



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

DETECCIÓN DE CARACTERES DE PLACAS DE AUTOMÓVILES MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL

George Mundaca-Vidarte

Piura, junio de 2016

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Mecánico-Eléctrica

Mundaca, G. (2016). *Detección de caracteres de placas de automóviles mediante técnicas de visión artificial* (Tesis de pregrado en Ingeniería Mecánico-Eléctrica). Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Mecánico-Eléctrica. Piura, Perú.



Esta obra está bajo una [licencia](#)
[Creative Commons Atribución-](#)
[NoComercial-SinDerivadas 2.5 Perú](#)

[Repositorio institucional PIRHUA – Universidad de Piura](#)

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA



**“Detección de caracteres de placas de automóviles mediante técnicas de
visión artificial”**

Tesis para optar el Título de
Ingeniero Mecánico - Eléctrico

George Antonio Mundaca Vidarte

Asesor: Dr. Ing. William Ipanaqué Alama

Piura, Junio 2016

**A Dios, a mi familia y
a mis amigos.**

Prólogo

La visión es una capacidad sensorial que le permite al ser humano interpretar el entorno a través de las habilidades de reconocer, distinguir objetos y calcular distancias, con la intención de dotar de las mismas habilidades a un computador surge el concepto de visión artificial como el conjunto de técnicas computacionales para el procesamiento y análisis de información que provienen de imágenes digitales.

Esta investigación desarrolla un sistema que implementa técnicas de visión artificial en Matlab, con el fin de proveer información instantánea de la identificación de placas de vehículos, facilitando la toma de acciones en el momento preciso que ocurren los sucesos ofreciendo así una solución a los problemas de seguridad que aqueja diariamente a la ciudad.

Así también, se busca fomentar el uso de algoritmos de visión artificial para el procesamiento digital de imágenes con enfoque en diversos campos como la agroindustrias, biomedicina, seguridad entre otros, que pueda generar propuestas de inversión y desarrollo para la comunidad.

Cabe mencionar que el sistema implementado tiene un costo computacional de 2.69 segundos, localiza al 100% la ubicación de las placas con fondo blanco sin importar la iluminación o perspectiva de la captura y el error de la identificación de los caracteres está sujeto al estado de deterioro en que se encuentre las placas.

Como resultado de esta investigación realizada en el laboratorio de sistemas automáticos de control, se realizó la publicación del artículo: “Sistema para detección de placas mediante el procesamiento digital de imágenes”, 15th IFAC Congreso Latinoamericano de control Automático, Octubre 23-26, 2012, Lima-Perú.

Es importante mencionar a las personas e instituciones que me han permitido cumplir la labor de investigador y por ende la redacción de este informe, por tal razón dirijo un agradecimiento especial a mi asesor Dr. Ing. William Ipanaqué Alama, a los colaboradores de este proyecto y laboratorio de sistemas automáticos de control que con su apoyo hicieron posible la elaboración de esta investigación y la redacción de esta tesis.

Resumen

Esta tesis se presenta bajo la modalidad de artículo científico y expone el desarrollo de un sistema para la identificación de caracteres en las placas de automóviles sin importar la perspectiva con la que la cámara capture la imagen, implementado con algoritmos de visión artificial en Matlab el sistema detecta la placa a través de la variación de gradientes, filtro Sobel y transformada de Bottom-Hat, corrige la perspectiva de captura de la placa localizada haciendo uso de la transformada de Hough y transformada proyectiva bidimensional, finalmente realiza el reconocimiento de los caracteres usando el método de correlación de Pearson.

El sistema implementado tiene un costo computacional total de 2,69 segundos, localiza el 100% la posición de la placa de automóvil, los resultados del reconocimiento de caracteres dependen en gran medida del estado en que se encuentra las placas, además están sujetos a que la evaluación de la placa se lleve a cabo en escenas estáticas y a una distancia prudente, cabe indicar que para obtener mejores resultados se debe trabajar con una cámara de óptima resolución.

Los resultados de esta investigación fueron presentados en el 15th IFAC Congreso Latinoamericano de control Automático, Octubre 23-26, 2012, Lima-Perú.

Índice

Introducción	1
Capítulo 1 Antecedentes.....	3
1.1. Motivación y justificación	3
1.2. Objetivos.....	4
1.3. Metodología	4
1.4. Conceptos generales	4
a) Imagen digital	4
b) Imagen digital RGB.....	5
Capítulo 2 Algoritmos de visión artificial	7
2.1. Histogramas	7
2.2. Filtros espaciales.....	8
2.3. Teorema de convolución.....	8
2.4. Operaciones morfológicas	8
2.4.1. Dilatación	9
2.4.2. Erosión.....	9
2.4.3. Apertura	9
2.4.4. Clausura	9
2.5. Derivada parcial y gradiente	9
2.5.1. Gradiente	9
2.5.2. Derivada parcial.....	10
2.6. Filtro Sobel	11
2.7. Gradiente Botton Hat	14
2.8. Binarización	15
2.9. Transformada de Hough	16
2.10. Transformada proyectiva bidimensional	16
2.11. Coeficiente de correlación de Pearson.....	18
Capítulo 3 Etapas del sistema de visión artificial	21
3.1. Captura de imagen	22
3.2. Pre-procesado	22
3.2.1. Escala de grises.....	22
3.2.2. Reducción de la imagen.....	23

3.2.3.	Complemento de la imagen	23
3.3.	Localización de la placa	24
3.3.1.	Umbral óptimo	25
3.3.2.	Aplicación de operaciones morfológicas	27
3.3.3.	Ubicación de concentración de gradientes	28
3.3.4.	Calibración del proceso	29
3.4.	Corrección de perspectiva	31
3.4.1.	Transformada de Hough.....	32
3.4.2.	Transformada proyectiva bidimensional	33
3.5.	Binarización.....	34
3.6.	Segmentación	35
3.6.1.	Recorte	35
3.6.2.	Segmentación	38
a)	Recorte horizontal	38
b)	Recorte vertical	39
c)	Problemas presentados.....	40
3.7.	Reconocimiento óptico de caracteres - OCR.....	41
3.7.1.	Reducción.....	42
3.7.2.	Generación de plantilla.....	42
3.7.3.	Comparación de matrices	42
Capítulo 4	Análisis del sistema de visión artificial	45
4.1.	Pruebas realizadas	48
4.1.1.	Prueba N°1	48
4.1.2.	Prueba N°2	48
4.1.3.	Prueba N°3	49
4.1.4.	Prueba N°4	49
4.2.	Discusión de resultados	50
Capítulo 5	Artículo científico	51
	Sistema para detección de placas mediante el procesamiento digital de imágenes	51
Conclusiones		57
Bibliografía		59

Introducción

La visión artificial por computadora es un campo de gran interés para distintos sectores de la industria, esto se debe a que un gran porcentaje de la información recibida por el hombre es visual, adquirir dicha información en un sistema computacional es un gran desafío en la actualidad y se logra mediante el método llamado “Procesamiento y reconocimiento de imágenes por computadora” [1], el mismo que ha presentado un grado de relevancia en los últimos años en el cual los investigadores vienen desarrollando aplicaciones en distintos campos como la agroindustria, la medicina, la seguridad entre otros [2]. El desarrollo de estos algoritmos busca dar solución a problemas como la detección, identificación y clasificación de objetos, entre otras diversas necesidades [3].

Esta investigación desarrolla un sistema en el entorno MATLAB que permita la identificación de los caracteres de las placas vehiculares, de modo que brinde una solución a la problemática que genera no tener información de manera instantánea de los vehículos en situaciones donde la seguridad lo amerita; evitando así por ejemplo, cualquier tipo de fraude o robo que se pueda generar.

La elaboración de esta tesis plasma de manera concisa y clara las etapas de esta investigación, presentándolo en 5 capítulos, en el Capítulo 1: “Antecedentes” explica brevemente la motivación y justificación del trabajo así como conceptos generales de imágenes digitales [4], en el Capítulo 2: “Algoritmos de visión artificial” explica los conceptos de los algoritmos que serán implementados en el desarrollo del sistema en el entorno Matlab [5], en el Capítulo 3: “Etapas del sistema de visión artificial” se explica las etapas del sistema, desde la adquisición de la imagen de las placas en formato RGB para pasar por la etapa de pre-procesado en base a algoritmos de reducción de imagen y conversión a escala de grises, para luego pasar por la etapa de localización de placa basado en algoritmos de transformada de Bottom-Hat y convolución lineal [6], luego se procede con la etapa de corrección de perspectiva con la finalidad de que la posición del automóvil con respecto a la cámara al momento de la captura no influya en el resultado, esto se realiza en base a algoritmos de transformada de Hough y transformada proyectiva bidimensional [7], luego se procede a la etapa de reconocimiento de caracteres (OCR) mediante el método de correlación de Pearson [8], en el Capítulo 4: “Análisis de los resultados” se analizará los resultados de las pruebas que se realizaron, finalmente en el Capítulo 5: “Artículo científico”, contiene el artículo científico resultado de la investigación y que se expuso en el Congreso

Latinoamericano de Control Automático en el año 2012. El sistema implementado tiene una carga computacional de 2,69 segundos y trabaja con un porcentaje de error de 23 % que se da en casos donde las letras y números presenten alguna similitud entre ellas o algún signo de deterioro.

Capítulo 1

Antecedentes

La realización de esta tesis, nace en el Laboratorio de Control de la Universidad de Piura en el año 2012, como trabajo del curso de Sistemas Automáticos de Control que puso en ese año especial interés en el campo de procesamiento de imágenes mediante algoritmos de visión artificial. Como resultado de esta investigación, se realizó el artículo científico “Sistema para detección de placas mediante el procesamiento digital de imágenes”, el cual fue seleccionado en el XV IFAC Congreso Latinoamericano de Control Automático en el 2012 (CLCA 2012).

Con la tesis propuesta se busca una solución a la obtención rápida y eficiente de la identificación de los caracteres de las placas de automóviles con fines de seguridad o de monitoreo, así mismo se busca incentivar el uso de algoritmos de visión artificial como medio de desarrollo tecnológico en diversas áreas como la agricultura, la medicina, etc.

1.1. Motivación y justificación

Establecer una estructura de trabajo que permita comprender el comportamiento de los sistemas de detección de caracteres en placas de automóviles, así como cooperar en el avance de la investigación en el campo del análisis de imágenes por computadora.

El desarrollo de software de procesamiento de imágenes en la Región Piura y en el Perú es nulo a pesar de las necesidades de seguridad, a esto se le suma los costos elevados de los equipos y software que existen en el mercado. De este modo, con esta investigación se busca desarrollar un sistema que detecte los caracteres de las placas de automóviles de manera inmediata, con la finalidad de implementarlo posteriormente en un hardware de modo que sea accesible al mercado.

1.2. Objetivos

Desarrollar un sistema en la plataforma Matlab que pueda identificar los caracteres en placas de automóviles, sin importar la perspectiva que este se encuentre delante de la cámara al momento de adquirir la imagen.

1.3. Metodología

Se abarcará aspectos prácticos de programación para la implementación de etapas de procesamiento de imágenes.

a) Adquisición de imágenes

Las imágenes son adquiridas y digitalizadas, las que serán adquiridas por medio de una cámara RGB.

b) Pre-procesamiento de imágenes

Se aplica algoritmos de visión artificial para pre-procesar las imágenes de muestras y obtener solo la información necesaria que posteriormente se procesará.

c) Implementación de plataforma en Matlab

Se implementará los algoritmos de visión artificial en el programa Matlab con la finalidad de reconocer los caracteres de las placas de automóviles.

d) Experimentación

Se realizará los experimentos de adquisición de nuevas imágenes

e) Validación y conclusiones

Se pone a prueba las imágenes adquiridas y se analiza la eficacia con la que trabaja el sistema implementado.

1.4. Conceptos generales

a) Imagen digital

Una imagen digital es una función de intensidad de luz que se expresa por la función $f(x, y)$, donde x y y son coordenadas llamadas píxeles y el valor de la función es un valor proporcional a la intensidad de luz de la imagen en dicho pixel [9]. La Figura 1, muestra la convención de coordenadas que se usará.

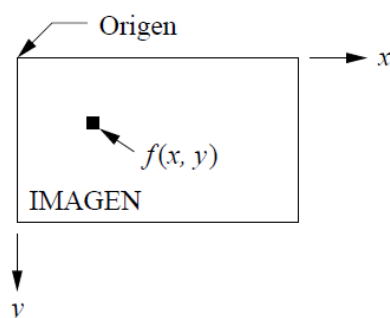


Figura 1. Convección de coordenadas en una imagen digital.
Fuente: Azriel Rosenfeld and Avinash C. KaK. Digital Picture Processing. Academic Press, 1982

b) Imagen digital RGB

La imagen digital RGB, es representada por una matriz de tres dimensiones $a \times b \times c$, donde a denota el número de pixeles del ancho de la imagen, b denota el número de pixeles del largo de la imagen y c denota el plano que puede tomar el valor de 1 para el rojo, 2 para el verde y 3 para el azul. En la Figura 2 se muestra la suma aritmética $I = R + G + B$ representada por un cubo. La recta que une el valor mínimo con el valor máximo muestra la escala de grises [4].

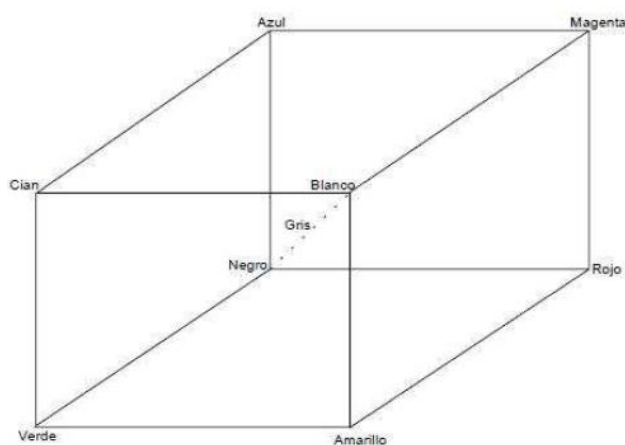


Figura 2 Espacio de colores RGB
Fuente: R.C. Gonzáles, R.E. Woods. Digital Image Processing. Pearson Prentice Hall, 2008

Capítulo 2

Algoritmos de visión artificial

A continuación, se describen algunos teoremas que respaldan la lógica algorítmica del sistema.

