnica de mejoramiento de contraste local, dividiendo la imagen en sub rectángulos sobre cada cual se aplicó una ecualización de histograma de manera independiente.

Logrando detectar la ubicación de la placa con una precisión del 85 % con un tiempo de respuesta promedio de 0.496s para una muestra de 200 imágenes con una cámara a color.

Los principales problemas de la detección de placa durante la noche es el bajo contraste debido a la poca iluminación o a la excesiva iluminación causada por la alta reflexión de la placa, los cuales afectan mayormente a los algoritmos que se basan en color. El hace uso del algoritmo basado en color de Wanli et al. quien obtuvo una precisión de 99.2 con un tiempo de computación promedio de 2.365s sobre una muestra de 1000 imágenes con una cámara a color, pero en el estudio se muestran únicamente imágenes con buena iluminación, por lo que el agrega un algoritmo de mejoramiento de contraste para casos en que no sea posible detectar la placa debido a pobres condiciones de iluminación.

Por otro lado, hace uso también del algoritmo estudiado por los autores de [3] quienes obtuvie-

ron una precisión de 98.7 % sobre una muestra de 64 imágenes usando un algoritmo morfológico basado en bordes con una cámara infrarroja.

De acuerdo a [3], los algoritmos de detección de placas en las últimas décadas se dividen en algoritmos basados en colores y bordes, los algoritmos basados en colores no son estables bajo iluminación variable, aunque ha habido estudios para mejorar su tolerancia, el analiza y profundiza el estudio de bordes por medio de un método de estudio de bordes con iluminación infrarroja variable para disminuir el ruido generado por la sobre/bajo exposición, ajustando la intensidad de la fuente de luz IR periódica e harmónicamente logrando capturar imágenes de buena calidad en un área mayor, aumentando el área monitorizada en ancho y profundidad.

“Existe un gran número de algoritmos de detección de placas basados en bordes, como análisis de histograma, morfológico y transformada de ondulaciones discretas [3].”

El hace un análisis general para determinar la ubicación de la placa y uno más fino para determinar la región de la placa y después los caracteres, en su estudio toma seis capturas de la misma imagen con diferentes intensidades de iluminación para enfatizar el efecto de la intensidad de iluminación y la distancia en la detección de placas.

Para la localización general de la placa o ROI (regiones de interés), analiza los bordes verticales de la imagen, computando los valores del gradiente se observa que los valores pico (valle) se encuentran a la derecha (izquierda) del carácter, toma las líneas horizontales entre un pico-valle de una determinada longitud como potenciales partes de carácter (PS), después realiza un histograma estadístico de la longitud de los PSs para encontrar las ROI.

Una vez ubicada la ROI se eliminan los candidatos que no son placas, para esto se considera la razón de sus dimensiones, la cantidad de caracteres en ellas y se recalculan los límites derechos e izquierdos para obtener una imagen com-

pacta de la placa.

Los autores de [5] utilizaron un algoritmo que combina detección de bordes, textura y colores, aplicando el filtro vertical de Sobel obtienen una imagen de borde, explota el espacio de color HSV para placas amarillas y un análisis de texturas para placas no amarillas, finalmente un análisis de componentes conexos se usa para ubicar la placa. Al aplicar el filtro vertical Sobel se detectan los bordes verticales, se binariza la imagen obteniendo la media de los valores de gradiente y se multiplica por un coeficiente como umbral.

Para placas amarillas se binariza la imagen a color HSV usando un umbral sobre el valor de hue y se aplica la operación lógica AND con la imagen binaria obtenida de la detección de bordes verticales.

Para placas no amarillas, se forma una imagen integral con la suma de bordes que es la cantidad de píxeles en un rectángulo de la imagen binaria y la cantidad de “saltos” de blanco a negro o negro a blanco en el rectángulo.

Después de dichas operaciones obtienen algunos candidatos a placa, aplica un análisis de componentes conexos (CCA) para identificar la placa, las regiones se combinan si tienen la misma posición en “y” y una pequeña distancia entre ellas, con esto se obtiene un nivel de pertenencia de los candidatos basados en área, alto, ancho y el componente con mayor pertenencia es seleccionado.

Obtuvieron una detección del 95 % para placas amarillas con un tiempo de procesamiento promedio de 92ms y 98 % para placas no amarillas con un tiempo de detección promedio de 47ms para un conjunto de prueba de 2943 imágenes localizando correctamente 2847.

Los autores de [7] buscan el patrón de múltiples caracteres con un alto contraste con su fondo, para esto usan una función de escalado Haar para una transformación wavelet para separar la imagen original en 4 sub-imágenes, LL, LH, HL y HH, la imagen LH contiene las características direccionales horizontales, se binariza y se bus-

can la fila con alta variación en la sub-imagen LH, encuentran una fila de referencia con la variación horizontal máxima al sumar la variación horizontal de cada fila, dentro de la fila seleccionada se encuentra el ancho de la placa obteniendo los picos en la curva de la proyección horizontal, se encuentra la región vertical buscando bajo la línea de referencia y se verifica el candidato de acuerdo a sus características geométricas como el cociente entre largo y ancho, finalmente se repite el proceso para buscar más candidatos, realizaron un análisis en 291 imágenes con un porcentaje de éxito de 92.4 % bajo múltiples condiciones como son ambientes nocturnos, placas inclinadas y pequeñas y múltiples placas en una imagen, fallando el proceso cuando el contraste entre la placa y el fondo no es claro o hay mucha distancia a la placa con un gran ángulo de inclinación.

Jose Luis [9], analizo el proceso de reconocimiento de placas de placas separándolo en los siguientes pasos, captación, Pre-procesamiento, segmentación y reconocimiento e interpretación.

