

**ANÁLISIS COMPARATIVO DE ALGORITMOS DE PROCESAMIENTO DE
IMÁGENES APLICADO AL RECONOCIMIENTO DE PLACAS DE
VEHÍCULOS**

YESID ALFONSO ANACONA BOLAÑOS

JOSÉ LUIS MIRANDA RODRÍGUEZ

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

POPAYÁN

2005

TABLA DE CONTENIDO

	Pág.
LISTA DE FIGURAS	4
1 INTRODUCCIÓN.....	1
1.1 INTRODUCCIÓN AL RECONOCIMIENTO DE PLACAS DE VEHÍCULOS	1
1.2 DEFINICIÓN DEL PROBLEMA.....	2
1.3 RESTRICCIONES	2
1.4 ASPECTOS TÉCNICOS	3
1.4.1 Placas.....	3
1.4.2 Equipos.....	4
2 PROCESAMIENTO DE LA IMAGEN.....	5
1.1 INTRODUCCIÓN	5
2.1 REPRESENTACIÓN DE UNA IMAGEN DIGITAL.....	5
2.2 HISTOGRAMAS	5
2.3 FILTRO ESPACIAL	6
2.4 SEGMENTACION BASADA EN FRONTERAS.....	7
2.4.1 Operador de Roberts.....	8
2.4.2 Operador de Sobel	8
2.4.3 Operador de Prewitt.....	9
2.4.4 Operador laplaciano.....	9
2.4.5 Detector de bordes de Canny.....	10
2.5 COLOR.....	11
2.5.1 Fundamentos del color.....	11
2.5.2 Modelo de color RGB.....	12
2.5.3 Modelo de color YIQ.....	13
2.5.4 Modelo de color HSI	14
2.5.5 Modelo de color CIE XYZ	16
2.6 TÉCNICAS BASADAS EN MORFOLOGÍA	17
2.7 UMBRALIZACIÓN	20
2.7.1 Umbralización por el método de Otsu	22
2.7.2 Umbralización adaptativa	23
3 LOCALIZACIÓN DE LA PLACA	24
3.1 INTRODUCCIÓN	24
3.2 TRANSFORMADA DE HOUGH	24
3.2.1 Acondicionamiento de la imagen	25
3.2.2 Detección de líneas con la transformada de Hough.....	25
3.3 MÉTODO DEL GRADIENTE.....	32
3.4 MÉTODO MORFOLÓGICO	33
3.5 DETECCIÓN POR COLOR	37
4 SEGMENTACIÓN	39
4.1 INTRODUCCIÓN	39
4.2 ANALISIS DE COMPONENTES CONECTADOS.....	41

4.2.1	Segmentación de caracteres por etiquetado de componentes conectados.....	44
4.3	RELACIÓN DE DISTANCIA	46
4.4	PICOS Y VALLES	49
5	EXTRACCIÓN DE CARACTERÍSTICAS	50
5.1	INTRODUCCIÓN	50
5.2	MOMENTOS	50
5.3	MÉTODO PARA EL RECONOCIMIENTO INVARIANTE A LA TRASLACIÓN, ROTACIÓN Y ESCALAMIENTO (RITRE)	53
5.4	EXTRACCIÓN DE CARACTERÍSTICAS BASADA EN COMPONENTES DIRECCIONALES DE KIRSCH	56
5.5	PUNTO FINAL	58
5.6	NUMERO DE EULER.....	59
5.7	AREA DE LOS BORDES	59
5.8	AREA	59
5.9	ZONIFICACIÓN	60
5.10	VALOR DE CADA PÍXEL.....	60
6	RECONOCIMIENTO DE LOS CARACTERES.....	61
6.1	INTRODUCCIÓN	61
6.2	FUZZY ARTMAP	61
6.2.1	ART1 Y Fuzzy ART.....	62
6.2.2	Resumen del algoritmo Fuzzy ART	65
6.2.3	Sistema Fuzzy ARTMAP	67
6.3	VECINO MÁS CERCANO HOLOGRÁFICO.....	70
6.4	RED MULTICAPA FEEDFORWARD.....	71
6.5	CLASIFICADORES ESTADÍSTICOS ÓPTIMOS	73
6.6	COMPARACIÓN DE PLANTILLAS	74
6.6.1	Construcción de la plantilla	75
6.6.2	Correlación Cruzada normalizada	76
6.6.3	Máximos - Mínimos	78
6.7	ARTMAP MODIFICADA	80
7	PRUEBAS.....	83
7.1	INTRODUCCIÓN	83
7.2	PRUEBAS DE TÉCNICAS DE LOCALIZACIÓN	83
7.2.1	Descripción de los datos de entrada.....	84
7.2.2	Transformada de Hough	84
7.2.3	Método del Gradiente	87
7.2.4	Método Morfológico.....	88
7.2.5	Color	89
7.2.6	Comparación de resultados de los métodos de localización	90
7.3	PRUEBAS DE SEGMENTACIÓN.....	91
7.3.1	Descripción de los datos de entrada.....	91
7.3.2	Análisis de componentes conectados.....	92
7.3.3	Relación de distancia	93
7.3.4	Picos y valles	93

7.3.5	Comparación de resultados de los métodos de segmentación	94
7.4	PRUEBAS DE LAS EXTRACCIÓN DE CARACTERÍSTICAS	94
7.5	PRUEBAS DE TÉCNICAS DE RECONOCIMIENTO	95
7.5.1	Descripción de los datos de entrada.....	96
7.5.2	Fuzzy ARTMAP.....	96
7.5.3	Vecino Más Cercano Holográfico	100
7.5.4	Red Multicapa <i>feedforward</i>	101
7.5.5	Clasificadores estadísticos óptimos	102
7.5.6	Comparación de plantillas	103
7.5.7	Fuzzy ARTMAP modificada.....	105
7.5.8	Comparación de resultados de los métodos de reconocimiento de caracteres	107
7.6	SISTEMA INTEGRADO.....	108
7.6.1	Resultados del sistema integrado.....	109
	CONCLUSIONES	116
	REFERENCIAS.....	118
	APÉNDICE A: LISTA DE FUNCIONES	120

LISTA DE FIGURAS

	Pág.
Figura 1.1. Ángulos y distancias máximas entre la cámara y el vehículo.....	3
Figura 1.2. Placa estándar para colombia.....	4
Figura 2.1 Histograma de una imagen de $m \times n$ con 256 niveles de gris	6
Figura 2.2. Máscara de 3x3	7
Figura. 2.3. Píxeles del entorno de un punto y dos ejemplos de máscaras.....	8
Figura 2.4. Máscaras del operador prewitt	9
Figura 2.5. Esquema de una imagen a color, con 3 planos R, G y B	13
Figura 2.6. Cubo RGB	13
Figura 2.7. Triangulo de color HSI	15
Figura 2.8. Componentes de CIE XYZ.....	16
Figura 2.9. Diagrama CIE	17
Figura 2.10. Ejemplo de elementos estructurantes.....	18
Figura 2.11. Ejemplo de erosión, (a) imagen inicial, (b) resultado de la erosión de (a).....	18
Figura 2.12. Ejemplo de dilatación, (a) imagen inicial), (b) resultado de aplicar dilatación a (a).	19
Figura 2.13. Imagen inicial.	19
Figura 2.14. Gradiente laplaciano del pez inicial.....	20
Figura 2.15. <i>Opening</i> de la imagen anterior (los círculos han sido dibujados manualmente para identificar los puntos).....	20
Figura 2.16. (a) Imagen en escala de grises, (b) histograma de una imagen en niveles de gris	21
Figura 3.1 imagen Inicial	24
Figura 3.2. (a) Escala de grises, (b) ajuste de la intensidad y (c) detección de bordes.....	25
Figura 3.3. (a) Espacio de x-y a (b) espacio de parámetros m-c, (c) a (d), lo invertido.....	27
Figura 3.4. Cuantificación del espacio m-c	28
Figura 3.5. (a) Representación normal de la recta, (b) cuantificación del plano (ρ, θ) en celdas	28

Figura 3.6. Transformada de Hough y las líneas correspondientes a los picos más altos.....	29
Figura 3.7. (a) Transformada de Hough con 7 picos encontrados (cuadros blancos), (b) líneas correspondientes.	30
Figura 3.8. (a) líneas encontradas con la transformada de Hough de los bordes verticales y horizontales (b) picos obtenidos de la imagen (a), (c) Transformada con solo bordes verticales, (d) picos de (c).	31
Figura 3.9 Imagen con Sobel horizontal	32
Figura 3.10. Detección de los caracteres.....	33
Figura 3.11. Resultado obtenido	33
Figura 3.12 Imagen original.....	34
Figura 3.13. Imagen aplicado el <i>closing</i>	34
Figura 3.14. Resta de la segunda imagen con la original.....	35
Figura 3.15. <i>Closing</i> con elemento estructurante horizontal.....	35
Figura 3.16. <i>Opening</i> con elemento estructurante horizontal.....	36
Figura 3.17. Obtención de la matrícula	36
Figura 3.18. Detección por color, (a) detección de color amarillo, (b) imagen después de aplicar un filtrado.	37
Figura 3.19. Regiones detectadas como amarillo y eliminación hasta encontrar la verdadera placa.....	38
Figura 4.1. Separación de caracteres	39
Figura 4.2. Corrección de ángulo, (a) imagen obtenida de la detección, (b) imagen girada 7 grados, (c) imagen obtenida aplacando una transformación espacial.	39
Figura 4.3 Umbralización con el método de Otsu.....	41
Figura 4.4. Tres objetos etiquetados.....	41
Figura 4.5. (a) píxel p y sus 4 vecinos, $N_4(p)$, (b) píxel p y sus vecinos diagonales, $ND(p)$, (c) píxel p y sus 8 vecinos, $N_8(p)$, (d) píxeles p y q son 4-adyacentes y 8-adyacentes, (e) píxeles p y q son 8-adyacentes pero no 4-adyacentes.	42
Figura 4.6. (a) imagen binaria, (b) imagen etiquetada usando conectividad 4, (c), imagen etiquetada usando conectividad 8.	42
Figura 4.7. Una iteración en el procesote encontrar un componente conectado.	43
Figura 4.8. Placa etiquetada por componentes conectados	44

Figura 4.9. Caracter unido al marco de la placa con las líneas de separación.....	45
Figura 4.10. Placa después de eliminar los pequeños componentes y separación de los caracteres 6 y 3.....	45
Figura 4.11. Placa colombiana	46
Figura 4.12. Negativo de la imagen original	46
Figura 4.13. Placa sin ruido.....	47
Figura 4.13. Placa sin el rectángulo alrededor de los números	47
Figura 4.15. Imagen dividida en los seis caracteres	47
Figura 4.16 Caracteres inclinados	48
Figura 4.17. Caracteres sin elementos de caracteres vecinos	48
Figura 4.18. Picos y valles de la placa	49
Figura. 5.1. Círculos sobre la letra “N” de una placa colombiana	54
Figura. 5.2. Igual número de cambios de intensidad en las letra N y M	54
Figura. 5.3. Diferencia entre los arcos de la letra N y M	55
Figura. 5.4. Código radial para diferentes tamaños del número 3.....	56
Figura. 5.5. Componentes direccionales de una imagen	57
Figura. 5.6. Notación para el entorno de un píxel, necesaria para el cálculo del gradiente de Kirsch.....	57
Figura 5.7. Máscara de Kirsch. (a) horizontal, (vertical), (c) diagonal derecha, (d) diagonal izquierda.....	58
Figura 5.8. (a) Caracter de entrada, (b) esqueletización, (c) eliminación de puntos finales falsos.	58
Figura. 5.9. Área exterior de los bordes, (a) longitud de borde 136, (b) longitud de borde 199.	59
Figura 5.10. Suma de los píxeles horizontales y verticales	60
Figura 5.11. Zonificación de una imagen de un caracter	60
Figura 6.1. Arquitectura Fuzzy ART.	63
Figura 6.2. Arquitectura ARTMAP.....	68
Figura. 6.3. Modelo de la neurona con R entradas.....	72
Figura. 6.4. Funciones (a) sigmoideal (0,1), (b) sigmoideal (-1,1), (c) lineal.....	72
Figura. 6.5. Arquitectura de la red Multicapa <i>Feedforward</i>	72

Figura 6.6. (a) Diferentes muestras de imágenes del caracter 3, (b) media de 10 muestras	75
Figura 6.7. Plantilla de los dígitos.....	76
Figura 6.8. Esquema de la correlación	76
Figura 6.9. (a) Imagen de entrada, (b), plantilla y (c) gráfica de correlación.....	78
Figura 7.1. Pasos empleados para la localización de la placa con la transformada de Hough y tiempo empleado.....	86
Figura 7.2. Sombras a causa de árboles	87
Figura 7.3. Imágenes tomadas a menor distancia.....	88
Figura 7.4. Variación del porcentaje de éxito respecto a ρ	97
Figura 7.5. Variación del porcentaje de éxito respecto la número de patrones de entrenamiento	98
Figura 7.6. Tiempo respecto al número de clases, (a) Fuzzy ARTMAP y (b) Fuzzy ARTMAP modificada.....	106
Figura 7.7. Confusión entre los números 2 y 5 utilizando los momentos	108
Figura 7.8. Sistema integrado.....	109
Figura 7.9. Interfaz con todos los métodos	110
Figura 7.10 Interfaz con los métodos de mejor resultado	112
Figura 7.11. Interfaz para agragar caracteres	114
Figura 7.12 Indica el proceso completo del reconocimiento de una placa.....	115
Figura 7.13 indica el funcionamiento de la interfaz 2	115

1 INTRODUCCIÓN

1.1 INTRODUCCIÓN AL RECONOCIMIENTO DE PLACAS DE VEHÍCULOS

En el reconocimiento automático de placas de automóviles se pueden emplear las tecnologías de procesamiento de imágenes y reconocimiento de caracteres para la identificación del vehículo.

Para el funcionamiento de un sistema de reconocimiento de placas de vehículos, se necesita una cámara CCD para la adquisición de la imagen, una tarjeta digitalizadora, un sistema de luces para poder obtener una buena imagen, tanto en el día como en la noche y un computador. Una vez instalado lo anterior se capturan imágenes de los vehículos para aplicarle el procesamiento y obtener los caracteres de la placa. Estos sistemas tienen muchas aplicaciones, como lo es en peajes para verificar si los vehículos o las personas tienen requerimientos por parte de los organismos judiciales; en los conjuntos cerrados, para controlar el ingreso de los vehículos; en los parqueaderos de buses, para saber la hora de entrada y salida de los automotores, etc.

Existen numerosas técnicas de procesamiento de imágenes aplicadas al reconocimiento de placas de automóviles, siendo unas más eficientes que otras. La eficiencia de una técnica depende del entorno en el cual se encuentra el vehículo y del tipo de placa de cada país. La idea principal en este trabajo es analizar las técnicas más sobresalientes, para definir la más adecuada en el reconocimiento de una placa colombiana.

En Colombia actualmente no se produce este tipo de sistemas para el mercado. El Ministerio de Transporte tiene un proyecto de instalar cámaras para el reconocimiento de placas en todos los peajes del país [1], por lo tanto es una tecnología que se va a empezar a utilizar, y se puede extender a muchas necesidades para el control de tráfico.

1.2 DEFINICIÓN DEL PROBLEMA

El reconocimiento de placas de vehículos se divide en cuatro etapas: El preprocesamiento de la imagen, localización de la placa, segmentación y reconocimiento de caracteres. Cada etapa tiene múltiples opciones de algoritmos, los cuales son genéricos de procesamiento digital de imágenes, siendo unos más eficientes que otros en una determinada aplicación.

El objetivo de este trabajo es investigar diferentes algoritmos para cada etapa, implementar los más destacados, realizar las respectivas pruebas, comparar la eficiencia de cada uno, y así encontrar la mejor alternativa para la solución al reconocimiento de una placa colombiana.

Por lo tanto, como el trabajo no consiste en realizar la implementación e instalación de un sistema automático de reconocimiento de placas, sino en el análisis de este, se parte de imágenes ya almacenadas. Pero para un estudio completo de estos sistemas se toman en cuenta características como: distancia de la captura de la imagen, posición de cámara y otros.

1.3 RESTRICCIONES

Para el reconocimiento de la placa es necesario que la cámara de adquisición de la imagen y el vehículo no estén a distancias muy largas, ni muy cortas, por las que se deben cumplir las siguientes condiciones:

- Máxima distancia del foco de la cámara al vehículo es de 2 mts.
- Mínima distancia del foco de la cámara al vehículo es de 0.5 mts

La posición de la cámara con respecto al vehículo puede colocarse frente a este o a un lado. En el caso de ubicar la cámara a un costado, el ángulo entre la posición de la cámara con respecto al vehículo no debe ser mayor a 40° como se ilustra en la figura 1.1.

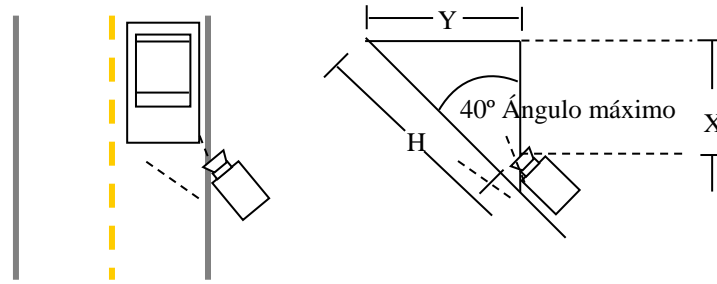


Figura 1.1. Ángulos y distancias máximas entre la cámara y el vehículo

La figura 1.1 indica el ángulo y distancias máximas que podría tener el sistema. Para un ángulo de 40° como máximo y H de 1,5mts las distancias X y Y serian:

- X debe ser como máximo de 1,14mts
- Y debe ser como máximo de 0,4mts

La captura de las imágenes no debe realizarse con el vehículo en movimiento, la escena para la captura de la imagen debe estar iluminada (luz día), los caracteres de la placa no pueden estar despintados (mas del 10%), ni entrecortados y la placa puede estar girada con un ángulo de inclinación máximo de 10° .

1.4 ASPECTOS TÉCNICOS

1.4.1 Placas

Todas las placas colombianas tienen forma rectangular como se muestra en la figura 1.2, sus dimensiones son: 16 cm de ancho por 33 cm de largo. El ancho para cada caracter es de 4cm y el grosor del trazo es de 1cm.

Tiene 6 caracteres escritos de color negro dentro de un marco negro, y están divididos en 3 letras y 3 números.



Figura 1.2. Placa estándar para Colombia

Los colores de las placas colombianas son amarillos, blancos y azules. El color amarillo está destinado a los vehículos particulares, los de placa blanca para vehículos de servicio público y los de placa azul para vehículos oficiales.

1.4.2 Equipos

La toma de imágenes se realizó con una cámara digital StyleCam Dv100, las imágenes tomadas presentan una resolución de 1024x768 píxeles y el formato es jpg. Para las pruebas se contaron con 90 imágenes. Los algoritmos fueron implementados en matlab 7.0 (ver lista de funciones, Apéndice A), porque es un lenguaje para el análisis de sistemas. Este lenguaje suministra un medio de programación de alto nivel, especialmente con las operaciones de matrices, lo cual es una ventaja para las tareas de procesamiento de imágenes. Se utilizó Matlab's Image Processing Toolbox v 4.2. Las pruebas fueron realizadas en un computador Intel® Pentium® 4 CPU 1.6 GHz y 256 RAM.

2 PROCESAMIENTO DE LA IMAGEN

1.1 INTRODUCCIÓN

El principal objetivo de las técnicas de procesamiento de imágenes es el de mejorar la imagen original, para que resulte más adecuada en las etapas siguientes de una aplicación específica. En este caso, las técnicas de procesamiento de imágenes presentadas en este capítulo, están enfocadas al problema del reconocimiento de placas de vehículos.

El capítulo describe los lineamientos básicos de histogramas y filtros, pero amplía la explicación en la detección de bordes, los fundamentos de color y la morfología matemática, debido a que son técnicas de mayor aplicación en el problema a resolver.

2.1 REPRESENTACIÓN DE UNA IMAGEN DIGITAL

Sea I una imagen digital y $f(x, y, c)$ la función que representa los puntos discretos que la conforman, donde x e y son las coordenadas espaciales y c un grupo de tres valores básicos de un modelo de color

Los valores de f dependen de la paleta de color utilizada por el sistema. Se manejan dos tipos de rangos: el de los valores normalizados en el rango $0 \leq f \leq 1$, y el de los no normalizados, que utiliza un *byte* para representarlos en el rango $0 \leq f \leq 255$. Cada valor de f , con la combinación de sus tres colores básicos o primarios, es conocido como un *píxel* de color.

2.2 HISTOGRAMAS

Básicamente el histograma es la representación gráfica de las frecuencias relativas con las que aparecen los distintos niveles de grises en una imagen determinada. En general se representa como un gráfico de barras, figura 2.1, en el que las abscisas son los distintos niveles de grises de la imagen y las ordenadas la cantidad de veces con la que cada color aparece en la imagen. El histograma proporciona información sobre el brillo

y el contraste de la imagen, y puede ser utilizado para ajustar estos parámetros, eliminar ciertas tonalidades molestas, etc.

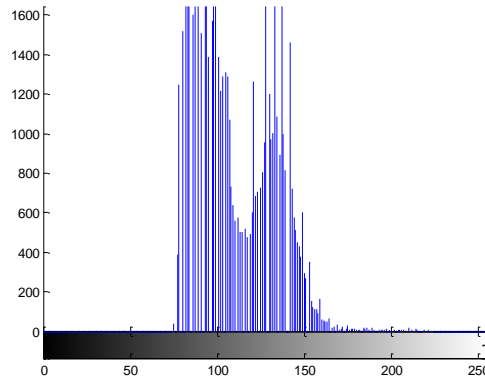


Figura 2.1 Histograma de una imagen de $m \times n$ con 256 niveles de gris

2.3 FILTRO ESPACIAL

El empleo de mascararas espaciales para el procesamiento de imágenes se denomina filtrado espacial. Entre los filtros espaciales lineales se encuentran los filtros pasa bajo, pasa alto, y pasa banda. Los denominados filtros pasa bajo atenúan o eliminan las componentes de alta frecuencia a la vez que dejan pasar las frecuencias bajas. Las componentes de alta frecuencia caracterizan los bordes y los restantes detalles muy marcados de la imagen, por ello el efecto de introducir un filtro de paso bajo es el de hacer mas borrosa la imagen. De forma similar, los filtros de paso alto atenúan o eliminan las componentes de baja frecuencia. Como estas componentes son las responsables de las características lentamente variables de la imagen, como el contraste global y la intensidad media, el resultado neto de un filtrado de paso alto es la reducción de estas características y, en correspondencia, una aparente agudización de los bordes y de los restantes detalles finos. Un tercer tipo de filtrado, denominado *filtrado de paso banda*, elimina unas regiones elegidas de frecuencia intermedias. Estos filtros se emplean para la restauración de imágenes.

La aproximación básica tomando la implementación de un filtro espacial consiste en sumar los productos entre los coeficientes de una máscara y las intensidades de los píxeles bajo la máscara en un punto determinado de la imagen. La figura 2.2. muestra una mascara de 3x3 general. Denominando a los niveles de gris de los píxeles bajo la

máscara de 3x3 en un punto determinado z_1, z_2, \dots, z_9 , la respuesta de una máscara lineal es:

w1	w2	w3
w4	w5	w6
w7	w8	w9

Figura 2.2. Máscara de 3x3

Si el centro de la máscara se encuentra en un punto (x, y) de la imagen, el nivel de gris del píxel situado en (x, y) se reemplaza por R . Luego se mueve la máscara hasta reemplazar el nuevo píxel de la imagen y se repite el proceso. Se continúa así hasta que se han cubierto todos los píxeles de la imagen. El valor de R se calcula empleando entornos parciales, para los píxeles que están colocados en los bordes de la imagen. También es habitual crear una nueva imagen para almacenar los valores de R , en lugar de ir cambiando los valores de cada píxel, ya que así se evita el empleo de niveles de gris que hayan sido alterados como resultado de una aplicación de la siguiente ecuación.

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

Estos filtros son utilizados para la segmentación basada en fronteras, que se utilizan en el acondicionamiento de la imagen en algunos métodos de detección de la placa.

2.4 SEGMENTACION BASADA EN FRONTERAS

Los bordes corresponden a cambios locales significativos en una imagen. Estos se encuentran en la frontera entre dos regiones diferentes (zonas de píxeles en las que existe un cambio brusco en la intensidad de la imagen, caracterizando las fronteras de los objetos). Tendrá gran importancia para la localización de la placa, al aplicar métodos de detección de líneas que conforman los bordes de la placa, como también para la etapa de segmentación y reconocimiento de los caracteres.

Hay variedad de algoritmos para la detección de bordes. Ahora se mostraran los tipos de detección de bordes que se utilizaron.

2.4.1 Operador de Roberts

Es un operador no lineal simple de calcular, ya que sólo usa una ventana (también llamada filtro, plantilla, máscara, kernel) de tamaño 2x2 [2].

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

La magnitud del gradiente se calcula mediante la siguiente aproximación, puesto que G_x y G_y son aproximaciones de las derivadas parciales con respecto a x e y respectivamente.

$$G(f(i, j)) = |f(i, j) - f(i+1, j+1)| + |f(i, j+1) - f(i+1, j)|$$

donde $f(x, y)$ es la imagen de entrada y $g(x, y)$ es la imagen procesada.

2.4.2 Operador de Sobel

Este operador toma en cuenta todos los píxeles vecinos al píxel de interés. Este operador funciona aplicando una ventana de tamaño 3x3 [2] sobre todos los píxeles de la imagen.

A ₀	A ₁	A ₂
A ₇	(i,j)	A ₃
A ₆	A ₅	A ₄

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figura. 2.3. Píxeles del entorno de un punto y dos ejemplos de máscaras

El operador de Sobel es la magnitud del gradiente calculado mediante:

$$M = \sqrt{s_x^2 + s_y^2}$$

donde las derivadas parciales se calculan mediante:

$$S_x = (A_0 + cA_7 + A_6) - (A_4 + cA_3 + A_2)$$

$$S_y = (A_0 + cA_1 + A_2) - (A_4 + cA_5 + A_6)$$

Con la constante $c=2$.

Estos operadores de gradiente pueden implementarse utilizando máscaras de convolución, tal como se vio en la sección 2.3.

Es importante notar que este operador le da más peso a los píxeles más cercanos al centro de la máscara.

2.4.3 Operador de Prewitt

Es muy similar al operador de Sobel, la diferencia es que cambia el valor de c a 1.

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Figura 2.4. Máscaras del operador prewitt

2.4.4 Operador laplaciano

Es el equivalente bidimensional de la segunda derivada [2], la fórmula para una función $f(x,y)$ es:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

La segunda derivada en la direcciones x e y se aproxima utilizando las ecuaciones en diferencias

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial G_x}{\partial x} \\ &= \frac{\partial(f(i+1, j) - f(i, j))}{\partial x} \\ &= f(i+2, j) - 2f(i+1, j) + f(i, j)\end{aligned}$$

Análogamente se tendría:

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+2) - 2f(i, j+1) + f(i, j)$$

Combinando las dos ecuaciones en un único operador, se puede usar la siguiente máscara para aproximar el laplaciano.

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

2.4.5 Detector de bordes de Canny

El algoritmo de Canny está considerado como uno de los mejores métodos de detección de contornos, mediante el empleo de máscaras de convolución, optimizando los tres criterios de detección, localización y respuesta única ante un borde.

El algoritmo consiste básicamente en cinco fases:

1. Suavizado de la imagen mediante una gaussiana para reducir los efectos del ruido.
2. Convolución de la imagen por operadores de gradiente direccionales.
3. Supresión de los gradientes que no son máximos en la dirección del contorno.
4. Aplicación de un umbral adaptativo con histéresis para distinguir los contornos reales del ruido.

5. Integración de los contornos detectados a diferentes escalas en un solo mapa de contornos consistentes.

Este operador es la combinación de un filtro gaussiano con una aproximación del gradiente [2], la cual es sensible al borde en la dirección del máximo cambio. El detector de bordes de Canny es la primera derivada de una gaussiana.

2.5 COLOR

Como las placas de los vehículos en Colombia siempre son de colores únicos, (ver sección 1.4.1), entonces, una forma de detectar la placa es buscar donde se encuentra el color de la placa, y luego buscar las coordenadas, aunque se puede obtener más de una región seleccionada como color de la placa, se puede ubicar la región correcta, buscando las características conocidas de una placa. Existen varios modelos de representación del color. En esta aplicación, se consideraron cuatro modelos de color, lográndose diferentes desempeños con cada uno de ellos.

2.5.1 Fundamentos del color

El espectro de color visible se puede dividir en seis regiones: violeta, azul, verde, amarillo, naranja y rojo. Sin embargo, cuando se ven todos los colores juntos, ningún color termina bruscamente, sino que cada color se mezcla suavemente con el siguiente.

Los colores que los seres humanos percibimos en un objeto, están determinados por la naturaleza de la luz reflejada del objeto. Un cuerpo que refleja luz relativamente equilibrada en todas las longitudes de onda, aparece como blanco para el observador, y un objeto que refleja en un rango limitado del espectro visible aparece como coloreado.

La caracterización de la luz es fundamental en la ciencia del color. Si la luz es acromática (sin color), su único atributo es la intensidad o cantidad de luz. Así el término nivel de gris se refiere a una medida escalar de la intensidad que va del blanco al negro pasando por los grises.

Para describir las características de una fuente cromática de luz se emplean tres magnitudes básicas: la radiancia, que es la cantidad de energía que sale de la fuente luminosa; la luminancia, que es la medida de la cantidad de energía que un observador percibe procedente de una fuente luminosa y el brillo, que incluye la noción acromática de la intensidad.

Los colores se ven como una combinación de los tres, determinados, colores básicos Rojo (R), verde (G), y azul (B) con longitudes de onda (designadas por la CIE) de 700 nm, 546.1 nm y 453.8 nm respectivamente.

Las características que generalmente se usan para distinguir un color de otro son el brillo, el color (matiz o tono) y la saturación. El tono es un atributo asociado con la longitud de onda dominante en la mezcla de las ondas luminosas (describe la pureza de un color) y la saturación se refiere a la cantidad de luz blanca mezclada con el tono.

