



# **UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**

La Universidad Católica De Loja

## **ÁREA TÉCNICA**

**TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y  
COMPUTACIÓN**

**Desarrollo de una aplicación para reconocimiento y extracción de placas  
de vehículos.**

**TRABAJO DE TITULACIÓN.**

**AUTOR:** Ordoñez López, Juan Pablo

**DIRECTOR:** Barba Guamán, Luis Rodrigo, Mgtr.

**LOJA – ECUADOR**

**2016**



*Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>*

2016

## **APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN**

Magister.

Luis Rodrigo Barba Guamán.

**DOCENTE DE LA TITULACIÓN**

De mi consideración:

El presente trabajo de titulación: Desarrollo de una aplicación para reconocimiento y extracción de placas de vehículos realizado por Ordoñez Lopez, Juan Pablo, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, junio de 2016.

f) .....

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Ordoñez Lopez, Juan Pablo declaro ser autor (a) del presente trabajo de titulación: **Desarrollo de una aplicación para reconocimiento y extracción de placas de vehículos**, de la Titulación Ingeniería en Sistemas Informáticos y Computación, siendo Luis Rodrigo Barba Guamán director (a) del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f. ....

Autor: Ordoñez Lopez, Juan Pablo

Cédula: 1104348972

## DEDICATORIA

Esta tesis va dedicada para las personas más importantes en mi vida, la cuales de cierta forma han estado en las buenas y en las malas junto a mí.

Especialmente para mi madre Esperanza Lopez que a través de su ejemplo, valor, sabiduría, amor ha sabido encaminarme y ha sido fuente de inspiración para cumplir mis objetivos, así como a toda mi familia que en cada paso me han estado brindando su apoyo y consejo.

Les dedico también este logro a todas aquellas personas que aportaron su granito de arena en el desarrollo del mismo y que sin ellas este proyecto no se hubiera concluido de forma satisfactoria.

Juan Pablo Ordoñez Lopez

## **AGRADECIMIENTO**

Mi gratitud a quienes me apoyaron en el desarrollo del presente trabajo de fin de titulación: A la Universidad Técnica Particular de Loja (UTPL), toda el área de la sección de Inteligencia Artificial por los conocimientos compartidos, y de manera especial al Mgtr. Luis Rodrigo Barba Guamán que me facilitó todo lo que estuvo a su alcance.

Juan Pablo Ordoñez Lopez

## ÍNDICE DE CONTENIDOS

CARATULA .....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN .....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS .....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
ÍNDICE DE CONTENIDOS .....	vi
ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS .....	ix
RESUMEN .....	1
ABSTRACT.....	2
INTRODUCCIÓN .....	3
ESTADO DEL ARTE .....	5
1.1    Antecedentes.....	6
1.2    Visión por computador. ....	6
1.3    Identificación de vehículos.....	7
1.4    Trabajos existentes.....	7
1.5    Estructura del sistema LPR (License plate recognition). ....	9
1.6    Desafíos en la detección de la placa.....	9
1.7    Detección de Vehículos. ....	10
1.7.1    Métodos de detección de vehículos.....	11
1.7.2    Selección del método para la detección de vehículos.....	21
1.8    Extracción de la región de la placa. ....	24
1.8.1    Métodos para la detección de placas en vehículos.....	25
1.8.2    Selección del método para la detección de la región de placa.....	32
1.9    Segmentación de los caracteres de la región de la placa. ....	37
1.9.1    Métodos para la segmentación de los caracteres de la placa.....	37
1.9.2    Selección del método para la segmentación de caracteres en la región de la placa. ....	42
1.10   Reconocimiento de los caracteres obtenidos en la segmentación de la región de la placa.....	43
1.10.1   Métodos para el reconocimiento de caracteres.....	44
1.10.2   Selección del método para el reconocimiento de los caracteres segmentados. ....	50
1.11   Sistemas de estimación de distancia de vehículos basados en visión estéreo.....	51
1.12   Cálculo de disparidad en un sistema de visión estéreo.....	52
1.13   Selección de cámaras para la visión estéreo.....	55
1.14   Configuración y Calibración de las cámaras del sistema estéreo.....	57
2    METODOLOGIA .....	59
2.1    Arquitectura del Sistema. ....	60
2.2    Metodología XP como metodología de desarrollo.....	61
2.3    Planeación - Historias de usuario. ....	62
2.3.1    Historia de usuario Capturar par de Imágenes.....	62
2.3.2    Historia de usuario Calibrar Cámaras. ....	63