El proceso de captación consiste en tomar una imagen digital por medio de un sensor o cámara.

La etapa de pre-procesamiento tiene un alto impacto en el resultado final ya que influye directamente en la calidad de la imagen que se analizara,consistiendo en filtros pasa bajas para eliminar el ruido de alta frecuencia, filtro mediano para eliminar el ruido sal y pimienta y binarización de la imagen lo cual convierte una imagen en escala de grises a otra en blanco y negro, la cual es pasada por un proceso de etiquetado que permite determinar regiones conexas, el cual disminuye el efecto del ruido en la imagen binaria, permitiendo obtener una imagen continua de la imagen binaria, la cual se ve afectada lo mínimo posible por el ruido y resalta las características propias de la placa.

Durante la segmentación se separan las regiones que ocupan cada uno de los caracteres para posteriormente analizarlas y determinar el carácter que corresponde.

Durante la interpretación se utiliza una red neuronal de retro-propagación para determinar los caracteres descritos en las diversas regiones, permitiendo formar una cadena de caracteres interpretables por una computadora.

Obteniendo un resultado de 92.1 % sobre 520 placas en la localización de la misma, 99.84 % sobre caracteres usados en el proceso de entrenamiento y 98.78 % sobre caracteres no usados durante el proceso de entrenamiento.

Los autores de [12] tomaron 2000 imágenes de 1280x720 píxeles de resolución a color, las cuales fueron sometidas a una conversión de la imagen a blanco y negro por medio del algoritmo YIQ, posteriormente fueron pasadas por una etapa de ecualización para ajustar el contraste en la imagen, un filtro gaussiano para eliminar ruidos para detectar las zonas candidatas a contener una placa usaron el algoritmo Haar Feature-based Cascade (HFC) con un clasificador AdaBoost para discriminar imágenes.

Una vez detectada la región de la placa para la extracción de caracteres primero binarizaron la imagen usando un umbral dinámico, la reescalaron para obtener un tamaño estándar, segmentaron los caracteres usando una búsqueda de contornos que respete una determinada proporción de ancho alto, obtuvieron el vector característico de cada carácter al segmentarlo en una matriz y obteniendo los cuadros en su mayoría negros como 1 y los blancos como 0, finalmente se implementó el algoritmo de memorias asociativas Alfa-Beta para reconocer cada uno de los caracteres.

Obteniendo una correcta localización de placa y reconocimiento de caracteres de 98.3 % al usar 18 AdaBoost sobre una base de datos de 707 imágenes.

Los autores de [11] analizaron el efecto de imágenes bajo condiciones de lluvia, niebla, nieve y falta de luz durante la etapa de entrenamiento de un sistema de detección de caracteres, obtuvieron que si bien una mayor cantidad de muestras en distintos ángulos y tamaños mejoran el rendimiento, no existen estudios que muestren

la diferencia de efectividad entre si dicha muestra es tomada con lluvia, nieve o de noche, ellos comparan la efectividad del mismo clasificador entrenado con muestras tomadas bajo condiciones de lluvia, condiciones ideales y poca iluminación, concluyendo sobre algoritmos basados en texturas comunes como, histogramas de gradientes orientados (HOG), patrones binarios locales (LBP) y características Haar combinadas con vectores de soporte maquina, probados a su vez bajo 3 clasificadores, el vecino mas K-Cercano, una red neuronal y una regresión lineal, que incluir imágenes afectadas por lluvia, nieve o niebla en el conjunto de entrenamiento no mejora la efectividad del clasificador bajo dichas circunstancias, los clasificadores entrenados bajo condiciones ideales mejoran la predicción bajo lluvia hasta un 19 % pero incluir imágenes con baja iluminación incrementa la precisión en un 29 % .

Existen múltiples métodos y una amplia investigación en el tema, los cuales reportan diversos índices de asertividad, sin embargo, los resultados son ampliamente dependientes de las condiciones en que se toman, por lo que es difícil realizar una comparación entre los múltiples métodos y determinar el mejor, de hecho, el mejor bajo determinadas circunstancias podría no ser el más adecuado con las condiciones de clima, tipo de carácter, color de placa o iluminación ambiental de otra región.

El tema de reconocimiento de caracteres en placas vehiculares a tenido una cantidad considerable de investigación, durante este proceso se analiza la placa vehicular como una cadena de caracteres y muchos algoritmos consisten en 2 etapas [13, 14]:

- Segmentación de caracteres: Consiste en delimitar el área de cada uno de los caracteres.
- Clasificación de caracteres: Usa métodos de reconocimiento óptico de caracteres (OCR) para identificar cada uno de los mismos.

Muchos de los algoritmos funcionan bien bajo condiciones controladas como imágenes de alta resolución, vehículos moviéndose lentamente sobre un acceso controlado o totalmente detenidos, aun es difícil reconocer caracteres de manera rápida y precisa en un ambiente abierto, lo cual se ve complicado por las distintas fuentes, colores, tamaños, diversidad de patrones, oclusión, distorsión por ángulo de visión o desenfoque [14]. Los autores de [13, 14, 15, 16] proponen LPRNet, LPRNet esta basada en una red neuronal convolucional profunda muy eficiente que permite soluciones en tiempo real, la cual tiene las siguientes características:

- Es un framework en tiempo real que sopor-

ta placas vehiculares de longitud de caracteres variable permitiendo entrenarse para diferentes placas nacionales.

- Es la primera aproximación que no usa redes neuronales recurrentes y es suficientemente ligera para correr en múltiples plataformas incluidos dispositivos embebidos.
- Los ejemplos muestran que es capaz de manejar casos difíciles como problemas de perspectiva, distorsión por cámara, condiciones de iluminación difíciles, etc.