El propósito de un modelo de color es facilitar la especificación de los colores en un formato estándar, en esencia un modelo de color es una especificación de un sistema en coordenadas tridimensionales y de un subespacio de este sistema donde cada color se representa por un punto único [3, 4].

2.5.2 Modelo de color RGB

Una imagen RGB es un arreglo de píxeles de color $M \times N \times 3$, donde cada píxel de color es un componente de la tripleta correspondiente a rojo, verde y azul en una localización espacial específica en una imagen RGB, figura 2.5 [4].

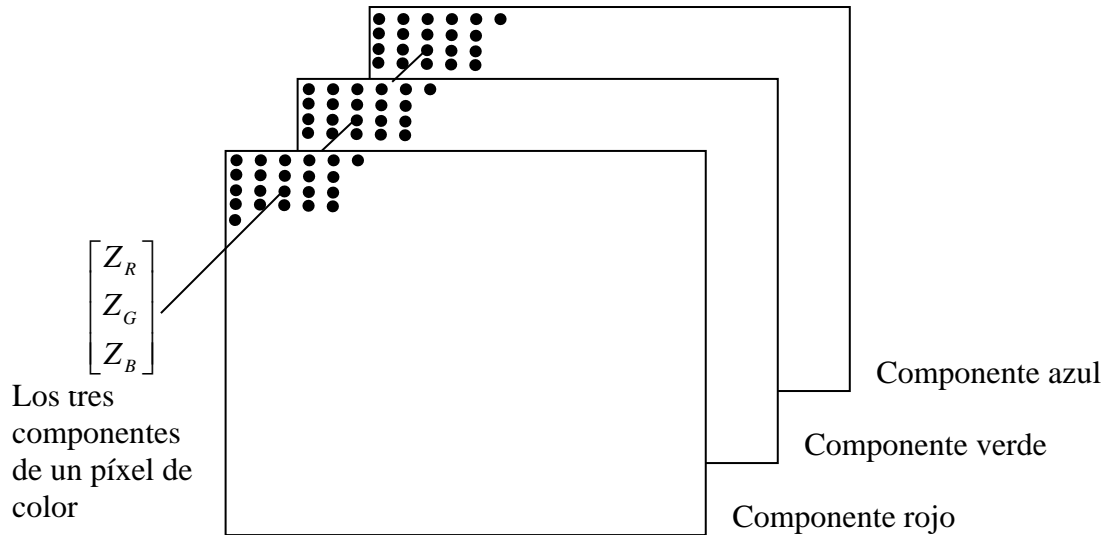


Figura 2.5. Esquema de una imagen a color, con 3 planos R, G y B

El espacio de color RGB usualmente es mostrado gráficamente como un cubo de color, ver figura 2.6. Los vértices del cubo son los colores primarios (rojo, verde y azul) y los secundarios (cyan, magenta y amarillo).

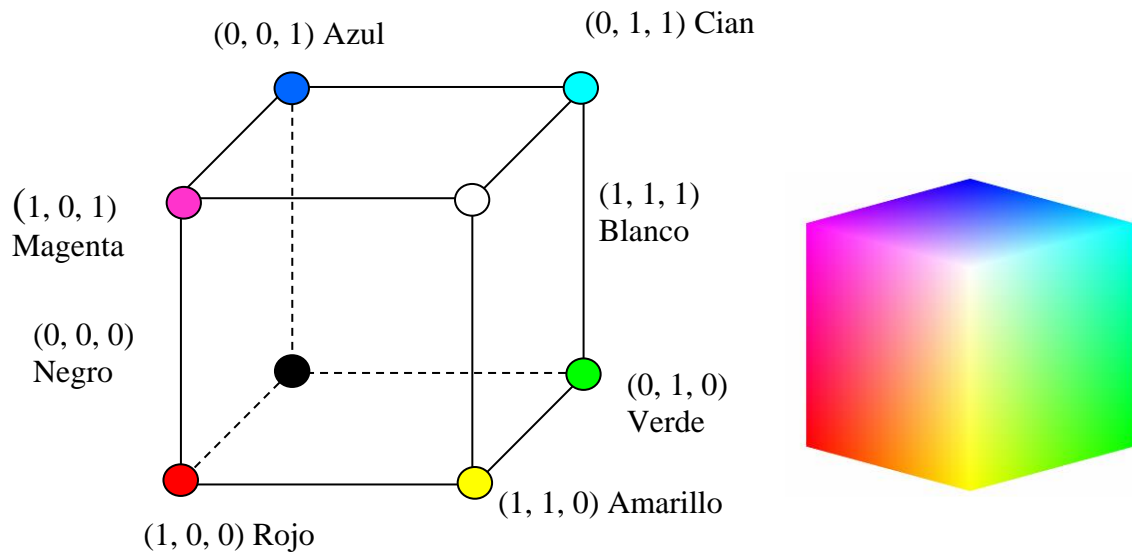


Figura 2.6. Cubo RGB

2.5.3 Modelo de color YIQ

Una de las principales ventajas de este formato es que la información en escala de grises es separada de los datos de color. Los datos de la imagen consisten en tres componentes:

Luminancia (Y), tono (I) y saturación (Q). El componente luminancia representa la información en escala de grises, y los otros dos componentes representan la información del color. Los componentes YIQ son obtenidos desde los componentes de una imagen RGB usando la transformación:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Note que los elementos de la primera fila suman 1 y los de las siguientes 2 filas suman 0, esto es porque para una imagen en escala de grises todos los componentes RGB son iguales y los componentes I y Q son cero.

2.5.4 Modelo de color HSI

En el modelo HSI (*Hue, Saturation, Intensity*), la componente de la intensidad (I), esta separada de la información cromática contenida en la imagen y las otras dos componentes están muy relacionadas con la forma en que el ser humano percibe el color. Estas características hacen que el modelo HSI sea una herramienta ideal para desarrollar algoritmos de procesamiento de imágenes basados en algunas propiedades de la percepción del color del sistema visual humano.

Las componentes de color tono y saturación del modelo HSI, están definidas con respecto al triángulo mostrado en la figura 2.7, se puede observar que el tono, H, es el ángulo del vector con respecto al eje rojo. Así cuando $H = 0^\circ$, el color es rojo, cuando $H = 60^\circ$, el color es amarillo y así sucesivamente. La saturación, S, es el grado en que el color no esta diluido con blanco y es proporcional a la distancia desde el centro hasta el triángulo [3,5], la intensidad se mide con respecto a una línea perpendicular al plano del triangulo y que pasa por su centro.

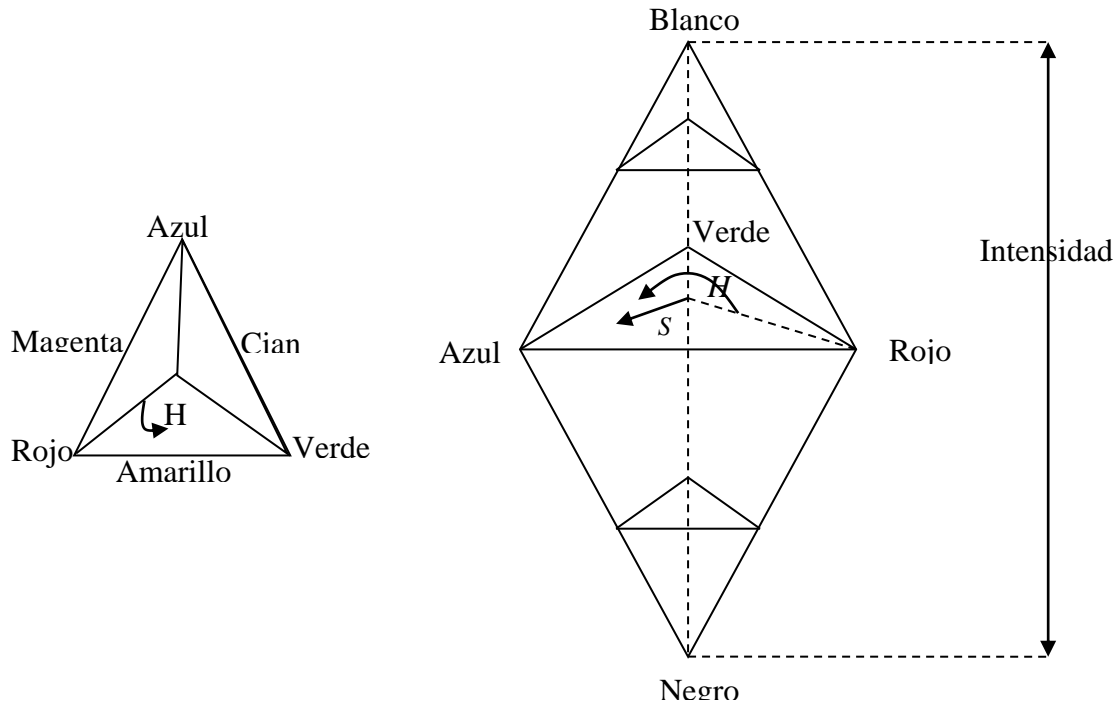


Figura 2.7. Triangulo de color HSI

La conversión de RGB a HSI es mas complicada que en los anteriores modelos, por lo tanto se pasaparece en detalle en [3]).

Dada una imagen RGB, el componente H de cada píxel RGB es obtenido usando la ecuación.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

donde

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\left[(R-G)^2 + (R-B)(G-B) \right]^{1/2}} \right\}$$

El componente de saturación esta dado por

$$s = 1 - \frac{3}{(R+G+B)} [\min(R,G,B)]$$

Finalmente, el componente de la intensidad esta dado por

$$I = \frac{1}{3}(R+G+B)$$

Esto es asumiendo que los valores RGB han sido normalizados al rango [0 ,1], y que el ángulo θ es medido con respecto a el eje rojo del espacio RGB, como se observa en la figura 2.7. El tono, H, puede ser normalizado al rango [0, 1] dividiendo los valores resultantes por 360 en la ecuación de H.

2.5.5 Modelo de color CIE XYZ

En el Sistema CIE (Commission Internationale de l'Eclairage), un color se define por las componentes de X, Y y Z, figura 2.8. La proyección en el eje X Y produce el diagrama de cromaticidad de la CIE, figura 2.9.

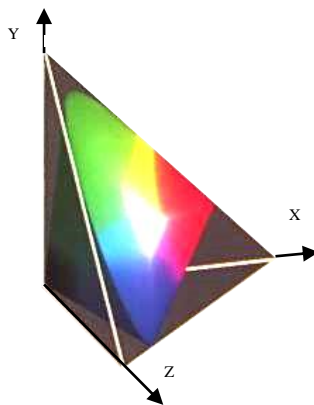


Figura 2.8. Componentes de CIE XYZ

Estas componentes no son reales sino imaginarias, pero cualquier color se puede definir como una combinación de estas. Asimismo, toda la información de intensidad esta comprendida únicamente en la componente Y.

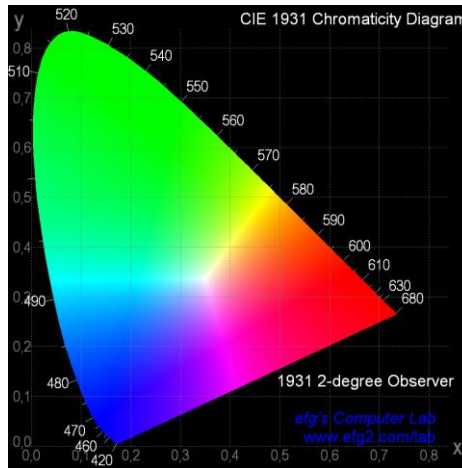


Figura 2.9. Diagrama CIE

El diagrama de cromaticidad esta basado en los valores triestimulos normalizados x , y y z donde, $x = X/(X+Y+Z)$, $y = Y/(X+Y+Z)$, y $z = Z/(X+Y+Z)$. Esta normalización remueve el brillo del diagrama, así que solamente dos coordenadas x y y son necesarias para describir la cromaticidad [6].

Los componentes se pueden extraer a partir de una imagen RGB como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 0.116 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.6 TÉCNICAS BASADAS EN MORFOLOGÍA

La morfología matemática fue desarrollada por un grupo de investigadores de la Escuela de Minas de París, dirigido por el profesor Jean Serra [7]. La morfología se basa en operaciones no lineales muy simples: se hallan máximos y mínimos en determinadas regiones de la imagen (aunque las operaciones morfológicas también pueden expresarse en términos de teoría de conjuntos). La morfología ofrece soluciones muy sencillas y rápidas a algunos problemas como se verá a continuación.

Las dos operaciones morfológicas básicas son la erosión y la dilatación. Todas las demás pueden expresarse en función de estas dos. Sin embargo, la erosión y dilatación

por sí solas tienen muy poca utilidad práctica. No así, las operaciones compuestas de estas.

Se van a considerar sólo imágenes en blanco y negro. Aunque es posible aplicar estas ideas a imágenes monocromas (y aún en color), el estudio de los conceptos básicos es conveniente en imágenes en blanco y negro.

Se llamará “Elemento Estructurante” a una figura geométrica conexa en la que se ha definido un origen o punto principal (habitualmente, en el centro). Dicha figura estará discretizada en píxeles igual que la imagen. Habitualmente, su tamaño será mucho menor que dicha imagen.



Figura 2.10. Ejemplo de elementos estructurantes.

Para realizar la erosión de una imagen, se colocará el elemento estructurante con el origen posicionado en cada uno de sus píxeles. Para cada una de estas posiciones (x,y) será posible calcular el mínimo de la intersección entre la imagen inicial y el elemento estructurante. Ese mínimo será el píxel de salida para la posición (x,y) . El efecto de la erosión será hacer decrecer a los objetos blancos (y crecer a los negros).



Figura 2.11. Ejemplo de erosión, (a) imagen inicial, (b) resultado de la erosión de (a).

En el ejemplo que se acaba de ver el elemento estructurante tiene forma cuadrada y su punto principal está en el centro.

Análogamente, se define la dilatación. En este caso lo que se calcula es un máximo y el efecto es contrario (los objetos blancos crecen y los negros decrecen):

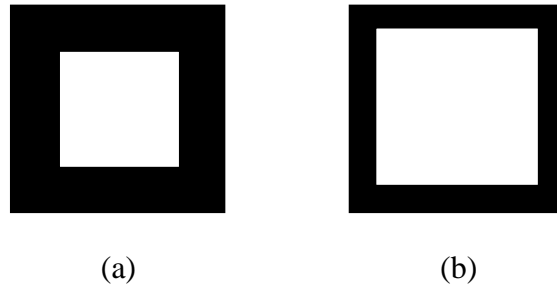


Figura 2.12. Ejemplo de dilatación, (a) imagen inicial), (b) resultado de aplicar dilatación a (a).

En este ejemplo se ha usado el mismo elemento estructurante.

Dos de las operaciones compuestas más habituales son el cierre o “*closing*” (se hace primero una dilatación y después una erosión) y la apertura u “*opening*” (primero erosión y después dilatación). El *closing* elimina los objetos pequeños negros (ruido) situados en zonas blancas (“cierra” los objetos blancos grandes). El efecto del *opening* elimina los objetos pequeños blancos situados en zonas negras (“abre” los objetos blancos grandes). Nótese, también, que el elemento estructurante es el que define qué es grande y qué es pequeño: los objetos pequeños son aquéllos dentro de los que el elemento no cabe entero.

Una forma fácil de entender estos métodos es mediante el siguiente ejemplo:



Figura 2.13. Imagen inicial.

Se parte de la imagen de un objeto (un pez) que pasa por una cinta transportadora de un sistema industrial. Se quiere hallar el perfil (silueta) del pez y, a partir de él, su tamaño. Lo primero es calcular un gradiente de la imagen. Se usa el operador laplaciano (basado en una máscara 3x3 que multiplica por 8 el píxel central y los demás por -1). El

gradiente da lugar a una imagen muy poco contrastada que se reduce a dos niveles usando el umbral de Otsu (sección 2.7.1). El resultado se muestra en la figura 2.14.

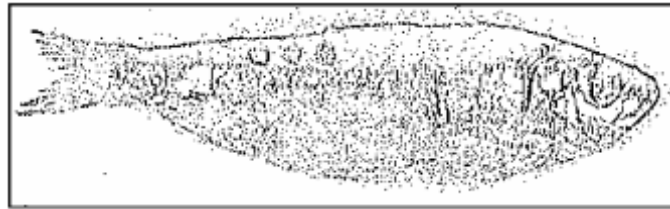


Figura 2.14. Gradiente laplaciano del pez inicial.

En la figura 2.14 se observa la silueta del pez perfilada por una nube de muchas manchas de pocos píxeles. El objetivo es juntarlas todas, esto es: cerrar un objeto negro grande. Se conoce que para eso hay que hacer un *opening*. Se hace con un elemento estructurante cuadrado de lado 11 píxeles, el punto principal estará situado en el centro. El resultado se ve en la siguiente imagen de la figura 2.15 [8].



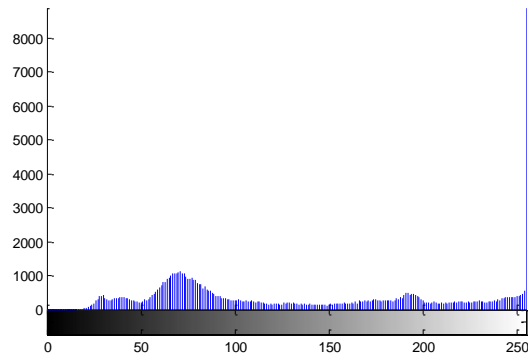
Figura 2.15. *Opening* de la imagen anterior (los círculos han sido dibujados manualmente para identificar los puntos).

2.7 UMBRALIZACIÓN

Se considera el histograma de niveles de gris mostrado en la figura 2.16 correspondiente a una imagen compuesta de objetos oscuros sobre un fondo luminoso, de forma que los píxeles en los objetos y el fondo tienen niveles de grises agrupados en dos modas dominantes.



(a)



(b)

Figura 2.16. (a) Imagen en escala de grises, (b) histograma de una imagen en niveles de gris

Una forma obvia de extraer los objetos del fondo es seleccionar un umbral T que separe estas modas. Es decir, obtener:

$$g(i, j) = \begin{cases} 1 & \text{si } f(i, j) \geq T \\ 0 & \text{si } f(i, j) \leq T \end{cases}$$

Esta ecuación puede tener muchas modificaciones. Una posible es segmentar la imagen en regiones de píxeles con nivel de gris en un conjunto D y el resto como fondo. También se puede utilizar umbrales múltiples, lo que produce imágenes no binarias con conjunto muy limitado de niveles de gris, el proceso sería:

$$g(i, j) = \begin{cases} 1 & \text{si } f(i, j) \in D_1 \\ 2 & \text{si } f(i, j) \in D_2 \\ \dots & \\ n & \text{si } f(i, j) \in D_n \\ 0 & \text{en el resto} \end{cases}$$

Es claro que para hacer la segmentación más robusta, el umbral debería ser seleccionado automáticamente por el sistema. Tanto el conocimiento sobre la escena como el problema a resolver y cualquier otro conocimiento debería ser usado para fijar el umbral. La información de la que se puede disponer puede venir proporcionada por las características de los objetos, el tamaño de los mismos, la fracción de la imagen

ocupada por los objetos y los diferentes tipos de objetos que pueden parecer en la imagen [4].

2.7.1 Umbralización por el método de Otsu

Se basa en el histograma de la imagen en escala de grises. Después de calcular el histograma se puede inferir algunos parámetros estadísticos. Primero se calcula la media y la varianza globales de los niveles de gris.

$$\mu = \sum lp(l) \quad \sigma^2 = \sum (l - \mu)^2 p(l)$$

Donde l son los niveles de intensidad (valores de 0 a 255) y $p(l)$ es la densidad de probabilidad discreta:

$$p(l) = \frac{n_q}{n}$$

Donde n es el numero de píxeles de la imagen y n_q es el número de pixeles que tienen un nivel de intensidad l .

Se supone un umbral U . Se calcula la media y varianza “por debajo” y “por encima” de U (lo que supone aplicar el concepto de probabilidad condicional a los casos $l < U$ y $l > U$).

$$w_1 = \sum_0^{U-1} p(l) \quad w_2 = \sum_{U+1}^{255} p(l)$$

$$\mu_1 = \sum_0^{U-1} \frac{lp(l)}{w_1} \quad \mu_2 = \sum_{U+1}^{255} \frac{lp(l)}{w_2}$$

$$\sigma_1^2 = \sum_0^{U-1} \frac{(l - \mu_1)^2 p(l)}{w_1} \quad \sigma_2^2 = \sum_{U+1}^{255} \frac{(l - \mu_2)^2 p(l)}{w_2}$$

Se define la “varianza conjunta después de aplicar el umbral U ”: $\psi^2 = w_1\sigma_1^2 + w_2\sigma_2^2$.

La diferencia $D = \sigma^2 - \psi^2$ es una medida de la calidad del umbral U . En un caso ideal, donde la imagen tuviera sólo dos niveles de gris se cumpliría $D = 0$ cuando U no está entre los niveles y $D = \sigma^2 > 0$ si está entre ellos.

El algoritmo consiste simplemente en calcular D para todos los posibles umbrales y usar el que produce el máximo. Se puede ver que las fórmulas son adecuadas para un cálculo iterativo y rápido (se puede calcular las sumas para un valor de umbral a partir de las anteriores: sumando o restando un valor).

Este método es el más robusto ante cualquier tipo de histograma y de hecho es el más referenciado y utilizado de la literatura.

2.7.2 Umbralización adaptativa

La iluminación desigual puede ser debida a sombras o la propia dirección de la iluminación. En todos estos casos, el mismo umbral no puede ser usado en toda la imagen. Los métodos adaptativos examinan regiones pequeñas de la imagen y entonces analizan la subimagen para obtener un umbral de sólo esa región. Una aproximación muy simple para segmentar este tipo de imágenes es particionar la imagen en $m \times n$ subimágenes y seleccionar un umbral para cada subimagen basado en su subimagen. La segmentación final de la imagen es la unión de las regiones de sus subimágenes.

3 LOCALIZACIÓN DE LA PLACA

3.1 INTRODUCCIÓN

Las técnicas de localización se refieren a los algoritmos utilizados para detectar la zona en la cual se encuentra la placa, si se parte de una imagen como se ilustra en la figura 3.1. Se presentan cuatro diferentes métodos en esta etapa: el primero se basa en la transformada de Hough, el segundo en el cálculo de los gradientes de la imagen (método del gradiente), el tercero se basa en morfología matemática, (método morfológico) y el cuarto se basa en color.



Figura 3.1 imagen Inicial

3.2 TRANSFORMADA DE HOUGH

Un método utilizado para localizar la placa es la transformada de Hough. La idea principal es encontrar las líneas que conforman el rectángulo de la placa. Partiendo de la imagen obtenida, se le hace primero un acondicionamiento para aplicarle la transformada de Hough, esta transformada entrega todas las líneas que hay en la imagen y luego se debe hacer una búsqueda para encontrar las líneas que verdaderamente forman la placa.

3.2.1 Acondicionamiento de la imagen

Para encontrar las líneas aplicando la transformada de Hough, se parte de una imagen binaria conteniendo los bordes de la imagen RGB. Como la transformada de Hough es lenta computacionalmente, entonces, se hace una reducción a la imagen del 50% de 1024 x 1280 x 3 a 512 x 640 x 3, luego se convierte la imagen de RGB a escala de grises, figura 3.2 (a), y se hace un aumento del contraste, figura 3.2 (b), para aplicar los métodos de detección de bordes (canny), figura 3.2 (c), [capítulo 2].

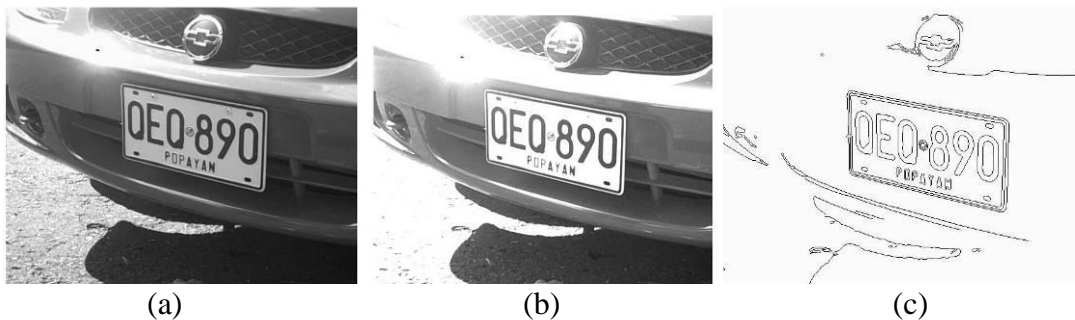


Figura 3.2. (a) Escala de grises, (b) ajuste de la intensidad y (c) detección de bordes.

3.2.2 Detección de líneas con la transformada de Hough

Una vez se tengan los bordes de la imagen, y aprovechando la geometría de la placa, se detectan las líneas, y luego se busca cuales son las líneas que conforman el rectángulo de la placa, para esto se utiliza la transformada de Hough. Aunque esta transformada puede detectar cualquier tipo de curvas, aquí solo se utiliza para detectar líneas rectas.

La formulación inicial de esta transformada se realizó sobre el espacio (m, c) , también denominado espacio de parámetros, que corresponde a la pendiente y el desplazamiento de la recta.

$$y = mx + c$$

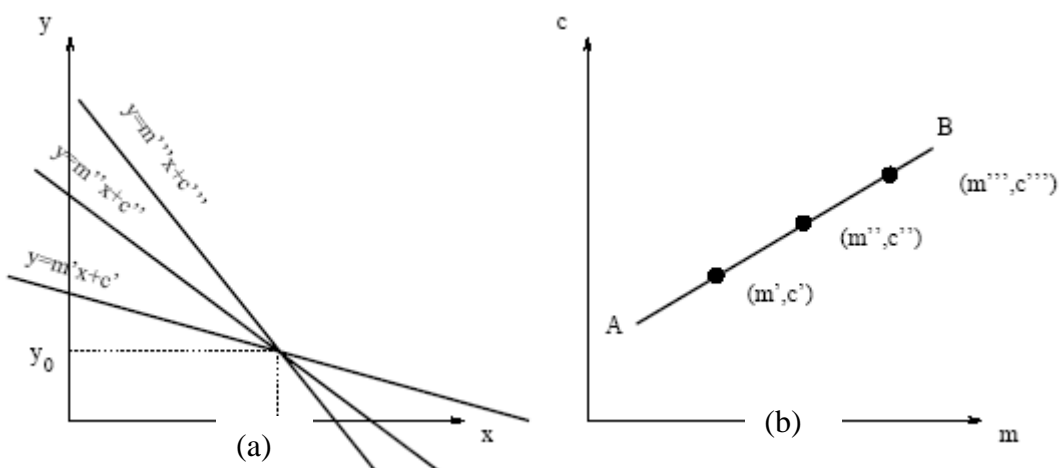
Entonces, la idea principal de la transformada de Hough es:

Recorrer todos los puntos de frontera (bordes), y para cada uno de ellos, acumular un voto en aquellas posiciones del espacio de parámetros que correspondan a una línea que pasa por dicho punto frontera [2, 9].

El objetivo será realizar una transformación de las evidencias del espacio de la imagen (espacio x - y) al espacio de rectas (espacio m - c). Un solo punto en el espacio m - c corresponde, a una recta en el espacio x - y , por otro lado un punto en el espacio x - y se puede definir como el punto donde cortan las infinitas líneas (para todos los valores de m) que pasan a través de él. Para un punto (x_o, y_o) las rectas que pasan a través de él son las rectas que cumplen $y_o = mx_o + c$, es decir, los puntos (m, c) que corresponden a la recta $c = -mx_o + y_o$ del espacio m - c .

Si se considera un conjunto de n puntos alineados P_i , sólo existe una recta $y = m_o x + c_o$ que pasa a través de todos ellos. Por lo tanto, al dibujar las n rectas del espacio m - c que corresponden a cada uno de los puntos se tiene que todas ellas tienen en común el punto (m_o, c_o) .

En la figura 3.3 (a) se puede ver que las líneas que pasan por un punto del espacio x - y corresponde a una línea en el espacio m - c , figura 3.3 (b). De igual forma, los puntos alineados del espacio x - y , figura 3.3 (c), corresponden a líneas que se cortan en un punto común, es decir, los puntos en una misma línea $y = m_o x + c_o$ corresponden a líneas en el espacio m - c sobre el punto (m_o, c_o) , figura 3.3 (d).



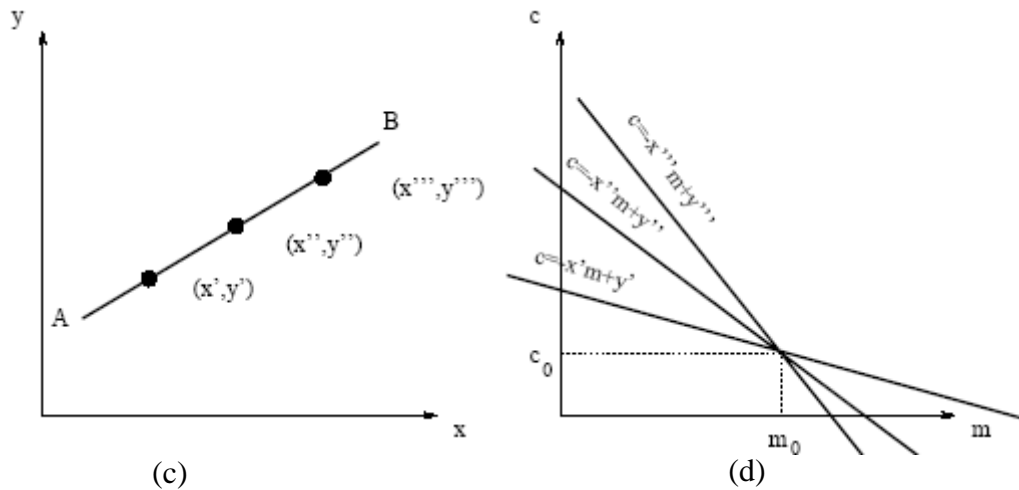


Figura 3.3. (a) Espacio de x-y a (b) espacio de parámetros m-c, (c) a (d), lo invertido

Por lo tanto un método para detectar líneas a partir de un conjunto de puntos P_i , es interpretar el espacio $m-c$ como un espacio de votación y para cada punto acumular un valor entero correspondiente al número de votos que indican la presencia de esa recta en la imagen. Para calcular el número de votos en cada posición, se hace que para cada punto de P_i se incremente en uno todos los puntos que corresponden a la recta del espacio $m-c$.