2.1. Histogramas

Los histogramas son medidas estadísticas que proporciona información de una imagen y son utilizadas como medio para evaluar propiedades en ella. Las distribuciones presentes en los histogramas, denotan la frecuencia con la que existen ciertos valores de intensidad de luz por cada pixel en la imagen [4]. La Figura 3 muestra el histograma de una imagen digital, donde el eje de las abscisas representa los valores de intensidad de luz y el eje de las ordenadas representan número de pixeles que presenta cierto valor.

Cada valor del histograma se define como $h(i)$ que representa al número de pixeles de I (imagen) con el valor de la intensidad i para todos los valores $0 \leq i < 255$. Esto queda expresado formalmente en la ecuación (1).

$$h(i) = \text{card}\{(u, v) | I(u, v) = i\} \quad (1)$$

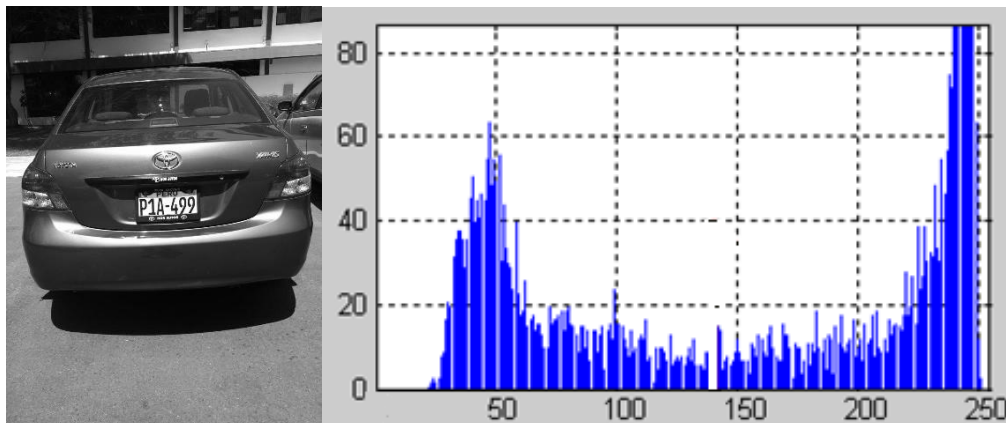


Figura 3 Histograma de una imagen capturada
Fuente: Elaboración propia

2.2. Filtros espaciales

Los filtros pueden ser considerados como operaciones de pixel por pixel donde el valor del nuevo pixel depende del pixel original y de los pixeles que tienen relación de vecindad con él. Los filtros lineales, son aquellos filtros en que los valores de intensidad de luz en los pixeles dentro de la región de procesamiento se combinan de manera lineal. La función o matriz que relaciona el pixel original con su vecindad se le denomina “mascara”. La ecuación (2) muestra un ejemplo de mascara.

$$H(i,j) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

2.3. Teorema de convolución

Es un filtro lineal que realiza operaciones de pixel por pixel que permite conectar dos funciones continuas o discretas en una sola [6]. El teorema de convolución está representada por la ecuación (3), donde I y H son dos funciones discretas bidimensionales.

$$I'(x,y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} I(x-i, y-j) \cdot H(i,j) \quad (3)$$

O de manera compacta dada por la ecuación (4).

$$I' = I * H \quad (4)$$

Esto aplicado al campo del procesamiento de imágenes, proporciona una herramienta que permite conectar una imagen (matriz) con una máscara (matriz).

2.4. Operaciones morfológicas

Son técnicas de procesado no lineal aplicado en una imagen, de tal forma que modifica la geometría de los objetos, proporcionando información sobre la forma o estructura de la imagen analizada [10]. Mediante este análisis morfológico, es posible extraer información que sea de utilidad para la representación, identificación y descripción de la forma que proporcionen características de forma y tamaño.

2.4.1. Dilatación

La dilatación, es una operación morfológica que añade píxeles a la estructura de la imagen, se controla a través, de una estructura de referencia llamada máscara, esta operación realiza el teorema de convolución y su notación se muestra en la ecuación (5).

$$I \oplus H = \{(x', y') = (x + i, y + j) | (x', y') \in P_I, (i, j) \in P_H\} \quad (5)$$

2.4.2. Erosión

Esta operación es la inversa de la dilatación de modo, que sustrae píxeles a la estructura de la imagen, su notación se expresa en la ecuación (6):

$$I \ominus H = \{(x', y') | (x + i, y + j) \in P_I, (i, j), \forall (i, j) \in P_H\} \quad (6)$$

2.4.3. Apertura

La apertura se da por una operación de erosión seguida por una operación de dilatación, se muestra en la ecuación (7).

$$I \circ H = (I \ominus H) \oplus H \quad (7)$$

2.4.4. Clausura

El cierre se da por una operación de dilatación seguida por una operación de erosión, se muestra en la ecuación (8).

$$I \bullet H = (I \oplus H) \ominus H \quad (8)$$

2.5. Derivada parcial y gradiente

2.5.1. Gradiente

El gradiente, es la diferencia de los píxeles vecinos en una dirección que puede ser vertical, horizontal, radial, etc. Se mostrara un ejemplo, imagine que tiene la siguiente matriz que puede ser una imagen en escala de grises 256 de niveles en el intervalo de [0 – 255]:

$$I = \begin{vmatrix} 240 & 239 & 130 & 131 & 130 & 45 & 44 \\ 239 & 240 & 130 & 130 & 129 & 45 & 44 \end{vmatrix}$$

Ahora para hallar los gradientes en cada pixel usaremos la siguiente matriz, $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$ entonces el proceso será el siguiente:

Gradiente (1,2)=240-239=1
 Gradiente (2,2)=239-240=-1
 Gradiente (1,3)=239-131=108
 Gradiente (2,3)=240-130=110
 Gradiente (1,4)=130-130=0
 Gradiente (2,4)=130-129=1
 Gradiente (1,5)=131-45=86
 Gradiente (2,5)=130-45=85
 Gradiente (1,6)=130-44=86
 Gradiente (2,6)=129-44=85

Entonces la matriz de gradientes quedaría:

$$G = \begin{bmatrix} 240 & 1 & 108 & 0 & 86 & 86 & 44 \\ 239 & -1 & 110 & 1 & 85 & 85 & 44 \end{bmatrix}$$

Al observar la matriz $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$, muestra que el gradiente realizado es de manera horizontal, esta matriz recibe el nombre de máscara y existen varias que permiten obtener los gradientes con mayor precisión o para distintas aplicaciones, como es resaltar bordes, filtrar una imagen, homogeneizar, etc.

En una imagen existen zonas en las que los pixeles poseen el mismo valor, al evaluar el gradiente en estas zonas se puede ver la diferencia entre pixeles con el mismo valor resultara 0, también existen zonas en las que los pixeles poseen valores distintos, es decir donde hay cambio de un color a otro, esto puede ser un borde, una línea que separa un objeto de otro, en esta zona la diferencia de los valores es bastante apreciable. Con esto se concluye que los gradientes miden la intensidad de cambio en los colores entre pixeles vecinos a lo largo de toda la imagen. Nótese que cuando hay un cambio de color blanco al negro, el gradiente es máximo (255) [10]. Por tanto, en esto se basa este análisis, en localizar zonas potenciales en las cuales exista grandes cantidades de gradientes elevados, es decir, zonas donde exista la mayor cantidad de cambio de color de blanco al negro (zona potencial de ubicación de la placa).

2.5.2. Derivada parcial

Es la derivada de una función multidimensional con respecto a una de las variables de la función, por ejemplo se tiene:

$$\frac{\partial I}{\partial x}(x, y) \quad , \quad \frac{\partial I}{\partial y}(x, y)$$

La Figura 4 expresa la derivada parcial de la función de la imagen $I(u, v)$ con respecto a la variable u o v .

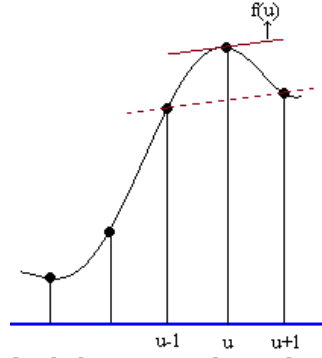


Figura 4 Cálculo de la primera derivada para una función.
Fuente: Elaboración propia

El vector de la ecuación (9) representa el vector del gradiente de la función I en el punto (x, y) .

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) \\ \frac{\partial I}{\partial y}(x, y) \end{bmatrix} \quad (9)$$

La ecuación (10) muestra el valor del gradiente que es indiferente a rotaciones de la imagen y la orientación de las estructuras contenidas en ella. Esta propiedad es importante para la localización de los puntos bordes de la imagen.

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (10)$$

2.6. Filtro Sobel

Este filtro utiliza una máscara para aproximar las derivadas parciales $\frac{dI}{dx}$ y $\frac{dI}{dy}$, la combinación de esas derivadas parciales muestra un valor de gradiente para cada pixel, las ecuaciones (11), (12), (13), muestran el filtro Sobel aplicado en diferentes direcciones.

$$\frac{dG(x, y)}{dx} = \frac{G(x - \Delta x, y) - G(x, y)}{\Delta x} \quad (11)$$

$$\frac{dG(x, y)}{dx} = \frac{G(x, y) - G(x - \Delta x, y)}{\Delta x} \quad (12)$$

$$\frac{dG(x, y)}{dx} = \frac{G(x + \Delta x, y) - G(x - \Delta x, y)}{2\Delta x} \quad (13)$$

De esto se puede deducir entonces, que el gradiente Sobel se puede determinar de manera horizontal (eje x), o en manera vertical (eje y) y en cuatro direcciones.

La ecuación (14) muestra una máscara del gradiente Sobel horizontal en sus 2 direcciones:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (14)$$

La ecuación (15) muestra una máscara del gradiente Sobel vertical en sus 2 direcciones:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (15)$$

Para determinar los gradientes de la placa, se aplica la máscara horizontal ya que las zonas de gradientes elevados tienen mayor longitud en sentido vertical y facilita la búsqueda de la zona de la placa en la imagen. La Figura 5 muestra una imagen de un automóvil luego de aplicarle el gradiente Sobel con máscara vertical, en ella no se puede diferenciar con claridad la zona placa del resto de la imagen mientras que la Figura 6 muestra la misma imagen del automóvil pero luego de aplicarle el gradiente Sobel con máscara horizontal donde la zona de la placa si se puede diferenciar.



Figura 5 Imagen luego de aplicar el gradiente Sobel vertical.
Fuente: Elaboración propia

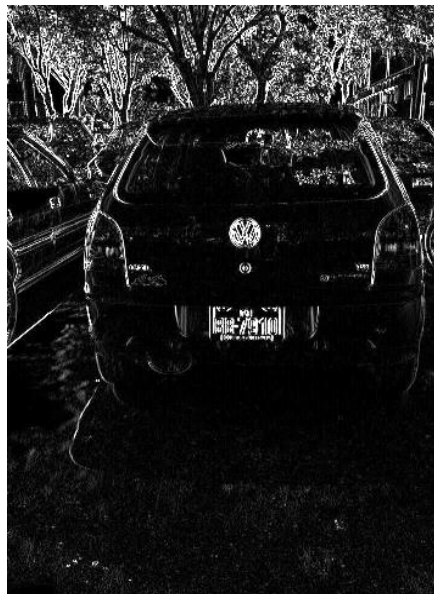


Figura 6 Imagen luego de aplicar el gradiente Sobel horizontal.
Fuente: Elaboración propia

En MATLAB se digitan las siguientes líneas para determinar los gradientes por el método sobel:

```
- Sobel horizontal:
hxe=[1 0 -1;2 0 -2;1 0 -1];
hxo=[-1 0 1;-2 0 2;-1 0 1];
ge=imfilter(iig,hxe);
go=imfilter(iig,hxo);
iir=abs(ge)+abs(go);
```

- Sobel vertical:

```
hxn=[1 2 1;0 0 0;-1 -2 -1];
hxs=[-1 -2 -1;0 0 0;1 2 1];
gn=imfilter(ii,hxn);
gs=imfilter(ii,hxs);
iit=abs(gn)+abs(gs);
```

2.7. Gradiente Botton Hat

La ecuación (16), denota el gradiente Bottom –Hat, donde h es la matriz resultante y b es la máscara Bottom-Hat que se emplea para determinar los gradientes y f la figura a analizar.

$$h = f.b - f \quad (16)$$

La transformada Bottom-Hat, tiene la característica de resaltar solo los gradientes máximos, es decir solo resalta en la imagen una zona blanca en un fondo negro o viceversa. MATLAB permite trabajar con distintas mascaras para determinar los gradientes, estas máscaras pueden ser líneas, cuadrados, círculos, octágonos, etc. Para este proyecto se determina una máscara circular porque es la que mejor se ajusta a la geometría de las letras en la placa.

La Figura 7 muestra la transformada Bottom-Hat aplicada a una imagen en escala de grises, en ella se puede visualizar la zona de la placa de color blanco en fondo negro.



Figura 7 Imagen en escala de grises (izquierda), con efecto Bottom-Hat (derecha).
Fuente: Elaboración propia

En MATLAB se digitan las siguientes líneas para hacer la operación Bottom-Hat con una máscara circular de radio 11 píxeles.

```
st=strel('disk',11);
ibh=imbothat(Imagen_en_escala_de_grises,st);
```

2.8. Binarización

Consiste en convertir una matriz de escala de grises, a una matriz binaria basándose en un número de umbral X [4].

La ecuación (17) muestra una matriz $[m \times n]$:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad (17)$$

Tal que:

$$0 \leq X \leq 255$$

La Figura 8, muestra un histograma de una imagen A con valores de 0 a 255 que muestran los niveles en escala de grises, el valor de X denota el valor umbral, donde los pixeles con valor menor al umbral se convertirá en 0 y los pixeles con valor mayor al umbral se convertirá en 1.

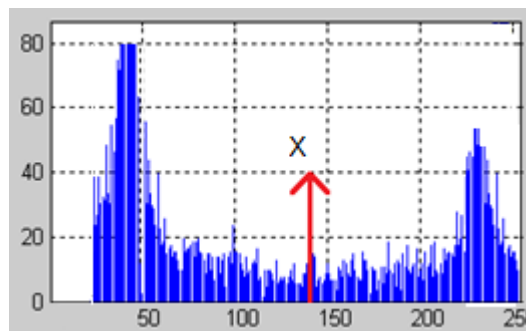


Figura 8 Histograma matriz A.
Fuente: Elaboración propia

Si:

$$\begin{aligned} 0 \leq a \leq X, & \text{ entonces } a = 0 \\ X \leq a \leq 255, & \text{ entonces } a = 1 \end{aligned}$$

Como resultado la Figura 9 muestra un histograma de la misma imagen A con todos los valores de los píxeles binarizados por el efecto umbral.