LPRNet es un algoritmo ampliamente usado y soportado por diversas fuentes como es NVIDIA [17] por ser versátil, ligero y preciso.

### 3. Implementación

Durante la implementación del sistema se consideraron los sistemas operativos linux y windows, finalmente optando por una distribución de Linux, esto debido a que es un sistema operativo libre, altamente versátil y escalable, permitiendo instalarlo desde dispositivos portables IOT como es el caso de una Raspberry PI, posibilitando realizar la detección en el lugar para aplicaciones que no requieren una alta velocidad de procesamiento, como es el caso de un estacionamiento o control de acceso, como en aplicaciones mas robustas sobre una computadora domestica y finalmente escalándolo hasta aplicaciones industriales sobre un servidor o un clúster, sin requerir cambios mayores en el programa u OS.

Para el lenguaje de programación se consideraron opciones como fue C++, Matlab y Python, eligiendo Python, ya que es libre, soportado por Linux, fácil de instalar y tiene amplio soporte y documentación para la biblioteca OpenCV, la cual es de código abierto y fue la principal herramienta utilizada para el procesamiento de las imágenes.

#### 3.1. Localización de placa

Para el proceso de localización de la placa se consideran imágenes o videos en HD de manera que al estandarizar un tamaño es posible realizar aproximaciones mínimas y máximas del tamaño relativo aproximado de la placa, descartando las formas demasiado pequeñas o demasiado grandes.

Durante el proceso primero se realiza un pre-procesamiento de la imagen, que consiste en conversión de la imagen a escala de grises, ya que se busca una aproximación que no dependa del color de manera que sea menos susceptible a las variaciones de iluminación y contraste del día y se realiza un filtrado Gaussiano, suavizando las variaciones de iluminación en la imagen para que esta tenga una transición suave entre píxeles de baja y alta intensidad sobre vecindades continuas como se muestra en la figura

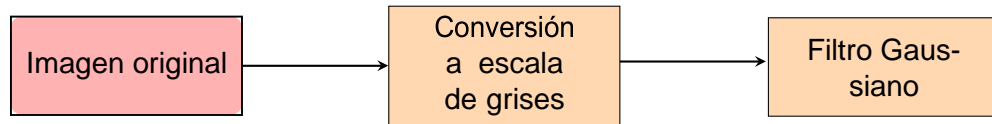


Figura 1: Pre-procesamiento de imagen.

1, las figuras 3, 5, 7, 9 y 11, muestran algunas imágenes pre procesadas para aplicar el algoritmo de localización de la placa.



(a) Imagen original.

(b) Escala de grises

(c) Filtro Gaussiano.

Figura 3: Pre-procesamiento de imagen.



(a) Imagen original.

(b) Escala de grises

(c) Filtro Gaussiano.

Figura 5: Pre-procesamiento de imagen.

Para la detección de la placa en una primera aproximación se aplicó el algoritmo de detección de bordes Canny, posteriormente se tomaron los 30 bordes cerrados con mayor superficie para finalmente seleccionar el que forma la figura más próxima a un polígono regular de 4 lados, la figura 12 describe la secuencia del algoritmo, mientras que las figuras 14, 16, 18, 20, 22 y 24, muestran el proceso aplicado a diversas imágenes.

Esto es un algoritmo muy ineficiente ya que únicamente considera bordes cerrados, por lo que como se aprecia en las figuras 14, 18 y 24, es posible que el borde que contiene el área de la placa sea descartado debido a que si este no es cerrado por motivo de ruido o oclusión no es considerado como un candidato, además si esta rodeada de bordes de contornos no cuadrangulares también será descartada por lo que el algoritmo es susceptible a la forma del marco o incluso a si existe una figura de





(a) Imagen original.



(b) Escala de grises



(c) Filtro Gaussiano.

Figura 7: Pre-procesamiento de imagen.



(a) Imagen original.



(b) Escala de grises



(c) Filtro Gaussiano.

Figura 9: Pre-procesamiento de imagen.



(a) Imagen original.



(b) Escala de grises



(c) Filtro Gaussiano.

Figura 11: Pre-procesamiento de imagen.

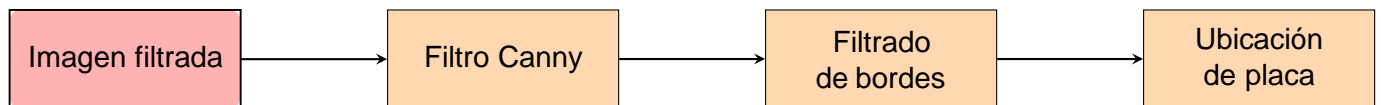
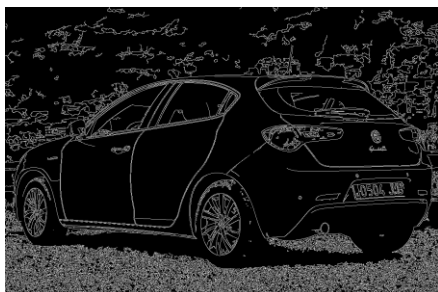


Figura 12: Algoritmo de localización de placa.

características geométricas similares en la imagen.

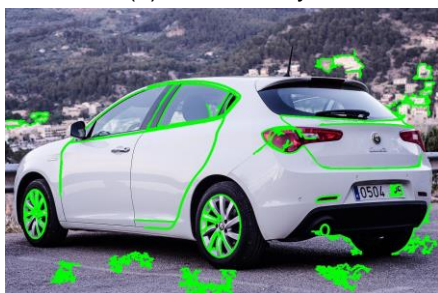
Incluso si no se considera esta deficiencia no se discrimina entre los múltiples contornos que podrían



(a) Filtro Canny.