Como resultado de lo anterior, en cada punto (m, c) obtenemos un número de votos igual al número de puntos situados sobre la recta $y = mx + c$. Es decir, la búsqueda de un conjunto de puntos alineados en el espacio $x-y$ se convierte en la búsqueda de un punto con un alto valor de acumulación en el espacio $m-c$

El atractivo de la transformada de Hough surge de la subdivisión del espacio parámetro en las denominadas celdas acumuladoras, figura 3.4, donde (c_{\max}, c_{\min}) , (m_{\max}, m_{\min}) son los rangos esperados de los valores de la pendiente y la ordenada [3].

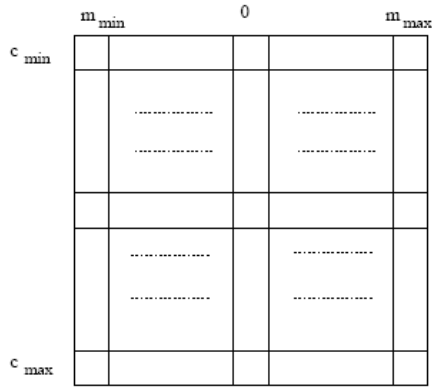


Figura 3.4. Cuantificación del espacio m-c

La celdas de coordenadas (i, j) , con valor de acumulador $A(i, j)$, corresponde al cuadrado asociado con el espacio de parámetro (c_o, m_o) .

Un problema que aparece al utilizar la ecuación $y = mx + c$ para representar una línea, es que tanto la pendiente como la ordenada en el origen tienden a infinito cuando la línea se acerca a la vertical, una forma de evitar esta dificultad es utilizar la representación normal de la recta, definida sobre los parámetros (ρ, θ)

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

En la figura 3.5 (a) se puede observar la interpretación de los parámetros de la recta y la subdivisión del espacio de parámetro, figura 3.5 (b), donde ρ es la distancia del origen a la recta y θ es el ángulo formado por la recta que une el origen con el punto más cercano y el eje de las abscisas.

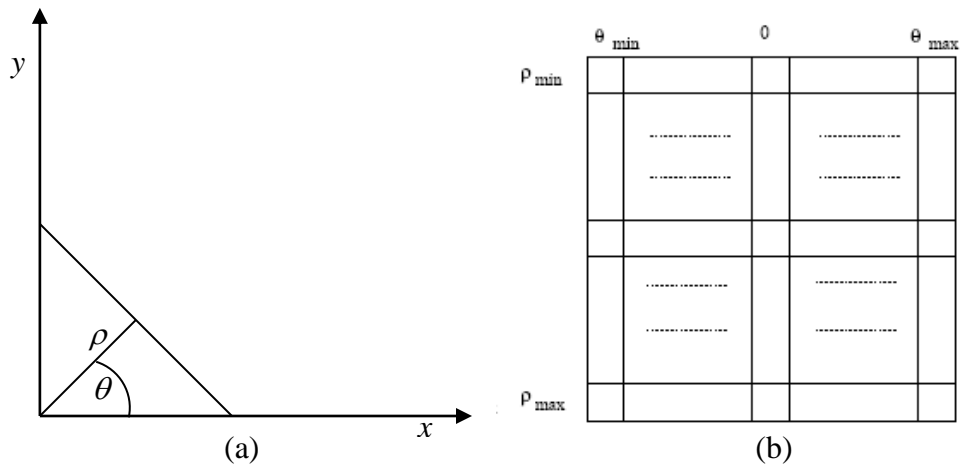


Figura 3.5. (a) Representación normal de la recta, (b) cuantificación del plano (ρ, θ) en celdas

El empleo de esta representación en la construcción de una tabla de acumuladores es idéntico al método expuesto para la representación punto – pendiente. Sin embargo, en vez de líneas rectas, los lugares geométricos son curvas sinusoidales en el plano (ρ, θ) , figura 3.6 (a)¹.

Una vez se haya aplicado la transformada de Hough, se verifica los puntos que tienen valores más altos (picos), estos contienen las líneas de mayor longitud, en la figura 3.6 (b) se puede observar que se obtiene una gran cantidad de líneas que cruzan toda la imagen, también se puede notar que hay muchas líneas juntas donde solo debería haber una que represente a la línea de borde de la imagen, esto se da porque las líneas que se obtuvieron con la detección de bordes no son completamente rectas o son de más de un píxel.

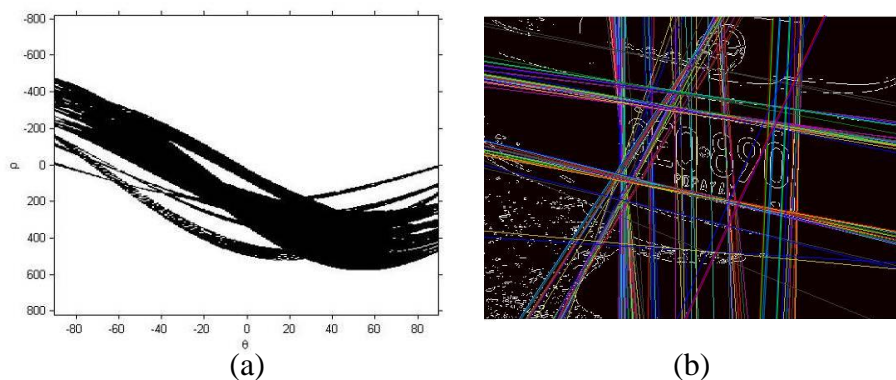


Figura 3.6. Transformada de Hough y las líneas correspondientes a los picos más altos

Con la obtención de los picos, figura 3.7 (a), se pasa a eliminar las celdas en la tabla de acumuladores que estén junto a los puntos más altos (para eliminar las líneas que están muy cerca de unas a otras) [4]. Con los picos que posiblemente conforman líneas se pasa a verificar si verdaderamente es una línea o es la acumulación de puntos aislados, también se debe encontrar dónde inicia y dónde termina cada línea, y finalmente se agrupan en segmentos de líneas los puntos encontrados, uniendo las discontinuidades en las líneas y suprimiendo los puntos que están aislados obteniendo únicamente las líneas más largas, figura 3.7 (b).

¹ en algunos casos los colores son invertidos para una mejor visualización.

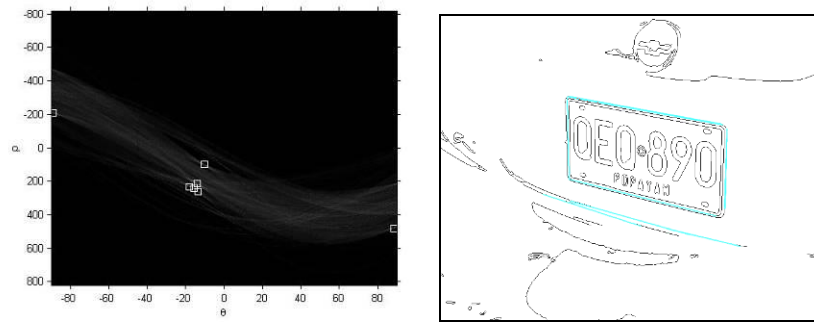
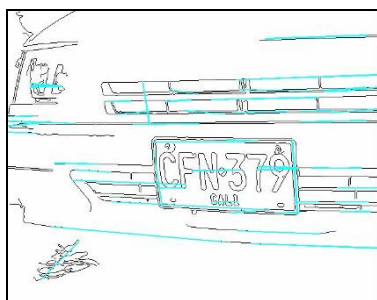


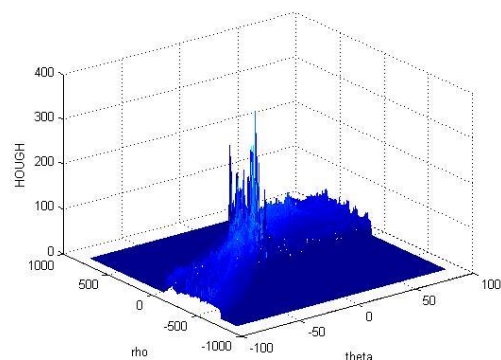
Figura 3.7. (a) Transformada de Hough con 7 picos encontrados (cuadros blancos), (b) líneas correspondientes.

En la figura 3.7 (b), se obtuvo casi únicamente las líneas que conforman el marco de la placa, por lo cual sería fácil obtener la región de la placa en este caso. En la mayoría de los casos se obtienen muchas líneas, ver figura 3.8 (a), por lo es complicado encontrar las líneas que conforman la placa, y si se disminuye el número de picos a encontrar es posible que se borren las líneas que se están buscando, porque se pueden encontrar líneas mas largas que las que conforman el marco de la placa. La figura 3.8 (b), muestra los picos obtenidos de la figura 3.8 (a), donde los valores más altos son los de las líneas horizontales que no forman parte de el marco de la placa.

Para hacer más rápida la transformada de Hough y para obtener menos líneas en las cuales buscar las que conforman la placa, se optó por aplicar un filtro para eliminar las líneas horizontales, y luego borrar las pequeñas partículas, obteniendo así únicamente las líneas verticales, figura 3.8 (c). Así, al aplicar la transformada únicamente se buscan líneas que están entre 80 a 100 grados, mejorando el tiempo de procesamiento y haciendo más fácil el encontrar las dos líneas verticales que forman el marco de la placa. La figura 3.8 (d), muestra los picos obtenidos de la figura 3.8 (c), donde los valores mas altos son los de las líneas verticales que conforman el marco de la placa.



(a)



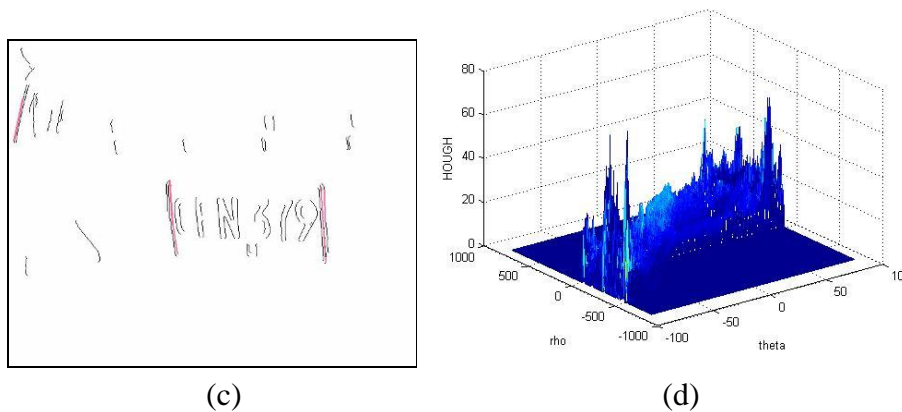


Figura 3.8. (a) líneas encontradas con la transformada de Hough de los bordes verticales y horizontales (b) picos obtenidos de la imagen (a), (c) Transformada con solo bordes verticales, (d) picos de (c).

Los resultados dependen en gran parte de qué tan bien se logren obtener los bordes. Cada detector de bordes trabaja de diferente manera, por lo tanto el resultado es diferente, de acuerdo a qué detector se utilice. En esta aplicación, se utiliza el detector de bordes de Canny, pero si no se logra encontrar la placa, se cambia de detector hasta que se pueda conseguir un buen resultado o hasta que haya fracasado en cuatro intentos. El orden en que se utilizan los detectores es el siguiente:

- Canny
- Sobel horizontal
- Sobel
- Prewitt

Para la búsqueda de las líneas de interés se hacen una serie de comparaciones:

- Se buscan los puntos de inicio y fin de cada línea hasta encontrar un par de líneas que aproximadamente² inicien y terminen en el mismo lugar.
- Se compara la longitud del par de líneas.
- Se compara si se tiene aproximadamente el mismo Angulo
- Se compara la relación ancho / alto

2. La placa no siempre esta paralela a los bordes de la imagen, normalmente se tiene un Angulo de giro y además cualquier ruido puede afectar los puntos de inicio y fin de cada línea.

- Si se obtiene más de un par de líneas que cumplan con los anteriores requerimientos, se hace mas precisa la relación ancho / alto hasta obtener un solo par de líneas
- En el caso de que no se encuentre ningún par de líneas que cumplan con las especificaciones, entonces se aplica un nuevo método de obtención de bordes³ y se realiza la búsqueda nuevamente hasta encontrar el par de líneas, si no se tiene éxito con cuatro (en el caso de esta aplicación) métodos de detección de bordes se termina la búsqueda con fracaso en la localización de la placa.
- Los puntos finales e iniciales del par de líneas son los cuatro puntos que conforman el ángulo de la placa.

3.3 MÉTODO DEL GRADIENTE

En este método se modela las zonas de caracteres presentes en la imagen como zonas de picos en la derivada de la imagen [8]. La idea es aplicar el gradiente Sobel Horizontal a toda la imagen (Figura 3.9).



Figura 3.9 Imagen con Sobel horizontal

3. Si no se tiene éxito con el primer intento, se tiene el problema de aumentar demasiado el tiempo de búsqueda, porque los algoritmos de detección de bordes y de la transformada de Hough son costosos computacionalmente

Una vez calculado el gradiente Sobel horizontal, la imagen que se obtiene presenta valores altos positivos seguidos de valores altos negativos (o viceversa). Son estos valores los que indican la ubicación de los caracteres, por ejemplo, un valor alto positivo puede indicar el comienzo del trazo de un carácter y un valor alto negativo el fin del carácter. También se pueden establecer el ancho del carácter y distancia entre ellos como se ilustra en la siguiente figura 3.10:

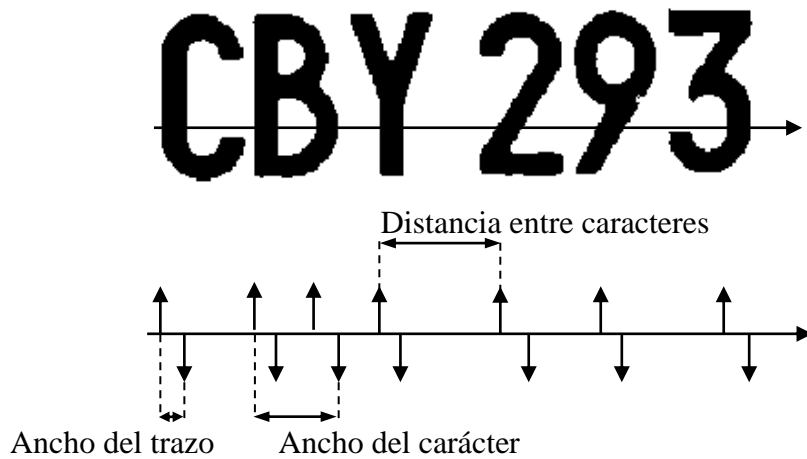


Figura 3.10. Detección de los caracteres

En la anterior figura 3.10 las líneas indican los valores altos positivos y negativos de la imagen, el método se basa en hacer una búsqueda en toda la imagen de un patrón que contenga las características ilustradas en la figura 3.10, es necesario que el patrón encontrado tenga varias líneas consecutivas, para así detectar un rectángulo candidato a contener una matrícula.

La figura 3.11 indica el resultado obtenido por este método.



Figura 3.11. Resultado obtenido

3.4 MÉTODO MORFOLÓGICO

El método morfológico [8], se basa en la morfología matemática (capítulo 2), en este caso se considera a los caracteres como objetos oscuros y de pequeño grosor.

Los pasos son los siguientes:

Se parte de la imagen mostrada en la figura 3.12.



Figura 3.12 Imagen original

Primer paso: Hacer un *closing* con un elemento estructurante circular y de diámetro mayor que el ancho del trazo, se consigue borrar las letras tal como se muestra en figura 3.13.



Figura 3.13. Imagen aplicado el *closing*

Segundo paso: Restar la imagen con *closing* con la original, los caracteres aparecerán resaltados, a esto se le conoce como *top-hat* de la imagen inicial.



Figura 3.14. Resta de la segunda imagen con la original

Ahora se realiza un preprocesado para eliminar zonas falsas.

Tercer paso: Se binariza la imagen y luego se hace un *closing* con un elemento estructurante lineal horizontal. Si la anchura es igual a la máxima separación entre caracteres, se consigue convertir la matricula en un rectángulo blanco (figura 3.15)

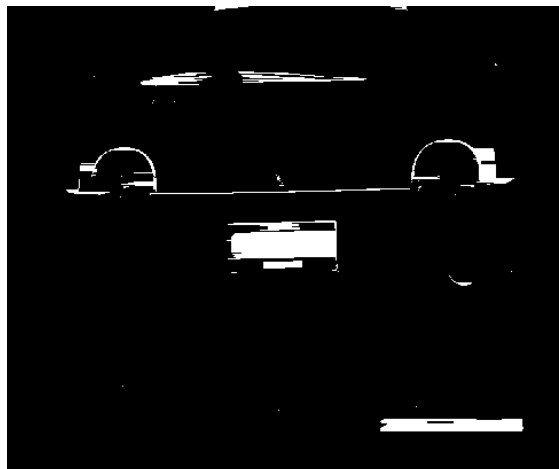


Figura 3.15. *Closing* con elemento estructurante horizontal

Cuarto paso: Hacer un *opening* con un elemento estructurante lineal vertical, se eliminan los objetos de altura inferior a la altura mínima de los caracteres (figura 3.16).

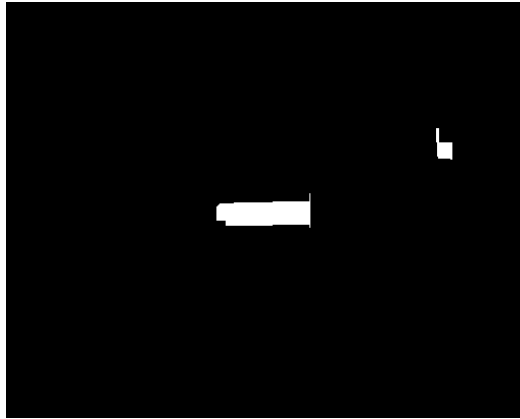


Figura 3.16. *Opening* con elemento estructurante horizontal

Quinto paso: Hacer otro opening que elimine los objetos de altura inferiores a la altura máxima de los caracteres, se elimina también la placa.

Sexto paso: Restar la imágenes del cuarto y quinto paso, se obtiene una imagen que contendrá sólo la placa y podrá presentar un poco de ruido residual.

El algoritmo implementado se probó con los pasos quinto y sexto pero se observó que el algoritmo funciona perfectamente hasta el paso 4. Por esta razón y debido a que los cálculos de opening y closing de una imagen representan tiempo computacional significativo el algoritmo se implementó hasta el paso cuarto.

El rectángulo blanco de la imagen obtenida en el paso cuarto (figura 3.16) brinda la ubicación de la placa y el cuadrado blanco en la esquina superior derecha se elimina como ruido residual.

En la figura 3.17 se ilustra el resultado obtenido, en la cual se emplea una dilatación pequeña del rectángulo de la imagen de la figura 3.16 para asegurar la totalidad de la placa.



Figura 3.17. Obtención de la matrícula

3.5 DETECCIÓN POR COLOR

Se parte de la imagen RGB, se hace una reducción del 50% (para reducir el tiempo de procesamiento), como en los anteriores métodos. Luego se hace una separación de los colores RGB, para hacer la conversión a los modelos descritos en la anterior sección 2.4. En esta aplicación se hicieron pruebas con los cuatro modelos, y se logro una mejor calibración con el modelo YIQ, por lo tanto las pruebas finales con el método de detección por color sólo se hicieron con este modelo.

Una vez se tenga separadas las componentes de luminancia (Y), tono (I) y saturación (Q) del modelo YIQ, se busca un umbral (el umbral fue encontrado haciendo pruebas con diferentes imágenes hasta obtener un buen resultado) que deje únicamente el color de la placa, ver figura 3.18 (a), (en esta aplicación se trabajó con el tono y la saturación). Hasta aquí se ha obtenido una imagen con unos en las regiones detectadas como color de placa y ceros en el resto de la imagen.

Con las regiones del color de la placa detectadas, se pasa a buscar las coordenadas de la placa. Para esto se le pasa a la imagen un filtro pasa bajos, para unir las regiones amarillas, figura 3.18 (b).

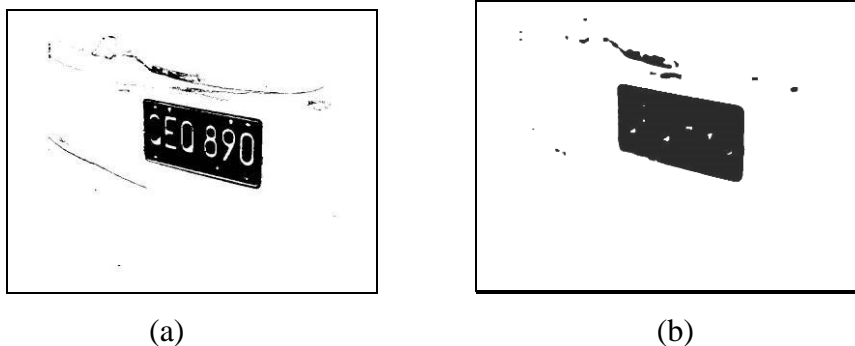


Figura 3.18. Detección por color, (a) detección de color amarillo, (b) imagen después de aplicar un filtrado.

La escogencia del tamaño de la máscara para el filtro es muy importante para lograr buenos resultados, si es muy grande puede unir regiones que no corresponde a la placa y si es pequeña puede que no una las regiones que forman la placa. Esto es porque la escogencia de la región se hace por análisis de componentes conectados (capítulo 4, sección 4.2). Por esta razón, primero se mide el porcentaje de las regiones que

corresponden a las encontradas como color de la placa con respecto a la imagen total, para aplicar la máscara adecuada, por defecto se utiliza una máscara de 10 x 10 y si el color detectado como color de placa es menor al 2 % (valor obtenido con pruebas a diferentes imágenes) entonces se aplica una máscara de 14 x 14.

El mayor problema de la detección por color es que si en la imagen de entrada hay colores muy parecidos o del mismo color de la placa, entonces se van a entregar muchas regiones en las cuales realizar la búsqueda de la placa, para solucionar algunos de estos casos se hace primero una verificación de la región encontrada así:

- Verificar si la región encontrada se aproxima a un rectángulo
- Ver si tiene variaciones de blanco a negro, las cuales corresponden a las letras
- Por ultimo, medir la relación ancho alto de la región encontrada
- Si no cumple con las especificaciones se borra la región y se pasa a la siguiente

En la figura 3.19, se observa un ejemplo donde se encuentran muchas regiones de color amarillo (mostradas en color negro) y la eliminación de regiones hasta encontrar la verdadera placa. En la búsqueda se inicia por la región más grande, sino es la placa, se elimina la región y se continúa con la siguiente más grande hasta lograr encontrar la placa.

No se obtienen resultados favorables si el color de la placa es del mismo color del carro, debido a que la placa y el vehículo quedan conectadas como una sola región.



Figura 3.19. Regiones detectadas como amarillo y eliminación hasta encontrar la verdadera placa.

Una vez se tenga la región escogida como la placa, se encuentran las fronteras de la región para localizar las coordenadas y hacer el recorte.

4 SEGMENTACIÓN

4.1 INTRODUCCIÓN

La segmentación de imágenes puede ser definida como la partición de una imagen en regiones. Una región (objeto), es un grupo de píxeles conectados que tiene propiedades similares. Es obvio que las regiones son importantes para la interpretación de las imágenes, pues pueden corresponder a objetos en la escena. Una imagen puede contener varios objetos y además cada objeto puede contener varias regiones, que corresponden a partes del mismo.

Para que una imagen pueda ser interpretada correctamente, tiene que ser dividida en regiones que correspondan a objetos o partes de ellos [2]. En este caso los objetos que se desean obtener de la imagen (placa ya localizada) son los seis caracteres separados, dando como resultado seis imágenes (figura 4.1).



Figura 4.1. Separación de caracteres

Antes de iniciar con el método de segmentación de los caracteres se hace una corrección del ángulo, ver figura 4.2



Figura 4.2. Corrección de ángulo, (a) imagen obtenida de la detección, (b) imagen girada 7 grados, (c) imagen obtenida aplicando una transformación espacial.

Para corregir el ángulo de inclinación se puede utilizar la transformada de Hough, se encuentran las líneas mas largas, que en este caso son las líneas horizontales del marco, y luego se obtiene el ángulo de inclinación de estas líneas.

Luego de tener el ángulo de inclinación, se aplica una transformación espacial [4], porque si se hace una rotación simple, se presenta el problema de que las letras quedan inclinadas, ver figura 4.2 (b).

La idea básica de la transformación geométrica es la siguiente:

Supóngase que una imagen f , definida sobre un sistema de coordenadas (w, z) , sufre una distorsión geométrica para producir una imagen g , definida sobre un sistema de coordenadas (x, y) , esta transformación de las coordenadas puede ser expresadas, como:

$$(x, y) = T\{(w, z)\}$$

Por ejemplo si $(x, y) = T\{(w, z)\} = (w/2, z/2)$, la distorsión es simplemente una reducción de la imagen a la mitad en ambas dimensiones espaciales.

Una de las formas de transformaciones espaciales mas comúnmente usadas es la transformada *affine*. La transformada *affine* puede ser escrita en forma de matriz de la forma

$$[x \ y \ 1] = [w \ z \ 1] T = [w \ z \ 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

Esta transformada puede escalar, rotar, trasladar, o *shear* un conjunto de puntos dependiendo de los valores escogidos en T.

Con esta transformación se corrige la imagen suponiendo los valores de corrección de acuerdo al ángulo encontrado, en la figura 4.2 (c) se puede observar el resultado obtenido aplicando esta transformación.

Luego de la corrección de la imagen se procede a umbralizar la imagen, ver figura 4.3



Figura 4.3 Umbralización con el método de Otsu

La umbralización obtenida juega un papel muy importante a la hora de la segmentación y el reconocimiento de los caracteres. En algunos casos se pueden borrar las letras o parte de estas, y al aplicar el borrado de ruido se eliminan como ruido porque son muy pequeñas; o al contrario se puede obtener regiones negras a parte de las letras, presentándose el problema de unir las letras entre si o al marco de la placa.

Con la imagen de la placa umbralizada se pasa a aplicar los métodos de segmentación de los caracteres.

4.2 ANALISIS DE COMPONENTES CONECTADOS

El etiquetado de componentes conectados, es un método que marca o enumera cada objeto (caracter o caracteres unidos) de una imagen binaria. La figura 4.4, muestra una imagen con tres objetos, donde, una vez aplicado el algoritmo de componentes conectados los píxeles de la A tendrán un valor de 1, la B de 2 y la C de 3.



Figura 4.4. Tres objetos etiquetados

Un píxel p de coordenadas (x, y) tiene dos vecinos horizontales y dos verticales, cuyas coordenadas son $(x+1, y)$, $(x-1, y)$, $(x, y+1)$ y $(x, y-1)$, este conjunto de 4 vecinos de p denotados por $N_4(p)$, figura 4.5 (a). Los 4 vecinos diagonales de p tienen coordenadas

$(x+1, y+1)$, $(x-1, y-1)$, $(x-1, y+1)$ y $(x+1, y-1)$ denotados por $N_D(p)$, figura 4.5 (b). La unión de $N_4(p)$ y $N_D(p)$, figura 4.5 (c), son los 8 vecinos de p denotados por $N_8(p)$ [10].

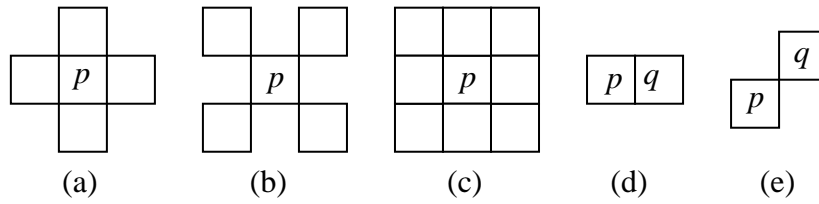


Figura 4.5. (a) píxel p y sus 4 vecinos, $N_4(p)$, (b) píxel p y sus vecinos diagonales, $N_D(p)$, (c) píxel p y sus 8 vecinos, $N_8(p)$, (d) píxeles p y q son 4-adyacentes y 8-adyacentes, (e) píxeles p y q son 8-adyacentes pero no 4-adyacentes.

Dos píxeles p y q son 4-adyacentes si $q \in N_4(p)$, similarmente p y q son 8-adyacentes si $q \in N_8(p)$, figura 4.5 (d) y (c) ilustran estos conceptos [4]. Un camino entre los píxeles p_1 y p_n es una secuencia de píxeles $p_1, p_2, \dots, p_{n-1}, p_n$ tal que p_k es adyacente a p_{k+1} , para $1 \leq k < n$. Un camino puede ser de *conectividad 4* o *conectividad 8*, dependiendo de la definición de adyacencia usada, figura 4.6.

Dos píxeles p y q de un objeto son de conectividad 4, si existe un camino de conectividad 4 entre ellos, y hay conectividad 8 si existe un camino de conectividad 8 entre ellos, el conjunto de todos los píxeles conectados de un objeto es llamado *componente conectado*.

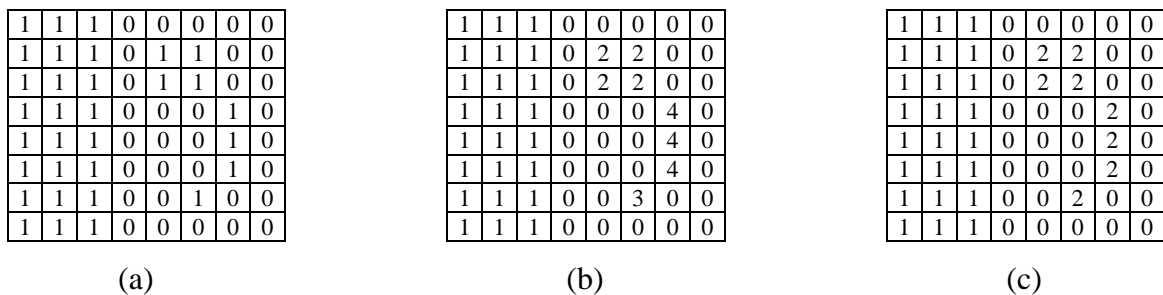


Figura 4.6. (a) imagen binaria, (b) imagen etiquetada usando conectividad 4, (c), imagen etiquetada usando conectividad 8.