2.3.3	Historia de usuario Detectar vehículos. ....	63
2.3.4	Historia de usuario Detectar área de la placa. ....	64
2.3.5	Historia de usuario Segmentar los caracteres de la placa. ....	64
2.3.6	Historia de usuario Reconocer los caracteres de la Placa. ....	64
<b>2.4</b>	<b>Planeación – Criterios de aceptación. ....</b>	<b>65</b>
<b>2.5</b>	<b>Diseño mediante tarjetas CRC.....</b>	<b>65</b>
2.5.1	Tarjeta CRC sistema detector de placas vehiculares.....	65
2.5.2	Tarjeta CRC calibración de cámaras. ....	66
2.5.3	Tarjeta CRC detector de vehículos. ....	67
2.5.4	Tarjeta CRC detector de región de la placa. ....	67
2.5.5	Tarjeta CRC segmentador e identificador de los caracteres de la placa. .	68
<b>2.6</b>	<b>Codificación mediante lenguaje de programación java.....</b>	<b>68</b>
2.6.1	Implementación Calibración de cámaras. ....	68
2.6.2	Implementación Detección de Vehículos. ....	69
2.6.3	Implementación Detección de la Región de la Placa. ....	71
2.6.4	Implementación Segmentación de los Caracteres de la Placa.....	75
2.6.5	Implementación Reconocimiento de Caracteres de la Placa. ....	76
<b>2.7</b>	<b>Pruebas de aceptación. ....</b>	<b>78</b>
2.7.1	Validar Captura del par de imágenes y Guardado. ....	78
2.7.2	Validar Calibración de cámaras. ....	79
2.7.3	Validar tasa de exactitud detección de vehículos.....	80
2.7.4	Validar tasa de exactitud de la región de la placa, segmentación y reconocimiento de caracteres en vehículos detectados. ....	81
<b>3</b>	<b>RESULTADOS .....</b>	<b>83</b>
<b>3.1</b>	<b>Versión 1. ....</b>	<b>84</b>
3.1.1	Detección de vehículos.....	84
3.1.2	Detección de la región de la placa. ....	86
3.1.3	Segmentación de los caracteres de la placa.....	88
3.1.4	Reconocimiento de caracteres. ....	90
<b>3.2</b>	<b>Versión 2. ....</b>	<b>91</b>
3.2.1	Detección de vehículos.....	91
3.2.2	Detección de la región de la placa. ....	92
3.2.3	Segmentación de la región de la placa. ....	93
3.2.4	Reconocimiento de caracteres. ....	94
<b>3.3</b>	<b>Exactitud del sistema en las versiones implementadas.....</b>	<b>94</b>
<b>4</b>	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>96</b>
<b>4.1</b>	<b>Conclusiones.....</b>	<b>¡Error! Marcador no definido.</b>
<b>4.2</b>	<b>Recomendaciones.....</b>	<b>¡Error! Marcador no definido.</b>
<b>5</b>	<b>REFERENCIAS .....</b>	<b>98</b>

## ÍNDICE DE TABLAS

Tabla 1. Comparación de sistemas basados en visión estéreo para la detección y estimación de distancia de vehículos. ....	12
Tabla 2. Comparación de métodos para la detección de vehículos. ....	22
Tabla 3. Tabla de escenarios con configuraciones diferentes para la detección de vehículos. ....	23
Tabla 4. Comparación de métodos para la extracción de la región de la placa, segmentación de caracteres y reconocimiento de los mismos. ....	34
Tabla 5. Principales características cámaras seleccionadas. ....	56
Tabla 6. Metodologías tradiciones vs metodologías Ágiles. ....	61
Tabla 7. Requerimientos y sus criterios de aceptación. ....	65
Tabla 8. Tasas de exactitud del sistema propuesto. ....	84
Tabla 9. Tasas de exactitud del sistema propuesto. ....	91
Tabla 10. Comparación de versiones implementadas. ....	95