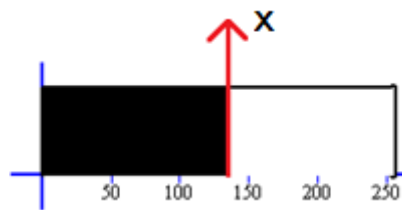


Figura 9 Binarización de matriz.
Fuente: Elaboración propia

2.9. Transformada de Hough

Método ideado por Paul Hough que permite localizar formas a partir de una distribución de puntos presentes en una imagen, por ejemplo: líneas, círculos o elipses. Una línea en el espacio se puede denotar por la ecuación (18).

$$x\cos\alpha + y\sin\alpha = r \quad (18)$$

Dicha ecuación permite una cuantificación lineal de sus parámetros r y α , por lo que el espacio de Hough (espacio donde se localiza las posibles rectas) y el punto $p_i = (x_i, y_i)$ se relacionan según la ecuación (19).

$$r_{x_i+y_i} = x_i\cos\theta + y_i\sin\theta \quad (19)$$

Donde el intervalo de los valores para α es $0 \leq \alpha \leq \pi$, si tomamos como referencia el punto (x_c, y_c) que coinciden con el centro de la imagen de la placa como se puede ver el intervalo de los valores de r se denota por la ecuación (20).

$$-r_{max} \leq r_{x,y}(\alpha) \leq r_{max} \quad (20)$$

La ecuación (21) muestra el valor de máximo de r , donde M y N representa el ancho y alto de la imagen.

$$r_{max} = \sqrt{\frac{M^2}{2} + \frac{N^2}{2}} \quad (21)$$

2.10. Transformada proyectiva bidimensional

Es una transformada geométrica que realiza operaciones pixel por pixel, logrando así operaciones sobre la imagen original como pueden ser: desplazamiento, rotación, escalamiento o distorsiones, tal como se muestra en la Figura 10.

La transformada aplicada a una imagen produce: $I(x, y) \rightarrow I'(x', y')$ donde el valor del pixel es cambiado, para cada coordenada de la imagen $I(x, y)$ original $x = (x, y)$ corresponde los puntos $x' = (x', y')$ de la imagen $I'(x', y')$ esto se denota según la ecuación (22).

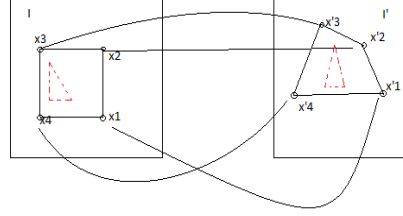


Figura 10 Transformación geométrica.
Fuente: Elaboración propia

$$x \rightarrow x' = T(x) \quad (22)$$

Para realizar esto se necesita primeramente una transformación de coordenadas en forma de una transformación geométrica [11]. Para distorsionar la imagen que en este caso es un rectángulo (geometría de la placa) se necesita trabajar con 4 puntos.

Sea T un matriz dado tal que:

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

$$\begin{bmatrix} \hat{x}' \\ \hat{y}' \\ \hat{h}' \end{bmatrix} = \begin{bmatrix} t'x' \\ t'y' \\ t' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Según referencia [11] se tiene las siguientes relaciones que se muestra en la ecuación (23) para cada punto del plano xy y su correspondiente en el plano $x'y'$:

$$x' = \frac{x'_1}{x'_3} = \frac{t_{11}x + t_{12}y + t_{13}}{t_{31}x + t_{32}y + t_{33}} \quad ; \quad y' = \frac{x'_2}{x'_3} = \frac{t_{21}x + t_{22}y + t_{23}}{t_{31}x + t_{32}y + t_{33}} \quad (23)$$

Los parámetros de la transformación proyectiva son determinados a partir de 4 coordenadas $(x_1, x'_1); (x_2, x'_2); (x_3, x'_3); (x_4, x'_4)$, cabe indicar que $x = (x_i, y'_i)$ son los puntos de la original imagen y $x'_i = (x'_i, x'_i)$ son los puntos de la imagen transformada como se muestra en ecuación (24) - (25).

$$x'_i = t_{11}x_i + t_{12}y_i - t_{13} - t_{31}x_ix'_i - t_{32}y_ix'_i \quad (24)$$

$$y'_i = t_{21}x_i + t_{22}y_i - t_{23} - t_{31}x_iy'_i - t_{32}y_iy'_i \quad (25)$$

Donde puede ser $i=1, 2, 3, 4$

Por ende se tiene la ecuación (26):

$$x' = T.x \quad (26)$$

2.11. Coeficiente de correlación de Pearson

Define un factor que mide el grado de igualdad entre dos matrices. Se tiene dos matrices binarias $[m \times 1]$:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

Se toma los pares (x_i, y_i) y se grafica en el plano cartesiano como se muestra en la Figura 11.

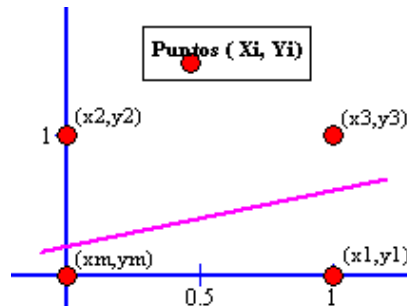


Figura 11 Plano cartesiano.
Fuente: Elaboración propia

La pendiente de la línea tendenci3n de dichos puntos, es el coeficiente de correlaci3n [8]. Seg3n se muestra en la ecuaci3n (27).

$$r = \frac{\frac{\sum_{i=1}^m (x_i \times y_i)}{m} - \bar{X} \times \bar{Y}}{S_x \times S_y} \quad (27)$$

Donde:

$$\bar{X} = \frac{\sum_{i=1}^m x_i}{m} \quad ; \quad \bar{Y} = \frac{\sum_{i=1}^m y_i}{m}$$

$\bar{X}$: Promedio de valores de matriz X.

$\bar{Y}$: Promedio de valores de matriz Y.

$$S_x = \sqrt{\frac{\sum_{i=1}^m x_i^2}{m} - \bar{X}^2} \quad ; \quad S_y = \sqrt{\frac{\sum_{i=1}^m y_i^2}{m} - \bar{Y}^2}$$

Donde:

S_x : Desviación estándar de variables de X.

S_y : Desviación estándar de variables de Y.

Capítulo 3

Etapas del sistema de visión artificial

La Figura 12 muestra un diagrama de bloques del sistema de visión artificial implementado, la entrada es la imagen capturada, la primera fase es la de Pre-procesado donde se acondiciona la imagen para el funcionamiento óptimo y eficiente del proceso, luego sigue la fase de Localización de la placa donde se selecciona solo la zona de la placa del automóvil, se procede luego a la fase de Corrección de perspectiva que traslada a un plano frontal cualquier imagen capturada con cierto ángulo de perspectiva, la siguiente fase es la de Binarización que convierte la imagen de la placa en una nueva imagen cuyo valores de pixeles solo contienen 0 ó 1, se prosigue luego a la fase de Segmentación donde los caracteres de la placa son separadas una a una para pasar finalmente por la fase de OCR(Reconocimiento Óptico de Caracteres) con el fin de reconocer y convertir a código alfanumérico los caracteres de la placa.

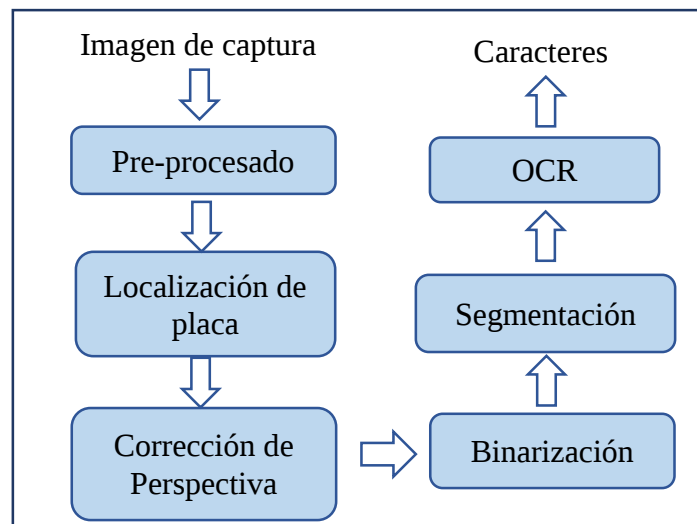


Figura 12 Diagrama de bloques del sistema implementado.
Fuente: Elaboración propia

3.1. Captura de imagen

El sistema de adquisición de imágenes en MATLAB posee su propio Toolbox para el procesamiento. La función que ofrece Matlab para iniciar la adquisición de imágenes por cámara y su debida configuración es **imaqtool**, que es una interfaz nativa del Matlab que permite configurar parámetros como resolución, fuente de la imagen, etc.

a) **Video:** Para adquirir videos por medio de una cámara web se usa el comando `winvideo`, que por defecto el Matlab le asigna el 1.

b) **Foto:** Para adquirir una foto almacenada en un directorio y trabajar con ella, se debe almacenar en una variable, que el Matlab leerá como una matriz, de la siguiente manera. `Imagen=imread ('imagen.jpg')`.

Para mostrar la imagen se usa el comando **imshow**. Matlab puede leer formatos comunes tales como: .jpg .png, etc.

3.2. Pre-procesado

El objetivo es poner las condiciones necesarias, para el funcionamiento óptimo y eficiente del proceso, es decir acondicionar la imagen para que al momento de aplicar el algoritmo funcione de manera adecuada cada una de sus etapas y se pueda obtener con precisión la posición de la placa.

3.2.1. Escala de grises

La representación de una imagen en escala de grises, es la simplificación de la imagen en una matriz bidimensional que solo contiene información relevante de la componente de luminancia del modelo, la cual presenta 256 niveles siendo 0 para el color negro (mínima intensidad) y 255 para el color blanco (máxima intensidad). Ver Figura 13.

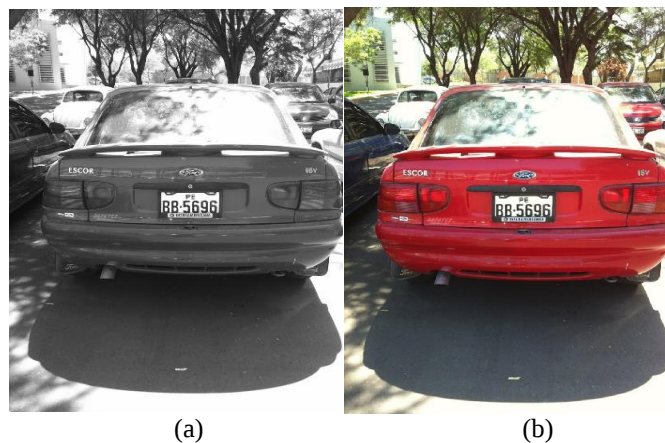


Figura 13 (a) Imagen en escala de grises (b) Imagen en formato RGB
Fuente: Elaboración propia

En MATLAB, la función que permite cambiar de RGB a escala de grises es `escala_de_grises= rgb2gray (imagen)` siendo 'imagen' una variable que contiene la matriz($M \times N \times 3$) de una imagen cualquiera en representación RGB y 'escala_de_grises' es una variable que almacena una matriz($M \times N$) con valores entre 0 y 255.

3.2.2. Reducción de la imagen

Es una operación que permite reducir las dimensiones de la matriz (imagen) que se está procesando, con la finalidad de reducir el número de iteraciones y operaciones que deba hacer el sistema y así reducir el tiempo de respuesta. Se sabe que MATLAB trabaja cada imagen como una matriz entonces el tiempo que demore el programa en realizar el procesamiento depende de las dimensiones de la matriz con la que se esté trabajando, de allí la finalidad del recorte, disminuir la cantidad de iteración que realiza el programa para hacerlo más rápido. Se debe encontrar un compromiso entre tiempo de demora del procesamiento y precisión de este, entonces es bueno recortar la imagen siempre y cuando no se pierda información relevante, para encontrar precisión en los cálculos.

De manera experimental se determinó un valor crítico para las dimensiones de la imagen: $M \times N = 1\,000\,000$ píxeles. Si la imagen tuviera menos píxeles que esta cantidad, al reducirla, se podría perder información relevante de la placa, pero si la imagen tuviese una cantidad mayor (1260x1080 píxeles) se puede recortar la imagen para reducir la cantidad de iteraciones sin perder información muy valiosa.

La función que me permite hacer esto en Matlab es **imresize**. O simplemente se pueden ir tomando los términos impares de la siguiente manera:

```
imagen =imagen(1:2:end,1:2:end,1:1:end);
```

3.2.3. Complemento de la imagen

El complemento de una imagen, se determina sustrayendo de 255 el valor de cada pixel. En este proceso, donde las placas tienen mayor área blanca (fondo) que la cantidad de área negra (letras). El algoritmo necesita números blancos en fondo negro por tal razón se procede a calcular el complemento de la imagen.

En MATLAB se puede obtener digitando las siguientes líneas:

```
[fil col]=size (imagen_escala_grises);
n=255*ones (fil,col);
imagen_escala_grises =double (imagen_escala_grises);
complemento_Imagen=n- imagen_escala_grises;
```

La Figura 14 muestra el complemento de una imagen en escala de grises.



Figura 14 Imagen en escala de grises (izquierda)- Complemento de imagen (derecha).
Fuente: Elaboración propia

La Figura 15 (a) muestra la imagen del automóvil en escala RGB, Figura 15 (b) muestra su correspondiente conversión a escala de grises y Figura 15 (c) muestra el complemento de la imagen.



Figura 15 (a) Representación de la imagen en formato RGB.
(b) Representación de la imagen en formato escala de grises.
(c) Complemento de la imagen.
Fuente: Elaboración propia

3.3. Localización de la placa

Esta etapa se basa en una característica fundamental de la placa que es tener letras negras en un fondo blanco, dado que para este proyecto se tiene en cuenta únicamente placas de este tipo es decir con fondo blanco. Cuando hay un cambio de color de blanco al negro el gradiente es máximo (255), por ende en esto se basa este análisis, en localizar zonas potenciales en las cuales exista grandes cantidades de gradientes elevados es decir zonas donde exista la mayor cantidad de cambio de color de blanco al negro.

El método usado para esto se basa en variación de gradientes, filtro Sobel y transformada de Bottom-Hat [4].