Para este algoritmo se han desarrollado diferentes métodos [2], uno de ellos es usando funciones de morfología. Este método consiste en hacer una búsqueda de un píxel negro, cuando el píxel es encontrado se asume que este píxel es parte de un componente y se toma como base para un proceso iterativo basado en la ecuación:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots$$

Donde X_k es el componente extraído, A es la imagen fuente, B es el elemento estructurante de tamaño 3×3 indicando conectividad 8 o 4 y X_0 es el píxel p de donde se inicio la iteración.

El algoritmo inicia creando una nueva imagen x_0 , conteniendo solamente el primer píxel negro, como se ilustra en la figura 4.7 (a), en el primer paso del algoritmo, se dilata x_0 usando un elemento estructurante B , figura 4.7 (b), esto significa que el área de interes es expandida desde el primer píxel negro a todos sus vecinos. El resultado x_1 , figura 4.7 (c), de la primera iteración es la intersección entre A y x_0 dilatada.

En la próxima iteración x_1 es dilatada, figura 4.7 (d), y se realiza el mismo procedimiento, figura 4.7 (d). Esto proceso iterativo continua hasta que el componente resultante sea igual al anterior.

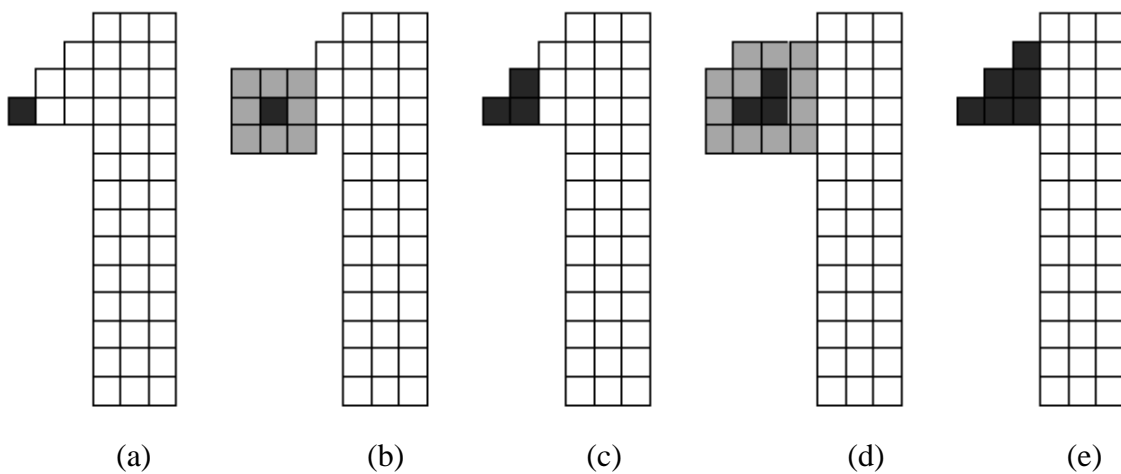


Figura 4.7. Una iteración en el procesote encontrar un componente conectado.

4.2.1 Segmentación de caracteres por etiquetado de componentes conectados

Partiendo de la imagen de la placa en binario (obtenida de la detección), se hace una eliminación de los pequeños componentes que no forman parte de los caracteres como manchas (objetos extraños), tornillos, etc. Para esto se hace un etiquetado de componentes conectados, figura 4.8, donde se pasa a eliminar los componentes que no cumplen con un porcentaje de área respecto a la placa y además que no cumplan una comparación de la relación ancho/alto de cada componente. Se elimina únicamente los componentes que sean inferiores a las especificaciones, en otro caso no se elimina porque se puede eliminar uno o más caracteres, en la figura 4.8 se muestra que los dos últimos caracteres de la placa (6 y 3) están unidos como un solo componente por lo cual no cumple con la relación ancho/alto pero no se los elimina porque no se tiene la seguridad de que sea ruido.

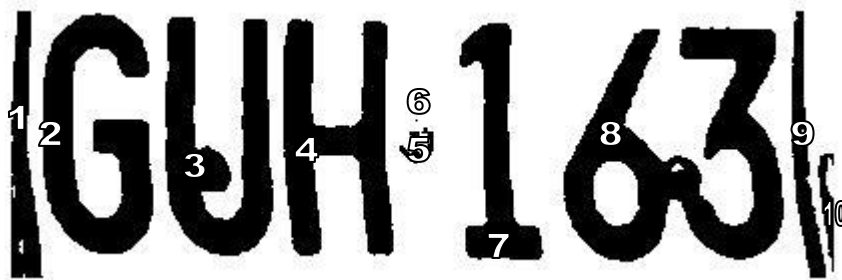


Figura 4.8. Placa etiquetada por componentes conectados

Otro caso que se puede presentar es cuando un carácter está unido al marco de la placa por una mancha o un tornillo, ver figura 4.9, para solucionar este problema después de aplicar el algoritmo para borrar ruido, se aplica un algoritmo que encuentra las líneas de el marco (por suma de píxeles) y traza una línea (de el mismo color del fondo) donde finalizan, separando así los caracteres del marco. En la figura 4.9 se muestra con líneas amarillas la separación del marco con el carácter 5, luego de esto se pueden eliminar las líneas de bordes como ruido.



Figura 4.9. Caracter unido al marco de la placa con las líneas de separación

Este método de eliminación de el marco de la placa, no presenta buenos resultados si la placa tiene un área grande de bordes, por lo cual no se pueden eliminar, porque se pueden encontrar objetos separados que no se borran porque no se está seguro de que no es un caracter.

Al eliminar los pequeños componentes y comprobar si se tiene líneas del marco, se hace nuevamente un etiquetado de componentes conectados, si el número de componentes es menor a 6 se revisa cual de los componentes es aproximadamente el doble de ancho de los demás y se hace una búsqueda del mínimo (valle) para ahí separar los caracteres. Ver figura 4.10 donde se separa el 6 de el 3, los cuales correspondían al mismo componente en la figura 4.8.



Figura 4.10. Placa después de eliminar los pequeños componentes y separación de los caracteres 6 y 3.

Luego de tener los 6 componentes separados y etiquetados se toma cada valor del etiquetado como un caracter obteniendo así los 6 caracteres separados.

4.3 RELACIÓN DE DISTANCIA

Debido a que los parámetros como: dimensiones (largo, ancho), ancho del carácter y separación entre caracteres de todas las placas colombianas son iguales, se puede dividir la imagen conociendo estos parámetros (ver figura 4.11).



Figura 4.11. Placa colombiana

Si se observa en la figura 4.11, se puede dividir la imagen en seis partes a una distancia igual al ancho del carácter, y una parte a una menor distancia que es la que se encuentra en el centro entre las letras y los números.

Puesto que las imágenes originales fueron tomadas a diferentes distancias entre el foco de la cámara y el vehículo, las imágenes resultantes (placa) tienen diferentes tamaños, y con esto un ancho de carácter diferente. Como el ancho del carácter disminuye linealmente como disminuye la longitud de la placa se puede hallar una relación entre la longitud de la placa y el ancho del carácter. Es decir, si se divide una longitud de placa y un ancho de carácter ya conocidos (calculados manualmente de alguna placa), se obtiene un “valor relación” el cual permite calcular el ancho del carácter de cualquier placa a partir de su longitud.

Ejemplo:

Se parte de la imagen umbralizada, se procede a calcular el negativo de la imagen y eliminar objetos pequeños de la imagen ver figura 4.12:



Figura 4.12. Negativo de la imagen original

Se procede a eliminar las zonas externas (ruido), es decir se eliminan áreas de píxeles pequeños de la imagen ver figura 4.13



Figura 4.13. Placa sin ruido

Se eliminan las líneas del rectángulo, para eliminar las líneas por encima y debajo se hace una sumatoria de filas en la imagen, y para las líneas verticales se hace una sumatoria de columnas en la imagen, ver figura 4.14



Figura 4.13. Placa sin el rectángulo alrededor de los números

De la figura 4.14 se calcula la longitud de la placa, la cual es la medida en píxeles desde el primer píxel negro (extremo izquierdo) hasta el último píxel (extremo derecho), se divide este valor por el *valor relación* y se obtiene el ancho del carácter para proceder a dividir la imagen, ver figura 4.15:



Figura 4.15. Imagen dividida en los seis caracteres

Para evitar los problemas de recortar partes del carácter a causa de que los caracteres estén inclinados, estén des pintados o tengan elementos en sus extremos, en la figura 4.15 se puede ver que se recorta parte del carácter por que los caracteres están inclinados, generando un cálculo no preciso de la distancia y/o un mal recorte, se aumenta en un porcentaje pequeño el valor de ancho del carácter y a cada imagen

recortada se le elimina los elementos de los otros caracteres (calculado de las áreas de píxeles pequeños), ver figura 4.16.



Figura 4.16 Caracteres inclinados

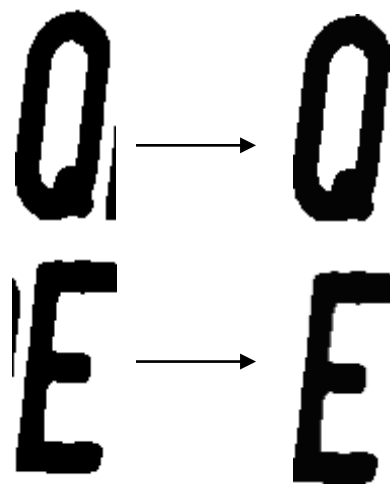


Figura 4.17. Caracteres sin elementos de caracteres vecinos

Para que el recorte de los caracteres de la placa sea satisfactorio, se debe garantizar que la imagen resultante de la figura 4.16 no posea elementos externos grandes, que conlleven a un error en el cálculo de la longitud de la placa, por esto la desventaja de este método es el alto riesgo de elegir un límite de corte erróneo, la ventaja es que el éxito de la separación de caracteres no depende de la calidad de la imagen.

También se pueden encontrar las líneas verticales del marco de la placa utilizando la transformada de Hough. Para esto se parte de la imagen localizada asegurándose que se tiene el marco de la placa, una vez se tenga las posiciones de las líneas se hace el recorte.

4.4 PICOS Y VALLES

El método conocido como picos y valles consiste en realizar una proyección horizontal en la cual se calcula el número de píxeles negros de las columnas. Si se suma lo píxeles negros a cada columna se obtienen valores altos donde se encuentra el carácter (picos) y valores bajos o nulos (valles), con estos valores se puede hallar el comienzo y final de cada carácter ver figura 4.18.



Figura 4.18. Picos y valles de la placa

El problema que presenta este método está en la calidad de la placa extraída, por ejemplo, si se observa la imagen de la figura 4.18 entre las letras *Q* y *E*, que están inclinadas, se pierde el fin y comienzo de los caracteres (valle), de igual forma el ruido en la imagen puede conllevar a errores.

5 EXTRACCIÓN DE CARACTERÍSTICAS

5.1 INTRODUCCIÓN

Para reconocer un objeto de la imagen es necesario extraer características del mismo, que permitan representarlo y describirlo matemáticamente.

Se entiende por representación de un objeto, previamente segmentado, cualquier proceso que conlleve la transformación de los píxeles que lo integran en un formato más compacto y de un nivel superior. Por ejemplo, en primera instancia el contorno de un objeto viene dado por los píxeles que integran dicha frontera. Una forma más eficaz de representar el contorno del objeto puede ser mediante un ajuste poligonal del mismo, que simplifica la información a manejar. A veces, los píxeles no son transformados sino que ellos mismos constituyen la representación que define el objeto.

A partir de una determinada representación es necesario describir matemáticamente el objeto. Esto es, extraer sus características (color, tamaño, posición, etc.), mediante un vector denominado *vector de características* que cuantifica el valor de las mismas.

En un gran número de ocasiones es conveniente que el *vector de características* sea invariante, si es posible, ante el escalado, posición y rotación del objeto en la imagen.

5.2 MOMENTOS

Dada una función continua $f(x, y)$, se define su momento de orden $(p+q)$ como:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots$$

El teorema de unicidad de Papoulis dice que, si $f(x, y)$ es continua a trozos y tiene valores no nulos sólo en una zona finita del plano x - y , entonces todos sus momentos existen y la aplicación que asigna una secuencia de momentos $\{m_{pq}\}$ a una función

$f(x, y)$ es biyectiva. Por consiguiente, tomando un número finito de momentos se obtiene una aproximación del objeto.

El problema de esta descripción es que, en principio, no es invariante. Centrando el sistema de referencia en el objeto se obtienen los momentos centrales (invariantes a la traslación) [4].

Los momentos se definen como:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \hat{x})^p (y - \hat{y})^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots$$

donde

$$\hat{x} = \frac{m_{10}}{m_{00}} \quad \hat{y} = \frac{m_{01}}{m_{00}}$$

son las coordenadas del centroide o centro de masa del objeto.

En el caso de una imagen digital se tiene:

$$\mu_{pq} = \sum_i \sum_j (i - \hat{i})^p (j - \hat{j})^q f(i, j)$$

Los momentos centrales se pueden poner en función de los no centrales. Por ejemplo, para los de orden hasta dos se tiene:

$$\begin{aligned} \mu_{00} &= \sum_i \sum_j (i - \hat{i})^0 (j - \hat{j})^0 f(i, j) = \sum_i \sum_j f(i, j) = m_{00} \\ \mu_{10} &= \sum_i \sum_j (i - \hat{i}) f(i, j) = m_{10} - \frac{m_{10}}{m_{00}} m_{00} = 0 \\ \mu_{11} &= \sum_i \sum_j (i - \hat{i}) (j - \hat{j}) f(i, j) = m_{11} - \frac{m_{10} m_{01}}{m_{00}} \end{aligned}$$

$$\mu_{20} = \sum_i \sum_j \left(i - \hat{i} \right)^2 \left(j - \hat{j} \right)^0 f(i, j) = m_{20} - \frac{2m_{10}^2}{m_{00}} + \frac{2m_{10}^2}{m_{00}} = m_{20} - \frac{m_{10}^2}{m_{00}}$$

$$\mu_{02} = \sum_i \sum_j \left(i - \hat{i} \right)^0 \left(j - \hat{j} \right)^2 f(i, j) = m_{02} - \frac{m_{01}^2}{m_{00}}$$

Los momentos μ_{20} y μ_{02} son los momentos de inercia del objeto con respecto a los ejes de coordenadas x e y , respectivamente.

Para conseguir, además, invarianza ante escalados, se definen los momentos centrales normalizados:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}$$

Donde $\gamma = \frac{p+q}{2} + 1$, para $p+q = 2, 3, \dots$

Finalmente, de los momentos segundo y tercero se deriva un conjunto de siete momentos invariantes a traslaciones, rotaciones y cambios de escala:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

$$\phi_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

Los siete momentos obtenidos son los valores que conforman el vector de características.

5.3 MÉTODO PARA EL RECONOCIMIENTO INVARIANTE A LA TRASLACIÓN, ROTACIÓN Y ESCALAMIENTO (RITRE)

Para obtener un vector de características que sea invariante a la rotación, escala y traslación se debe seguir los siguientes pasos:

Primero se debe calcular el momento de inercia de la imagen. En general, el momento de inercia cuantifica la inercia de un objeto rotativo considerando la distribución de su masa. El momento de inercia es normalmente calculado dividiendo el objeto en N piezas de masas m_1, m_2, \dots, m_N . Cada pieza está a una distancia r_1, r_2, \dots, r_N desde el eje de rotación. El momento de inercia del objeto es:

$$I = m_1 r_1^2 + m_2 r_2^2 + \dots + m_N r_N^2.$$

El momento de inercia depende de la posición del eje de rotación y de la forma y masa del objeto rotativo. Este es invariante a la traslación y rotación. Una imagen bidimensional no es un objeto con masa, pero se puede representar a la imagen por una función continua $f(x, y)$ en la cual cada píxel de la imagen puede ser considerado una partícula con masa igual al valor de intensidad del píxel. Para imágenes binarias, el momento de inercia del objeto con respecto a su centroide (momento de inercia central) es:

$$I = \sum_{i=1}^N d_i^2 = \sum_{i=1}^N \left((x_i - C_x)^2 + (y_i - C_y)^2 \right)$$

Donde C_x, C_y son los centroides de las coordenadas, x_i, y_i son las coordenadas de los píxeles de la imagen y N es el total de píxeles en la imagen. La invarianza a la traslación y rotación es lograda calculando el momento de inercia central y si se divide a I por N^2 (I_N), se logra la invarianza a la escala. Debido a que existe la posibilidad de que dos o más imágenes diferentes obtengan un mismo I_N , se propone un método más preciso. Se

basa en la creación de círculos sobre la imagen. La primera parte de este método considera el número de cambios de intensidad que tiene la imagen por donde pasa el círculo (ver figura 5.1).



Número de cambios de intensidad: 1 2 2 4 4 4 4

Figura. 5.1. Círculos sobre la letra “N” de una placa colombiana

Este simple código extrae características del objeto independientemente de la posición, orientación y tamaño. Además, para obtener una representación más robusta, se usa varios círculos proporcionalmente arreglados sobre el objeto. Sin embargo, en algunos casos dos diferentes objetos pueden tener el mismo código radial (por ejemplo las letras *M* y *N*, ver figura 5.2). Por lo tanto la segunda parte de este método consta de calcular los arcos más largos para cada círculo d_1 y d_2 ver figura 5.3. Para lograr un tamaño normalizado, se divide la diferencia de d_1 y d_2 por la circunferencia.



Cambios de intensidad:

0 2 2 2 2 2 4



Cambios de intensidad:

0 2 2 2 2 2 4

Figura. 5.2. Igual número de cambios de intensidad en las letra N y M

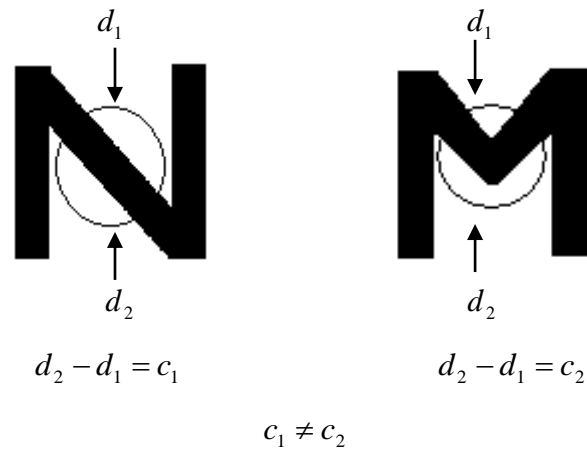


Figura. 5.3. Diferencia entre los arcos de la letra N y M

Para este tipo de letra ocurre que el número de cambios de intensidad dan iguales, pero en el caso de las letras M y N de la placa colombiana son diferentes.

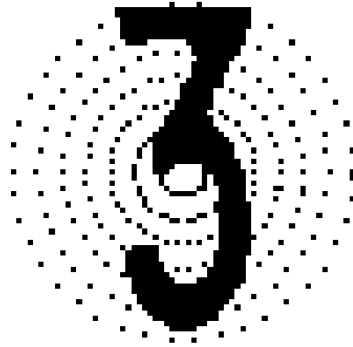
Los pasos a seguir para obtener el código radial de una imagen binaria es:

1. Obtener el centroide del objeto (centro en x y y)
2. Generar K círculos (C_i) concéntricos equidistantes alrededor del centroide. El espacio entre cada círculo es igual a la distancia entre el centroide y el píxel mas lejano del objeto dividido por K .
3. Para cada círculo, contar el número de cambio de intensidad (de cero a uno o de uno a cero) que ocurren en la imagen, este es R_i .
4. Para cada círculo, obtener los dos arcos más largos. La medida de cada arco se calcula con el número de píxeles que contiene el arco, se obtiene la diferencia entre las dos distancias d_1 y d_2 , y se divide por la circunferencia ($D_i = \frac{(d_1 - d_2)}{dc}$), donde d_1 es el arco mas largo, d_2 es el segundo arco mas largo y dc es la circunferencia [11].

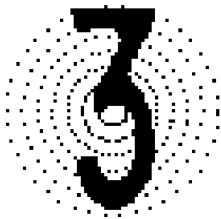
El código radial puede ser representado por el siguiente vector:

$$R_1, R_2, \dots, R_K, D_1, D_2, \dots, D_K$$

Considerando K círculos. R_i enteros positivos, y D_i valores reales entre $[0,1]$. La Figura 5.4, muestra el código radial para diferentes tamaños del número 3 de una placa colombiana.



$$D_i = [1 \ 1 \ 2 \ 3 \ 2 \ 2 \ 2 \ 0.068 \ 0.013 \ 0.087 \ 0.013 \ 0.048 \ 0.039 \ 0.81]$$



$$\begin{matrix} 1 & 1 & 2 & 3 & 2 & 2 & 2 \\ 0.06 & 0.02 & 0.08 & 0.01 & 0.04 & 0.03 & 0.79 \end{matrix}$$



$$\begin{matrix} 1 & 1 & 2 & 3 & 2 & 2 & 2 \\ 0.06 & 0.01 & 0.08 & 0.01 & 0.04 & 0.03 & 0.8 \end{matrix}$$

Figura. 5.4. Código radial para diferentes tamaños del número 3.

5.4 EXTRACCIÓN DE CARACTERÍSTICAS BASADA EN COMPONENTES DIRECCIONALES DE KIRSCH

Sobre la imagen inicial se definen 5 componentes. La primera es la propia imagen inicial, las otras 4 son componentes direccionales: horizontal, vertical, diagonal primera y diagonal segunda. Estas se ilustran mejor en la figura 5.5:

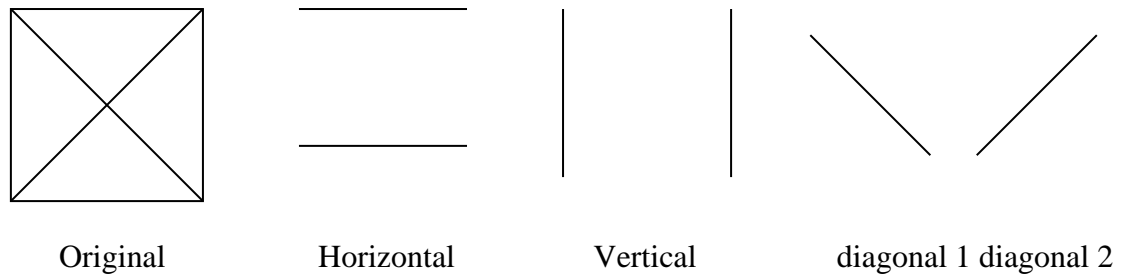


Figura. 5.5. Componentes direccionales de una imagen

El operador gradiente de Kirsch utiliza la siguiente ventana para ser aplicada sobre la imagen (figura 5.6):

A ₀	A ₁	A ₂
A ₇	(i,j)	A ₃
A ₆	A ₅	A ₄

Figura. 5.6. Notación para el entorno de un píxel, necesaria para el cálculo del gradiente de Kirsch.

La figura 5.7 indica las mascararas utilizadas para calcular las cuatro componentes direccionales de una imagen: horizontal, vertical, diagonal derecha y diagonal izquierda. Estas cuatro componentes junto con la imagen inicial son las entradas a los algoritmos de reconocimiento de patrones, en al practica se hace una reducción a cada imagen obtenida para reducir el tamaño del vector de características.

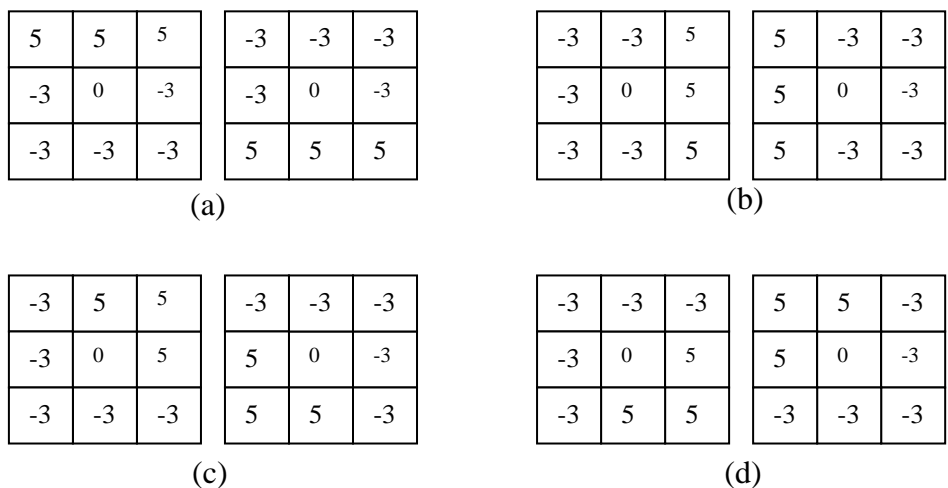


Figura 5.7. Máscara de Kirsch. (a) horizontal, (vertical), (c) diagonal derecha, (d) diagonal izquierda.

5.5 PUNTO FINAL

Un punto final es el píxel que únicamente tenga un vecino. El número de puntos finales de cada caracter y la ubicación de estos también puede ser una característica a tomar en cuenta, por ejemplo, se podría diferenciar el 6 del 9, teniendo en cuenta que el punto final del 6 esta en la parte superior y el del 9 en la parte inferior.

Para encontrar los puntos finales, primero se hace una *esqueletización* [4], figura 5.8 (b). En la mayoría de los casos se tienen puntos finales donde no deberían aparecer, para eliminar estos puntos se eliminan las líneas que tengan menos de 4 píxeles de longitud, figura 5.8(c), el problema que se puede presentar es que se elimina puntos finales reales como el de la 'Q'.

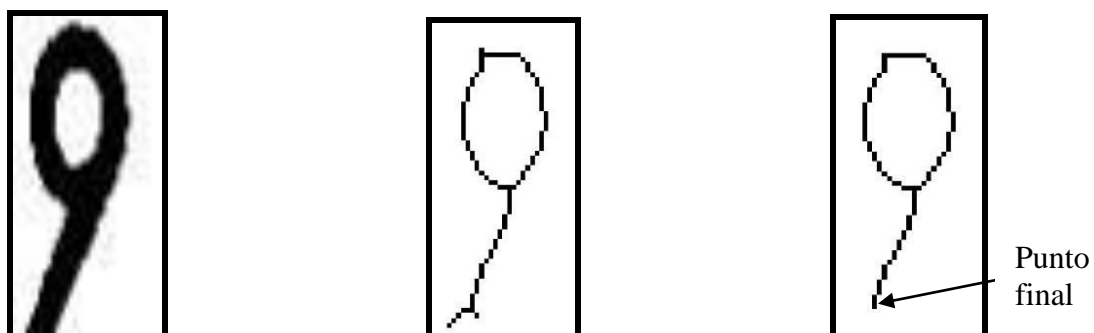


Figura 5.8. (a) Caracter de entrada, (b) esqueletización, (c) eliminación de puntos finales falsos.

5.6 NUMERO DE EULER

El número de Euler se define como el número de objetos en una imagen menos el número de huecos, un ejemplo de esta característica es que '8' o 'B' se pueden reconocer únicamente con este valor, o también se puede clasificar las letras y números en 3 grupos (0, 1 o 2 huecos) antes de pasar al algoritmo de reconocimiento.

Una forma de encontrar este valor es por componentes conectados (tomando como componentes el fondo). Entonces el número de Euler es igual a dos (el carácter que es un componente mas el fondo exterior) menos el número de componentes conectados (fondo mas huecos).

5.7 AREA DE LOS BORDES

Aquí se mide la longitud del borde exterior de cada caracter, en la figura 5.10, se observan dos letras 'Q' y 'E', las cuales tienen gran diferencia en su longitud de borde.

Primero se encuentran las fronteras o límites exteriores [10], figura 5.9, y luego se suma el número de píxeles encontrados.



Figura. 5.9. Área exterior de los bordes, (a) longitud de borde 136, (b) longitud de borde 199.

5.8 AREA

Se hace un conteo de los píxeles horizontales y verticales, como se indica en la figura 5.10, para hacer el conteo se asume el fondo como ceros y el carácter como unos, se puede observar que en la suma de las filas se tiene más diferencia de un carácter a otro, la suma de las columnas es más uniforme aunque se tiene unas pocas variaciones.

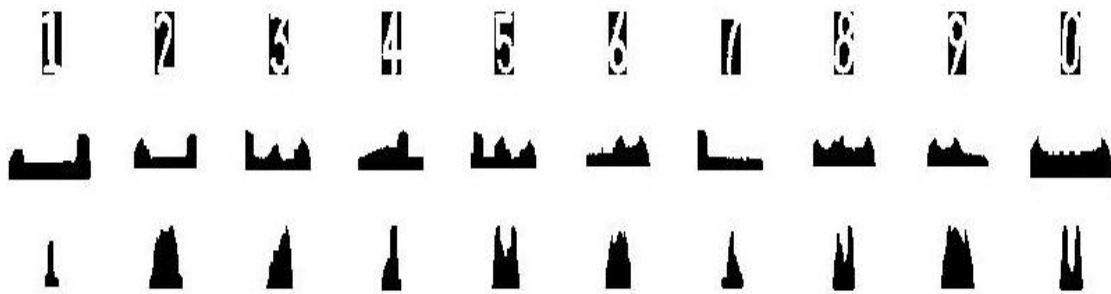


Figura 5.10. Suma de los píxeles horizontales y verticales

5.9 ZONIFICACIÓN

Se realiza una división de la imagen en zonas [12], a cada zona se le encuentra el promedio para obtener un único valor, en la figura 5.11 (a) se muestra una imagen de 60 x 25 y el resultado de la zonificación, figura 5.11 (b) una imagen de 12 x 5.