(b) Bordes.



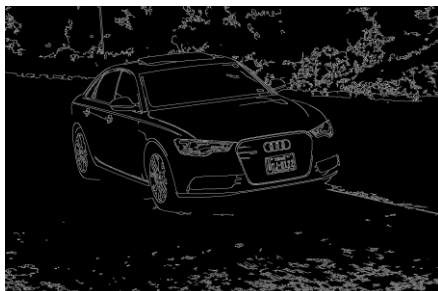
(c)

Selección de bordes.

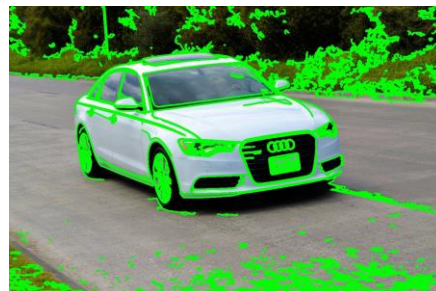


(d) Candidato a placa.

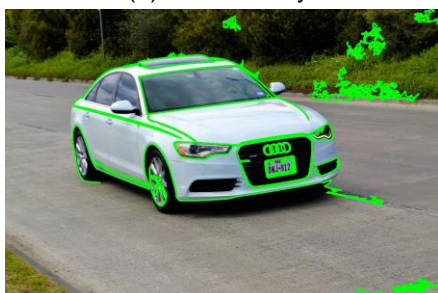
Figura 14: Proceso de localización de placa.



(a) Filtro Canny.

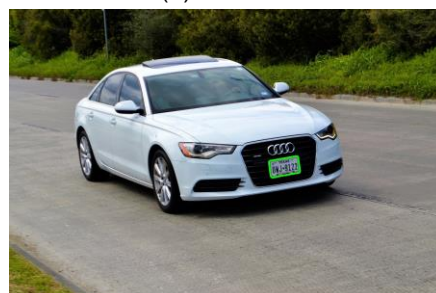


(b) Bordes.



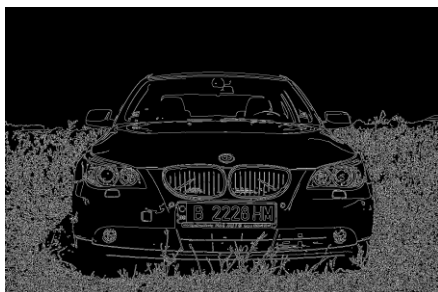
(c)

Selección de bordes.

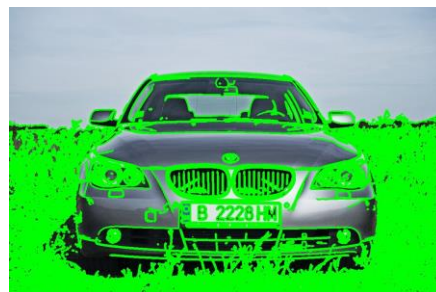


(d) Candidato a placa.

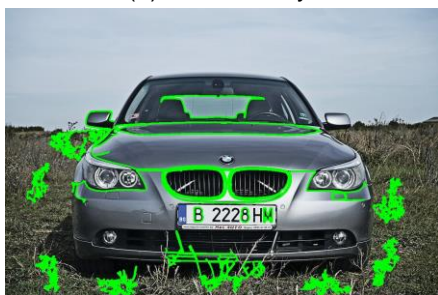
Figura 16: Proceso de localización de placa.



(a) Filtro Canny.



(b) Bordes.



(c)

Selección de bordes.

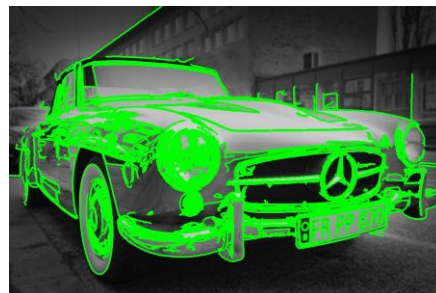


(d) Candidato a placa.

Figura 18: Proceso de localización de placa.



(a) Filtro Canny.



(b) Bordes.



(c)

Selección de bordes.



(d) Candidato a placa.

Figura 20: Proceso de localización de placa.





(a) Filtro Canny.



(b) Bordes.



(c)

Selección de bordes.



(d) Candidato a placa.

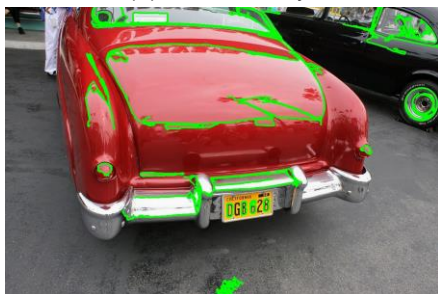
Figura 22: Proceso de localización de placa.



(a) Filtro Canny.



(b) Bordes.



(c) Selección de bordes.



(d) Candidato a placa.

Figura 24: Proceso de localización de placa.

ser candidatos a contener una placa por lo que solo elige el candidato con una figura de cuadrilátero de mayor área descartando la posibilidad de buscar múltiples placas en una imagen por lo que es necesario profundizar en un segundo paso de selección que permita un análisis mas profundo.