Esta etapa busca zonas potenciales que cumplan ciertas características que permita encontrar la posición de la placa en la imagen. El análisis que se plantea está relacionada con el valor de la placa en escala de grises que en este proyecto son placas de fondo blanco con letras negras, esto genera en los bordes de la placa y en las letras gradientes con valores elevados, los filtros que se evalúa para extraer esta característica fueron el operador Sobel como se muestra en la Figura 16 y la transformada Bottom-Hat como se observa en la Figura 17.



Figura 16 Filtro Sobel aplicada a imagen de placa de automóvil.
Fuente: Elaboración propia



Figura 17 Transformada Bottom-Hat aplicada en imagen capturada.
Fuente: Elaboración propia

3.3.1. Umbral óptimo(X)

Al evaluar los valores de los gradientes, se puede identificar que los que se encuentran en la zona de la placa tienen valores muy altos en comparación con los demás, pero alrededor de toda la imagen existen zonas en las que los gradientes también podrían ser elevados, por lo tanto se elaboró un procedimiento que permita diferenciar los gradientes de la zona de la placa de otros, para ello se planteó el siguiente análisis: Debido que los gradientes en la placa son grandes pues, la diferencia entre los niveles de blanco y negro es muy alta, se diseña un filtro para pasar solamente gradientes muy elevados, es decir que dependiendo de un valor crítico(umbral) gradientes mayores a él darán 1 lógico(blanco) y valores menores dará 0 lógico(negro) [12]. Se aclara que, este proceso aún no es lo suficientemente robusto, porque el umbral no puede ser constante, debido a que diferentes imágenes poseen diferente iluminación, contraste, brillo, de forma que se genera valores de gradientes en la placa tanto menores como mayores que el valor umbral y por lo tanto, se perdería información de la placa, lo que se planteó es que el valor umbral dependa de las características de cada imagen, para esto se usó el histograma en escala de grises que muestra la cantidad de píxeles vs intensidad de una imagen, con esta herramienta podemos determinar el umbral óptimo que permita diferenciar los gradientes de la placa (muy elevados), por lo cual se deduce que el filtro pasa altos localiza solo el área bajo la curva del histograma que contenga cierta cantidad de píxeles, como se muestra en la Figura 18.

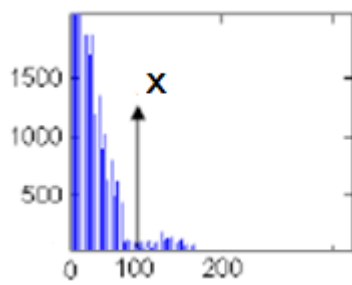


Figura 18 Histograma de gradientes de una imagen.
Fuente: Elaboración propia

Luego se busca el porcentaje bajo la curva del histograma, que sea lo suficiente para eliminar la información extra y solo capturar la información que pertenece a la zona donde se encuentra la placa. Este parámetro depende de algunos factores externos, siendo el más relevante la distancia a la cual se encuentra el vehículo del punto en cual se ha tomado la fotografía, esto quiere decir que si el vehículo se encuentra muy alejado la zona de la placa respecto a la imagen total es muy pequeña y por lo tanto requiere de una pequeña porción del área bajo la curva del histograma, y si se encontrase muy cerca el área de la placa respecto a la imagen es grande, lo cual indicaría que se necesita un porcentaje mayor de área bajo la curva del histograma.

Teniendo en cuenta que este umbral no puede ser constante pues, diferentes entornos generan diferentes gradientes, por lo cual habría que ajustar el umbral para cada situación, el problema queda resuelto al sujetar el valor del umbral a un valor probabilístico del área bajo la curva del histograma, este valor puede ser hallado experimentalmente, dicho valor fue determinado luego de varias pruebas y se determinó 3.5 % menor que el valor máximo de los gradientes.

La Figura 19 muestra la imagen binarizada con umbral de 3.5 %, en ella se puede ver que la zona de la placa puede ser diferenciada del entorno de la imagen.



Figura 19. Imagen binarizada con umbral de 3.5 %.
Fuente: Elaboración propia

3.3.2. Aplicación de operaciones morfológicas

Al observar los resultados de aplicar el filtro del umbral óptimo, se observa que aún no se obtiene como resultado únicamente la placa, sino que se encuentra con cierto “ruido”.

Para eliminar este ruido y limpiar la imagen, es necesario usar operaciones morfológicas, de erosión, apertura, dilatación y clausura [6]. La dilatación agrega a la imagen píxeles en cada 1 lógico o pixel blanco que encuentre, mientras que la erosión extrae píxeles en cada 1 lógico o pixel blanco que encuentre. El número de píxeles que extrae o agrega depende de la máscara que se le asigne a la operación por ejemplo si se trabaja con una máscara $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ esta agregará o extraerá un pixel a cada lado del pixel central.

MATLAB permite resaltar características importantes de la imagen, tales como bordes, rellenos, eliminar información extra, detectar objetos, etc. La manera de hacer llamado a estas funciones es de la siguiente manera:

```
h=ones (3,3); %Matriz de 1 de 3x3
Ie=imerode(ibw,h); %erosión
Id=imdilate (ie,h); %dilatación
```

Donde h es la máscara e ibw es la imagen lógica o en blanco y negro que se desea erosionar o dilatar.

La Figura 20 muestra la imagen Binarizada, la Figura 21 (a) muestra la imagen luego de aplicar la operación de erosión con una máscara de 5x5, mientras que la Figura 21 (b) muestra la imagen luego de la operación erosión con una máscara de 3x3, nótese que en este caso se ha eliminado gran parte del ruido sin descuidar la geometría rectangular de la placa.



Figura 20 Imagen binarizada.
Fuente: Elaboración propia

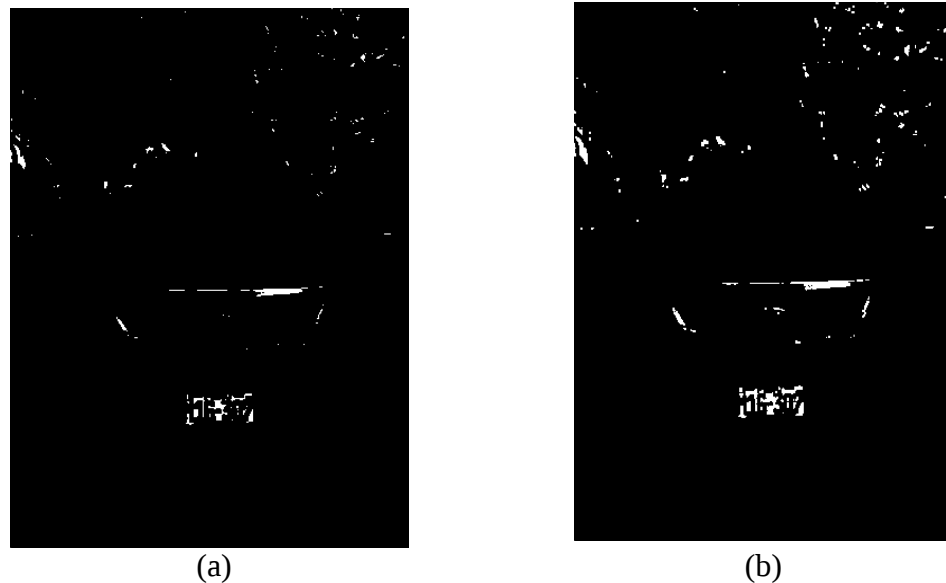


Figura 21 (a) Imagen erosionada con una máscara de 5x5; (b) Imagen dilatada con una máscara de 3x3.

Fuente: Elaboración propia

3.3.3. Ubicación de concentración de gradientes

En este punto del proceso muchas de las imágenes puede que ya no exista ruido y aparezca únicamente la placa, pero existen aún algunas imágenes que poseen suficiente ruido como para poder asegurar que la placa es la única zona en ella.

Para continuar en el proceso se diseñó un filtro que pueda determinar donde se encuentra la mayor concentración de gradientes, esto se hace con ayuda de una máscara la cual tendrá por función ir promediando los pixeles vecinos en el rango de la máscara y así localizar la zona específica. Las especificaciones de la máscara dependen de la geometría que se desee localizar, es decir como la placa es de geometría rectangular se usará una máscara que sea rectangular y con dimensiones proporcionales a la de una placa. Esto es conveniente pues suponiendo que la máscara alcanza un rango exactamente igual al de la placa el pixel central a estas tendrá un promedio muy alto pues la zona de la placa posee muchos 1 que al sumarlo y dividirlo entre el total de pixeles de la máscara obtendremos un valor aproximado a 1, de la misma manera los pixeles más cercanos al centro de la placa tendrán promedios altos mientras que los que se encuentran alejados su promedio será menor. La Figura 22 muestra una imagen y el resultado al promediar sus pixeles con una máscara con la geometría de la placa.

Cabe resaltar que la imagen obtenida es una imagen normalizada en escala de grises, nótese que la zona que posee mayor intensidad es la zona de la placa y otras zonas un poco menos intensas, ahora para poder distinguir entre estas zonas, solo basta con umbralizar nuevamente la imagen, por lo cual simplemente se hace el llamado a la función del “umbral óptimo”.



Figura 22 Imagen con píxeles promediados con una máscara con la geometría de la placa.
Fuente: Elaboración propia

Se diseñó una máscara en base a la convolución lineal para encontrar cuales son los que cumplen con la característica de pertenecer a una zona rectangular (con proporciones de la placa) y además se encuentren lo suficientemente agrupados.

3.3.4. Calibración del proceso

El número de píxeles de la foto esta denominado por $M \times N$, cada parámetro asignado al proceso así como el tiempo de demora para procesar dependen de los píxeles. Una fotografía tomada muy cerca del automóvil generara una placa de tamaño grande, ver Figura 23, mientras que una imagen del auto a una distancia prudente entre 4-8 metros generarán una imagen de la placa de tamaño medio, ver Figura 24, y si se toma una imagen muy lejana la placa será de tamaño pequeño.



Figura 23 Imagen capturada con placa de tamaño grande.
Fuente: Captura fotográfica



Figura 24 Imagen capturada con placa de tamaño medio
Fuente: Captura fotográfica

Experimentalmente, se calculó que para placas grandes y de tamaño medio el valor es de 7 % para encontrar la concentración de gradientes, el parámetro que se debe ajustar es la máscara, que deberá tener un valor aproximado al área de la placa en la imagen y una proporción geométrica igual a la de la placa, por tal razón la máscara a utilizar para placas grandes debe tener una área del 7 % del número de píxeles y una disposición de 2 a 4.5 quedando $h = \text{ones}(2k, 4.5k)$, siendo $k = \sqrt{0.07MN/9}$. La Figura 25 muestra la máscara para convolución lineal.

$$\frac{1}{9k^2} \left[\begin{array}{ccc} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{array} \right] \left. \vphantom{\begin{array}{ccc} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{array}} \right\} 2k$$

Figura 25 Dimensiones de la matriz máscara para convolución lineal.
Fuente: Elaboración propia

La Figura 26 muestra una imagen de automóvil luego de aplicarle la máscara para convolución lineal.



Figura 26 Imagen luego de aplicar convolución lineal.
Fuente: Elaboración propia

Para poder eliminar el resto de zonas se binariza la imagen, para esto se usó un umbral de 2% como se muestra en la Figura 27.



Figura 27 Imagen binarizada con umbral de 2%.

Fuente: Elaboración propia

Ahora si se puede decir, que se localizó la zona de la placa, posteriormente se pasa a dilatar un poco la imagen obtenida para asegurar que se capturó toda la placa y no correr el riesgo de capturar solo una parte. Para esto se escogió una máscara h=ones (10,60) es decir, se expandirá la zona encontrada 10 pixeles en dirección vertical y 60 en dirección horizontal. Por último, se realiza el corte a la zona localizada, ver Figura 28, esto se realiza con ayuda de la función imcrop la cual permite realizar un corte rectangular asignando la coordenada de inicio y las longitudes de corte.



Figura 28 Imagen recortada de la placa.

Fuente: Elaboración propia

3.4. Corrección de perspectiva

Esta fase del proceso, traslada a un plano frontal cualquier placa que haya sido capturada con alguna perspectiva [7]. Esta fase localiza las coordenadas de los vértices de la placa de automóvil mediante la transformada de Hough y realiza un traslado geométrico mediante la transformada proyectiva bidimensional.

La Figura 29 muestra la imagen de una placa con perspectiva.



Figura 29 Imagen original con perspectiva al momento de la captura.

Fuente: Elaboración propia

3.4.1. Transformada de Hough

Este proceso detecta la posición de la placa a través de la detección de líneas. Se utiliza como entrada al algoritmo una imagen binaria $I(x, y)$, la cual contiene información de los bordes (extraídos anteriormente). Se utiliza el valor de uno en la imagen que determina que el pixel es parte de las aristas de la placa y cero que es parte del fondo, como primer paso se colocan todas las celdas de la matriz de registros de acumulación a cero, después se recorre la imagen binaria de tal forma de que cada vez que se encuentre un pixel borde se obtiene los valores de r , realizando un barrido de α de 0 a π , sin embargo para encontrar los valores de r se emplea como coordenadas de referencia el centro de la imagen de este modo se va formando una nueva imagen $I(x, y)$ matricial donde aparece la detección del líneas.

Con el propósito de reducir el tiempo de iteración solo se utiliza los puntos de los bordes horizontales.

Se crea una matriz en Matlab que contiene los valores r y las columnas contienen los valores de α , luego se ubica la mayor cantidad de puntos que coinciden con la misma recta y se localiza los 4 vértices que envuelven a la placa, en algunos experimentos los cuatro vértices no forman un rectángulo que encierre la placa, si no formaron imágenes trapezoidales. La solución que se planteó, fue determinar las 4 líneas que envuelve la placa, luego se encuentra las intersecciones de dichas líneas y se tiene obtienen los 4 vértices que se utilizarán en la fase de la trasformada proyectiva.

Para encontrar los bordes de la imagen se separó en 2 mitades una superior y otra inferior llamadas mitad A y mitad B, luego se pudo obtener un mallado más representativo. Con la finalidad de reducir el costo computacional de esta fase se reduce el campo de iteración de alpha de 30 a -30 (datos calculados estadísticamente). De este modo, es como primero se analiza el borde superior de la Izquierda, luego se analiza el borde inferior de la Derecha, para luego pasar a analizar las líneas verticales izquierda y derecha.