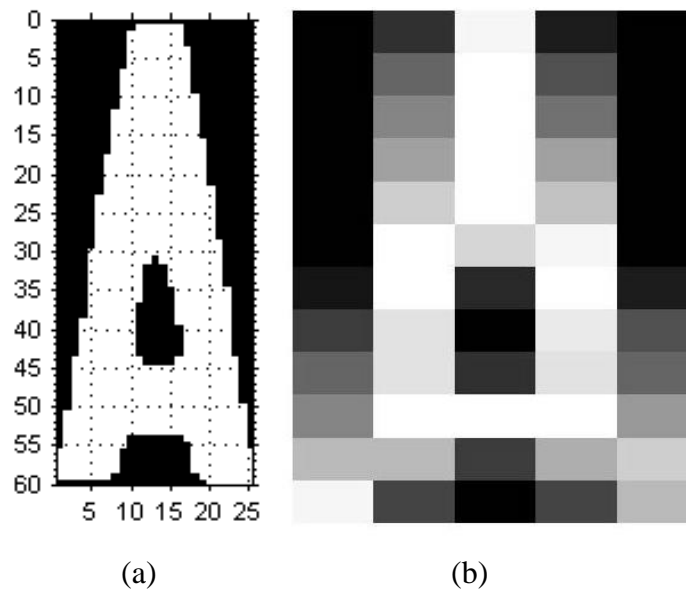


Figura 5.11. Zonificación de una imagen de un caracter

Donde el vector de características es el valor de cada píxel obtenidos de la zonificación.

5.10 VALOR DE CADA PÍXEL

El valor de cada píxel también puede ser usado como vector de características, aunque se tiene el problema de hacer lento el procesamiento, debido a la longitud del vector.

6 RECONOCIMIENTO DE LOS CARACTERES

6.1 INTRODUCCIÓN

Una vez que los caracteres segmentados son representados mediante un *vector de características*, se procede a realizar el reconocimiento. Este proceso también llamado de clasificación, asigna cada vector de características como perteneciente a un grupo representativo de características comunes, llamadas clases, por lo tanto cada clase representa un tipo diferente de carácter.

En este capítulo se explican varias técnicas para la solución a este problema, como son las redes neuronales, el algoritmo del vecino más cercano y el método de “*template matching*”.

6.2 FUZZY ARTMAP

Fuzzy ARTMAP es un modelo de red neuronal dinámica que presenta ciertas características útiles en la solución de este problema, esta red implementa una clase de aprendizaje supervisado y una posterior auto-organización de vectores multidimensionales arbitrarios de entrada (vector de características, capítulo 5), que representan características propias de los vectores de salida mediante mecanismos de predicción o clasificación.

La primera red ARTMAP [10] desarrollada se basó en ART1 [13], y sirve para clasificar mapas binarios. Si se reemplaza la red ART1 por una red Fuzzy ART [14] en el sistema ARTMAP, la arquitectura que se obtiene es una Fuzzy ARTMAP, que aprende en forma estable las relaciones entre valores de entrada y salida ya no sólo binarios, sino también continuos.

Los elementos vectoriales se notarán con letras mayúsculas y cada uno de sus componentes en minúscula. Un patrón es el vector que forma la representación computacional de un objeto (vector de características), I es el patrón de entrada, M es el

número de componentes del patrón I , además el código de complemento definido para un vector u es $U = [u, I-u]$. La norma vectorial se maneja como $|U| = \sum_{i=1}^M u_i$.

6.2.1 ART1 Y Fuzzy ART

Los modelos neuronales intentan resolver el dilema de estabilidad y plasticidad: ¿cómo se pueden aprender rápidamente nuevos patrones (Plasticidad del aprendizaje) sin perder aquellos previamente almacenados (estabilidad del aprendizaje)? Una solución se conoce como Teoría de la Resonancia Adaptativa (ART), a partir de la cual se desarrollaron los modelos ART que se han mencionado. En la arquitectura funcional se distinguen dos tipos de flujo entre las capas constitutivas: uno de entrada a salida (feedforward) y otro de realimentación (feedback).

La arquitectura de ART1 y Fuzzy ART, figura 6.1, consta de dos subsistemas: orientación y atención. El subsistema de orientación está conformado por dos capas (F_1 y F_2). F_1 es una capa conformada por M neuronas y se encarga de la recepción del patrón de entrada, F_2 está conformada por N neuronas que componen el conjunto de categorías que se forman durante el trabajo de la red, solo para Fuzzy ART existe una tercera capa F_0 que trabaja un preprocesamiento, esta capa sirve para codificar la entrada mediante el *código de complemento*, y entregar su salida a la capa F_1 ; en este caso se sigue hablando de una entrada de M componentes por patrón pero la capa F_1 consta de ahora de $2*M$ neuronas.

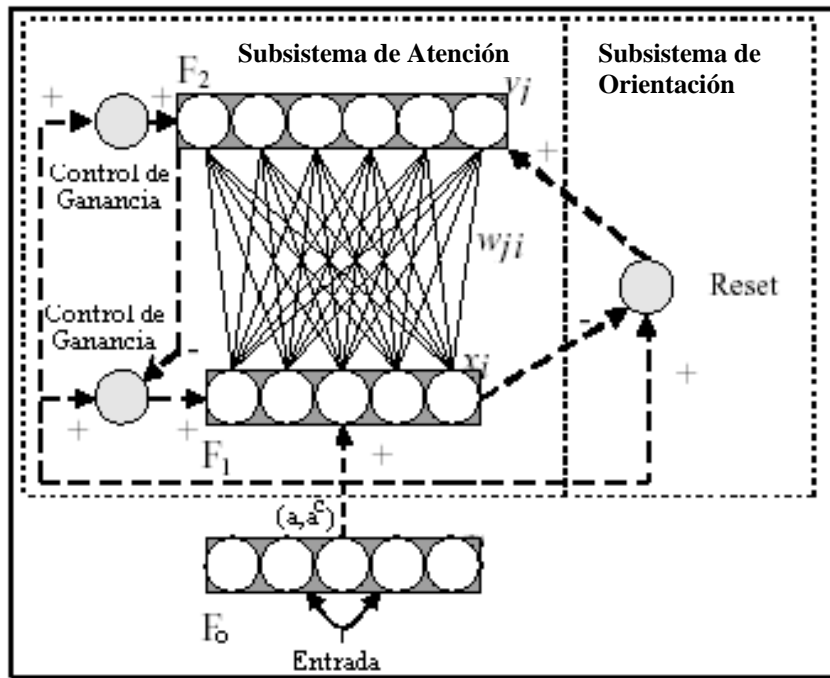


Figura 6.1. Arquitectura Fuzzy ART.

La función *Subsistema de atención* es el reconocimiento y la clasificación de la información. La memoria a largo plazo de la red localizada en este subsistema esta compuesta por los pesos w_{ji} y v_{ij} que corresponden a las neuronas de la capa de entrada y salida, logrando la estabilidad de la red.

El *Subsistema de orientación* es un módulo encargado de validar si una entrada pertenece o no a la categoría seleccionada por la red. El nodo Reset lleva registro de todas aquellas neuronas que fueron rechazadas por no satisfacer los requerimientos de semejanza.

Los siguientes pasos describen el proceso de trabajo de una red ART1 y Fuzzy ART.

- Se le proporciona a la red un patrón arbitrario: cada una de las neuronas de la capa de entrada recibe un componente que es transmitido a cada neurona de la capa de salida.
- Las neuronas de la capa F_2 compiten hasta que una se activa: la vencedora. La competencia consiste en obtener la mayor entrada neta producida por las conexiones desde la capa F_1 a F_2 (conexiones *feedforward*).

- La neurona vencedora envía su salida a la capa F_1 a través de las conexiones *feedback* (conexiones hacia atrás). Como sólo hay una neurona activa (las neuronas no activas no influyen con sus conexiones *feedback*) el valor que llega a cada una de las neuronas de la capa F_1 , es el valor del peso de la conexión entre cada una de ellas y la vencedora.
- En la capa F_1 se compara el patrón de entrada y el patrón formado por los pesos de las conexiones *feedback*; éste último es el prototipo de la categoría representada por la neurona que resultó vencedora. La comparación se hace mediante la *relación de semejanza*: $|X| / |I|$. Tal que si la semejanza entre la entrada y el prototipo de salida es de coincidencia total dará uno (100%) y cero (0%) si son completamente diferentes.
- La *relación de semejanza* se compara con el *parámetro de vigilancia* ρ , ART1 y Fuzzy ART tiene dos parámetros muy importantes que controlan su funcionamiento: el parámetro de vigilancia ρ y un parámetro de aprendizaje β . ρ es un número real cuyo valor es definido por el usuario dentro de un rango $[0, 1]$ (rango compartido con la *relación de semejanza*) que determina cuán parecida debe ser una entrada para clasificarse como de una u otra categoría que ya esté almacenada. Es decir, si es cercano a uno implica una clasificación muy exigente, cuando se especifica ρ igual a 1 solo se reconocerán como pertenecientes a una determinada categoría los patrones de entrada que sean exactamente iguales al prototipo de salida; y si es cercano a cero permite la categorización de entradas poco parecidas como la misma clase. El parámetro β controla el ritmo de aprendizaje, tal que si es igual a uno se tiene un aprendizaje rápido, y si es cero el aprendizaje es nulo. Se pueden presentar dos casos.
 - a) $|X|/|I| < \rho$. El prototipo no representa bien al patrón de entrada. Entonces la neurona ganadora es llevada a un estado de *reset*. Se reinicia el proceso desde el segundo paso, excluyéndose de la competencia a las neuronas en *reset*, con ayuda del nodo *reset*. Este proceso puede generar una condición de desborde, cuando todas neuronas de la capa de salida están bajo el estado de *reset*, la red no podrá crear una nueva categoría, ya que no quedan neuronas para asociarla. Entonces la red se ha saturado y la única solución será agregar más neuronas a la capa de salida.

- b) $|X|/|I| < \rho$. La categoría asociada a la neurona ganadora es la que mejor describe al patrón de entrada. En este punto, se deben hacer ajustes a los pesos de las conexiones *feedback* y *feedforward* (memoria a largo plazo) para que el prototipo memorizado se acomode mejor a las características del patrón reconocido.

Se concluye diciendo que una red ART puede hacer una discriminación de tantas categorías como neuronas tenga en su capa de salida (N es el número de neuronas de salida).

6.2.2 Resumen del algoritmo Fuzzy ART

El vector de entrada se denota por: $a = (a_1, \dots, a_M)$, donde a_i está en el intervalo $[0,1]$, $i = 1, \dots, M$, con código de complemento $I = (a_M, a^c_M)$. En la capa F_1 se genera un vector de activación denotado por: $x = (x_1, \dots, x_M)$ construido a partir de la función intersección entre el vector de entrada con el vector de pesos ganador de la capa F_2 , y donde el vector de activación de la capa F_2 se denota por $y = (y_1, \dots, y_N)$, calculado a partir del nodo ganador.

Vector de pesos: Representación numérica interna de la red para cada categoría almacenada sobre la capa F_2 asociado con el nodo j de cada categoría ($j = 1, \dots, N$). Es un vector de pesos adaptativo $w_j \equiv (w_{j1}, \dots, w_{jM})$, inicialmente:

$$w_{j1}(0) = \dots = w_{jM}(0) = 1$$

Parámetro de selección α : Modula el proceso de escalamiento en el calculo de la selección de categoría. Debe ser muy cercano a cero y $\alpha > 0$.

Selección de categoría: para cada entrada I y cada nodo J de F_2 , la función de selección T_j se define como:

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}$$

donde el operador AND (\wedge) en lógica difusa se define como:

$$A \wedge B = \min(A, B)$$

y la norma se define como:

$$|I| = \sum_{i=1}^{2M} I_i$$

el nodo J ganador se encuentra en un proceso de búsqueda como sigue:

$$T_j = \max\{T_j : j = 1, \dots, N\}$$

Resonancia / reset: se determina que existe resonancia si se cumple:

$$\frac{|I \wedge w_j|}{|I|} \geq \rho$$

En la resonancia se activa la categoría J como respuesta al vector de entrada I ; es decir la red clasifica el vector como clase J , además se ejecuta la activación de los pesos de dicha categoría.

El *reset* ocurre sí el nodo J ganador:

$$\frac{|I \wedge w_j|}{|I|} < \rho$$

Entonces el valor del nodo J en el vector T_j es llevado a cero temporalmente y se inicia la búsqueda de otro nodo que cumpla con la resonancia. Si ninguna categoría resuena, se crea una nueva categoría para dicho vector de entrada.

Aprendizaje: una vez finalice la búsqueda, el vector de pesos w_j es actualizado de acuerdo a la ecuación:

$$w_j^{(Nuevo)} = \beta(I \wedge w_j^{Anterior}) + (1 - \beta)w_j^{Anterior}$$

6.2.3 Sistema Fuzzy ARTMAP

Cada sistema Fuzzy ARTMAP [15] esta formado por dos módulos Fuzzy ART (Fuzzy ART_a y Fuzzy ART_b), Figura 6.2, los cuales son conectados por el mapa de campo F_{ab} , el mapa de campo es un modulo de mapeo que relaciona los nodos de salida de ART_a con los nodos de salida de Fuzzy ART_b . Durante el aprendizaje supervisado, Fuzzy ART_a recibe un conjunto de patrones de la forma $a^{(p)}$ (donde p denota el número de vectores de entrada) y Fuzzy ART_b recibe una serie de patrones $b^{(p)}$, donde $b^{(p)}$ (define cada categoría del conjunto de entrenamiento) es la predicción dado $a^{(p)}$.

Cada vez que se suministra un patrón de entrenamiento, la red Fuzzy ART_a activa un nodo y la red Fuzzy ART_b activa otro. El mapa de campo relaciona los nodos activados; de esta forma, a cada nodo nuevo que se active en Fuzzy ART_a se le asocia un nodo en Fuzzy ART_b .

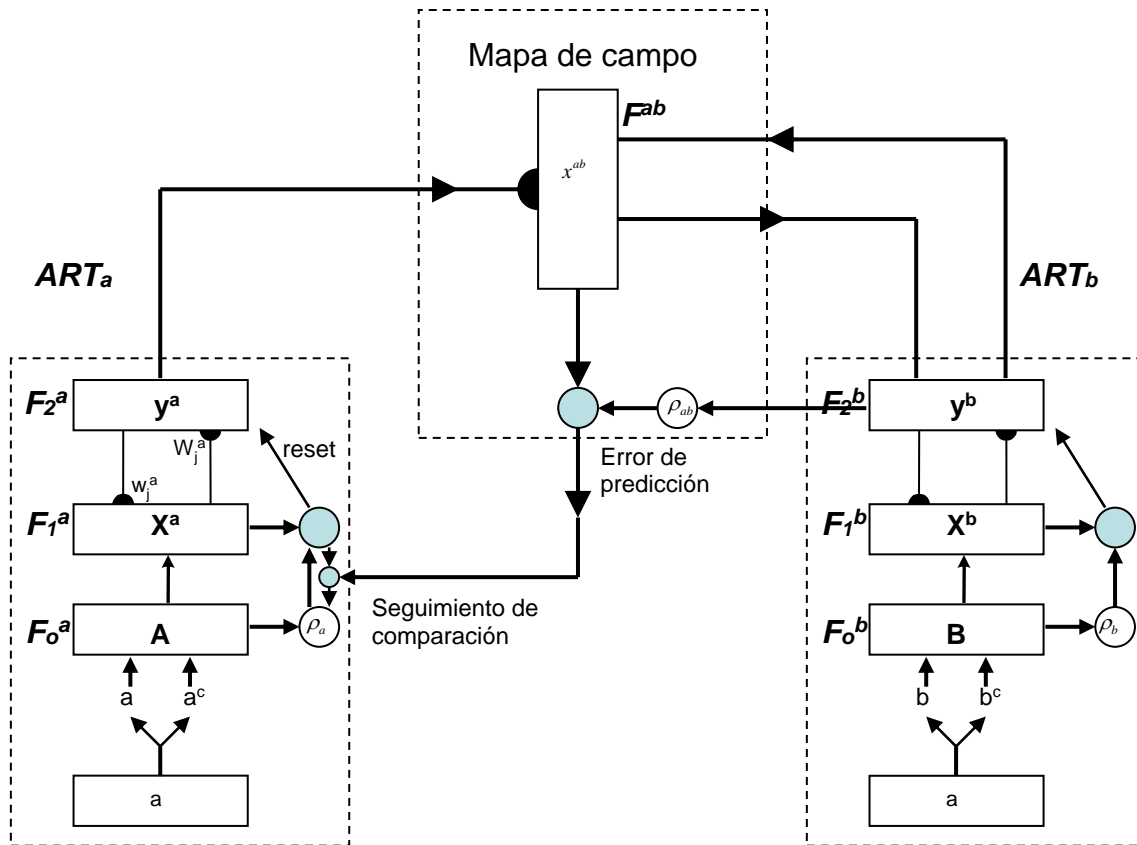


Figura 6.2. Arquitectura ARTMAP.

Cuando el nodo activo de Fuzzy ART_b no representa correctamente al nodo de Fuzzy ART_a , una inhibición proveniente del mapa de campo induce un proceso de seguimiento de comparación, este proceso incrementa la vigilancia ρ_a lo justamente necesario para disparar una búsqueda en Fuzzy ART_a , lo que lleva a encontrar una categoría que predice a $b^{(p)}$ en forma correcta o a un nodo no entrenado previamente, lo que genera una nueva categoría.

Un parámetro de vigilancia base $\bar{\rho} = \bar{\rho}_a$ calibra un nivel mínimo de confianza en el cual Fuzzy ART_a va a aceptar una categoría elegida. Valores bajos de $\bar{\rho}_a$ permiten formar categorías más grandes, inicialmente $\rho = \bar{\rho}$. Durante el entrenamiento, una falla predictiva en Fuzzy ART_b incrementa ρ lo justamente necesario para disparar una búsqueda en Fuzzy ART_a y mediante el seguimiento de comparación se encuentra una categoría que represente mejor a la salida $b^{(p)}$.

En la fase de entrenamiento la red necesita que se le presente un conjunto de patrones y un vector de salida que codifica la categoría que se le debe asignar. Posteriormente, en la fase de ejecución sólo se le presenta los vectores de entrada y la red clasificará dichas entradas de acuerdo a lo aprendido en el entrenamiento.

Las interacciones mediadas por el mapa de campo pueden ser operacionalmente caracterizadas como sigue:

ART_a y *ART_b*: Las entradas a *ART_a* y *ART_b* son en código de complemento figura 6.2 las variables en *ART_a* y *ART_b* son designadas por subíndices o superíndices *a* y *b*. Para *ART_a*, $x^a \equiv (x_1^a \dots x_{2Ma}^a)$ denota el vector de salida de la capa F_1^a ; $y^a \equiv (y_1^a \dots y_{Na}^a)$ denota el vector de salida de F_2^a y $w_j^a \equiv (w_{j1}^a, w_{j1}^a, \dots, w_{j,2Ma}^a)$ denota el *k*th nodo del vector de pesos de Fuzzy *ART_a*. Para *ART_b*, $x^b \equiv (x_1^b \dots x_{2Mb}^b)$ denota el vector de salida de F_1^b ; $y^b \equiv (y_1^b \dots y_{Nb}^b)$ denota el vector de salida de F_2^b ; $w_k^b \equiv (w_{k1}^b, w_{k1}^b, \dots, w_{k,2Mb}^b)$ denota el vector de pesos del *j*th nodo de F_2^a a F_2^{ab} .

Activación del Mapa de Campo: El mapa de campo F^{ab} , es activado principalmente si *ART_a* predice o no la misma categoría de *ART_b* a través de los pesos w_j^{ab} . El vector de salida x^{ab} obedece

$$x^{ab} = \begin{cases} y^b \wedge w_j^{ab} & \text{Si el } j\text{-ésimo nodo de } F_2^a \text{ está activo y } F_2^b \text{ está activo} \\ w_j^{ab} & \text{Si el } j\text{-ésimo nodo de } F_2^a \text{ está activo y } F_2^b \text{ está inactivo} \\ y^b & \text{Si } F_2^a \text{ está inactivo y } F_2^b \text{ está activo} \\ 0 & \text{Si } F_2^a \text{ está inactivo y } F_2^b \text{ está inactivo} \end{cases}$$

$x^{ab} = 0$ si la predicción w_j^{ab} es desconformada por y^b . Este acontecimiento provoca una nueva búsqueda en *ART_a* para encontrar una mejor categoría como sigue.

Seguimiento de comparación: al inicio de cada presentación de entrada, el parámetro de vigilancia de ART_a ρ_a es igual al parámetro de vigilancia base, $\overline{\rho_a}$. Si $|x^{ab}| < \rho_{ab}|y^b|$, entonces ρ_a es incrementado ligeramente mas grande que $|A \wedge w_j^a| |A|^{-1}$, donde A es la entrada a F_1^a . Entonces

$$|x^a| = |A \wedge w_j^a| < \rho_a |A|$$

Donde J es el índice del nodo activo F_2^a , cuando esto ocurre se hace una nueva búsqueda en ART_a para la activación de otro nodo F_2^a con

$$|x^a| = |A \wedge w_j^a| \geq \rho_a |A|$$

y

$$|x^{ab}| = |y^b \wedge w_j^{ab}| \geq \rho_{ab} |y^b|$$

Pesos: El vector de pesos de cada categoría j de Fuzzy ART_a se denomina w_a , para Fuzzy ART_b se denomina w_b y para el Mapa de Campo w_{ab} .

6.3 VECINO MÁS CERCANO HOLOGRÁFICO

El algoritmo del vecino más cercano holográfico (VMCH) está basado en los principios de Memoria Asociativa Holográfica (MAH) [16]. La MAH basa su concepto en la holografía óptica sobreponiendo múltiples asociaciones de estímulo-respuesta en una misma unidad de procesamiento. Todo esto en el dominio de los números complejos. El vector de datos externos es representado por el conjunto de estímulos-respuestas [17]:

$$S = (s_1, s_2, \dots, s_M)$$

donde s_i son valores escalares definidos en el conjunto de estímulos-respuestas y M es el número de variables de entrada.

El vector S_i de la MAH puede ser el código radial calculado en la sección 5.3 (Método para el reconocimiento invariante a la traslación, rotación y escalamiento (RITRE)), o cualquier vector de características calculado en el capítulo 5.

Cada variable de entrada es mapeada a variables polares por el término de relación sigmoïdal:

$$\theta_i = 2\pi \left(1 + e^{(\mu - s_i)/\sigma} \right)^{-1}$$

Donde μ es la media y σ la desviación estándar para cada variable. La anterior ecuación mapea cada s_i a valores de fase entre cero y 2π .

El algoritmo VMCH se calcula como la menor distancia euclidiana de las θ_i , en donde cada una se obtiene utilizando la siguiente expresión:

$$MinEuclidiana = \sqrt{\sum_{i=1}^M (\theta_{exp}^i - \theta_i^i)^2}$$

θ_i es el nuevo parámetro de entrada

θ_{exp} es el parámetro de entrenamiento

Los anteriores parámetros son los vectores de características que han sido mapeados a variables polares por el término de relación sigmoïdal. Para identificar cual es el vector de características que representa al caracter, se calcula la distancia euclidiana entre todos los θ_{exp}^i de entrenamiento y el nuevo parámetro de entrada, el menor valor de este cálculo será el vector de características o caracter.

6.4 RED MULTICAPA FEEDFORWARD

La Red multicapa *feedforward* es la red más comúnmente usada con el algoritmo *backpropagation*.

Este método se basa en el modelo neuronal mostrado en la figura 6.3, donde R es el número de elementos del vector de entrada, $a = f(W_p + b)$.

La función f usada puede ser: sigmoideal y lineal. Sus graficas se indican en la figura 6.4:

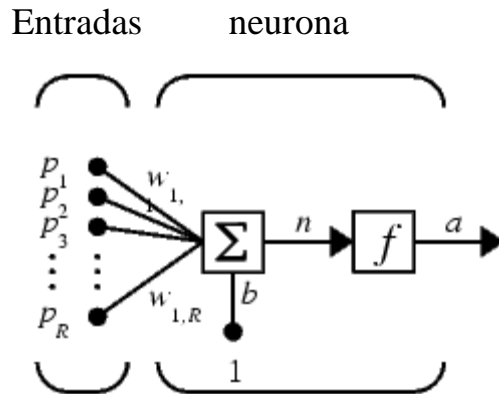


Figura. 6.3. Modelo de la neurona con R entradas.

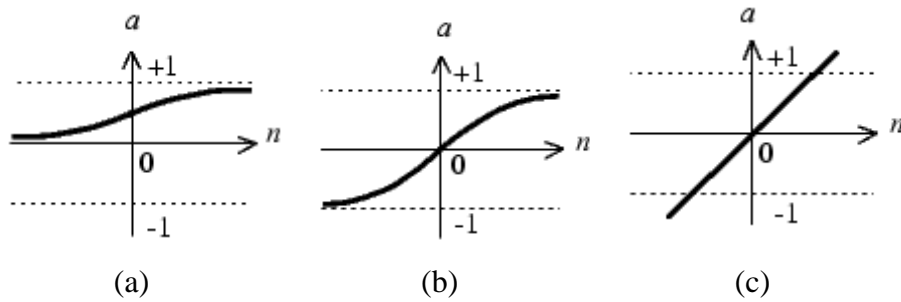


Figura. 6.4. Funciones (a) sigmoideal (0,1), (b) sigmoideal (-1,1), (c) lineal

La arquitectura de la red Multicapa *Feedforward* usada se indica en la figura 6.5

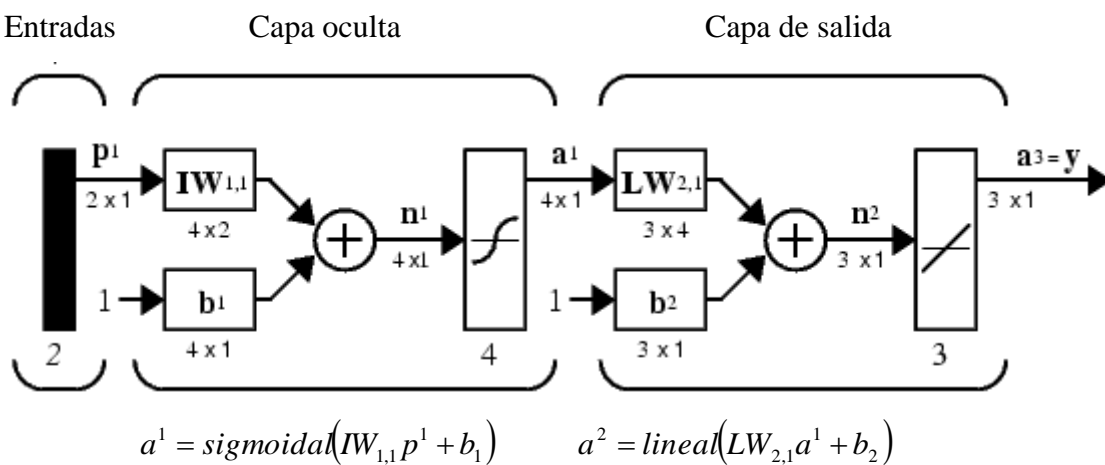


Figura. 6.5. Arquitectura de la red Multicapa *Feedforward*

Se crearon 2 redes por cada método de extracción de características, en las cuales se usaron las 3 capas descritas en la figura 6.5. Las configuraciones de neuronas se explican en la sección 7.5.4.

6.5 CLASIFICADORES ESTADÍSTICOS ÓPTIMOS

En esta sección se presenta una aproximación probabilística al reconocimiento de caracteres.

Dado un patrón desconocido x , este puede ser clasificado como perteneciente a una clase j de un conjunto de M clases. Esto se puede hacer usando un clasificador estadístico; en esta sección se presenta el clasificador de Bayes como función discriminante para dividir el espacio de características.

El clasificador de Bayes [18] para funciones de error 0-1 (1 para decisiones incorrectas y 0 para decisiones correctas), tiene funciones de decisión de la forma:

$$d_j(x) = p(x/w_j)P(w_j) \quad j = 1, 2, \dots, M$$

Donde $p(x/w_j)$ es la función de densidad de probabilidad (PDF) de los vectores patrón de la clase w_j y $P(w_j)$ es la probabilidad de ocurrencia de la clase w_j [4].

Entonces, un patrón vectorial x se asigna a la clase w_j si $d_i(x) > d_j(x)$ para todo $j \neq i$.

Cuando las funciones de densidad de probabilidad son (o se asumen) gaussianas, las PDF gaussianas n -dimensionales tienen la forma:

$$p(x/w_j) = \frac{1}{(2\pi)^{n/2} |C_j|^{1/2}} e^{\left[-\frac{1}{2}(x-m_j)^T C_j^{-1}(x-m_j) \right]}$$

Donde C_j es la matriz de covarianza y m_j es el vector medio del vector población de la clase w_j , y $|C_j|$ es el determinante de C_j .

Puesto que el logaritmo es una función monótona creciente, escoger el más grande $d_j(x)$ para clasificar los patrones es equivalente a utilizar el más grande $\ln[d_j(x)]$, así se puede utilizar las funciones de decisión de la forma:

$$d_j(x) = \ln[p(x/w_j)P(w_j)] = \ln p(x/w_j) + \ln P(w_j)$$

que siempre es real porque $p(x/w_j)$ y $P(w_j)$ son positivas. Sustituyendo la expresión para la PDF gaussiana se tiene la ecuación:

$$d_j(x) = \ln p(w_j) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |C_j| - \frac{1}{2} (x - m_j)^T C_j^{-1} (x - m_j)$$

El término $(n/2)\ln 2\pi$ es un valor constante positivo para todas las clases, por lo que es posible eliminarlo, obteniendo entonces:

$$d_j(x) = \ln p(w_j) - \frac{1}{2} \ln |C_j| - \frac{1}{2} (x - m_j)^T C_j^{-1} (x - m_j)$$

6.6 COMPARACIÓN DE PLANTILLAS

La comparación de plantillas (*Template Matching*), es uno de los primeros métodos de clasificación desarrollados para la comparación de patrones. En este método, los píxeles individuales de la imagen se utilizan como características. La clasificación se lleva a cabo comparando la imagen de un carácter de entrada contra un conjunto de patrones que representan a cada una de las clases de caracteres. En el caso de reconocimiento de caracteres es posible también utilizar más de un patrón por cada carácter, aunque esto aumentará el tiempo de ejecución.