## ÍNDICE DE FIGURAS

Figura 1. Elementos de un sistema de reconocimiento de placas.....	6
Figura 2. Representación del mundo real. ....	6
Figura 3. Fases de un Sistema para la detección de los caracteres de la placa en vehículos. ....	9
Figura 4. (a) Placa estándar Alberta. b) Placa Alberta extendida.....	10
Figura 5. Imagen de salida en el proceso de detección de vehículos. ....	11
Figura 6. Representación en el espacio de un vehículo. ....	14
Figura 7. Esquema de un sistema estereoscópico. ....	15
Figura 8. Representación de un cromosoma.....	16
Figura 9. Proceso para detectar y seguir vehículos. ....	17
Figura 10. Imagen resultante de la detección de la región de la placa.....	24
Figura 11. Aplicando el método de la gradiente. ....	25
Figura 12. Imagen a y b son dos imágenes de secuencia y c elimina los estáticos. ....	26
Figura 13. Línea recta en el sistema cartesiano.....	27
Figura 14. Región de la placa representada como rectángulos. ....	28
Figura 15. Coincidencia de plantillas en la detección de la placa.....	29
Figura 16. Proceso Detección de la Región de la Placa. ....	30
Figura 17. a) Suma valores al punto (x,y) b) Calculo Imagen Integral .....	31
Figura 18. Cálculo del área para una región de la imagen original, basados en la imagen integral. ....	32
Figura 19. Placas Taiwanesa y Ecuatoriana. ....	33
Figura 20. Caracteres segmentados de la región de la placa. ....	37
Figura 21. Caracteres como rectángulos. ....	37
Figura 22. Extracción de caracteres utilizando el método de proyección horizontal y vertical.....	39
Figura 23. Picos y Valles basados en el histograma de la imagen.....	40
Figura 24. a) Placas compactas b) Caracteres distintos c) Caracteres indistintos .....	40
Figura 25. a) Placa Taiwanesa, b) Placa Ecuatoriana y c) Placa China.....	43
Figura 26. Caracteres obtenidos para identificar un vehículo. ....	43
Figura 27. a) Capas de la ANN y b) Número de ejemplos de entrenamiento usado. ....	44
Figura 28. Esquema de movimiento para realizar coincidencia mediante plantilla. ....	46
Figura 29. Características prototipos OCR a) CC y b) DC. ....	47
Figura 30. Esquema para la verificación de un carácter. ....	48
Figura 31. Extracción característica CC. ....	48
Figura 32. Extracción característica PBA. ....	49
Figura 33. a) Características de imagen b) Características de imagen mapeadas al espacio. ....	50
Figura 34. a) Regiones compactas de placa, b) Caracteres distintos y c) Caracteres Indistintos.....	51
Figura 35. Cámaras en escena. ....	52
Figura 36. Imágenes Rectificadas. ....	53
Figura 37. Correspondencia de puntos entre las imágenes izquierda y derecha. ....	54
Figura 38. Valores de disparidad relativos para las cámaras. ....	54