En el análisis de las rectas verticales se encuentra varios problemas por la sensibilidad del Sobel con aquellas placas que tienen letras al comienzo como K o L o M donde el programa cree es el comienzo de la placa y el mallado lo hace erróneamente, por tal razón se cambia la sensibilidad del Sobel de forma que desprecie esos tipos de línea no continua para procesar solo las líneas verticales más grandes y así mejorar dicho problema.

El borde superior Izquierda almacena los puntos de intersección de la línea del borde superior con la línea vertical izquierda, borde superior derecha almacena los puntos de intersección de la línea borde superior con la línea vertical derecha, Borde inferior izquierda almacena los puntos de intersección de la línea borde inferior con la línea vertical izquierda y Borde inferior Derecha almacena los puntos de intersección de la línea borde inferior con la línea vertical derecha, es así como se obtiene los 4 puntos que formaran los vértices necesarias para envolver la placa y realizar la transformada proyectiva de la imagen como se muestra en la Figura 30.



Figura 30 Transformada de Hough aplicada a imagen de placa de automóvil.
Fuente: Elaboración propia

3.4.2. Transformada proyectiva bidimensional

Esta fase se encarga de trasladar a un plano frontal una imagen adquirida con cierta perspectiva con respecto a la cámara, esto se logra mediante la aplicación de una transformada T que es una matriz que adopta los 4 puntos encontrados por la transformada de Hough y transforma la imagen distorsionada a una imagen trasladada a un plano frontal [7]. La Figura 31 muestra cómo actúa la transformada proyectiva bidimensional.

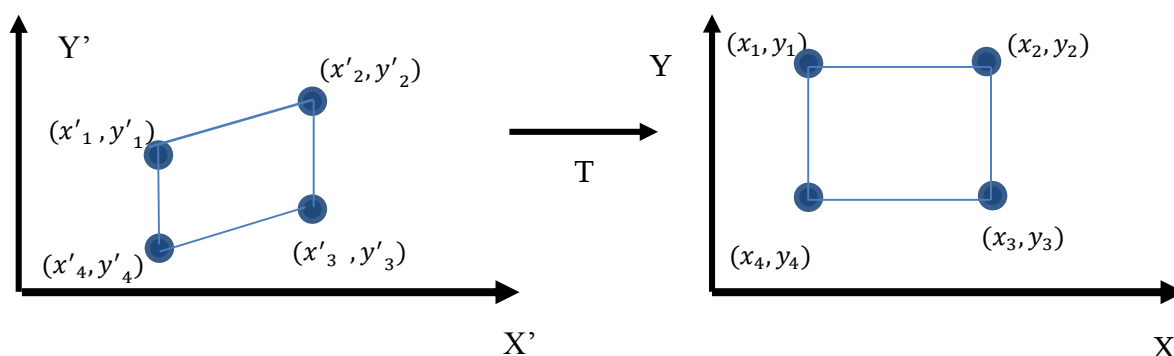


Figura 31 Transformada Proyectiva Bidimensional.

Fuente: Elaboración propia

Se puede establecer la matriz “ T ” según la teoría de geometría proyectiva

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

Luego que los vértices fueron hallados, se calcula la matriz T con la relación entre las imágenes originales I' y corregida I . Para generar las dimensiones de la nueva imagen generada se realiza a partir del ancho y el largo de la placa original tomando como referencia las dimensiones más grandes, luego se calcula la matriz de transformación T y se procede a encontrar la nueva imagen corregida.

Para la realización de la función corrige perspectiva se tiene lo siguiente:

Función corregida perspectiva: Esta función toma como entrada una imagen donde es cortada en 2 mitades, una inferior y otra superior y luego se analiza con función Sobel y transformada de Hough los bordes inferior, superior, derecha e izquierda para obtener los puntos de intersección, esta función dará como resultado la imagen corregida, esta función hace un llamado de una función llamada verificación1.

Función verificación 1: Función que verifica que los vértices hallados encierren realmente la región de los caracteres y de este modo realizar la transformación, para que esta función cumpla sus tareas de manera correcta llama a una función llamada Corrección.

Función Corrección: Función que realiza la ubicación de los vértices, el ordenamiento de los mismos y la aplicación de la transformada proyectiva, mediante esta función se transforma una imagen de una determinada perspectiva a una imagen trasladada al plano frontal utilizando la transformada T.

La Figura 32 muestra la imagen que entrada y final en el proceso de corrección de perspectiva.



Figura 32 Corrección de la perspectiva de la placa de automóvil.
Fuente: Elaboración propia

3.5. Binarización

Luego que se tiene la imagen corregida, se binariza la imagen, este proceso consiste en hacer 1 ó 0 los pixeles que están en orden matricial de cierto rango en la escala de grises 0-255, las funciones que se usan para esto se basa en una estadística avanzada donde mide el gradiente de la figura en escala de grises encontrando así un valor X, la función que binariza convierte a en 1 a todos los pixeles mayores que el número X y convierte en 0 a todos los pixeles de la imagen que sea menores a ese valor X en la escala de grises, obteniendo así una imagen binarizada, es decir un matriz de 0 y 1 que facilita su posterior análisis.

Luego se aplica el filtro Bottom-Hat con la finalidad de eliminar cierto ruido que puede presentar aun dicha placa como por ejemplo pernos letras, figuras pegadas en la placa o las letras donde señala el país de procedencia de la placa, que no son necesarios en el análisis de la placa.

Se puede visualizar en la Figura 33 que se ha eliminado la palabra Perú, así como unos pernos en la parte superior que tenía al inicio, es claro que la limpieza de la placa no es completa, sin embargo pasa por otro tipo de filtros.



Figura 33 Imagen de la placa de automóvil binarizada.
Fuente: Elaboración propia

La función en Matlab que realiza lo propuesto es:

```
function [ ibw ] = Binariza( iig )
%BINARIZA Summary of this function goes here
% Detailed explanation goes here
[fi col]=size(iig);
%%//hallamos negativo en grises//%%
n=255*ones(fi,col);
iig=double(iig);
niig=n-iig;
niig=uint8(niig);
%figure;imshow(niig)
%%//-----//%%
%%//Operación Botton-Hat para resaltar gradientes b&w//%%
st=strel('disk',11);
ibh=imbothat(niig,st);
%figure;imshow(ibh)
%%//-----//%
%=====
%
%%//Hallamos el umbral para la binarizacion//%%
u=umbral_bin(iig);
%%//-----//%%
%%//binarizamos imagen//%%
ibw=iig>u;
%ibw1=1-ibw;
%figure;imshow(ibw1)
%%//-----//%%
%%//hacemos las operaciones morfológicas de erosión y dilatación para
%%eliminar acumulación de gradientes//%%
%h=ones(3,3);
%ie=imdilate(ibw,h);%erosion
%figure;imshow(ie)
End
```

3.6. Segmentación

3.6.1. Recorte

Luego de limpiar correctamente la placa, se encuentra espacio que no tiene información de los caracteres de la placa, este espacio sin información hará que el sistema realice iteraciones sin sentido y se vuelva más lento, por ende se pasa a recortar la imagen y obtener el área de la imagen original que se tratara como se muestra en la Figura 34.



Figura 34 Recorte de la imagen.
Fuente: Elaboración propia

La Figura 35 muestra la imagen del automóvil adquirida, la Figura 36 muestra la placa localizada, corregida y binarizada, la Figura 37 muestra el resultado de la función recorte.



Figura 35 Imagen de Entrada.
Fuente: Captura fotográfica



Figura 36 Imagen de placa de automóvil binarizada.
Fuente: Elaboración propia



Figura 37 Imagen final de la función recorte.
Fuente: Elaboración propia

El código en Matlab que realiza lo propuesto es:

```
function [ rct ] = recorte( iig,n )
[fi col]=size(iig);
%%//mostramos la imagen de la placa%%
L=bwlabel(iig); %Crea regiones
stats=regionprops(L,'area'); %Estadísticas de las regiones
Ac=n*fi*col/100; %área crítica de la placa en la foto
Idx=find([stats.Area]>(Ac)); %Diferencia las regiones las mejores están entre
0.8% y 1.2%
co=ismember(L,Idx); %Crea una imagen con dichas regiones
limcol1=round(0.1*col);
limcol2=round(0.9*col);
lim1=round(fi/2-0.15*fi);
lim2=round(fi/2+0.15*fi);
k=1;
i=1;
while k<=1 & i<limcol1
    P=sum(co(lim1:lim2,i));
    if P==0
        k=i;
    end
end
```

```

        i=i+1;
    end
    l=col;
    j=col;
    while l>=col & j>limcol2
        P=sum(co(lim1:lim2,j));
        if P==0
            l=j;
        end
        j=j-1;
    end
    refile=round(fi/2);
    m=1;
    q=refil;
    a=l-k;
    while m==1 & q>0
        P=sum(co(q,k:l));
        if P==0 || P==a
            m=q;
        end
        q=q-1;
    end
    p=fi;
    r=refil;
    while p==fi & r<fi
        P=sum(co(r,k:l));
        if P==0
            p=r;
        end
        if P>l-k-1
            p=r;
        end
        r=r+1;
    end
    b=p-m; corte=[k m a b];
    rct=imcrop(co,corte);
end

```

Se muestra a continuación las diferentes etapas del sistema. La Figura 38 muestra la imagen adquirida, Figura 39 muestra la imagen luego de haber sido sometida por la función corrige perspectiva, la Figura 40 muestra la placa luego de pasar por la función Binariza y la Figura 41 muestra el resultado de la función recorte.



Figura 38 Imagen Original.
Fuente: Captura fotográfica.



Figura 39 Imagen con corrección de perspectiva.
Fuente: Elaboración propia



Figura 40 Imagen Binarizada.
Fuente: Elaboración propia



Figura 41 Imagen Recortada.
Fuente: Elaboración propia

3.6.2. Segmentación

Implica el fraccionamiento de una imagen en segmentos de píxeles que son homogéneos con respecto a algún criterio.

Se tiene un matriz $P=[m \times n]$ binaria (Figura 42), la cual es la imagen de la placa, se procederá al recorte vertical y luego horizontal.

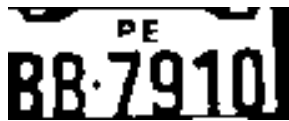


Figura 42 Placa vehicular binarizada.
Fuente: Elaboración propia

a) Recorte horizontal

Localiza la parte que se encuentran los caracteres. Realiza lo siguiente:

Se obtiene la sumatoria de los valores de cada fila como muestra la ecuación (29)

$$RH = \begin{bmatrix} \sum_{i=1}^n p_{1i} \\ \sum_{i=1}^n p_{2i} \\ \dots \\ \sum_{i=1}^n p_{3i} \end{bmatrix} \quad (29)$$

Cada valor del vector RH puede tomar valores desde 0 hasta el número de columnas de la matriz P según ecuación (30)

$$0 \leq \sum_{i=1}^n p_{ji} \leq n \quad (30)$$

Se binariza el vector columna RH, ver Figura 43.



Figura 43 Binarización matriz RH.
Fuente: Elaboración propia

Recorto la parte que contiene la letra, ver Figura 44.

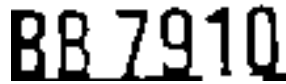


Figura 44 Recorte horizontal.
Fuente: Elaboración propia

b) **Recorte vertical**

Separa los caracteres de la placa. Para esto realiza lo siguiente:

Se obtiene la sumatoria de los valores de cada columna, según ecuación (28).

$$RV = \left[\sum_{i=1}^m p_{i1} \quad \sum_{i=1}^m p_{i2} \quad \dots \quad \sum_{i=1}^m p_{i3} \right] \quad (28)$$

Cada valor del vector RV puede tomar valores desde 0 hasta el número de filas de la matriz P, según ecuación (29).

$$0 \leq \sum_{i=1}^m (m_i, n_j) \leq n \quad (29)$$

Aunque la placa ya está filtrada, a veces pueden existir objetos no deseados, para desaparecer esos objetos lo que se hace es obtener la suma de casi todas las filas de cada una de las columnas de manera que se tendrá una matriz de tamaño [1 x número de columnas de imagen], donde los valores obtenidos indican si el valor obtenido en cada columna se acercan al número de filas tomadas, si el contenido de casi toda la columna es blanco por tanto no existirá carácter, en cambio si se acerca a 0 en la columna existe carácter, es decir, se tendrá una distribución de grises, ver Figura 45.

Figura 45 Distribución de nivel del gris de la placa de automóvil
Fuente: Elaboración propia

Ahora lo que se hace es comparar los números obtenidos en la matriz con el número de filas tomadas, si ese valor se acerca al número de filas, se convierte en blanco (1 lógico) y si no se convierte en negro (0 lógico), ver Figura 46.

Figura 46 Distribución binaria de la placa de automóvil.
Fuente: Elaboración propia

Al ya tener una imagen lógica, se procede a ver cuándo existe una variación de color, si hay una variación positiva, entonces significa que de negro pasa a blanco ($1-0=0$) por tanto podemos decir que el carácter finaliza allí, en cambio si hay una variación negativa, significa que empieza un carácter. La Figura 47 muestra la matriz RV y el recorte respectivo de cada uno de sus caracteres. Por ende se obtiene la matriz de cada carácter de la placa.

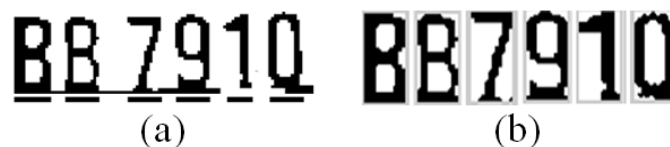


Figura 47 (a) Binarización matriz RV. (b) Recorte Vertical.
Fuente: Elaboración propia

c) Problemas presentados

Se tiene que coger un número razonable de filas por los siguientes motivos:

Si se toma todas las filas se pierde información. Comúnmente los “ruidos” se presentan en la parte inferior por tanto para tener una mejor aproximación se desprecia, como se muestra en la Figura 48.



Figura 48 Error por tomar todas las filas.
Fuente: Elaboración propia

Si se toma pocas filas, hay caracteres como la letra J, donde la información importante desaparece, como se muestra en la Figura 49.

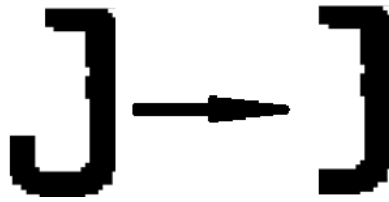


Figura 49 Error por tomar pocas filas.
Fuente: Elaboración propia

3.7. Reconocimiento óptico de caracteres - OCR

También conocido como OCR, consiste en el reconocimiento digital de imágenes para la obtención de distintos caracteres, es decir si se tiene un texto en formato de imagen se convierte a texto en formato ASCII (código estándar para el intercambio de información).