Cada comparación resulta en una medida de similitud entre el carácter de entrada y el patrón, la similitud se incrementa cuando un píxel del carácter es idéntico al mismo píxel en el patrón y decrece si los píxeles son diferentes. Después que se comparan todas las plantillas contra la imagen del carácter de entrada, la clase a la que pertenece

el caracter es la clase del patrón más similar. La forma más común para ejecutar la comparación entre plantillas es la correlación cruzada normalizada [19], tambien se planteó un método de máximo y mínimos que presenta resultados muy similares.

La plantilla es pasada paso por paso sobre toda la imagen, durante cada paso la plantilla es correlacionada con un área de la imagen de entrada, la cual tiene el mismo tamaño de la plantilla. Donde se encuentre el mayor valor, se encuentra la mayor similitud.

6.6.1 Construcción de la plantilla

La construcción de la plantilla tiene un papel muy importante en el éxito de la comparación de plantillas, aunque no hay reglas establecidas para la construcción de las plantillas, éstas tienen que representar todas o las características más relevantes de los objetos a reconocer.

Como las imágenes son tomadas desde diferentes distancias, diferentes iluminaciones, diferentes ángulos y además algunos vehículos tienen las placas más gastadas que otros, se tiene que, después de la umbralización se obtienen diferentes imágenes para los mismos caracteres, ver figura 6.6 (a)

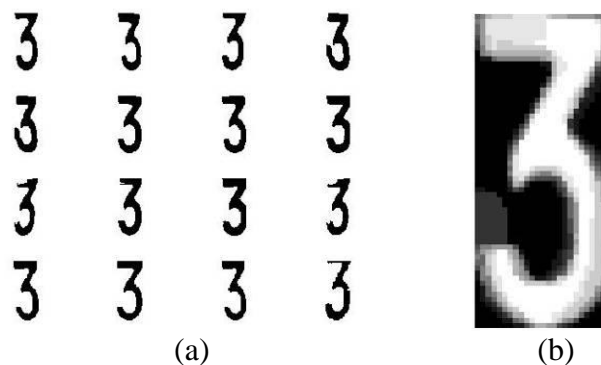


Figura 6.6. (a) Diferentes muestras de imágenes del caracter 3, (b) media de 10 muestras

Por las anteriores razones se tomó como plantilla para cada caracter la media de diez muestras, figura 6.6 (b), luego de tener las plantillas de todos los dígitos, se hace una pila, como la figura 2.3, de todas las plantillas. La figura 6.7 muestra la pila de plantillas de los dígitos. Para las letras se hace el mismo procedimiento.



Figura 6.7. Plantilla de los dígitos.

6.6.2 Correlación Cruzada normalizada

La correlación calcula la similitud de la plantilla $w(x,y)$ dentro de una imagen $f(x,y)$, figura 6.8, dependiendo de la correlación se tiene un alto o bajo valor de similitud.

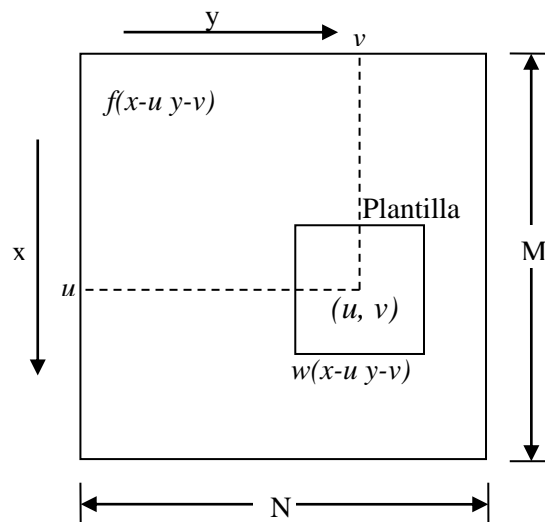


Figura 6.8. Esquema de la correlación

La correlación cruzada es basada sobre el cuadrado de la distancia euclidiana de la forma:

$$d_{f,t}^2(u, v) = \sum_x \sum_y [f(x, y) - w(x-u, y-v)]^2$$

La anterior ecuación es una expresión de la suma del cuadrado de la distancia entre los píxeles en la plantilla y los píxeles en la imagen cubierta por la plantilla.

Expandiendo la expresión se tiene

$$d_{f,t}^2(u, v) = \sum_x \sum_y [f^2(x, y) - 2f(x, y).w(x-u, y-v) + w^2(x-u, y-v)]$$

El término $\sum_x \sum_y w^2(x-u, y-v)$ es la suma de los píxeles de la plantilla al cuadrado y es constante, asumiendo que el término $\sum_x \sum_y f^2(x, y)$ también puede ser estimado como constante, entonces el término restante llega a ser una medida de la similaridad entre la imagen y la plantilla. Esta medida de similaridad es llamada la correlación cruzada

$$c(u, v) = \sum_x \sum_y [f(x, y).w(x-u, y-v)]$$

La correlación cruzada tiene algunas desventajas en la comparación de plantillas como:

- Si la energía de la imagen $\sum_x \sum_y f^2(x, y)$ varia con la posición puede que no se obtenga un resultado correcto. Por ejemplo el valor de correlación puede ser más alto en regiones de brillo que en el áreas donde la plantilla esta realmente correlacionada.
- El rango de $c(u, v)$ es dependiente del tamaño de la plantilla.
- Es invariante a cambios en la imagen como los causados por los cambios en las condiciones de luz a través de la secuencia de la imagen.

La correlación cruzada normalizada supera estas dificultades [9].

$$\gamma(u, v) = \frac{\sum_x \sum_y [f(x, y) \cdot \bar{f}_{u,v}] [w(x-u, y-v) - \bar{t}]}{\sqrt{\sum_x \sum_y [f(x, y) \cdot \bar{f}_{u,v}]^2 \sum_x \sum_y [w(x-u, y-v) - \bar{t}]^2}}$$

Donde $\bar{f}_{u,v}$ es la media de los píxeles de la imagen de entrada en la región cubierta por la plantilla y \bar{t} es la media de la plantilla el valor de γ esta entre -1 y 1, siendo 1 cuando hay una perfecta relación, 0 no hay relación y -1 perfecta relación girada 180 grados.

En el caso de esta aplicación donde el tamaño de la imagen es igual al de la plantilla, entonces se tiene.

$$\gamma = \frac{\sum_x \sum_y [f(x, y) - \bar{f}][w(x, y) - \bar{t}]}{\sqrt{\sum_x \sum_y [f(x, y) - \bar{f}]^2 \sum_x \sum_y [w(x, y) - \bar{t}]^2}}$$

Para los números se hace una comparación del caracter con las 10 plantillas, la plantilla del caracter que arroje el valor más alto es el caracter de salida.

Este método también podría ser utilizado para la localización de la placa, tomando como plantilla la forma de la placa, el problema que se presenta es que no se tiene el mismo tamaño de la placa en todas las imágenes, por lo cual se necesitaría partir de una plantilla (placa) pequeña e ir aumentándola de tamaño hasta obtener el tamaño de la placa de la imagen de entrada, lo cual entregará una alta correlación. En la figura 6.9, se observa un ejemplo simple, tomando como plantilla la placa de la misma imagen a la cual se le aplicó la correlación, en la figura 6.9 (c) se observa el pico donde se encuentra la placa en la imagen de entrada.

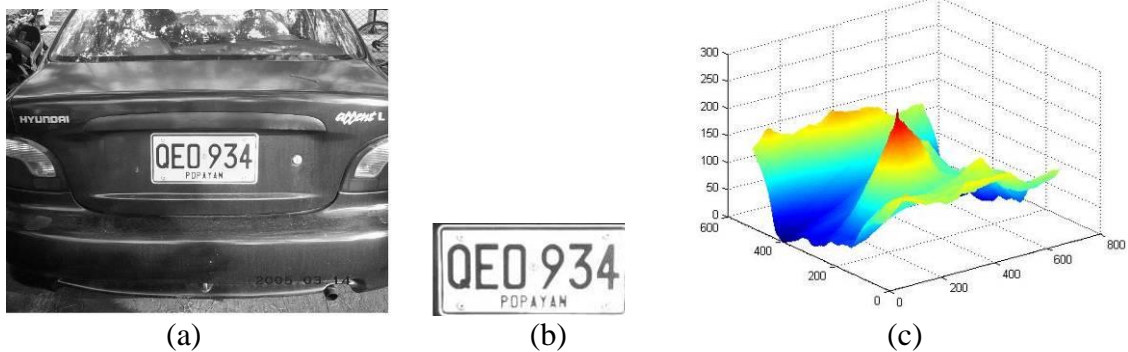


Figura 6.9. (a) Imagen de entrada, (b), plantilla y (c) gráfica de correlación

6.6.3 Máximos - Mínimos

Este método desarrollado por el estudio de este trabajo, consiste en hacer una búsqueda de la mínima diferencia entre las plantillas y la imagen del caracter de entrada. El caracter de salida se obtiene del mínimo valor encontrado.

Este método se elaboró para esta aplicación en especial, por lo tanto se explicará la forma en que se utilizó.

Máximos – Mínimos calcula la mínima diferencia entre una imagen f y unas plantillas $T_1, T_2, T_3, \dots, T_m$, donde m es el número de clases representados por las plantillas, dependiendo de que tan pequeño sea esta diferencia así mismo es la similaridad entre la imagen de entrada y la platilla. Esta diferencia se calcula de la siguiente manera:

Se calcula el mínimo y el máximo entre la imagen de entrada y cada plantilla

$$A = \min(f, T_i)$$

$$O = \max(f, T_i)$$

Luego se calcula la mínima diferencia para cada plantilla

$$D = O - A$$

Por la forma de la construcción de las plantillas, sección 6.6.1, se tiene que si todas las muestras utilizadas en la construcción de la plantilla coinciden entonces se tiene el valor de 1, si 9 coinciden se tiene un valor de 0.9 y así sucesivamente. Por esta razón se eleva al cubo la diferencia, para disminuir el peso de los valores que no coinciden totalmente, lográndose obtener mejores resultados.

$$S = \sum_x \sum_y D^3_{xy}$$

y el caracter de salida es el mínimo S , de las m salidas

$$C = \min(S_i)$$

En la figura 6.10, se puede observar el proceso de esta búsqueda, (a) caracter de entrada, (b) plantilla, (c) mínimo entre la imagen y la platilla, (d) máximo entre la imagen y la plantilla, (e), diferencia y (f) diferencia elevada al cubo. Con las restantes plantillas se va a tener una diferencia más grande, figura 6.11.

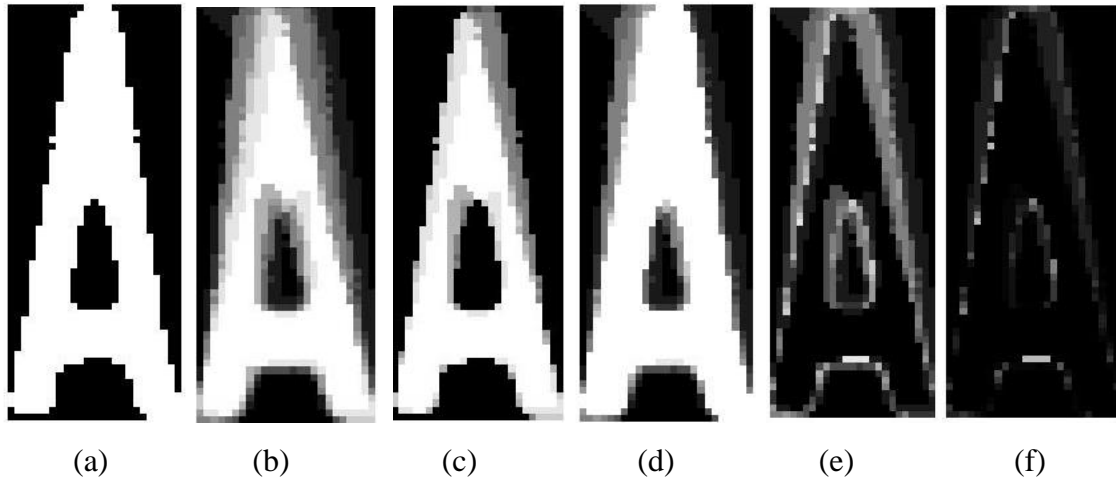


Figura 6.10. Búsqueda de la mínima diferencia

En la figura 6.11, se observa el proceso de selección utilizando Máximos - Mínimos, tomando como entrada la letra 'A', donde fácilmente se puede notar que la primera plantilla logra la mínima diferencia.



Figura.6.11 Selección utilizando Máximos - mínimos

6.7 ARTMAP MODIFICADA

En esta sección se plantea una modificación al algoritmo de la red neuronal Fuzzy ARTMAP en base del desarrollo de este trabajo, sección 6.2. Donde el objetivo es minimizar los tiempos de ejecución, principalmente en la etapa de entrenamiento.

Con este planteamiento se elimina el código de complemento, el cual hace que la red Fuzzy ARTMAP trabaje con vectores de dos veces la longitud del vector de entrada.

Fuzzy ARTMAP hace una búsqueda de la categoría en *ARTa* de la siguiente forma:

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}$$

El nodo ganador es el máximo T_j , el cual se compara para ver si cumple con el porcentaje mínimo de semejanza, si no cumple se inhibe el nodo y se hace una nueva búsqueda. En el momento en que la categoría seleccionada cumpla con la semejanza se envía el nodo ganador al mapa de campo, para comprobar si el nodo correspondiente en *ARTb* también cumple con los requerimientos de semejanza, si no es así se inhibe el nodo y se hace una nueva búsqueda en *ARTa*.

En la etapa de entrenamiento cuando se tiene un número considerable de clases a clasificar, es posible que cada que se ingrese una nueva clase, se pase por una gran cantidad de nodos antes de que se ocupe un nodo libre para crear la nueva categoría.

Entonces a partir de esto lo que se hace es entregar la categoría que mejor represente al vector de entrada y si esta categoría no cumple con la semejanza se inhiben todos los nodos ocupados siguientes para crear una nueva categoría, esto conlleva a eliminar el *seguimiento de comparación*, figura 6.12, y a crear un nuevo parámetro η que lleva la información del siguiente nodo libre.

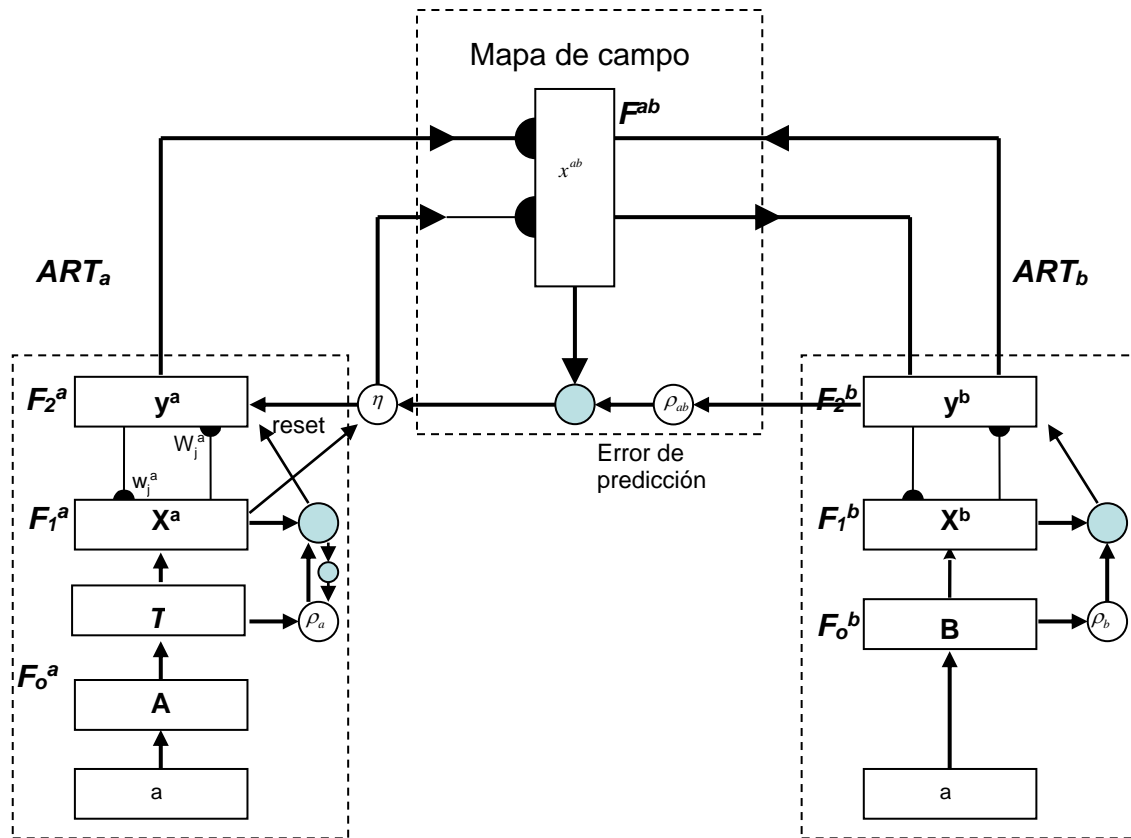


Figura 6.2 Arquitectura de Fuzzy ARTMAP modificada

Para encontrar la categoría que mejor represente al vector de entrada, se reemplaza el método de selección por:

$$T_j(I) = \left| (I \vee w_j) - (I \wedge w_j) \right|$$

donde el nodo ganador J es

$$T_J = \min \{ T_j : j = 1, \dots, N \}$$

Para hacer la relación de semejanza se procede de la misma forma que en ARTMAP, pero en el caso de que no cumpla con la semejanza se envía una señal a ART_a y al mapa de campo del siguiente nodo libre, figura 6.12, para crear una nueva categoría.

Esto evita el hacer la búsqueda por todos los nodos ocupados, que es lo que más tiempo emplea en la etapa de entrenamiento, además de reducir la longitud del vector de entrada a la mitad que hace que se reduzca el tiempo en la etapa de prueba.

7 PRUEBAS

7.1 INTRODUCCIÓN

Se realizaron pruebas para cada etapa, desde la localización de la placa hasta la identificación de caracteres. Primero se hizo una evaluación independiente para cada etapa, es decir, por ejemplo para las pruebas de segmentación se toma como entrada sólo las placas que resultaron localizadas.

Se ilustra en tablas la eficiencia y el tiempo de cada método, cada uno con sus respectivas ventajas y desventajas y al final de cada etapa se realiza una comparación de todos los métodos.

Después de observar los resultados de cada etapa, se concluye con la evaluación del sistema completo, teniendo en cuenta los métodos que presentaron mayor eficiencia.

7.2 PRUEBAS DE TÉCNICAS DE LOCALIZACIÓN

Cada método de localización presenta diferencias en la forma de buscar la placa, por esta razón, si un método tiene problemas con alguna característica de la imagen de entrada, que lleve a no poder detectarla, es posible que con otro método se tenga éxito.

También los métodos presentan diferencia en los tiempos de procesamiento. Como este tipo de aplicación exige un rápido resultado, se podría decir que algún(os) método(s) no cumple(n) con este requerimiento.

Si la imagen de entrada cumple con las especificaciones que exige cada método, se puede obtener con cualquiera de ellos una precisión del 100%, pero esas especificaciones son difíciles de cumplir para una aplicación real, por lo cual las pruebas se hacen con imágenes tomadas aleatoriamente, y son las mismas imágenes para todos los métodos, excepto para el método de detección por color donde no se hace pruebas con algunas imágenes, lo cual se explica en la sección 7.2.5.

7.2.1 Descripción de los datos de entrada

Se tomaron fotografías de vehículos desde diferentes distancias, bajo diferentes condiciones de iluminación y como fueron tomadas manualmente se tiene variación en el ángulo de inclinación de la placa.

Otro aspecto importante es que algunas imágenes obtenidas presentan brillo debido a los árboles que se encuentran cerca y generan sombras, esto ocasiona problemas en algunos métodos de localización.

El deterioro de la placa también ocasiona problemas con algunos métodos de localización que se acumulan y afectan las etapas siguientes.

Se tomaron dos conjuntos de imágenes, 80 imágenes para el entrenamiento y 90 para las pruebas, ambos fueron tomadas con las anteriores características.

7.2.2 Transformada de Hough

Con este método se logró un porcentaje de éxito de 85%, el cual no es un buen resultado. Esto es porque es muy dependiente del tamaño de la placa con respecto a la imagen total. Para obtener un buen resultado es necesario que en la imagen de entrada esté únicamente la parte delantera del vehículo.

Se hicieron pruebas con 90 imágenes, se logró la detección correcta en 77 imágenes y en 13 imágenes no se logró obtener un buen resultado, en 4 casos se la detectó pero recortando un carácter, contándose como error de detección.

El porcentaje de error se dio entre imágenes con demasiado ángulo de inclinación y bordes de la placa deteriorados. En este caso se presentaron 4 imágenes de taxis blancos los cuales tienen placa blanca, sin marco negro, por lo cual no se logró obtener bordes, presentando error de localización en las 4 imágenes.

Como las pruebas se hicieron con las mismas imágenes para todos los métodos, entonces hay imágenes donde la región de la placa es muy pequeña, por lo cual las líneas que se están buscando son muy cortas con respecto a todas las líneas encontradas,

por esta razón, para este método se asume que la placa está en el centro de la imagen y se hace un recorte de todos los lados a la imagen, para intentar tener únicamente la parte delantera del vehículo.

Si en una aplicación real es posible enfocar la parte delantera del vehículo, con este método se obtendrían muy buenos resultados, sin depender del color de la placa ni del vehículo

Éste método emplea un largo tiempo en entregar el resultado, 2.88 segundos, por lo cual es un problema para este tipo de aplicación.

La mayor cantidad de tiempo se emplea en el acondicionamiento de la imagen, por lo que vale la pena hacer una ilustración, figura 7.1. El detector de bordes de Canny es lento y es donde se gasta el mayor tiempo, pero con este algoritmo se lograron mejores resultados.

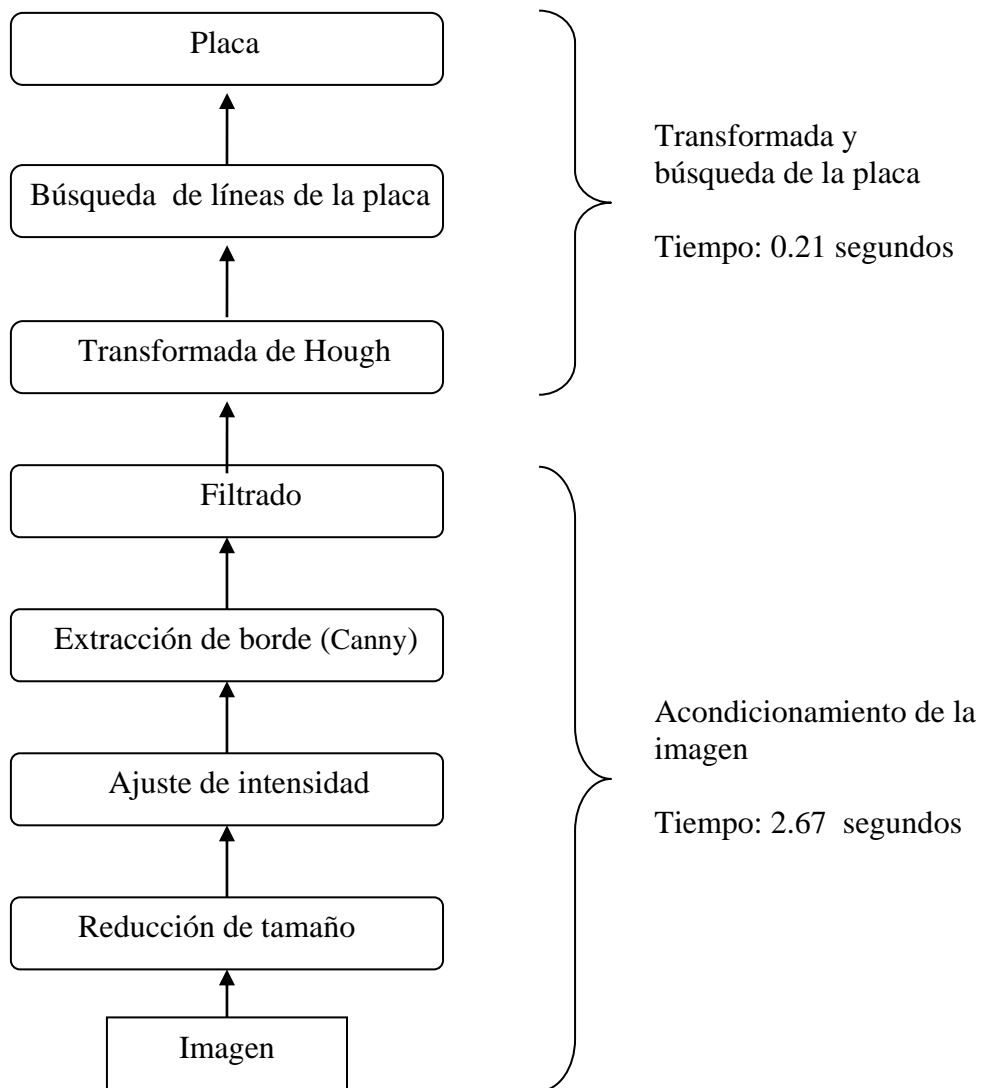


Figura 7.1. Pasos empleados para la localización de la placa con la transformada de Hough y tiempo empleado

Ventajas: La principal ventaja es que se puede localizar la placa sin importar el tipo de vehículo¹, además no depende de la rotación de la placa, se puede encontrar justamente el rectángulo que conforma la placa y puede variar en escala (hasta cierta distancia).

Desventajas: emplea demasiado tiempo para localizar la placa y depende demasiado de la imagen de entrada y del deterioro de la placa.

1. El color de la placa varía de un carro particular a uno de servicio público

7.2.3 Método del Gradiente

El método del gradiente depende mucho de la calidad de la imagen en cuanto a brillo y sombras, debido a que los errores que se presentaron en las pruebas corresponden a las imágenes en las cuales el vehículo se encuentra bajo árboles en un día soleado (como se ilustra en el parabrisas del vehículo de la figura 7.2), lo cual genera un aumento en bordes conllevando a localización de zonas falsas (las zonas falsas son a causa de que el patrón buscado en la imagen, sección 3.3, corresponde a un “patrón falso”).



Figura 7.2. Sombras a causa de árboles

El porcentaje de éxito por el método del gradiente fue de 88 %, las pruebas se realizaron con 90 imágenes de las cuales localizó perfectamente 80 imágenes.

El porcentaje de éxito aumenta para imágenes no tan complejas, como lo es en ambientes sin sol intenso y sin sombras muy marcadas, el porcentaje de éxito para este caso es de 95%.

Para imágenes tomadas a una distancia menor, (figura 7.2.3) el porcentaje de éxito es de 100% de éxito. Este porcentaje es alcanzado por las siguientes razones:

- La búsqueda de zonas disminuye.
- Las sombras como se ilustra en la figura 7.3 (b), ya no conllevan a elegir un patrón falso, porque el patrón buscado está bien definido.



(a)

(b)

Figura 7.3. Imágenes tomadas a menor distancia

El tiempo de ejecución de éste método fue de 0.99 segundos, el cual se puede considerar como bueno.

Ventajas. Este método presenta buen resultado en el tiempo de ejecución.

Desventajas. El porcentaje de error aumenta para imágenes bajo sobras muy marcadas generadas por las ramas de árboles en días con sol intenso.

7.2.4 Método Morfológico

El método morfológico depende mucho de las variaciones de distancia, debido que el algoritmo se basa en la altura de los caracteres, por esta razón el porcentaje de éxito para este método fue de 83 %.

Aunque el método funciona bien para distancias fijas, puede haber un pequeño rango de variación, pero para aumentar su precisión se aumentó este rango de variación de distancia, creando dos algoritmos uno para distancias grandes y otro para distancias cortas, mejorando el porcentaje de éxito a 92%

Las pruebas se realizaron con 90 imágenes de las cuales 75 fueron correctas para el caso de un solo algoritmo y 83 correctas para el caso de dos algoritmos.

El tiempo de ejecución de este método con los dos algoritmos es de 2.7 segundos, el cual para este tipo de aplicaciones se puede considerar como grande.

Ventajas La mayor ventaja de este método es la precisión de la localización, es decir el recorte de la placa es muy preciso, lo cual ayuda a la etapa de segmentación.

Desventaja Una desventaja de este método es que posee corto rango de variación de distancia, y aunque se desarrollaron dos algoritmos el método sigue teniendo errores con algunas imágenes tomadas a una distancia muy grande. Otra desventaja es que el tiempo de ejecución es grande, y este aumenta si se adicionan algoritmos para suplir el problema de la poca variación de distancia.

7.2.5 Color

Este método de detección presento un porcentaje de éxito del 100%, además soporta una gran variación en la escala o rotación de la placa y tampoco es afectado por el deterioro de la placa.

Como con este método lo que se busca es un color específico (color de la placa) en la imagen de entrada, entonces este método puede únicamente buscar placas de un determinado color (blanca o amarilla), por lo cual no se puede distinguir la placa de un carro particular y de uno público con el mismo código. En el caso de los taxis, esto se puede corregir haciendo primero una medida de qué tanto color amarillo hay en la imagen, para luego saber si se busca una placa de color amarillo o de color blanco. Esto nos lleva al problema de aumentar el tiempo de ejecución.

Pero el mayor problema que presenta este método es si el vehículo es del mismo color de la placa, en este caso no se puede (o sería difícil) encontrar una forma de corregir este problema.

Las pruebas solo se hicieron con 56 imágenes de carros particulares, por lo que este método es incapaz de detectar placas del mismo color del vehiculo, y no es fácil conseguir bastantes imágenes de taxis para entrenamiento (imágenes con las cuales se hace el ajuste de parámetros) y pruebas.

Este método presenta un tiempo de ejecución de 1.78 segundos que es relativamente corto, Aquí se esta tomando en cuenta el tiempo de lectura de la imagen desde el disco duro, lo cual es un tiempo de aproximadamente 0.8 segundos.

Ventajas: La mayor ventaja es que se obtiene una alta precisión, además encuentra placas de imágenes donde la placa es una región muy pequeña con relación a la imagen completa, tampoco depende de variaciones en el ángulo rotación ni de la calidad de la placa.

Desventajas: dependencia del color de la placa, en algunos casos se puede obtener un área mayor a la de la placa y no se puede detectar placas que sean del mismo color del vehículo.