### 3.2. Segmentación de caracteres

Una vez localizada el área donde posiblemente se encuentre una placa vehicular el siguiente paso es localizar la ubicación de cada uno de los caracteres alfanuméricos dentro de la misma, para esto se realiza un re-escalado de la imagen a un ancho fijo, de manera que sin importar el tamaño de la placa en la imagen original es posible trabajar con tamaños relativos para discriminar ruido, se filtra nuevamente para suavizar el ruido generado por el re-escalado y se binariza la imagen con un valor de threshold adaptativo medio, permitiendo una mejor definición de la imagen binaria.

Después se descartan las áreas no deseables tomando solo aquellas áreas conexas que están dentro de un rango determinado de área, se determinan los contornos de las superficies restantes, se traza un rectángulo que los contiene y se obtienen los diferentes candidatos a contener un caracter individual.

Como se muestra en las figuras 27 y 35, este procedimiento no discrimina entre caracteres y figuras conexas con un área dentro del rango correcto, para solucionar esto se toma la media de ancho y alto de los rectángulos generados y se descartan todos aquellos candidatos que no estén dentro de un margen alrededor de la media, descartando de esta manera aquellos candidatos que no cumplen con una proporción aproximada a la media.

También se puede apreciar de las figuras 27 y 35 que el algoritmo no determina si lo que esta detectando es un caracter o si lo que esta recibiendo es una placa, esto no es un problema ya que al realizar el reconocimiento de los caracteres es posible descartar la placa completa en caso de determinarse que los resultados obtenidos hasta este punto no son caracteres dentro de una placa o específicamente si no contiene un determinado numero de candidatos a caracteres.

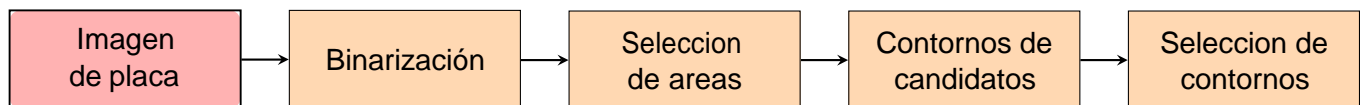


Figura 25: Proceso de segmentación de caracteres.

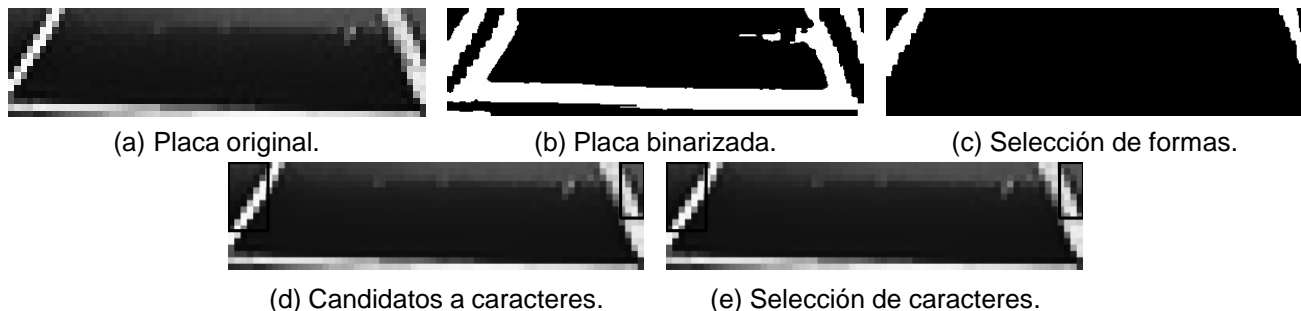


Figura 27: Proceso de segmentación de caracteres.

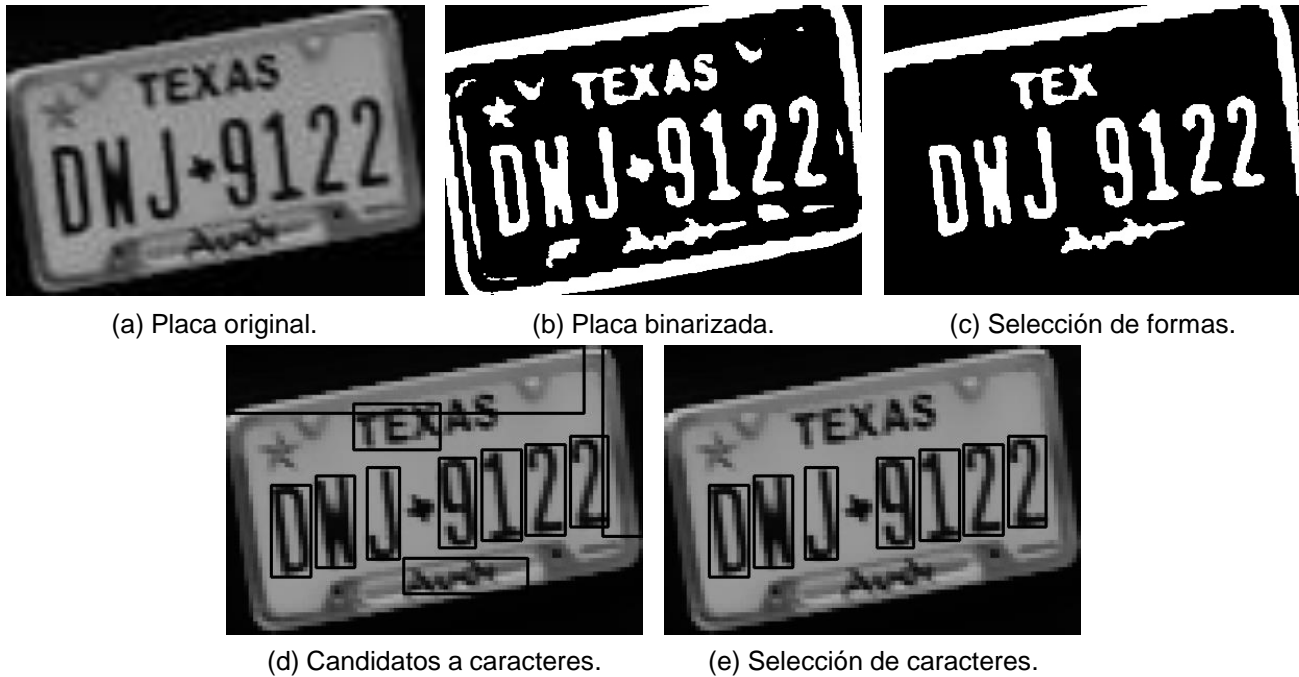


Figura 29: Proceso de segmentación de caracteres.