El OCR desarrollado se basa en plantillas que contienen todos los números y letras del abecedario, lo cual al tener segmentadas los caracteres, se puede comparar con dichas plantillas a través de la función correlación de Matlab para así saber cuál es la plantilla indicada.

Se crea plantillas de todas las letras y números, ver Figura 50.

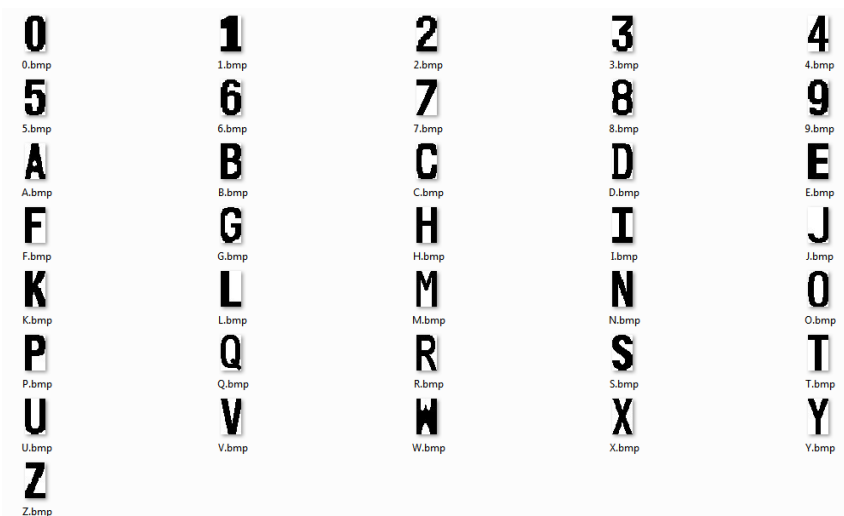


Figura 50 Plantilla de letras.
Fuente: Elaboración propia

3.7.1. Reducción

Ya que las plantillas creadas tienen una dimensión 42x24 pixeles, se debe reducir los caracteres de la placa a dicha dimensión, ver Figura 51.



Figura 51 Reducción de imagen.
Fuente: Elaboración propia

3.7.2. Generación de plantilla

Se importa las imágenes de las plantillas que se encuentran en formato de imagen .bmp a formato lógico, para tener la matriz de cada una de las plantillas, ver Figura 52.

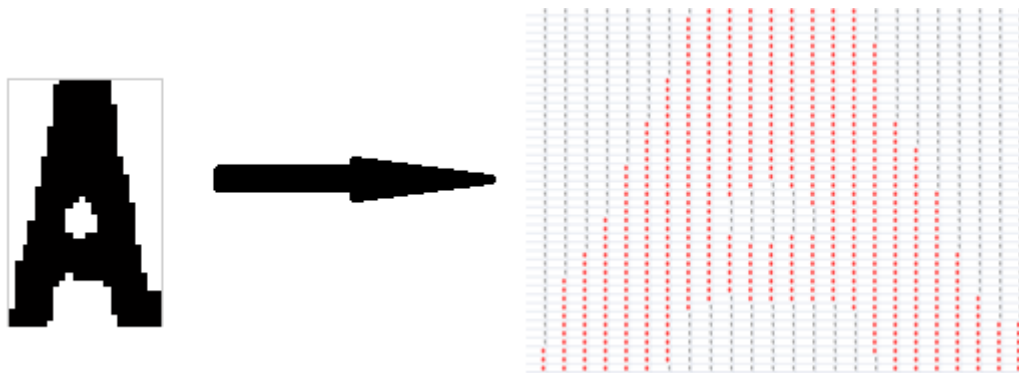


Figura 52 Formato lógico (blanco es 1 y negro es 0).
Fuente: Elaboración propia

3.7.3. Comparación de matrices

Luego se compara la matriz de cada carácter obtenidos de la placa con las plantillas creadas.

Para la comparación se usó el método de correlación de Pearson que es una técnica para investigar la relación entre dos variables continuas y cuantitativas [8]. El coeficiente de Pearson (r) mide la fuerza de asociación entre dos variables aleatorias, para este caso una variable es la matriz de cada uno de los caracteres obtenidos de la placa y la otra son las matrices de cada una de las plantillas.

La función para hallar la correlación de dos matrices en Matlab es “corr2”, la cual toma la correlación de Pearson, según ecuación (30).

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A}))^2 (\sum_m \sum_n (B_{mn} - \bar{B}))^2}} \quad (30)$$

Donde $\bar{A}$ y $\bar{B}$ son los promedios de los valores de A y B respectivamente.

Si el término r es igual a 1, significa que existe una relación directa entre A y B, es decir las matrices comparadas son iguales. Si el término es -1 significa que la relación es inversa. En cambio si el resultado es 0, significa que no existe relación entre las matrices.

Mientras mayor es r, significa que mayor es la igualdad de las matrices.

Se realizó una plantilla para letras y otra para números, con la finalidad de reducir el tiempo de búsqueda (reducir el costo computacional) y aumentar la fiabilidad del análisis, para eso se analizó la distribución de los caracteres en la placa de los automóviles en Perú, ver Figura 53. Se puede ver los siguientes patrones en las placas:

- 1° carácter es una letra.
- 2° y 3° carácter puede ser tanto letra como número.
- 4°, 5° y 6° carácter son sólo números.



Figura 53 Distintos tipos de placas.

Fuente: Elaboración propia

Debido a la simplicidad lógica el resultado es inmediato y su varianza es constante durante todos los experimentos. Este método es simple y el costo computacional es menor, el proceso se realiza de forma rápida.

Capítulo 4

Análisis del sistema de visión artificial

El sistema implementado tiene un costo computacional de 2.6 segundos cuando analiza imágenes que han sido previamente capturadas y guardadas en un computador, la iniciativa de este estudio es implementar este sistema en un análisis de video a tiempo real, para lograr este objetivo se diseñó una nueva etapa que tiene como fin buscar la presencia de una placa de fondo blanco haciendo un análisis de gradientes del entorno.

La adquisición de la imagen por la cámara de video se denomina frame, se debe buscar un tiempo adecuado de muestreo entre los frames con la finalidad de bajar el costo del sistema sin que se pierda información, para esta investigación se usó un intervalo de 10 frames por segundo es decir el tiempo de muestreo fue de 0.1 segundos, el costo computacional de esta nueva etapa es de 0.09 segundos por análisis de frame y debido a que es menor que el tiempo de muestreo permite analizar cada frame del video a tiempo real sin ninguna interferencia.

En el Toolbox de Matlab se encuentran diferentes funciones que permiten establecer el tiempo de muestreo entre frame y frame, así como el número máximos de frames que se desea analizar.

A continuación se exponen las funciones utilizadas para el muestreo.

-adquisición del video:

```
handles.vidobj = video input ('winvideo', 1, 'YUY2_640x480');
```

-tiempo entre frame y frame:

```
handles.vidobj.FrameGrabInterval = 10;
```

-inicio de video:

```
start(handles.vidobj);
```

El análisis de los gradientes es un patrón para detectar la presencia de una placa de automóvil, esta nueva etapa analiza la presencia de gradientes muy altos en la placa generado por la diferencia entre el fondo blanco (255) y las letras negras (0), por ende cuando encuentra gradientes elevados que sobrepasen un cierto umbral determinado por auto sintonía indica la presencia de una placa en el entorno y luego la procesa con el fin de identificar los caracteres de la placa.

La Figura 54 expone gráficamente lo explicado.

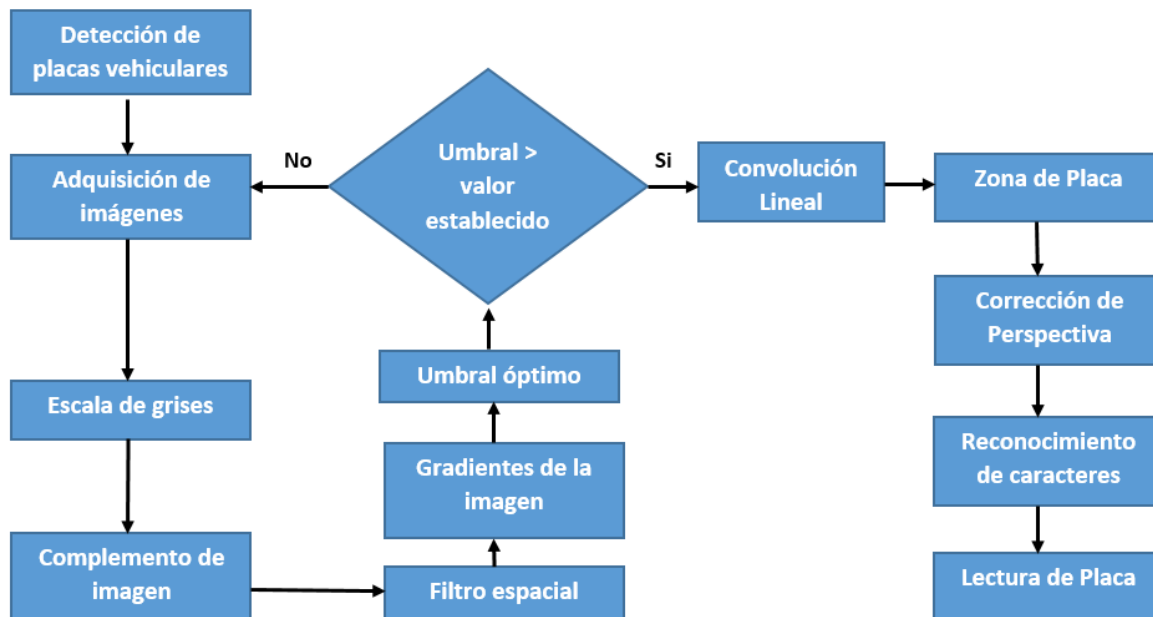


Figura 54 Diagrama de flujo del sistema en tiempo real.
Fuente: Elaboración propia

Como se explicó en el apartado 2.5.1 el gradiente es la diferencia de los píxeles vecinos en una dirección que puede ser vertical, horizontal, radial, etc, la Figura 55 muestra los máximos gradientes como un número adimensional resultado del análisis frame a frame del video conforme transcurre el tiempo, los puntos dentro de los círculos rojos son los instantes donde aparece una placa frente a la cámara y en el cual se visualiza con facilidad la presencia de mayores gradientes.

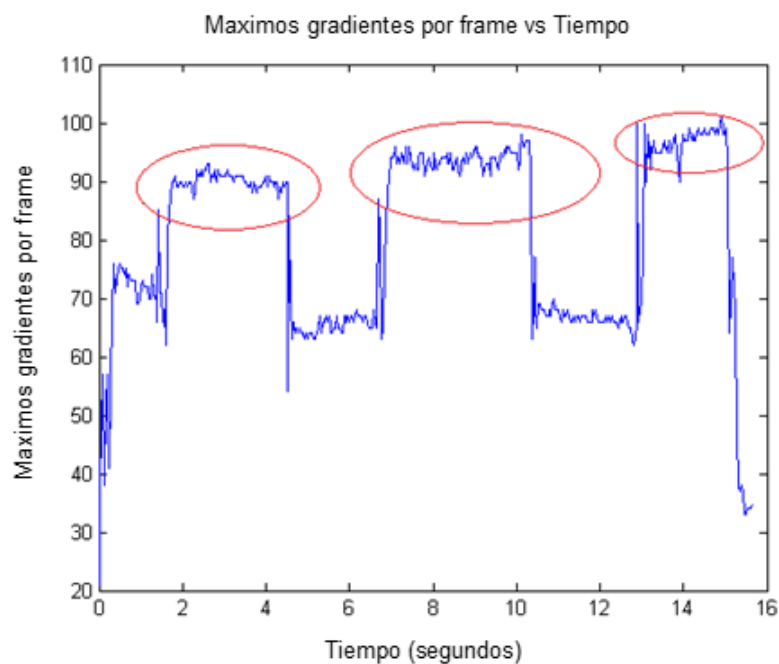
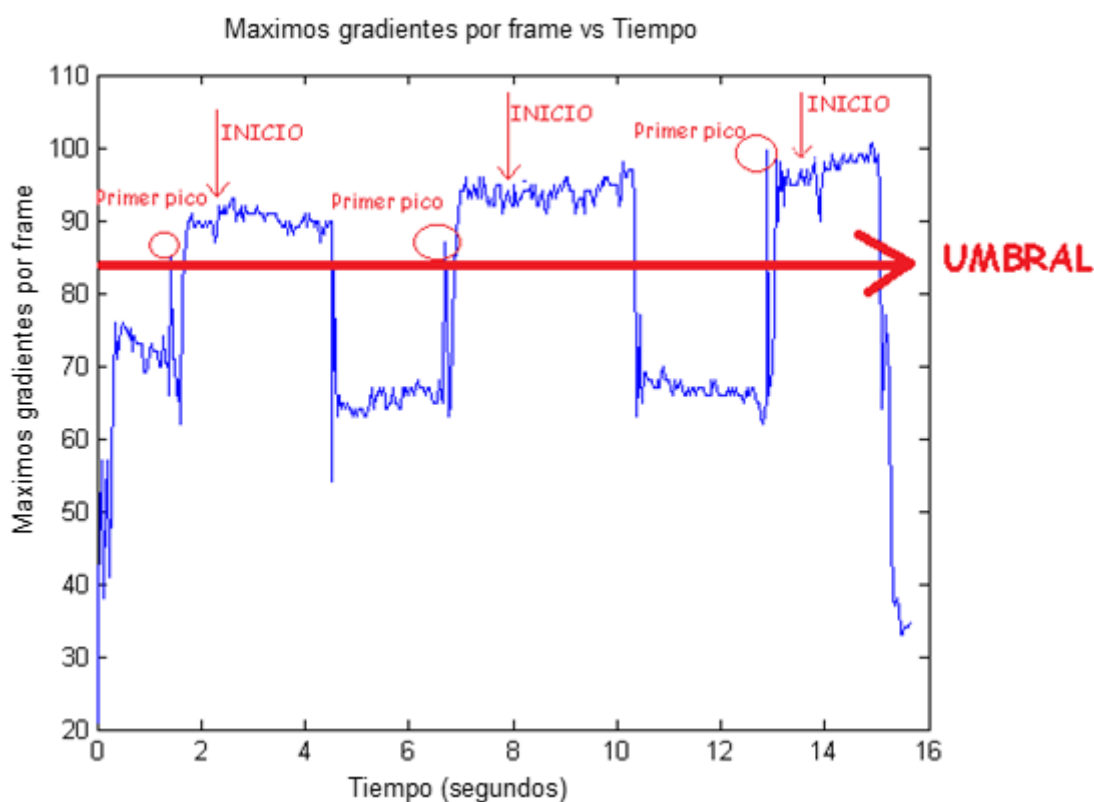


Figura 55 Gradientes vs tiempo.
Fuente: Elaboración propia

Se buscó un gradiente crítico (Umbral), para detectar la presencia de la placa, este Umbral no es un valor constante ya que depende del entorno donde se adquiere la imagen, para lograr esto se diseñó un algoritmo que analiza el entorno, este algoritmo denominado auto sintonía determina el gradiente máximo del entorno y determina un valor de Umbral que proporcionara de manera acertada la presencia de la placa.