Figura 39. Webcams en una barra trípode.....	55
Figura 40. Posibles rangos de detección de vehículos .....	55
Figura 41. Detección de vehículo.....	56
Figura 42. Par de imágenes para calibración estéreo.....	57
Figura 43. Módulos proceso asilado – Calibración de cámaras.....	60
Figura 44. Módulos proceso en tiempo real – Detección de placas.....	61
Figura 45. Fases Programación extrema.....	62
Figura 46. Prototipo Detector de Placas Vehiculares.....	66
Figura 47. Prototipo Detector de vehículos.....	67
Figura 48. Par de imágenes del conjunto de calibración de cámaras.....	69
Figura 49. Par de cámaras para simular el sistema estéreo.....	69
Figura 50. Par de imágenes para generar el mapa de disparidad.....	70
Figura 51. Par de imágenes rectificadas.....	70
Figura 52. Creación de la población inicial de cromosomas.....	71
Figura 53. Vehículo Detectado.....	72
Figura 54. Bordes Verticales.....	72
Figura 55. Generación de áreas candidatas.....	73
Figura 56. Extracción de áreas candidatas – Algoritmo un solo paso.....	73
Figura 57. Región de placa detectada.....	75
Figura 58. Etiquetado de componentes conectados.....	75
Figura 59. Caracteres de placa segmentados.....	76
Figura 60. Extracción característica CC.....	76
Figura 61. Características prototipos OCR a) CC y b) DC.....	77
Figura 62. Extracción característica PBA.....	77
Figura 63. Aplicación para asociar una imagen a un carácter o dígito.....	78
Figura 64. Mapas de disparidad para un vehículo ubicado a diferentes distancias.....	85
Figura 65. a) Rectificación imagen izquierda, b) Rectificación imagen derecha y c) Correspondencia de imágenes.....	85
Figura 66. Problema en la extracción de las regiones de placa inclinada.....	86
Figura 67. Problema al generar las regiones candidatas a placa en imagen con brillos.....	86
Figura 68. Problema en regiones de placa laterales no reconocidas por el clasificador en cascada.....	87
Figura 69. Problema en regiones de placa en condiciones ideales no reconocidas por el clasificador en cascada.....	87
Figura 70. Detección de falsas alarmas en vehículos con textos.....	88
Figura 71. Problema en la segmentación de caracteres en placas antiguas....	88
Figura 72. Problema con los caracteres enlazados entre sí mediante borde de la región de la placa.....	89
Figura 73. Problema en la segmentación de caracteres para imágenes con brillos.....	89
Figura 74. Problema en la segmentación de caracteres en regiones de placa con fondo de color fuerte.....	90
Figura 75. Problema en regiones de carácter con grietas y ruido como tornillos. .....	90
Figura 76. Merge o correspondencia del par de imágenes izquierda y derecha para diferentes pruebas de calibración.....	92
Figura 77. a) Imagen ecualizada, b) Imagen dilatada y c) Generación de áreas candidatas.....	93

Figura 78. Vehículo y detección de la región de placa parcial.....	93
Figura 79. a) Región de placa detectada, b) Placa sin inclinación y c) Región sin sobrantes.....	93
Figura 80. a) Región de carácter con grietas y b) Carácter después de aplicar opening. ....	94
Figura 81. Exactitud del sistema por versión.....	94

## RESUMEN

El actual proyecto de tesis es una concentración de los diversos conocimientos aprendidos durante la Carrera de Ingeniería en Sistemas que imparte la Universidad Técnica Particular de Loja (UTPL).

Se implementará un sistema que permita detectar y extraer placas de vehículos a partir de una imagen utilizando un sistema de visión artificial, se realizará un análisis de los métodos existentes, se seleccionará los mejores métodos para proceder a la implementación del reconocimiento de placas, se utilizará el lenguaje de programación java, las librerías de inteligencia artificial opencv y su interfaz javavc.

**PALABRAS CLAVES:** Reconocimiento de placas de matrícula, cascada de clasificadores, estimación de distancia, análisis de componentes conectados, máquina de vectores de soporte

Juan Pablo Ordóñez López

## **ABSTRACT**

The current thesis project is a concentration of the various skills learned during Race Systems Engineering offered by the Technical University of Loja (UTPL).

A system to detect and extract license plates from an image using a machine vision system will be implemented, an analysis of existing methods is made, the best methods to proceed with the implementation of plate recognition is selected, the Java programming language, opencv and javacv libraries will be used.

**KEYWORDS:** Recognition of license plates, cascade classifiers, distance estimation, connected component analysis, support vector machine

Juan Pablo Ordonez Lopez

## INTRODUCCIÓN

En el presente trabajo se implementa un sistema que permite automatizar las tareas de identificación de los caracteres de la placa mediante técnicas de visión artificial, para ello se realiza una selección de los mejores métodos existentes.

El trabajo se encuentra compuesto por tres capítulos:

El primero contiene la exploración o estado del arte sobre los métodos existentes y la selección de los mismos para la extracción de los caracteres de la placa, en general contiene toda la teoría necesaria para realizar la implementación en una etapa posterior.