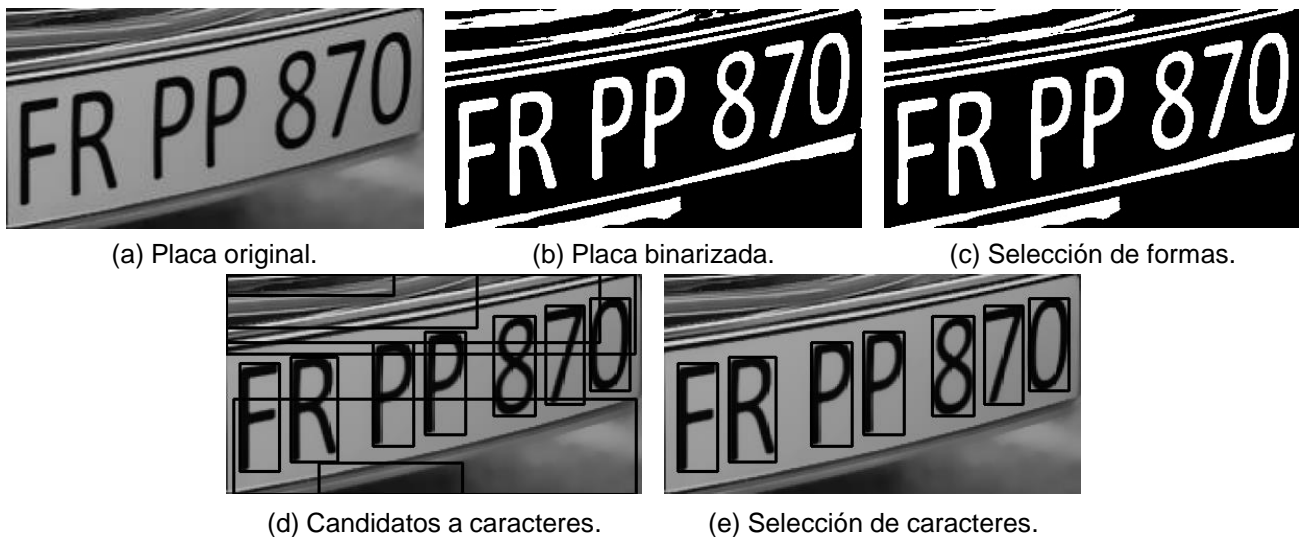


Figura 31: Proceso de segmentación de caracteres.

#### 4. Reconocimiento de caracteres

Reconocimiento de caracteres Una vez localizado cada una de los caracteres dentro de la imagen se utilizó una red neuronal convolucional para reconocerlos, como datos de entrenamiento se generó una



Figura 33: Proceso de segmentación de caracteres.

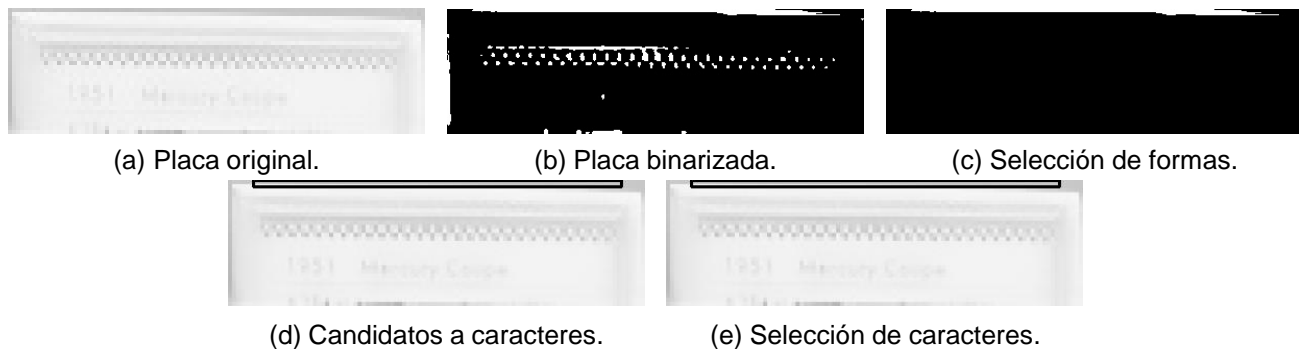


Figura 35: Proceso de segmentación de caracteres.