Para determinar cuál es el momento apropiado para empezar el análisis del “frame”, se debe saber que para que el proceso sea exitoso la placa debe estar estática ya que si estuviese en movimiento la imagen puede ser distorsionada y no se leería correctamente la placa, entonces el instante inicial no es el instante en que el primer gradiente supera el umbral pues cabe la posibilidad de que este sea un pico de luminosidad o que recién el automóvil se esté deteniendo frente a la cámara, por lo tanto un instante prudente para determinar el inicio, es esperar a que unos 4 o 5 gradientes (de los “frames” o muestras) ya hayan superado el valor umbral. Así también esta etapa del programa debe dejar de procesar luego de reconocida la placa, sino se caería en un procesamiento constante en todo el tiempo que el auto se encuentra detenido enfrente de la cámara. A continuación la Figura 56 muestra el análisis gráficamente.



1.

Figura 56 Instante de ocurrencia de evento e inicio del proceso completo.

Fuente: Elaboración propia

El programa solo debe procesar cuando se da el evento inicio y esperar hasta que vuelva ocurrir un nuevo evento para procesar nuevamente, así se procesa solo cuando es necesario, haciendo el programa más eficiente y rápido.

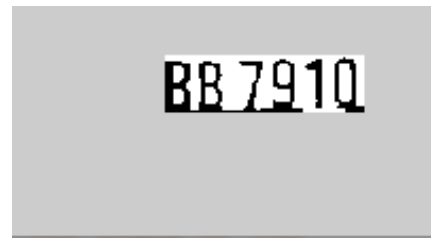
4.1. Pruebas realizadas

4.1.1. Prueba N°1

La Figura 57 muestra la adquisición de la imagen, la Figura 58 muestra el resultado del sistema de detección de placa, se puede visualizar que la detección y el reconocimiento de caracteres se realizaron de manera exitosa.



Figura 57 Imagen Inicial - Prueba N°1
Fuente: Elaboración propia



BB7910
Figura 58 Resultado –Prueba N°1
Fuente: Elaboración propia

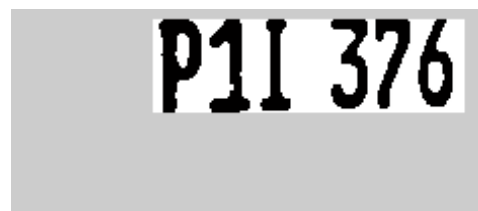
Prueba Exitosa: Se pudo reconocer todos los caracteres de la placa.

4.1.2. Prueba N°2

La Figura 59 Muestra la adquisición de la imagen, la Figura 60 muestra el resultado del sistema de detección de placa, se puede visualizar que la detección y el reconocimiento de caracteres se realizaron de manera exitosa.



Figura 59 Imagen Inicial - Prueba N°2.
Fuente: Elaboración propia



P1I376
Figura 60 Resultado –Prueba N°2.
Fuente: Elaboración propia

Prueba Exitosa: La imagen se tomó de noche, la luminosidad del flash ayuda a recalcar la zona de la placa del contexto de la imagen. Se pudo reconocer todos los caracteres de la placa.

4.1.3. Prueba N°3

La Figura 61 muestra la adquisición de la imagen con perspectiva, Figura 62 muestra el resultado del sistema de detección de placa, se puede visualizar que la detección y el reconocimiento de caracteres se realizaron de manera exitosa.



Figura 61 Imagen Inicial –Prueba N°3.
Fuente: Elaboración propia

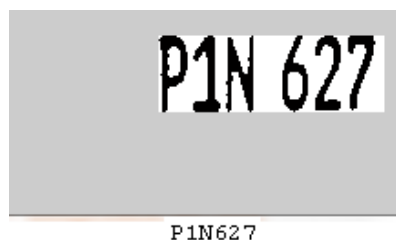


Figura 62 Resultado –Prueba N°3.
Fuente: Elaboración propia

Prueba Exitosa: Se pudo reconocer todos los caracteres de la placa

4.1.4. Prueba N°4

La Figura 63 muestra la adquisición de la imagen, Figura 64 muestra el resultado del sistema de detección de placa, se puede visualizar que detecto la placa pero fallo al momento de reconocer los caracteres.



Figura 63 Imagen Inicial –Prueba N°4.
Fuente: Elaboración propia

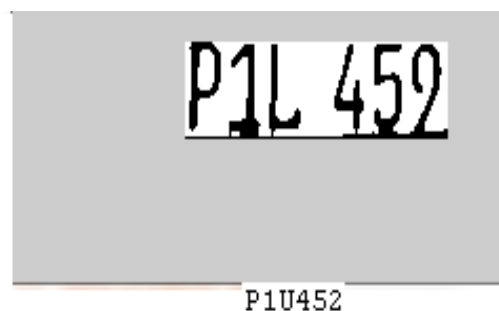


Figura 64 Resultado –Prueba N°4
Fuente: Elaboración propia

Prueba Error: Reconoció la Letra L por la letra U y el numero 9 por 2.

4.2. Discusión de resultados

Como se puede observar el método es muy efectivo ya que en la mayoría de las pruebas hechas ha reconocido la placa del vehículo.

La etapa implementada para detectar la presencia de placa de automóvil en ocasiones presento problemas, en especial cuando en el entorno a analizar existía presencia de objetos con colores blanco y negro contiguos produciendo valores grandes de gradientes muy parecidos a los gradientes presentes en las placas de automóviles con fondo blanco y letras negras, de manera que analizaba imágenes donde no existía presencia alguna de placa y dando como resultado que no se encontró placa, este problema puede ser solucionado como trabajo futuro incluyendo el filtro Canny para detectar bordes al análisis que se realiza a los frames en el video, este nuevo filtro sin duda alguna aumentara el costo computacional de esta etapa, solo se deberá cuidar que este nuevo tiempo de procesamiento sea menor al tiempo de muestreo entre frames por segundo del video.

En todos los casos vistos anteriormente, se puede observar que el problema principal es el reconocimiento de caracteres como se observa en el caso 4, lo que sucede es que algunas ocasiones la matriz de algún carácter tiene ruido y que al compararla con una plantilla, el coeficiente de correlación es más bajo, por ende otra plantilla puede ser elegida de manera errónea como su correspondiente.

Cabe indicar con este proceso que las únicas placas que reconoce son las de fondo blanco, ya que al cambiar de color el gradiente obtenido es muy bajo y no permite localizar la zona de la placa.

Capítulo 5

Artículo científico

Sistema para detección de placas mediante el procesamiento digital de imágenes

Mundaca G., Ipanaqué W., Arbulú J., Otero L.

(e-mail: george@mundaca.pe, william.ipanaque@udep.pe, jlac_12_3@hotmail.com,
lotero191@hotmail.com)

Resumen: En este artículo se desarrolla un método para el reconocimiento de caracteres en placas de automóviles, basado en filtrado espacial, operaciones morfológicas y transformadas matemáticas. El sistema está estructurado por etapas y desarrollado completamente con la herramienta “Video and Image Processing Toolbox” que opera en el entorno Matlab. Debido al costo computacional del proceso, para que este se desarrolle de manera óptima, es necesario que la evaluación de la placa se lleve a cabo a velocidad baja y a una distancia prudente, cabe indicar que para obtener mejores resultados se debe trabajar con una cámara de óptima resolución.

Palabras claves: Reconocimiento de placas vehiculares, Filtros espaciales

1. INTRODUCCION

Debido a su importancia en la actualidad el desarrollo de software de procesamiento de imágenes es utilizado en diferentes ámbitos tales como el campo vehicular, la seguridad, la topografía, etc. El desarrollo del software se dio en el entorno MatLAB que nos permita la identificación en código alfanumérico de las placas vehiculares en un centro de parqueo vehicular, dando una solución a la problemática que genera no llevar un control automatizado del flujo de autos en los centros de parqueo, evitando así cualquier tipo de fraude.

Este artículo expone de manera concisa las etapas de este proyecto de manera que el lector tenga un panorama nuevo del desarrollo del software de procesamiento de imágenes e incentive su desarrollo

2. MARCO TEÓRICO:

A continuación se describen algunos teoremas que respaldan la lógica algorítmica del sistema.

Filtros espaciales:

Filtros son operaciones de pixel por pixel donde el valor del nuevo pixel depende del pixel original y de los pixeles que están en relación de vecindad con él. La función o matriz que relaciona el pixel original con su vecindad se le denomina “mascara”.

Ejm:
$$H(i, j) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Teorema de la Convolución:

Operación que me permite conectar dos funciones continuas o discretas en una sola. Para dos funciones discretas bidimensionales I y H la convolución lineal es definida como:

$$I'(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} I(x-i, y-j) \cdot H(i, j)$$

O de manera compacta: $I' = I * H$

Esto es una herramienta que me permite conectar una matriz (imagen) con otra matriz (filtro o máscara).

Derivada parcial y Gradiente:

Es la derivada de una función multidimensional a lo largo de un eje coordenado, por ejemplo:

$$\frac{\partial I}{\partial x}(x, y) \quad , \quad \frac{\partial I}{\partial y}(x, y)$$

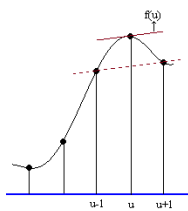


Figura 1. Primera derivada de una función

Denota la derivada parcial de la función de la imagen $I(u, v)$ con respecto a la variable u o v . El vector gradiente de la función I en el punto (x, y) es:

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) \\ \frac{\partial I}{\partial y}(x, y) \end{bmatrix}$$

El valor es: $|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

Esta propiedad es importante para la localización de los puntos bordes de la imagen.

Filtro Sobel:

Es una máscara que combina de las derivadas parciales $\frac{dI}{du}$ y $\frac{dI}{dv}$ obtenemos para cada pixel un valor de gradiente.

$$\frac{dG(x, y)}{dx} = \frac{G(x - \Delta x, y) - G(x, y)}{\Delta x}$$

$$\frac{dG(x, y)}{dx} = \frac{G(x, y) - G(x - \Delta x, y)}{\Delta x} = \frac{G(x + \Delta x, y) - G(x - \Delta x, y)}{2\Delta x}$$

Binarización:

Consiste en convertir una matriz de escala de grises, a una matriz binaria basándose en un número de umbral U .

Se tiene una matriz A [$m \times n$]:

Tal que: $0 \leq U \leq 255$

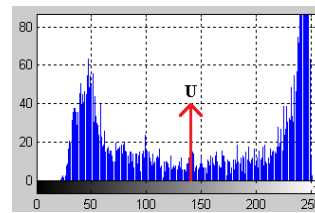


Figura 2. Histograma matriz A.

Si: $0 \leq a \leq U$, entonces $a = 0$; $U \leq a \leq 255$, entonces $a = 1$

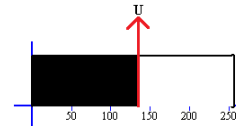


Figura 3. Binarización de matriz

Trasformada de Hough:

Con la ayuda de esta transformada localizamos formas paramétricas a partir de una distribución de puntos presentes en una imagen.

Esta transformada localiza las líneas sobre imágenes binarias producidas por la umbralización del gradiente. Una línea en el espacio es de la forma:

$$x \cos \theta + y \sin \theta = r$$

Dicha ecuación permite una cuantificación lineal de sus parámetros r y θ , por lo que el espacio de Hough (espacio donde se localiza las posibles rectas) y el punto $p_i = (x_i, y_i)$ se relacionan según:

$$r_{x_i+y_i} = x_i \cos \theta + y_i \sin \theta$$

Donde el intervalo de los valores para θ es $0 \leq \theta \leq \pi$, tomamos de referencia el centro de la imagen (x_c, y_c) el intervalo de los valores de r es:

$$-r_{max} \leq r_{x,y}(\theta) \leq r_{max}$$

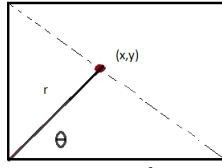


Figura 4. Parámetros

Transformada Projectiva Bidimensional:

La operación aplicada a una imagen produce: $I(x, y) \rightarrow I'(x', y')$ en donde el valor del pixel es cambiado además de la posición en la nueva imagen.

La cual para cada coordenada de la imagen $I(x, y)$ original $x = (x, y)$; $x' = (x', y')$ de la imagen $I'(x', y')$ esto es: $x \rightarrow x' = T(x)$

Tenemos la imagen A de la forma

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}$$

$$\begin{bmatrix} \hat{x}' \\ \hat{y}' \\ \hat{h}' \end{bmatrix} = \begin{bmatrix} h'x' \\ h'y' \\ h' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = \frac{1}{h'}(a_{11}x + a_{12}y + a_{13}) = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + 1}$$

$$y' = \frac{1}{h'}(a_{21}x + a_{22}y + a_{23}) = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + 1}$$

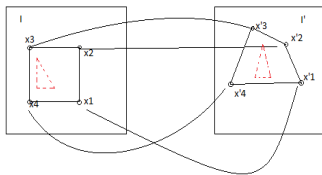


Figura 5. Transformación geométrica

Los parámetros de la transformación projectiva son determinados a partir de 4 coordenadas (x_1, x'_1) ; (x_2, x'_2) ; (x_3, x'_3) ; (x_4, x'_4) , cabe indicar que $x = (x_i, y_i)$ son los puntos de la original imagen y $x'_i = (x'_i, y'_i)$ son los puntos de la imagen transformada de modo que:

$$\begin{aligned} x'_i &= a_{11}x_i + a_{12}y_i - a_{13} - a_{31}x_ix'_i - a_{32}y_ix'_i \\ y'_i &= a_{21}x_i + a_{22}y_i - a_{23} - a_{31}x_ix'_i - a_{32}y_ix'_i \end{aligned}$$

Donde puede ser $i=1, 2, 3, 4$

Por ende tenemos $x' = A \cdot x$; $x = A^{-1} \cdot x'$

La inversión de la matriz A puede calcularse a través de la matriz inversa T^{-1} , para eso es necesario los puntos del polígono P_1 a otro polígono cualquiera de cuatro puntos P_2 de la manera siguiente

$$P_1 \rightarrow T_1^{-1} \rightarrow C \rightarrow T_2^{-1} \rightarrow P_2$$

$$x' = T(x) = T_2(T_1^{-1}(x))$$

Coefficiente de correlación de Pearson (r):

Define un factor que mide el grado de igualdad entre dos matrices. Tenemos dos matrices binarias

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

Tomamos los pares (x_i, y_i) en el plano cartesiano.