El capítulo dos describe la arquitectura del sistema, la metodología utilizada, el análisis y descripción de los módulos, el diseño de la solución, la implementación de los mismos, las herramientas y librerías manejadas.

El capítulo tres describe la evaluación de los principales problemas encontrados en la primera versión, en una segunda versión se intenta incrementar la exactitud de los módulos mediante la corrección o mejoras a las fases implicadas.

En los últimos años el flujo de vehículos en las ciudades ha crecido de manera impresionante de tal forma que es necesario realizar un control estadístico, de seguridad, control de velocidad de los sistemas de transporte, es por ello que se crea la necesidad de contar con sistemas que ejecuten estas tareas tediosas y en cierta manera cansadas y monótonas en las que una persona tiende a cometer errores en la recolección de datos.

Para resolver el problema de la identificación de vehículos mediante el reconocimiento de los caracteres de la placa, se explora en búsqueda de investigaciones previas que centran su trabajo en la comparación de los diversos métodos existentes y la selección de los mismos para la tarea plateada.

Para el desarrollo del proyecto final de carrera se considera el siguiente objetivo general: Desarrollar un aplicativo con la capacidad de detectar y extraer los caracteres de la región de placa en un vehículo, a partir de éste se extraen los siguientes objetivos específicos:

- Realizar un análisis y selección de los mejores métodos existentes para el proceso de extracción y reconocimiento de los caracteres de la placa en vehículos.
- Implementar los métodos seleccionados para la extracción y reconocimiento de los caracteres de la placa en vehículos.
- Mejorar la exactitud del reconocimiento mediante una refactorización del sistema a partir de una primera versión implementada.

En el desarrollo del proyecto uno de los principales bloqueos, fue la falta del par de cámaras ip para la implementación del método para la detección de vehículos, necesarias para utilizar un sistema de visión estéreo mediante el cual es posible determinar o estimar la distancia de un objeto hacia el sistema de cámaras, para solucionar el problema se utilizó aplicaciones para el sistema operativo Android que permiten simular la emisión de video como si se tratase de una cámara ip.

## **ESTADO DEL ARTE**

## Antecedentes.



Figura 1. Elementos de un sistema de reconocimiento de placas.  
Fuente: ("Vivotek," 2015).

Las vías son una parte importante para el desarrollo de un país, cuando el parque automotor crece de forma tan acelerada, se enfrenta a la necesidad de disponer de una gran cantidad de recurso humano para monitorear y hacer cumplir las leyes de tránsito, así como empresas privadas necesitan elaborar estudios para poder realizar mantenimientos futuros o mejoras a las vías. En ambos casos las tareas de monitorizar variables como velocidad, cantidad de autos que pasan por día, se vuelve tedioso y sobre todo cansado para las personas encargadas que realizan el control y seguimiento en prolongadas jornadas, las cuales no aseguran que el personal este concentrado totalmente durante su periodo de trabajo, luego estos datos serán pasados a un experto para que los analice. Entonces surge la necesidad de crear o automatizar procesos de monitorización y seguimiento de tráfico, para guardar información y explotar esos datos de forma eficiente.

## Visión por computador.

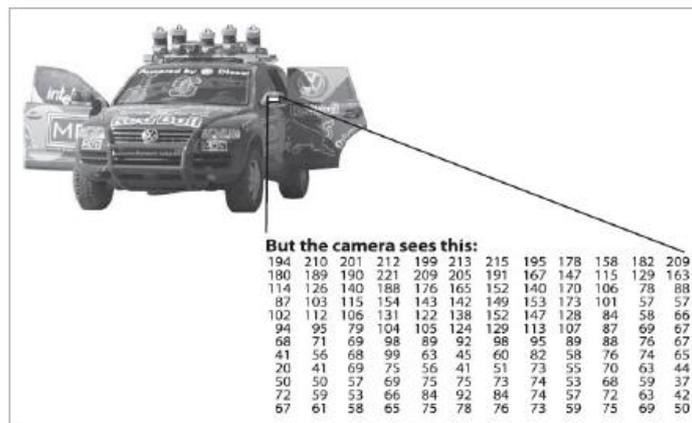


Figura 2. Representación del mundo real.  
Fuente: (Valencia, 2014)