base de datos propia utilizando 3315 fuentes de google, a partir de las cuales se generaron imágenes binarias de los siguientes caracteres "0123456789ABCDEFGHIJKLMNPRSTUVWXYZ", se omitieron los caracteres "IOQ" ya que estos no se usan en placas vehiculares en México por su similitud con los números "10", cada imagen generada tiene unas dimensiones de 36 píxeles de ancho por 90 de alto, generando con esto un conjunto de 98472 datos de entrenamiento y 10923 datos de pruebas. Con dichos datos se entreno una red neuronal con la configuración de capas mostrada en la tabla 36 en 60 épocas de 32 lotes, obteniendo un porcentaje de asertividad del 96.72 %, como datos de pruebas se descargaron 10000 placas de la pagina "<https://matriculasdelmundo.com/mexico.html>" generados de manera aleatoria por medio de un script, al aplicar la red neuronal a las placas generadas se observo un claro patrón, alser imágenes generadas por computadora todas cuentan con el mismo tamaño, misma fuente y misma posición por lo que todas las placas que no contenían ni 2 ni Z se reconocieron correctamente mientras que dichos caracteres fueron reconocidos de manera inversa, el 2 como z y viceversa, para evitar esto se realizo un nuevo intento pero esta vez generando imágenes rotadas de las placas con un ángulo de

inclinación aleatorio entre 0 y 25 grados con esto el porcentaje de asertividad bajo de 93 % sobre las imágenes normales a 64 % sobre las imágenes rotadas, esto nos indica que con altas rotaciones la red neuronal tiene problemas para identificar los caracteres, para solucionar dicho problemas se aplico un sencillo algoritmo que permite rotar una placa hasta obtenerla en su posición horizontal para que de esta forma sea mas fácil para la red neuronal reconocer los caracteres ya que esta fue entrenada con caracteres generados horizontalmente.

Tipo de capa	Neuronas	Activacion	Caracteristicas
Entrada convolucional	32	Relu	3 x 3
Oculto convolucional	64	Relu	3 x 3
Oculto	100	Relu	Densa
Salida	33	Softmax	Densa

Figura 36: Características de la red neuronal.

El algoritmo de rotación considera las siguientes 2 hipótesis, los caracteres dentro de la placa tienen igual altura y todos están colocados en la misma altura relativa a la placa, bajo estos conceptos se observa en la figura 38 a) que podríamos trazar una línea desde el extremo superior izquierdo del cuadrado que contiene el primer caracter hasta el que contiene el ultimo y esta sería paralela a la parte inferior de la placa por lo que al aplicar la tangente inversa sobre esta podremos obtener el ángulo de inclinación de la placa, pero este algoritmo esta sujeto a 2 errores, el primero se puede apreciar en la imagen b) de la figura 38, donde resulta evidente que aunque todos los caracteres son de igual altura no necesariamente tienen igual ancho por lo que tomar un extremo del marco que lo contiene no garantiza que ambos extremos seleccionados tengan el mismo ángulo de la línea formada hacia el centro del respectivo caracter, por otro lado como se aprecia en la imagen c) de la figura 38 también existe un error por perspectiva, si bien todos los caracteres tienen igual altura, por el ángulo en que se tomo la imagen el "0" se encuentra mas lejos de la "F", haciendo que sea mas pequeño en la imagen, para solucionar estos dos errores lo que se hace es tomar la línea que va del centro del caracter, de esa manera garantizamos que sin importar si los caracteres son de ancho o alto distinto debido a la perspectiva al rotarlos su centros quedara alineado paralelamente al eje x, obteniendo caracteres horizontales.



Figura 38: Error de rotación.

Tras el anterior análisis se agregan pasos al algoritmo, una vez localizados los caracteres se rota la placa y se vuelven al buscar los caracteres antes de pasar a la red neuronal para su reconocimiento.



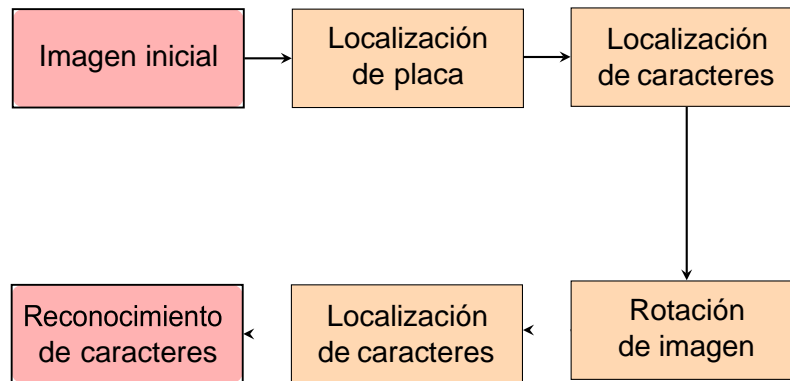


Figura 39: Proceso de reconocimiento de placas.

La figura 41 muestra algunos resultados del algoritmo de rotación de caracteres.



Figura 41: Algoritmo de rotación.

## 5. Resultados

Durante el estudio del estado del arte se analizaron múltiples técnicas de pre-procesamiento de imágenes [6, 7], los métodos de mitigar los factores de ruido que afectan el proceso [1, 2, 3, 10], las técnicas de localización de placas [4, 5] y de reconocimiento de caracteres [13, 14], permitiendo analizar las características positivas y negativas de los mismos, lo cual mejora la implementación de un algoritmo que

cumpla con las condiciones de precisión, resiliencia, tiempo de respuesta, costo, etc. para aplicaciones específicas.

Se aplicaron múltiples técnicas a los algoritmos programados obteniendo los resultados del cuadro 42, para los algoritmos de localización de placa, segmentación y reconocimiento de caracteres.

Proceso	Total	Correctos	Porcentaje
Localización de placa	24	7	30
Segmentación de caracteres	7	7	100
Reconocimiento de caracteres	50	49	98

Figura 42: Resumen de resultados.