7.2.6 Comparación de resultados de los métodos de localización

En la siguiente tabla 7.1 se ilustra el resultado de todos los métodos, estos resultados pueden ser mejorados enfocando la parte delantera del vehículo.

Método de detección	Porcentaje de éxito (%)	Tiempo (s)
Transformada de Hough	85	2.88
Método del Gradiente	88 ²	0.99
Método Morfológico	92	2.7
Color	100³	1.78

Tabla 7.1. Porcentaje de éxito de los métodos de detección

En el caso de una aplicación en la cual se esté haciendo el control de vehículos particulares, con el método de detección por color se obtendrán muy buenos resultados.

2. Hay que tener en cuenta que teniendo una distancia mas corta entre la cámara y el vehículo el resultado de éxito es mucho mejor.

3. Este resultado se obtuvo con pruebas de imágenes de carros particulares, tampoco se presentaron casos de vehículos amarillos.

En una aplicación donde no se generen sombras y la distancia entre la cámara y el vehículo no sean grandes, podría ser utilizado el método del Gradiente.

En donde se tenga una distancia fija o poca distancia entre el vehículo y la cámara, se puede utilizar la Transformada de Hough o el método Morfológico

Otro aspecto importante en tener en cuenta es el tiempo de ejecución. De la tabla 7.1 se puede observar que con el método del gradiente se tiene un muy buen tiempo con respecto a los otros métodos.

7.3 PRUEBAS DE SEGMENTACIÓN

Los resultados de la segmentación de los caracteres dependen por una parte de la exactitud de la localización, es decir que no se recorte o no se obtenga un área sobrante grande, por otra parte influye también la umbralización y el ángulo de inclinación.

Aunque en la mayoría de imágenes se obtiene buenos resultados de umbralización, hay factores que influyen en contra de esto como: deterioro de la placa, manchas, tornillos muy visibles, demasiadas sombras o imágenes muy oscuras. También si se tiene un ángulo de inclinación relativamente alto, aunque se lo intente corregir se presentan deformaciones al rotar la imagen, porque es complicado obtener el ángulo exacto de inclinación vertical y horizontal.

7.3.1 Descripción de los datos de entrada

Las entradas para las pruebas de este módulo fueron el resultado de la etapa de localización, sin tener en cuenta las placas no detectadas o recortadas. Se aceptan entradas con áreas sobrantes para probar la robustez de los métodos.

Para las pruebas de esta etapa se utilizaron las 56 placas localizadas por el método de color.

7.3.2 Análisis de componentes conectados

El porcentaje de éxito de este método fue de 100 %, para el método de localización por color, el cual entrega la imagen de la placa sin demasiados bordes, con los otros métodos de localización en los cuales la placa es entregada con regiones grandes aparte de la placa, este método no presenta buenos resultados

Aunque en las 56 imágenes de prueba se obtuvo una perfecta segmentación, este método puede borrar caracteres en el caso de que alguno esté separado (borrado en algún sector y el método de umbralización lo separe en dos o mas partes). Se borra el caracter, porque se toma como ruido al ser muy pequeño con respecto a los demás, este error se presenta en la etapa de borrado de ruido.

Si en el borrado de ruido, no se logra eliminar alguna línea del marco o algún ruido conectado al caracter, entonces en la separación se toma como caracter todo el componente, lo cual lo degenera y es muy difícil que los algoritmos de reconocimiento logren obtener buen resultado.

El tiempo empleado por ese método, teniendo en cuenta el tiempo de eliminación de ruido, fue de 0.81 segundo, y puede variar de acuerdo a que tanto ruido se tenga después de la umbralización.

Ventajas: Si se logra una buena umbralización, lo cual ocurre en la mayoría de los casos, se obtiene una rápida y precisa separación de caracteres y se pueden eliminar con facilidad pequeñas partículas que estén separadas de los caracteres

Desventajas: Demasiada dependencia de la umbralización⁴ y posibilidad de borrar uno o más caracteres y dejar en lugar de eso el ruido reemplazando al caracter borrado.

4. El problema se da cuando el ruido o el marco de la placa se une a uno o mas caracteres, los caracteres están unidos entre sí y/o los caracteres están demasiado borrados.

7.3.3 Relación de distancia

Con éste método se obtuvo un porcentaje de éxito de 92.5 %, el cual es bueno, y se tiene la ventaja de no eliminar caracteres pequeños a los cuales se les ha borrado parte de ellos en la umbralización.

Los errores presentes en este método son debidos al mal cálculo de la longitud de la placa como se explico en la sección 4.3. También este método presenta errores en el momento de eliminar el ruido grande de la imagen, como son tornillos o manchas, puesto a que al eliminar este ruido se corre el riesgo de eliminar algún carácter que esté demasiado borrado o esté muy pequeño.

Para imágenes que tengan poca variación de tamaño, el éxito de separación es de 96%.

El tiempo que gasta desde la umbralización, corte de caracteres y eliminación de ruido fue de 0.65 segundos, el cual es un tiempo bueno.

Ventaja: Tiempo de ejecución bajo

Desventajas: Para imágenes que poseen mucho ruido hay un mal cálculo de la longitud de la placa lo cual genera una mala precisión del recorte del carácter.

7.3.4 Picos y valles

El porcentaje de éxito obtenido por este método fue de 66.6%, el cual es muy pobre.

Este método es el más deficiente debido a que, además de depender de la calidad de la imagen, depende mucho de que los caracteres no estén inclinados, sección 4.4, es decir el método no es capaz de separar correctamente imágenes con caracteres inclinados.

El tiempo que de ejecución fue de 0.8 segundos, el cual es relativamente corto.

Desventajas: El método no es capaz de segmentar imágenes con caracteres inclinados y depende mucho de la calidad de la imagen en cuanto al contenido de ruido.

7.3.5 Comparación de resultados de los métodos de segmentación

La mayoría del porcentaje de error en cada método se presentó en las mismas imágenes, las cuales tienen un problema en localización, umbralización o corrección de ángulo, otro factor que influye es que en algunos casos no se puede separar el ruido del carácter, los resultados de éstos métodos se ilustran en la tabla 7.2.

Método de segmentación	Porcentaje de éxito	Tiempos (s)
Componentes conectados	100	1.10
Relación de distancia	92	0.65
Picos y Valles	66.6	0.8

Tabla 7.2. Comparación de los métodos de segmentación

Como se puede ver en la anterior tabla la separación por análisis de componentes conectados es muy eficiente.

Aunque el método de relación de distancia presenta un tiempo menor al de componentes conectados, es más importante la diferencia de resultados de éxito en la separación entre estos métodos.

7.4 PRUEBAS DE LAS EXTRACCIÓN DE CARACTERÍSTICAS

En esta etapa no se pueden obtener resultados de la eficiencia de cada método extractor de características. Pero los resultados de los algoritmos de reconocimiento dependen en gran parte de qué tan bien represente un vector de características a los caracteres.

Entre más pequeño sea el vector de características, el algoritmo de reconocimiento será más rápido, pero puede tener resultados desfavorables si no se logra representar en el vector adecuadamente las características de cada carácter.

Este vector también depende de la calidad de la imagen de entrada. En imágenes de caracteres que presenten uniformidad (sin problemas de ruido, rotación, escala o

cualquier problema que deforme el caracter), es fácil que cualquier algoritmo de extracción de características represente adecuadamente los diferentes caracteres sin tener problemas en el reconocimiento de estos, pero en este caso, donde cada imagen de un mismo caracter es diferente, puede ser difícil lograr que un vector represente a todas las imágenes.

Aunque en algunos vectores de características no importa la variación de escala, se prefirió normalizar todas las imágenes a un tamaño de 60x25.

En la etapa de reconocimiento de caracteres se hacen pruebas con los diferentes vectores de características, donde se observa la variación de los resultados en el mismo algoritmo de reconocimiento.

7.5 PRUEBAS DE TÉCNICAS DE RECONOCIMIENTO

Obtener buenos resultados en esta etapa depende de el vector de características de entrada, si el vector de características logra representar correctamente a cada caracter, cualquiera de los algoritmos implementados debería obtener buenos resultados.

En algunos algoritmos de reconocimiento, se toma como característica directamente los píxeles de la imagen del caracter, estos algoritmos son más lentos para entregar los resultados, y además, sólo permiten rotaciones en la imagen que se hayan incluido en la etapa de entrenamiento.

Los principales problemas que se presentan, se dan por las deformaciones de los caracteres en las imágenes, por lo cual el algoritmo de extracción de características no logra representar correctamente algunas imágenes.

Para evaluar esta etapa, se toma en cuenta cada caracter, no los seis caracteres de la placa, así el porcentaje de éxito y el tiempo empleado se presenta por caracter. En el sistema integrado se evalúa por placa, es decir si hay un caracter errado entre los seis de una misma placa, se toma como error toda la placa.

7.5.1 Descripción de los datos de entrada

Algunos algoritmos de reconocimiento de caracteres soportan cualquier vector de características (capítulo 5) incluyendo la imagen completa de cada caracter, pero algunos no soportan vectores de entrada muy grandes y otros están diseñados para comparar la imagen completa.

Se toman como entrada las imágenes que no presentaron problemas de segmentación, para evaluar esta etapa independiente, como se ha hecho con las anteriores.

Para la etapa de entrenamiento de los algoritmos, se utilizaron diez patrones por cada clase, hay algunas letras que no son comunes en Popayán, como lo son: O, P, R, S, T, V, W, X, Y, Z, por lo cual se tienen menos muestras para estos caracteres, tabla 7.3. Para las pruebas se utilizaron 328 imágenes de caracteres distintos a los de la etapa de entrenamiento.

Nro Clase	1	2	3	4	5	6	7	8	9	10	11	12	13
Caracter	A	B	C	D	E	F	G	H	I	J	K	L	M
Muestras/clase	10	10	10	10	10	10	10	10	10	10	10	10	10

Nro Clase	14	15	16	17	18	19	20	21	22	23	24	25	26
Caracter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Muestras/clase	10	5	4	10	4	4	2	10	3	2	2	3	2

Tabla 7.3. Número de muestras por clase utilizadas en la etapa de entrenamiento

7.5.2 Fuzzy ARTMAP

Se realizaron diferentes pruebas, variando el parámetro de vigilancia ρ y variando el número de patrones, además se realizaron pruebas con distintos vectores de características. Todas las pruebas realizadas se hicieron con una época de entrenamiento.

El número de categorías que crea *ARTa* depende del parámetro de vigilancia, por lo tanto varían los resultados. En la tabla 7.20, se pueden observar los resultados obtenidos utilizando como entradas el valor de cada píxel con diferentes valores de ρ , en la etapa

de entrenamiento para las letras se utilizaron las muestras dadas en la tabla 7.3 y para los números se utilizaron diez muestras por cada uno.

Valores de ρ	Porcentaje de éxito (%)
0.90	(301/328) 91.76
0.94	(325/328) 99.08
0.98	(327/328) 99.69

Tabla 7.4. Resultados obtenidos con diferentes valores de ρ utilizando como entradas el valor de cada píxel.

Los resultados obtenidos son muy dependientes de ρ , como se observa en la figura 7.4, donde se logra el mejor resultado con $\rho = 0.98$, considerándose este resultado como aceptable por lo que si se aumenta ρ se incrementa el número de categorías, lo cual hace mas lento el funcionamiento, por lo tanto en las pruebas siguientes, utilizando otros vectores de características, se hicieron únicamente para este valor de ρ .

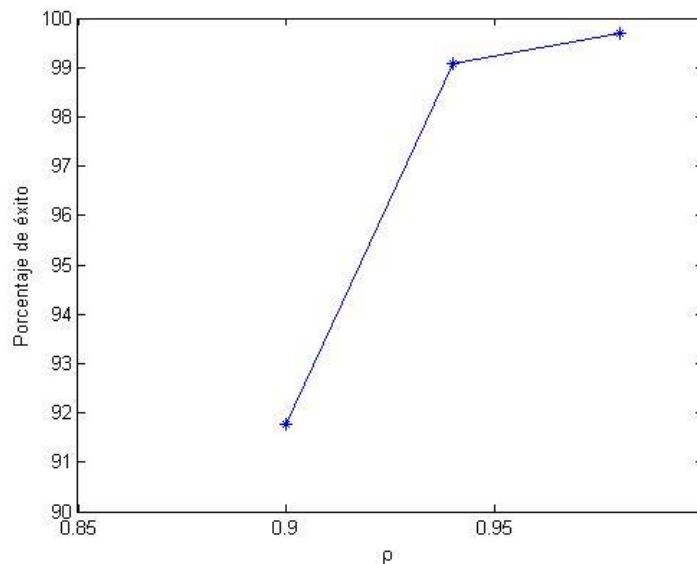


Figura 7.4. Variación del porcentaje de éxito respecto a ρ

Los resultados también son dependientes del número de patrones de entrada para representar a cada categoría en la etapa de entrenamiento, esto se puede observar en la tabla 7.5.

Numero de patrones por cada clase	Porcentaje de éxito (%)
3	(318/328) 95.79
7	(321/328) 97.86
10	(327/328) 99.69

Tabla 7.5. Resultados obtenidos variando el numero de patrones de entrada en la etapa de entrenamiento.

Si el número de patrones de entrada en la etapa de entrenamiento es alto, entonces se crean más categorías en *ARTa*, para representar a una misma clase, por lo cual se logran mejores resultados entre más patrones se tengan en la etapa de entrenamiento, figura 7.5.

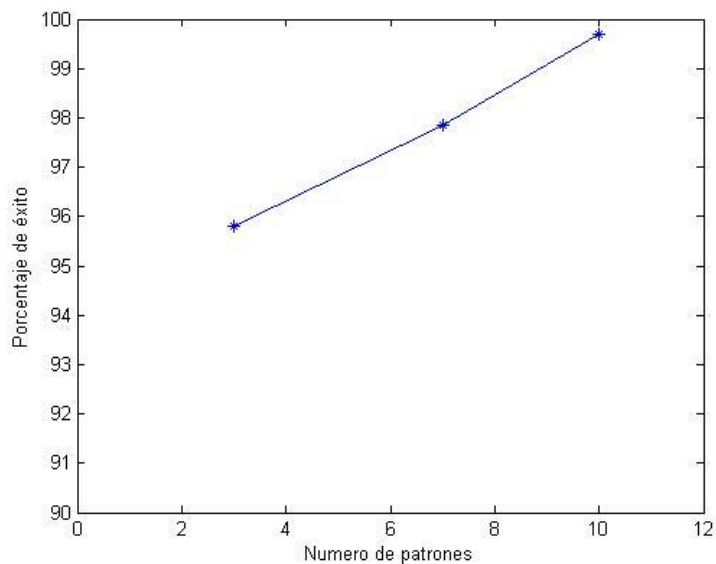


Figura 7.5. Variación del porcentaje de éxito respecto la número de patrones de entrenamiento

Los resultados también dependen notablemente del vector de características utilizado, como se muestra en la tabla 7.6, estos resultados fueron obtenidos con $\rho = 0.98$ y con la misma cantidad de patrones de entrenamiento para cada clase.

Vector de características	Longitud del vector	Porcentaje de éxito (%)
Zonificación	30	(328/328) 100
Área	90	(323/328) 98.47
Momentos	7	(285/328) 86.89
RITRE	14	(246/328) 75.00

Tabla 7.6. Resultados obtenidos con diferentes vectores de características

En la tabla 7.6 se puede observar que utilizando el método de zonificación para obtener el vector de características, se logró un correcto reconocimiento para los 328 caracteres, por lo que para este caso se realizaron pruebas variando el número de patrones de entrada, tabla 7.7.

Número de patrones por cada clase	Porcentaje de éxito
4	(321/328) 97.86
7	(323/328) 98.47
10	(327/328) 99.69

Tabla 7.7. Resultados obtenidos variando el número de patrones de entrada en la etapa de entrenamiento, utilizando zonificación.

Utilizando como vector de características el obtenido por Zonificación, además de presentar buenos resultados, es más rápido, debido a que la longitud del vector es mucho menor que al utilizar el valor de cada píxel, en este caso se dividió la imagen del caracter en 30 zonas y en el caso del valor de cada píxel la longitud del vector es de 1500.

El tiempo de procesamiento empleado se muestra en la tabla 7.8. Este tiempo es dependiente de la longitud del vector de características.

Vector de características	Tiempo (s)
Valor de cada píxel	0.11
Otros vectores de características	0.015 - 0.020

Tabla 8. Tiempo empleado por Fuzzy ARTMAP.

Ventajas: se lograron buenos resultados con algunos vectores de características y bajo tiempo de procesamiento, se pueden presentar nuevos patrones sin borrar los ya almacenados.

Desventajas: algoritmo complejo, se necesita duplicar la longitud del vector de entrada debido al código de complemento.

7.5.3 Vecino Más Cercano Holográfico

Las pruebas realizadas con los vectores de características del capítulo 5 se ilustran en la tabla 7.9.

Vector de características	Porcentaje de éxito (%)	Tiempo (s.)
Valor de cada píxel	(327/328) 99.69	0.063
Área	(303/328) 92.37	0.0045
RITRE	(263/328) 80.18	0.0326
momentos	(201/328) 61.28	0.0044
Zonificación	(327/328) 99.69	0.0052

Tabla 7.9. Resultado de la pruebas con el método del Vecino Más Cercano Holográfico

El código RITRE es eficiente para el reconocimiento de caracteres invariante a la rotación, traslación y escalamiento pero sólo si las letras presentan uniformidad. Este método no es adecuado para reconocer caracteres de diferentes tipos, por ejemplo el vector de característica resultante de una letra “N” ancha es diferente al de una letra “N” delgada y estirada, y debido a que en los datos de entrada hay caracteres despintados, inclinados, estirados, etc, el porcentaje de éxito de este método es bajo.

Los momentos tienen un error más alto que el código RITRE, porque además de tener las mismas causas de error que tiene el código RITRE, los valores de los momentos son más parecidos entre letras distintas.

El tiempo de ejecución de este método, es muy bueno como se ilustra en la tabla 7.9

Ventajas: el algoritmo es sencillo y se tiene un buen resultado con los vectores de características de Zonificación y con el valor de cada píxel, el tiempo de ejecución es corto.

Desventajas: No tiene la capacidad de discriminar información repetida en la etapa de entrenamiento, por lo cual si se tiene muchas muestras por cada caracter, se obtendrán matrices demasiado grandes

7.5.4 Red Multicapa *feedforward*

Las pruebas realizadas se hicieron con vectores de características del capítulo 5, los resultados se ilustran en la tabla 7.10.

Para las pruebas se crearon dos Redes Multicapa *Feedforward*, una para las letras y otra para los números.

Vector de características	Porcentaje de éxito (%)	Tiempo (s)
RITRE	78.65	0.031
momentos	56.70	0.031
Zonificación	97.86	0.031

Tabla 7.10. Resultado de las pruebas con la Red Multicapa Feedforward

La red para letras utilizando el código RITRE consta de 14 neuronas para la capa de entrada, 25 para la capa oculta y 26 neuronas en la capa de salida, la red para números tiene la misma configuración, para las dos primeras capas y en la capa de salida tiene 10 neuronas.

La red para letras utilizando los momentos consta de 7 neuronas para la capa de entrada, 15 para la capa oculta y 26 neuronas en la capa de salida, la red para números tiene la

misma configuración para las dos primeras capas y en la capa de salida tiene 10 neuronas.

La red para letras utilizando el método de zonificación costa de 30 neuronas para la capa de entrada, 50 para la capa oculta y 26 neuronas en la capa de salida, la red para números tiene la misma configuración para las dos primeras capas y en la capa de salida tiene 10 neuronas.

Los resultados de éste método depende mucho del numero de neuronas de cada capa de la red, y la mejor configuración, de diez realizadas, se consiguió con las arquitecturas mencionadas.

Ventajas: El tiempo de ejecución es corto, además se puede reconocer los seis caracteres de la placa en una sola ejecución

Desventajas: Es difícil entrenarla con vectores muy grandes porque se requiere equipos con alta velocidad, el tiempo de entrenamiento es muy grande y no hay una regla para definir la configuración óptima de la red.

7.5.5 Clasificadores estadísticos óptimos

En este método es necesario calcular la inversa de la matriz de covarianza de las muestras, sección 6.5, lo cual no es posible en este proyecto⁵, esto se debe a que el determinante de la matriz es cercano o igual a cero, porque esta mal condicionada, es decir, el número de condición es muy grande.

Esto puede ser comprobado calculando el número de condición para todas las matrices el cual es mayor a 10^{30} , el número de condición de una matriz es definido como el producto de la norma de A por la norma de A^{-1} .

5. Para poder invertir la matriz se necesita que el número de muestras por cada clase sea mayor que la longitud del vector de entrada.

Como una alternativa a la medida de distancia, puede ser usada la distancia euclidiana, además, como se dijo anteriormente, el determinante de la matriz de covarianza es cercano a cero, y en este caso la probabilidad es igual para todas las entradas, por lo tanto se tiene:

$$d_j(x) = (x - m_j)^T (x - m_j)$$

Cuando se usa la distancia euclidiana, la inversa de la matriz es remplazada por la matriz identidad.

En la tabla 7.11. Se muestran los resultados que se obtuvieron con este método

Vector de características	Porcentaje de éxito (%)	Tiempo (seg)
Zonificación	(324/328) 98.78	0.007
Área	(313/328) 95.42	0.018
Momentos	(225/328) 68.59	0.003
RITRE	(274/328) 83.53	0.010

Tabla 7.11. Resultados obtenidos con clasificadores estadísticos óptimos

No se logran buenos resultados debido a que únicamente se está utilizando la media de cada carácter para hacer la comparación y selección. Este método presenta muy bajo tiempo de ejecución como se muestra en la tabla 7.11.

Ventajas: Bajo tiempo de ejecución.

Desventajas: necesita muchas muestras por cada clase para poder utilizar éste método correctamente.

7.5.6 Comparación de plantillas

Este método de reconocimiento de caracteres acepta únicamente la imagen de cada carácter como entrada (sección 6.6.1), por lo cual no es invariante a la rotación de la imagen. Aunque en la creación de cada plantilla, se utilizaron muestras de imágenes de caracteres con pequeños ángulos de rotación, el porcentaje de error se presentó debido a este problema y la gran semejanza entre la 'O' y la 'Q'.

En la tabla 7.12 se muestran los resultados de este método, los cuales presentan un alto porcentaje de precisión. Este resultado depende principalmente de la corrección del ángulo que se puede conseguir.

Método de selección	Porcentaje de éxito	Tiempo (s)
Correlación	(326/328) 99.39	0.051
Máximos - Mínimos	(326/328) 99.39	0.016

Tabla 7.12. Porcentaje de éxito y tiempo obtenido con el método de comparación de plantillas

El error que se presentó fue por imágenes de entrada bastante inclinadas y a una confusión entre la ‘O’ y la ‘Q’, el primer caso se ocurre, porque en la construcción de la plantilla no se incluyeron muestras de imágenes con estas características, sólo se tomaron en cuenta pequeños ángulos de rotación. Para corregir este problema se podría hacer más de una plantilla o patrón para representar a cada caracter. Por ejemplo, hacer tres patrones para cada caracter, la primera con la imagen del caracter con pequeños ángulos de rotación, la segunda con imágenes rotadas hacia la derecha con inclinaciones considerables, y la tercera igual que la segunda pero hacia la izquierda.

En el caso de la confusión entre la ‘O’ y la ‘Q’, al tener como salida cualquiera de estas dos letras, se podría hacer una comparación únicamente de la parte inferior derecha, que es donde esta la diferencia entre estas.

El tiempo de procesamiento empleado se muestra en la tabla 7.12. Es relativamente bajo porque el tamaño de la imagen es igual al de la plantilla.

Con los dos métodos de selección se logró obtener el mismo resultado de eficiencia, pero con el método de Máximos – Mínimos se consigue un procesamiento mas rápido.

Ventajas: es un algoritmo simple, se lograron buenos resultados y no necesita etapa de entrenamiento.

Desventajas: no es capaz de reconocer imágenes rotadas y solo acepta como entrada la imagen completa por lo cual podría ser lenta para imágenes grandes.

7.5.7 Fuzzy ARTMAP modificada

Con este método de reconocimiento se obtuvieron exactamente los mismos resultados de eficiencia que con la red Fuzzy ARTMAP, pero se logró reducir en gran cantidad el tiempo de procesamiento como se muestra a continuación.

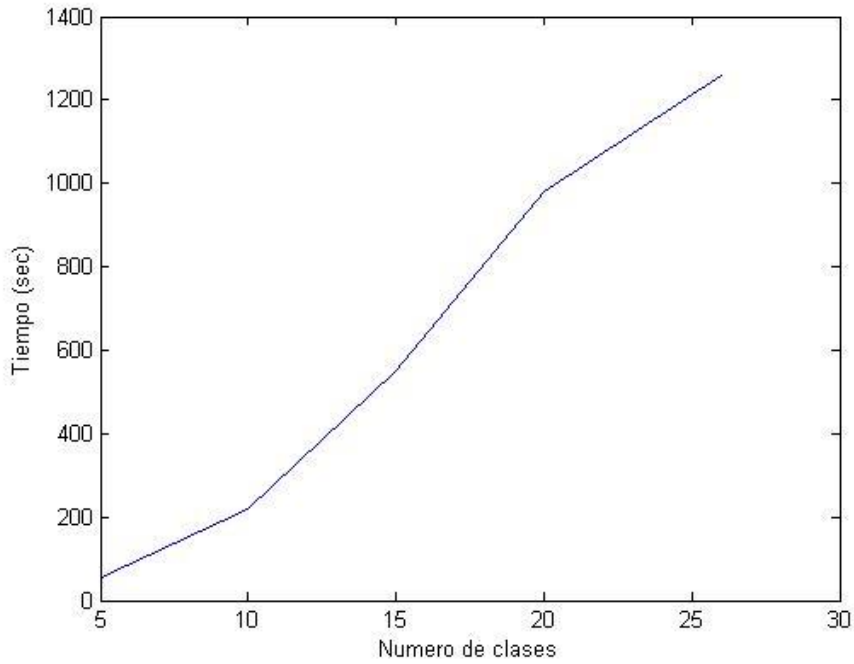
La comparación de los tiempos se realizó con los 26 caracteres que corresponden a las letras, el número de muestras por cada clase se muestra en la tabla 7.3.

El resultado de los tiempos empleados en la etapa de entrenamiento para una época se muestra en la tabla 7.13.

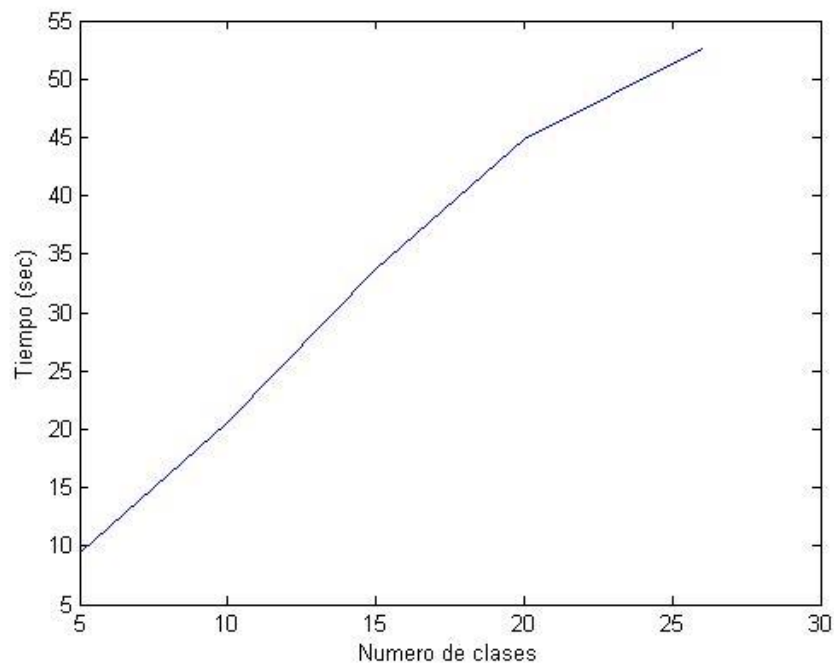
Numero de clases	Fuzzy ARTMAP (s)	Fuzzy ARTMAP modificada (s)
5	54.28	9.34
10	217.53	20.64
15	551.27	33.67
20	979.07	44.90
26	1260.32	52.53

Tabla 7.13. Comparación de tiempo entre Fuzzy ARTMAP y Fuzzy ARTMAP Modificada

En la figura 7.6 (a) se muestra que la pendiente de la recta de Fuzzy ARTMAP aumenta al aumentar el número de clases, mientras que en Fuzzy ARTMAP modificada, figura 7.6 (b), permanece constante (la disminución de la pendiente a partir de 20 clases se debe al número de muestras por clase que se muestra en la tabla 7.19).



(a)



(b)

Figura 7.6. Tiempo respecto al número de clases, (a) Fuzzy ARTMAP y (b) Fuzzy ARTMAP modificada

En la etapa de prueba Fuzzy ARTMAP modificada emplea 0.085 segundos, por cada caracter, utilizando como entrada el valor de cada píxel, contra 0.11 segundos empleado por Fuzzy ARTMAP. Esta reducción se debe a la eliminación del código de

complemento, donde la longitud del vector de entrada para Fuzzy ARTMAP modificada es de 1500 x 1 mientras que para Fuzzy ARTMAP es de 3000 x 1

Ventajas: Las mismas de Fuzzy ARTMAP, con disminución del tiempo de ejecución.

Desventajas: Las mismas de Fuzzy ARTMAP.

7.5.8 Comparación de resultados de los métodos de reconocimiento de caracteres

En la siguiente tabla 7.14 se ilustran los porcentajes más altos de cada método, la mayoría de estos resultados son obtenidos utilizando como vector de características obtenido por Zonificación.

Método de Reconocimiento	Porcentaje de éxito (%)	Tiempos (seg)
Fuzzy ARTMAP	100	0.015
Vecino Más Cercano Holográfico	99.69	0.0052
Red Multicapa Feedforward	97.86	0.031
Clasificadores estadísticos óptimos	98.78	0.003
Comparación de Plantillas	99.39	0.016
Fuzzy ARTMAP Modificada	100	0.085

Tabla 7.14 Comparación de los métodos de Reconocimiento

Como se puede ver, las redes Fuzzy ARTMAP presentan el mejor resultado, además se puede agregar nuevas muestras de las clases cuando el sistema esté en funcionamiento.