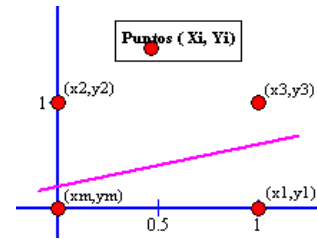


Figura 6. Plano cartesiano

El coeficiente de correlación es la pendiente de esta lineal

$$r = \frac{\sum_{i=1}^m (x_i \times y_i) - \bar{X} \times \bar{Y}}{S_x \times S_y}$$

Dónde:

$$\bar{X} = \frac{\sum_{i=1}^m x_i}{m} \quad ; \quad \bar{Y} = \frac{\sum_{i=1}^m y_i}{m}$$

$$S_x = \sqrt{\frac{\sum_{i=1}^m x_i^2}{m} - \bar{X}^2} \quad ; \quad S_y = \sqrt{\frac{\sum_{i=1}^m y_i^2}{m} - \bar{Y}^2}$$

$\bar{X}$: Promedio de valores de matriz X; $\bar{Y}$: Promedio de valores de matriz Y; S_x : Desviación estándar de variables de X; S_y : Desviación estándar de variables de Y.

3. ETAPAS DEL SISTEMA:

Pre-Procesado

El objetivo es poner las condiciones necesarias, para el funcionamiento óptimo y eficiente del proceso.

Reducción de la imagen:

Es una operación que me permite reducir las dimensiones de la matriz (imagen), con la finalidad de reducir el número de iteraciones y poder reducir el tiempo de la respuesta.

Escala de grises:

La representación de una imagen en escala de grises, es la simplificación de la imagen en una matriz bidimensional que solo contiene información relevante de la componente de luminancia del modelo a color YIQ, presentando niveles de intensidad entre 0-255 siendo 0 para el color negro y 255 para el blanco.

Complemento de la imagen:

El complemento de la imagen se determina restando a 255 el valor de intensidad de cada pixel perteneciente a la imagen.



Figura 7. Representación de la imagen (a) RGB (b) Escala de grises (c) Complemento de la imagen

Localización de la Placa

La finalidad de esta etapa es encontrar zonas potenciales que cumplan ciertas características morfológicas, aplicando filtros espaciales que me permitan extraer características específicas de la placa. El análisis planteado en esta etapa enfoca dos características fundamentales de las placas. La primera característica está relacionada con el valor de la intensidad de los colores de la placa en escala de grises que en general en una placa es de fondo claro con letras oscuras, esto genera en los bordes de la placa y de las letras gradientes con valores elevados. Los filtros que evaluamos para extraer esta característica fueron el operador Sobel y la transformada Bottom-Hat.



Figura 8. (a) Ordenador Sobel (b) Escala de grises



Figura 9. (a) Ordenador Sobel (b) Transformada Bottom-Hat (c) Escala de grises

A partir de la imagen filtrada se puede separar aquellas zonas de pixeles donde los gradientes son más elevados, esto es posible binarizando la imagen con un umbral que sea lo suficientemente alto para poder descartar zonas de bajos gradientes pero suficientemente bajo para no descartar los mismos gradientes de la placa, teniendo en cuenta que este umbral no puede ser constante pues diferentes entornos generan diferentes gradientes por lo cual habría que ajustar el umbral para cada situación, el problema queda resuelto al sujetar el valor del umbral a un valor probabilístico del área bajo la curva del histograma, este valor puede ser hallado experimentalmente, dicho valor fue determinado luego de varias pruebas y se determinó 3.5% menor que el valor máximo de los gradientes.

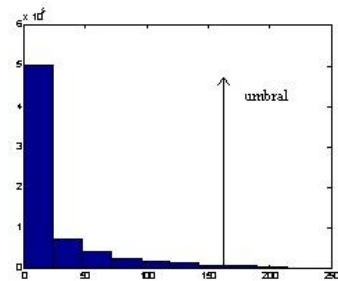


Figura 10. Histograma de gradientes de una imagen



Figura 11. Imagen Binarizada con Umbral de 3.5%

Observamos que aún existe ruido que no permite distinguir claramente la placa en toda la imagen. La segunda característica está relacionada con la geometría de la placa y la concentración de los gradientes en la misma zona. Apoyándonos en la convolución lineal se diseñó una máscara o filtro promediador el cual me permitirá encontrar cuales son los que cumplen con la característica de pertenecer a una zona rectangular (con

proporciones de la placa) y además se encuentren suficientemente agrupados.

$$\frac{1}{9k^2} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \begin{matrix} 4.5k \\ 2k \end{matrix}$$

Fig.12 dimensiones de la matriz mascara para convolución lineal

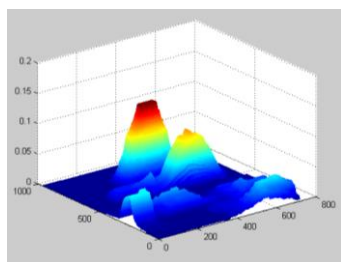


Figura 13. Zonas de concentración de gradientes

Podemos observar como la zona de la placa es la zona con mayor concentración de gradientes. Para poder eliminar el resto de zonas binarizamos la imagen, para esta Binarización se usó un umbral de 2%.



Figura 14. Imagen binarizada con umbral de 2%

Luego se puede realizar un recorte de la zona localizada para su posterior trabajo.



Figura 15. Imagen recortada de la placa

Corrección de Perspectiva

Para realizar este paso se necesita de la transformada de Hough para determinar las coordenadas de los vértices y la Transformada proyectiva Bidimensional para la distorsión de perspectiva.

Transformada de Hough:

Esta transformada nos ayuda detectar líneas sobre imágenes binarias producidas por la umbralización del gradiente. Luego de efectuada esta parte podemos ubicar la mayor cantidad de puntos que coinciden con la misma recta de parámetros (θ, ρ) , una vez que tenemos el ángulo de inclinación pasamos a encontrar el vector que encierra la recta horizontal.



Figura 16. Transformada de Hough

Transformada Proyectiva Bidimensional:

Luego de los vértices hallados en la transformada de Hough, se calcula la matriz de transformación H con la correspondencia entre las imágenes originales I' y corregida I . Para generar las dimensiones de la nueva imagen lo hacemos a partir del ancho y el largo de la placa original tomando como referencia sus dimensiones más grandes, una vez que tenemos los puntos de la matriz original y los puntos de la matriz donde va abarcar la nueva imagen corregida se calcula la matriz de transformación H .



Figura 17. Corrección de la perspectiva de la placa

Reconocimiento de Caracteres (OCR):

Segmentación:

Implica el fraccionamiento de una imagen en segmentos de píxeles que son homogéneos con respecto a algún criterio, en este caso separar cada carácter de la placa.

$$RV = \left[\sum_{i=1}^m p_{i1} \quad \sum_{i=1}^m p_{i2} \quad \dots \quad \sum_{i=1}^m p_{i3} \right]$$

El vector RV es un vector que contiene la sumatoria de los valores de intensidad de cada pixel a lo largo de las columnas de la matriz imagen de la placa. Cada valor del vector RV puede tomar valores desde 0 hasta el número de filas de la matriz P:

$$0 \leq \sum_{i=1}^m (m_i, n_j) \leq n$$



Figura 18. Vector RV.

Al binarizar el vector RV con un umbral del 80%, podemos separar los caracteres de la imagen.



Figura 19. Binarización matriz RV.



Figura 20. Recorte Vertical

Conclusiones

Se logró el objetivo de desarrollar una plataforma en Matlab que pueda reconocer los caracteres de las placas de automóviles sin importar la posición en la que se encuentra ante la cámara, el reconocimiento de caracteres de la placa del automóvil se realizó mediante OCR (Optical character recognition) y correlación de Pearson.

El sistema implementado trabaja con un costo computacional de 2.69 segundos y un nivel de aceptación medio con un error de 23 % en un universo de 80 pruebas realizadas, este error está sujeto a placas en deterioro las cuales contenían caracteres que no se podían visualizar a simple vista.

Existen aún en circulación vehículos que contienen placas en deterioro a pesar del decreto supremo 017-2008-MTC, capítulo iv - placas de exhibición, publicado el día sábado 19 de abril del año 2008 en el diario oficial el peruano que dicta que las placas en deterioro deben de ser cambiadas por unas nuevas con la intención que los caracteres presentes en las placas puedan ser visibles y reconocibles a simple vista, en caso la ley fuera acatada el error de 23% del sistema implementado se vería reducido.

La iluminación y la distancia al vehículo fueron los factores más importantes a tener en cuenta al momento de analizar cada imagen, en ocasiones cuando el vehículo se encuentra a grandes distancias resta resolución, a pesar de ello se superaron los problemas.

El sistema implementado permite localizar la placa en 100 % de los experimentos con placas de fondo blanco sin importar la perspectiva que se encuentra esta con la cámara al momento de capturar y sin importar las diferentes circunstancias de iluminación que presente el entorno.

En algunos experimentos se presentaron problemas en la etapa del reconocimiento de caracteres, lo que sucede es que en algunas ocasiones la matriz de algún carácter tiene ruido o la función recorte no se realiza con efectividad por ende al comparar con la plantilla el coeficiente de correlación es más bajo, por tanto otra plantilla puede ser elegida de manera errónea como su correspondiente.

Se realizó una plantilla para letras y otra para números, con la finalidad de reducir el tiempo de búsqueda (reducir el costo computacional) y aumentar la fiabilidad del análisis, para eso se analizó la distribución de los caracteres en la placa de los automóviles en Perú y se dedujo que el primer carácter es una letra, segundo y tercer carácter puede ser letra o número y los caracteres 4°, 5° y 6° solo son números.

La etapa de pre-procesado no fue demasiada extensa, principalmente fue obtener una imagen fácil de procesar que contenga la información de la imagen.

En la etapa de segmentación se debe de tomar pocas filas para el recorte ya que hay caracteres como la letra J donde la información importante desaparece y al final no puede ser reconocida.

En el momento que se realiza la corrección de perspectiva de la imagen capturada se encontraron varios problemas por la sensibilidad del Sobel con aquellas placas que tienen letras al comienzo como K o L o M donde el programa detectaba como el comienzo de la placa y el mallado lo hacía erróneamente, este error fue superado cambiando la sensibilidad del filtro Sobel de manera que desprecie las líneas no continuas y solo procese las líneas verticales grandes, de este modo este tipo de incidentes fueron solucionados.

El sistema implementado solo puede reconocer placas con fondo blanco y letras negras esto se debe a que solo se analizó la variación de gradientes en una sola dirección, esto puede ser solucionado si se opta por el uso de filtro Canny y densidad de gradientes que analiza la segunda derivada de los gradientes en todas las direcciones permitiendo reconocer placas con fondos de distintos colores pero a esto se le suma una carga computacional aún más elevada.

El sistema implementado reconoce los caracteres previamente segmentados apoyándose del teorema de correlación de Pearson, como resultado se tiene un reconocimiento de baja carga computacional con resultado rápidos, varianza constante y depende mucho de la calidad de la imagen de entrada, con la finalidad de reducir el porcentaje de error se puede optar por “Reconocimiento por Redes Neuronales” ya que a estas las puedes entrenar y así hacer el proceso más efectivo, debido a su compleja lógica demora para que el resultado final sea correcto ya que las redes neuronales pasa por un proceso de entrenamiento, el coste computacional es más elevado que el método de OCR pero se puede sacrificar a coste de aumentar la fiabilidad de los resultados, su implementación puede permitir reconocer placas que estén incluso deterioradas y disminuir la dependencia con respecto a la calidad de la imagen de entrada.

Bibliografía

- [1] H Ernest , "Computer Image Processing" , Academic Press, N. Y., c1979
- [2] Vázquez Alcaraz Gilberto Enrico, "Realce de imágenes utilizando técnicas de Mascaras", Capitulo 3, Tesis de Maestría, pág. 31-36. CITEDI - IPN, Mexico, 1995.
- [3] R. S. Rangel "Nueva metodología para el conteo de objetos en imágenes", Tesis de maestría, CIC- IPN, México, junio 2000.
- [4] R. C. Gonzáles, R.E. Woods. *Digital Image Processing*. Pearson Prentice Hall, 2008
- [5] "Image Processing Toolbox User's Guide For use with MATLAB" Version 8 -2012 by The MathWorks, Inc.
- [6] Aitzol Zuloaga Izaguirre, José Luis Martín González. "Visión artificial dinámica determinación de movimiento a partir de secuencias de imágenes" Doctorado En Tecnologías De La Información Universidad Del País Vasco, setiembre 1998
- [7] Domingo Mery, "Visión Artificial", Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Santiago de Chile ,09 de enero 2002.
- [8] Edwards, A. L. "The Correlation Coefficient." Ch. 4 in an Introduction to Linear Regression and Correlation. San Francisco, CA: W. H. Freeman, pp. 33-46, 1976.
- [9] Azriel Rosenfeld and Avinash C. KaK. Digital Picture Processing. Academic Press, 1982.
- [10] Fernando Martín Rodríguez, Xulio Fernández Hermida. "Un Nuevo Método, Basado en Morfología, para la Localización de Matrículas", Departamento de Tecnologías de las Comunicaciones. Universidad de Vigo. E.T.S.I.T. Ciudad Universitaria S/N. 36200 Vigo. Researchgate, setiembre de 2000.
- [11] Tesis Doctoral de F. Martín Rodríguez (Director: X. Fernández Hermida): Cap. 1 y Cap. 2 "Reconocedor Automático de Matrículas de Automóviles" Universidad de Vigo. España 1993-1995.
- [12] N. Otsu. "A Threshold Selection Method for Gray Level Histograms". IEEE Transactions on System, Man and Cybernetics. Enero, 1979.