## 6. Conclusiones

En el estudio del reconocimiento de placas es difícil tener una conclusión objetiva respecto a cuál técnica es mejor que el resto, ya que si bien existen múltiples estudios, los resultados de los mismos están sujetos a las condiciones en que fueron realizados, desde condiciones de iluminación, capacidad de procesamiento del sistema empleado, calidad de las imágenes, ángulo de visión, etc. son algunos de los factores que intervienen en la precisión del sistema, algunos algoritmos pueden ser los mas adecuados para ciertas aplicaciones mientras que deficientes en otras, por lo que es importante conocer condiciones en que se pretende aplicar, así como dominar las distintas técnicas de detección para poder proponer un algoritmo adecuado para cada aplicación específica.

En los algoritmos aplicados es importante trabajar sobre el algoritmo de localización de placas ya que cuenta con un muy bajo porcentaje de asertividad debido a que tiende a discriminar el área de la placa, el enfoque siguiente sera mejorar el algoritmo de discriminación aplicando una segunda etapa que permita analizar la densidad de los contornos mas que solo en su forma, mientras que el algoritmo de segmentación de caracteres mostró buenos resultados localizando exitosamente todos los caracteres en las placas usadas de ejemplo, el algoritmo de reconocimiento de caracteres también mostró un porcentaje de asertividad del 98 % acercándose al resultado obtenido por medio de la verificación del mismo durante su entrenamiento.

## 7. Referencias

- [1] Savneet Kaur. 2014. An Automatic System for Detecting the Vehicle Registration Plate from Video in Foggy and Rainy Environments using Restoration Technique. International Journal of Computer Applications.
- [2] Noppakun Boonsim. 2017. License Plate Localization at Night. International Journal of Computer Applications.
- [3] Chen, Y. T., Chuang, J. H., Teng W. C., Lin, H. H., and Chen, H. T. 2012. Robust license plate detection in nighttime scenes using multiple intensity IR-illuminator. In Proc. IEEE International Symposium on Industrial Electronics, 893-898.

- [4] Saidallah M., Taki F., Alaoui A. E. B. E., Fergougui A. E. 2021. CLASSIFICATION AND COMPARISON OF LICENSE PLATES LOCALIZATION ALGORITHMS. *Signal & Image Processing: An International Journal*.
- [5] L. Luo, H. Sun, W. Zhou, and L. Luo, "An Efficient Method of License Plate Location," in 2009 First International Conference on Information Science and Engineering, Dec. 2009, pp. 770-773, doi: 10.1109/ICISE.2009.250.
- [6] A. Boudjella, B. B. Samir, H. B. Daud, and R. Syahira, "License plate recognition part II: Wavelet transform and Euclidean distance method," in 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012), Jun. 2012, vol. 2, pp. 695-700, doi: 10.1109/ICIAS.2012.6306103.
- [7] Ching-Tang Hsieh, Yu-Shan Juan, and Kuo-Ming Hung, "Multiple license plate detection for complex background," in 19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers), Mar. 2005, vol. 2, pp. 389-392 vol.2, doi: 10.1109/AINA.2005.257.
- [8] INEGI. Parque vehicular. Parque vehicular. <https://www.inegi.org.mx/temas/vehiculos/>
- [9] Delgado, Jose. 2010. RECONOCIMIENTO DE PLACAS VEHICULARES [Tesis de maestria, Instituto Politécnico Nacional]. <https://tesis.ipn.mx/bitstream/handle/123456789/9880/221.pdf?sequence=1&isAllowed=y>
- [10] M. Rhead, S. Ramalingam, R. Gurney, W. E. Martin and A. Jones, "Implications of the Method of Construction of UK Number Plates for Infrared Reflectance and Camera Settings on ANPR System," 2019 International Carnahan Conference on Security Technology (ICCST), 2019, pp. 1-5, doi: 10.1109/CCST.2019.8888400
- [11] Rio-Alvarez, A. & Andrés, Javier & Gonzalez-Rodriguez, Martin & Lanvin, Daniel & Lopez, Benjamin. (2019). Effects of Challenging Weather and Illumination on Learning-Based License Plate Detection in Noncontrolled Environments. *Scientific Programming*. 2019. 1-16. 10.1155/2019/6897345.
- [12] Alanís-Carranza, L. E., Márquez-Olivera, M. V., Hernández-Herrera-Olivera, V. G., & Sánchez-García, O. (2018, December 31). Sistema de reconocimiento de placas vehiculares haciendo uso de modelos asociativos. *Research in Computing Science*, 147(12), 127-136. <https://doi.org/10.13053/rcs-147-12-12>
- [13] Zherzdev, Sergey and Gruzdev, Alexey. 2018. LPRNet: License Plate Recognition via Deep Neural Networks. doi: 10.48550/ARXIV.1806.10447
- [14] H. Li and C. Shen, "Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs," arXiv:1601.05610 [cs], Jan. 2016, arXiv: 1601.05610. 2, 4
- [15] Zhichao Wang, Yu Jiang, Jiaxin Liu, Siyu Gong, Jian Yao, Feng Jiang, "Research and Implementation of Fast-LPRNet Algorithm for License Plate Recognition", *Journal of Electrical and Computer Engineering*, vol. 2021, Article ID 8592216, 11 pages, 2021. <https://doi.org/10.1155/2021/8592216>
- [16] Wang, D.; Tian, Y.; Geng, W.; Zhao, L.; Gong, C. LPR-Net: Recognizing Chinese license plate in complex environments. *Pattern Recognit. Lett.* 2020, 130, 148–156
- [17] NVIDIA. LPRNet - TAO Toolkit 3.22.05 documentation. LPRNet. [https://docs.nvidia.com/tao/tao-toolkit/text/character\\_recognition/lprnet.html](https://docs.nvidia.com/tao/tao-toolkit/text/character_recognition/lprnet.html)