En esta aplicación cada que se reciben los seis caracteres de una placa estos son almacenados temporalmente, con el objetivo de que si se desea que la red aprenda un nuevo caracter con distintas características, se pueda almacenarlo indicándole a qué salida corresponde, para en la siguiente vez que se presente uno parecido lo pueda reconocer.

Se puede observar que los tiempos son cortos para que todos los métodos, por lo tanto lo más importante es el porcentaje de éxito.

Los métodos de reconocimiento no consiguen buenos resultados con los métodos de extracción de características invariantes a la rotación, escala y traslación como los momentos y la técnica RITRE. A continuación se presenta un ejemplo del error más común cometido en los números, utilizando los momentos.

Como se ilustra en la figura 7.7 (a), si se toma el cinco, se lo gira 180° y se lo refleja se obtiene una figura muy parecida a la de un 2, figura 7.7 (b). Puede ocurrir casos parecidos con otros caracteres.

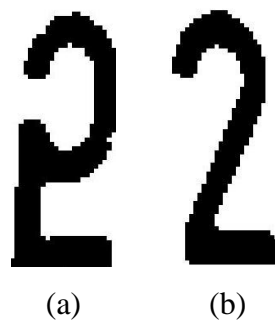


Figura 7.7. Confusión entre los números 2 y 5 utilizando los momentos

7.6 SISTEMA INTEGRADO

En la figura 7.8 se ilustra los métodos con los cuales se ha decidido realizar el sistema integrado a partir de las pruebas realizadas. Para la primera etapa, localización, se proponen dos métodos, para utilizarlos dependiendo de la aplicación, en las otras etapas sólo se tiene un método porque es el que ha presentado la mayor eficiencia.

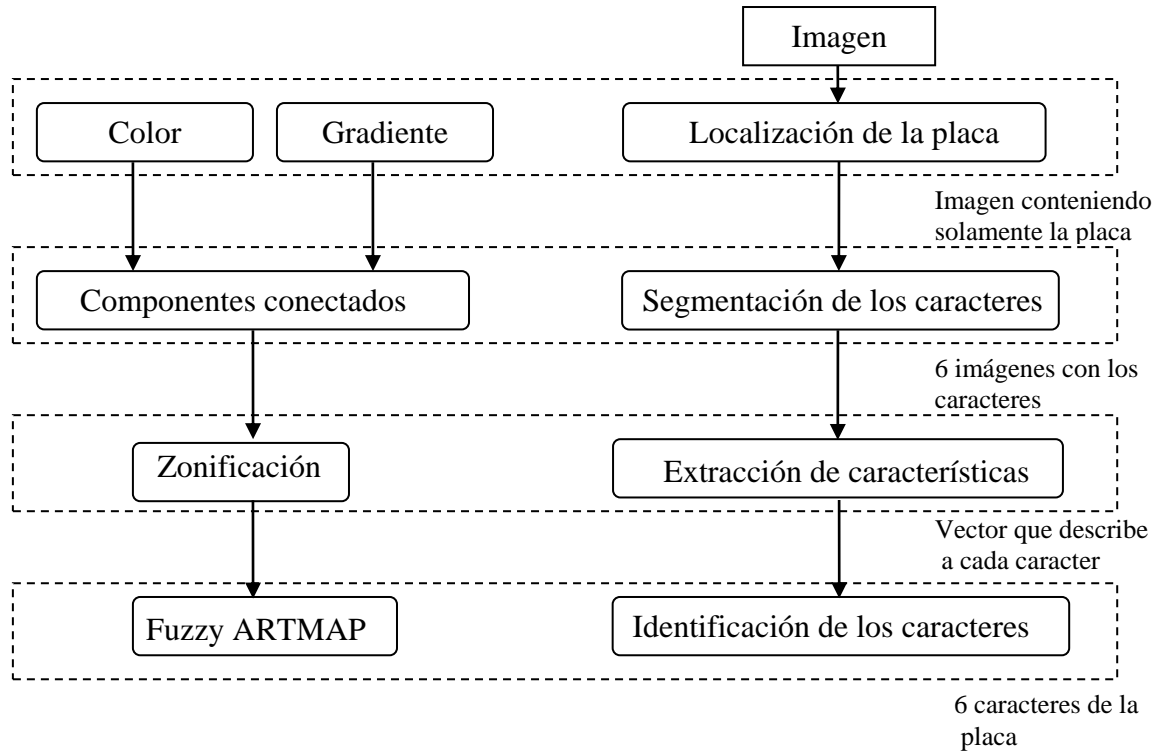


Figura 7.8. Sistema integrado

7.6.1 Resultados del sistema integrado

El porcentaje de carros amarillos particulares es bajo, aproximadamente del 2%⁶, para los cuales el método de detección por color no tendrá éxito, con esto se tiene los resultados de la tabla 7.35, teniendo en cuenta que la detección por color sólo es para aplicaciones que sólo tengan vehículos particulares.

Sistema integrado	Porcentaje de éxito	Tiempos (s)
Color	97.77	3.39
Gradiente	88	2.6

Tabla 7.15. Resultados del sistema integrado

6. este porcentaje se obtuvo de un conteo en 4 parqueaderos

En una aplicación real, si la imagen es tomada a una distancia corta, o si el vehículo pasa aproximadamente por el mismo sitio se puede recortar la imagen, suponiendo que la placa se encuentra en el centro, para mejorar la detección con el método del gradiente.

Para ilustrar de una mejor forma el funcionamiento de este proyecto se realizó una interfaz, la cual se explica a continuación:

La interfaz se desarrolló con la herramienta GUIDE de Matlab, se divide en dos, una contiene todos los métodos que se utilizaron en las pruebas y la segunda posee los métodos que tuvieron el mejor resultado, estas se explican a continuación, ver figura 7.9:

A continuación se ilustra la interfaz en la cual están todos los métodos de prueba:

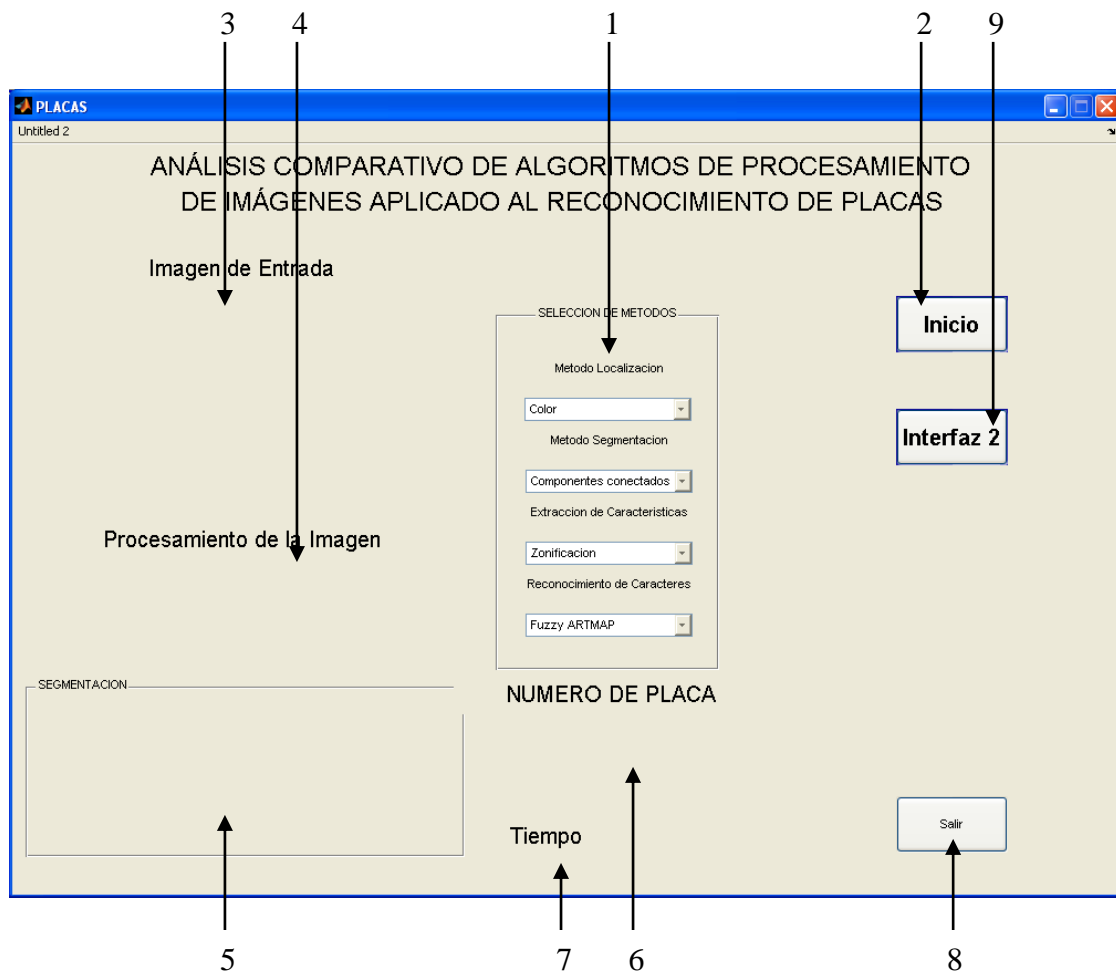


Figura 7.9. Interfaz con todos los métodos

1 Selección de métodos: contiene 4 pop-menú en los cuales se encuentran los métodos de localización, segmentación, extracción de características y reconocimiento de caracteres. Con un click del mouse se selecciona el método que se quiere ejecutar, esto permite al usuario observar el funcionamiento de un método con otro, es decir hacer una combinación entre todos los métodos.

2 Inicio: el botón inicio ejecuta los métodos que ya fueron seleccionados, en el caso de no seleccionar el método, el programa ejecuta por defecto los métodos de Color (localización), Componentes conectados (segmentación), Zonificación (extracción de características) y Fuzzy ARTMAP (reconocimiento de caracteres).

3 Imagen de Entrada: visualiza la imagen correspondiente al vehículo seleccionado aleatoriamente por el programa.

4 Procesamiento de la Imagen: visualiza el proceso que esta pasando la imagen de entrada: la localización y umbralización.

5 Segmentación: se visualiza el recorte de los seis caracteres de la placa.

6 Número de la placa: indica en el número de la placa que el sistema da por reconocido.

7 Tiempo: se indica el tiempo que tarda el sistema en reconocer una placa.

8 Salir: finaliza la aplicación

9 Interfaz 2: abre la aplicación en la cual se encuentran los métodos que tuvieron mejor resultado.

Interfaz 2

Los métodos seleccionados en esta interfaz son: color (método de localización), componentes conectados (método de segmentación), zonificación (método de extracción de características) y Fuzzy ARTMAP Modificada (método de reconocimiento). La interfaz se ilustra en la figura 7.10

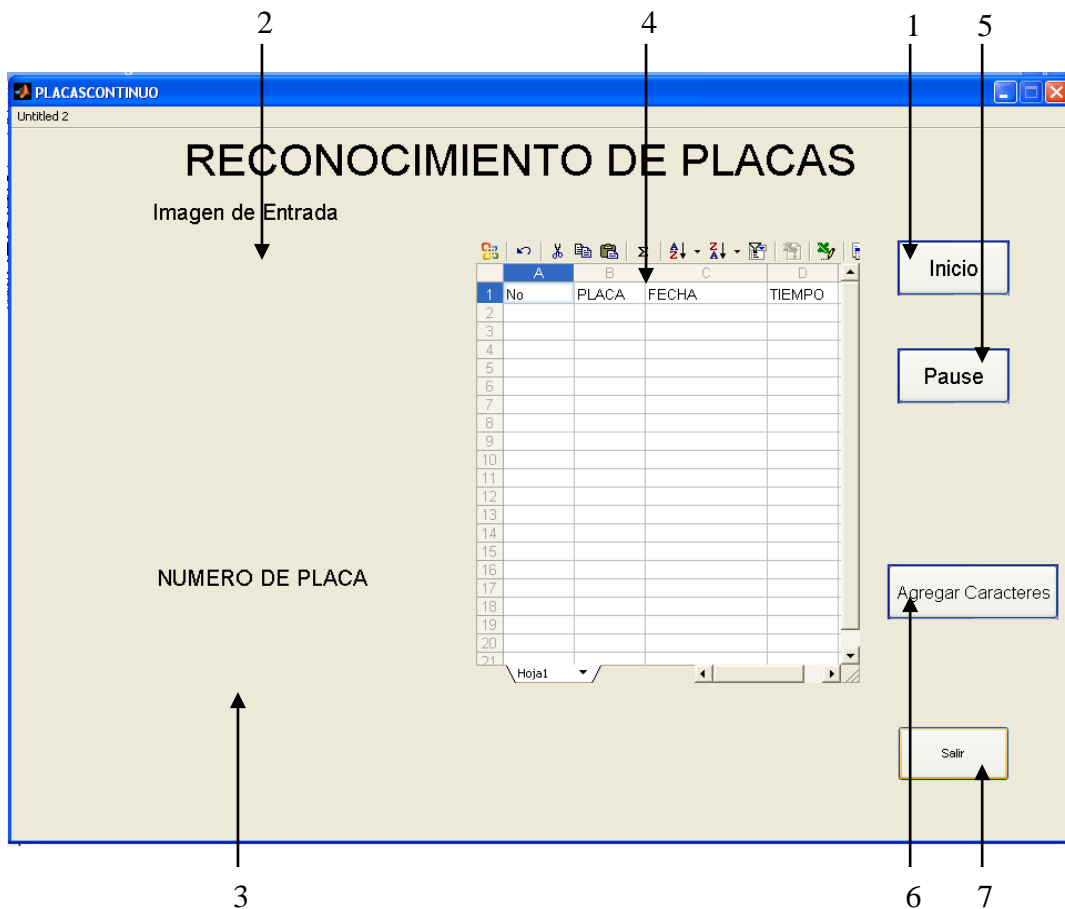


Figura 7.10 Interfaz con los métodos de mejor resultado

1 Inicio: ejecuta el programa con los métodos mencionados arriba, la selección de la imagen de entrada es de forma aleatoria, y el programa procesa imágenes continuamente.

2 Imagen de Estrada: visualiza la imagen correspondiente al vehículo seleccionado aleatoriamente por el programa.

3 Número de la placa: indica en el número de la placa que el sistema da por reconocido

4 No, Placa, Fecha y Tiempo: corresponde a una tabla de Excel en la cual se ilustra los siguientes datos: “No” indica el numero de vehículos que el sistema ha reconocido, “Placa” ilustra el numero de la placa, “Fecha” indica la fecha y el tiempo en la cual se

reconoció la placa, y “Tiempo” es el tiempo que se tarda el sistema en procesar una imagen. La tabla almacena estos datos sir ser remplazados.

5 Pause: detiene la aplicación pero al oprimir Enter el programa sigue donde estaba.

6 Salir: finaliza la aplicación

7 Agregar caracteres: corresponde a otra interfaz para agregar caracteres en el caso de que se produzca un error en el reconocimiento de algún carácter, esto se explica a continuación:

Agregar caracteres

Debido a que el método con el mejor resultado en la etapa de reconocimiento es Fuzzy ARTMAP Modificada es posible agregar caracteres en el caso de un error.

Fuzzy ARTMAP Modificada es capaz de almacenar y aprender nuevos caracteres que el usuario desee ingresar, por lo tanto el sistema irá obteniendo mayor robustez con el tiempo en que se vaya ejecutando y agregando caracteres que produzcan errores, la interfaz se ilustra en la figura 7.11.

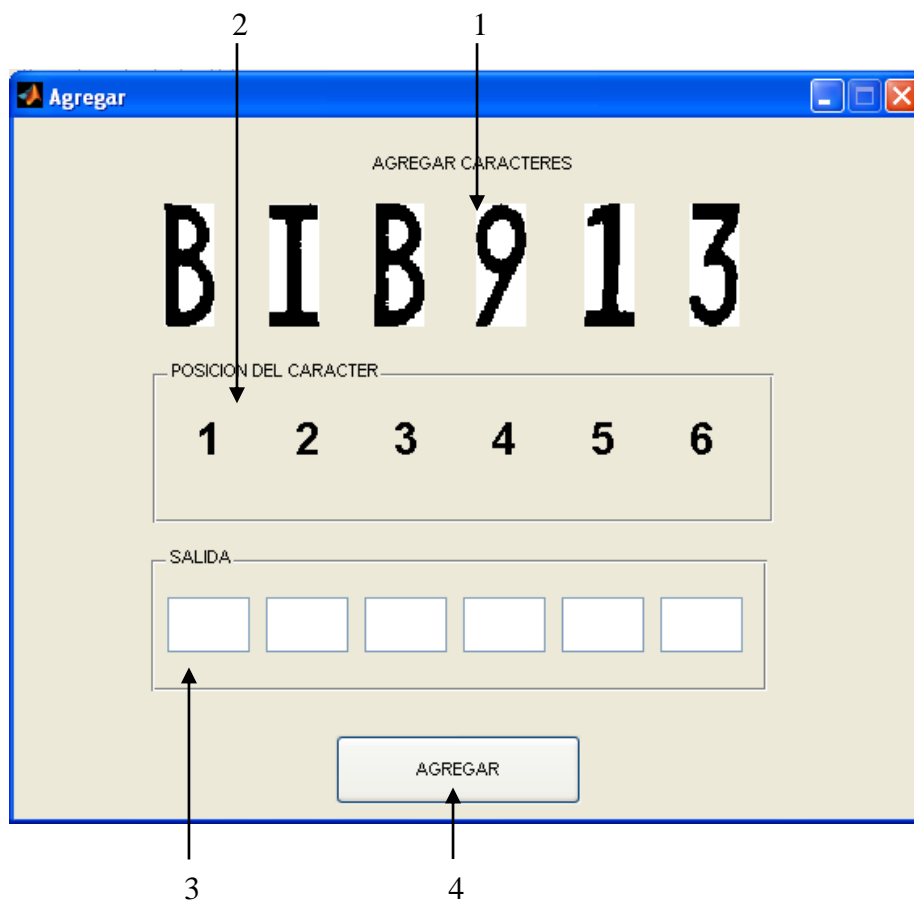


Figura 7.11. Interfaz para agragar caracteres

1 Ilustra los caracteres que se han segmentado.

2 Posición del carácter: indica el número de la posición correspondiente a los caracteres ya segmentados.

3 Salida: cajas de texto en las cuales el usuario debe ingresar el carácter que el sistema no reconoció, por ejemplo, si el la letra “I” ilustrada en la figura 7.11 no fue reconocida por el sistema el usuario debe escribir la letra “I” en la caja de la posición 2, con lo cual la red Fuzzy ARTMAP Modificada recibirá el carácter segmentado (posición 2) y la salida ingresada por el usuario.

4 Agregar. Agrega a Fuzzy ARTMAP el carácter de error.

A continuación se ilustra un ejemplo del funcionamiento de las interfaces:



Figura 7.12 Indica el proceso completo del reconocimiento de una placa



Figura 7.13 indica el funcionamiento de la interfaz 2

CONCLUSIONES

El propósito de este trabajo ha sido el análisis de diferentes métodos utilizados para el reconocimiento de placas de vehículos, de los cuales se han encontrado los más adecuados para el reconocimiento de placas en Colombia.

Para la etapa de detección de la placa se presentaron 4 métodos, siendo unos más eficientes que otros, dependiendo de las características de la imagen de entrada. Se podría decir que esta es la etapa más importante del sistema, porque es donde se obtuvo el mayor porcentaje de error, porque cada método requiere unas especificaciones en la entrada que no se cumplieron.

La segmentación depende de la calidad de la placa y de la precisión en la detección. En esta etapa se utiliza el método de picos y valles mencionado en la mayoría de la literatura, presentando un alto porcentaje de error, donde la causa fue la poca capacidad de segmentar caracteres inclinados, por lo tanto se decide que no es apropiado para resolver este problema. Con el método de relación de distancia se obtuvo un porcentaje de error no tan bajo, con el método de los componentes conectados se obtuvo un resultado satisfactorio y se decide que este último método es el apropiado.

La eficiencia de la etapa de extracción de características se vio reflejada en la etapa de reconocimiento, de la cual se observó que los métodos de Momentos y el código RITRE no son los más adecuados para esta clase de aplicaciones. Se demostró que el método de zonificación, siendo un algoritmo menos complejo como lo son RITRE y los momentos, fue el que obtuvo un resultado muy bueno.

Los métodos de reconocimiento de caracteres desarrollados obtuvieron porcentajes de éxito altos para el caso de que el vector de características represente a cada carácter correctamente.

Los métodos de reconocimiento de caracteres pueden presentar errores si se presentan caracteres con características, que no se han utilizado en el entrenamiento, por lo cual será importante recolectar más patrones para entrenar los algoritmos.

Se observó que aunque unos métodos presentaron menor tiempo de ejecución, pero con errores de éxito más altos que otros. Dado que el porcentaje de éxito es lo más primordial y la diferencia de tiempo no es grande, se escogen estos métodos para la implementación del sistema total.

En una aplicación real, se pueden ajustar los parámetros para determinada aplicación, de acuerdo a las condiciones que se estén presentando, para lograr mejores resultados, teniendo en cuenta que cada aplicación presenta distintas características

En todos los algoritmos, desde detección hasta reconocimiento, si la entrada no tiene problemas se pueden obtener buenos resultados; uno de los mayores problemas es el ángulo de rotación, porque si es muy alto, no se logra hacer una buena restauración de la imagen y las letras se deforman.

REFERENCIAS

- [1] www.eltransporte.com/modules.php?name=News&file=article&sid=394
Fecha de consulta 10 de marzo de 2005 3:45 PM
- [2] Molina R, Introducción al Procesamiento y Análisis de Imágenes Digitales, 1998.
- [3] Gonzalez Rafael, Woods Richard, Digital Image Processing, 1993, Addison-Wesley, ISBN 0-201-60078-1.
- [4] Gonzales Rafael, Woods Richard y Eddins Steven, Digital Image Processing using Matlab, 2004.
- [5] Díaz Acosta Beatriz, Experiments in image segmentation for automatic us license plate recognition, 2004
- [6] www.normankoren.com
Fecha de consulta 17 de agosto de 2005 4:00 PM
- [7] Serra Jean. “image Análisis Using Mathematical Morphology”. Ed: Academic Press. 1984.
- [8] Rodríguez Fernando, Fernández Xulio, Contribución al reconocimiento de caracteres en imágenes complejas, 2003.
- [9] http://www.vision.auc.dk/%7Etbm/Student_projects/02-automatic.pdf
Fecha de consulta 22 de noviembre de 2004 5:30 PM
- [10] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, “ARTMAP: Supervised real-time learning and Clasificación of Nonstationary data by self organizing neural network”, Neural Networks, vol 4, pp. 565-588, 1991. IEEE.
- [11] Torres, Translation, Rotation, and Scale-Invariant Object Recognition, Applications and Reviews, vol 30, NO. 1, February 2000, IEEE.
- [12] Trier, Jain y Taxt. Feature extraction methods for character recognition – a survey Shahaf Oshrat and Timor Avi, Real Time License Plate Recognition in a Video Movie, 2002

- [13] James A. Freeman/David M. Skapura, (1991), redes neuronales algoritmos, aplicaciones y técnicas de programación, Addison-Wesley/Diaz de Santos.
- [14] G. A. Carpenter, S. Grossberg, and D. B. Rosen “Fuzzy ART: Fast stable Learning and categorization of analog patterns by an adaptive resonance system”, Neural Networks, vol 4, pp. 759-771, 1991. IEEE.
- [15] G. A. Carpenter, S. Grossberg, and N. M. Markuson, J. H. Reynolds and D. N. Rosen, (1992) “Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog multidimensional Maps” vol. 3, pp. 698-713. IEEE Transactions on neural Networks.
- [16] Sutherland, J. The holographic neural method. En Fuzzy, Holographic and Parallel Intelligence. John Wiley And Sons, New York, 1992.
- [17] Torres, R. Suárez, G. Gómez, y Sucar E. Translation, Rotation and Scale Invariant Object Recognition. Memorias de la XII Reunión Nacional de Inteligencia Artificial, México. Pp 70-76.1995.
- [18] Gail A. Carpenter, Natalya Markuzon, ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases
- [19] Moritz StÄoring and Thomas B. Moeslund An Introduction to Template Matching

APÉNDICE A: LISTA DE FUNCIONES

- Function PlacaS = PrincipalSobel_H(I)

La función “PrincipalSobel_H” localiza la placa por el método del Gradiente.

“I” es la imagen (vehículo) de entrada a la función “PrincipalSobel_H”.

“PlacaS” es la imagen (placa) de salida de la función “PrincipalSobel_H”.

- Function PlacaM=M_Morfológico (I)

La función “M_Morfológico” localiza la placa por el método Morfológico

“I” es la imagen (vehículo) de entrada a la función “M_Morfológico”.

“PlacaM” es la imagen (placa) de salida de la función “M_Morfológico”.

- Function PlacaC = detecolor_H(I)

“I” es la imagen (vehículo) de entrada a la función “detecolor”.

“PlacaC” es la imagen (placa) de salida de la función “detecolor”.

- Function PlacaC = deteHough_H(I)

“I” es la imagen (vehículo) de entrada a la función “deteHough”.

“PlacaC” es la imagen (placa) de salida de la función “deteHough”.

- Function Ang = angulo (placa)

“placa” es la imagen de entrada a la función “angulo”.

“Ang” es el angulo de la línea mas larga.

- Function ImagRest = Restauracion (placa, angulo)

“placa” es la imagen de entrada a la función “Restauracion”.

“Angulo” es de rotación de la placa.

“ImagRest” es la imagen de la placa restaurada.

- function P_sin_ruido = borrar_ruido(PlacaBin)

La función “borrar_ruido” elimina pequeñas partículas en proporción a los caracteres

“PlacaBin” es la imagen de entrada en binario, esta variable es la entrada a la función “FuzzyARTMAPMod”.

“P_sin_ruido r” es la salida de la función “borrar_ruido”..

- Function ImagUmbral = Umbralización (I)

“I” es la imagen de entrada a la función “Restauracion”.

“ImagUmbral” es la imagen de salida en binaria.

- Function ImagFilt = Filtrado(placa)

“placa” es la imagen de entrada a la función “Filtrado”.

“ImagFilt” es la imagen de la placa con los seis caracteres.

- Function caracteres = SepararCarac(placa)

“placa” es la imagen de entrada a la función “Filtrado” conteniendo únicamente seis objetos.

“caracteres” estructura conteniendo las seis imágenes que corresponden a los seis caracteres.

- Function Caracteres = RelaciónDistancia (placa)

La función “RelaciónDistancia” segmenta la imagen (placa) en los seis caracteres por el método Relación de distancia.

“placa” es la imagen de la placa del vehículo, esta variable es la entrada a la función “RelaciónDistancia”.

“Caracteres” son las imágenes de los caracteres de la placa, esta variable es la salida de la función “RelaciónDistancia”.

- Function Caracteres=PicosValles (placa)

La función “PicosValles” segmenta la imagen (placa) en los seis caracteres por el método de Picos y Valles.

“placa” es la imagen de la placa del vehículo, esta variable es la entrada a la función “PicosValles”.

“Caracteres” son las imágenes de los caracteres de la placa, esta variable es la salida de la función “PicosValles”.

- Function VectorC=RITRE (Caracterer)

La función “RITRE” calcula el código RITRE (vector de características) a partir de la imagen de un carácter

“Carácter” es la imagen de un carácter, esta variable es la entrada a la función “RITRE”.

“VectorC” es la salida de la función “RITRE”, entrega el vector de características.

- Function VectorC=momentos (Caracterer)

La función “momentos” calcula los siete momentos los cuales conforman al vector de características, a partir de la imagen de un carácter.

“Carácter” es la imagen de un carácter, esta variable es la entrada a la función “momentos”.

“VectorC” es la salida de la función “momentos”, entrega el vector de características.

- Function VectorC=Zonificación(Caracterer)

La función “Zonificación” calcula los 30 valores promedio de la imagen de un carácter los cuales conforman al vector de características.

“Caracterer” es la imagen de un carácter, esta variable es la entrada a la función “Zonificación”.

“VectorC” es la salida de la función “Zonificación”, entrega el vector de características.

- Function VectorA = Area(Caracterer)

La función “Area” calcula los valores de la suma horizontal y vertical de la imagen

“Caracterer” es la imagen de un carácter, esta variable es la entrada a la función “Area”.

“VectorA” es la salida de la función “Area”, entrega el vector de características.

- Function caracter=VecinoHolográfico(VectorC)

La función “VecinoHolográfico” calcula el Vecino Mas Cercano Holográfico, e identifica el carácter.

“VectorC” es el vector de característica, esta variable es la entrada a la función “VecinoHolográfico”.

“caracter” es la salida de la función “VecinoHolográfico”, entrega el carácter identificado.

- Function caracter = FuzzyARTMAP(VectorC)

La función “FuzzyARTMAP” identifica el carácter utilizando la red neuronal Fuzzy ARTMAP.

“VectorC” es el vector de característica, esta variable es la entrada a la función “FuzzyARTMAP”.

“caracter” es la salida de la función “FuzzyARTMAP”, entrega el carácter identificado.

- Function caracter = FuzzyARTMAPMod(VectorC)

La función “FuzzyARTMAPMod” identifica el carácter utilizando la red neuronal Fuzzy ARTMAP modificada.

“VectorC” es el vector de característica, esta variable es la entrada a la función “FuzzyARTMAPMod”.

“caracter” es la salida de la función “FuzzyARTMAPMod”, entrega el carácter identificado.

- Function $caracter = RecoPlantillas(VectorC)$

La función “RecoPlantillas” identifica el carácter utilizando el método de comparación de plantillas

“VectorC” es el vector de característica, esta variable es la entrada a la función “RecoPlantillas”.

“caracter” es la salida de la función “RecoPlantillas”, entrega el carácter identificado.

- Function $caracter = BayesG(VectorC)$

La función “BayesG” identifica el carácter utilizando el método de comparación de plantillas

“VectorC” es el vector de característica, esta variable es la entrada a la función “BayesG”.

“caracter” es la salida de la función “BayesG”, entrega el carácter identificado.