La visión por ordenador es un subcampo de la inteligencia artificial que incluye métodos para adquirir, procesar, analizar y comprender las imágenes, la representación del mundo real se la hace mediante la generación de información numérica o simbólica. Un argumento en el desarrollo de este campo es la de duplicar la capacidad de la visión humana mediante la percepción por vía electrónica para comprender una imagen. Las imágenes pueden ser vistas como relaciones numéricas que son creadas a partir de sus características utilizando modelos construidos con la ayuda de la geometría, la física, la estadística y la teoría del aprendizaje.

### **Identificación de vehículos.**

La identificación automática de vehículos es una etapa esencial en los sistemas de tráfico inteligentes. Hoy en día los vehículos juegan un papel muy importante en el transporte. También el uso de los mismos se ha incrementado debido al crecimiento de la población y las necesidades humanas en los últimos años. Por lo tanto, el control de los autos se está convirtiendo en un gran problema y con muchas dificultades por resolver. Estos sistemas automáticos de identificación de vehículos se utilizan para el propósito de un control efectivo.

El reconocimiento de placas de matrículas (LPR o License plate recognition) es una forma de identificación automática de vehículos.

Se trata de una tecnología de procesamiento de imagen que se utiliza para identificar vehículos mediante sus placas de matrícula.

Un sistema LPR en tiempo real juega un papel muy importante en la vigilancia y seguimiento automático de las normas de tráfico y el mantenimiento del orden en la vía pública.

### **Trabajos existentes.**

Muchas investigaciones de identificación de vehículos han sido abordados por la detección y reconocimiento de placas, algunos de los trabajos más relevantes se exponen en (Du, Ibrahim, Shehata, & Badawy, 2013), en el mismo se hace mayor referencia en la comparación de métodos para la detección de la región de la placa, las segmentación de caracteres y el reconocimiento de los mismos. Otros trabajos relacionados a la detección de vehículos y seguimiento de estos basados en técnicas de visión estéreo se exponen en (Nguyen, Nguyen, Nguyen, Lee, & Jeon, 2013) y (Chiu, Chung, & Chen, 2010), otro método en (Lee, Chen, & Wang, 2004) propone realizar la

identificación de vehículos por eliminación de bloques estáticos mediante la comparación de una secuencia de imágenes, en (Negri, Clady, Hanif, & Prevost, 2008) se hace la identificación de vehículos utilizando el método de cascada de clasificadores Haar-Like.

Así también se encuentra investigaciones relacionadas a la detección de la región de la placa (Wang, Member, & Lee, 2007) basadas en las características de una área de placa mediante la búsqueda de tramos de píxeles horizontales muy cercanos y verificación de las mismas mediante una cascada de clasificadores Haar-Like, en otro trabajo en (Gazcón, Chesñevar, & Castro, 2012) la detección se lleva a cabo mediante la búsqueda de rectángulos, el método propuesto soporta ligeras inclinaciones que son rectificadas antes de segmentar los caracteres de la región detectada.

Trabajos relacionados a la segmentación de los caracteres de la placa se puede encontrar en (Wang et al., 2007), en el mismo se propone una combinación de técnicas, la primera etapa lo hace mediante la búsqueda de componentes conectados y en la segunda etapa se realiza el corte de componentes enlazados entre sí mediante un análisis de picos y valles, en (Duan, Duc, & Du, 2004) la tarea se la realiza utilizando las técnicas de proyección vertical y horizontal.

Con respecto al reconocimiento de caracteres se encuentra los siguientes trabajos, en (Ozbay & Ercelebi, 2005) se lleva a cabo por plantillas de coincidencia, en (Wang et al., 2007) se lo hace aplicando una máquina de vectores de soporte, en donde los vectores están representados por tres características de los caracteres, en (Gazcón et al., 2012) se utiliza la técnica de redes neuronales artificiales para realizar la tarea.

## Estructura del sistema LPR (License plate recognition).



Figura 3. Fases de un Sistema para la detección de los caracteres de la placa en vehículos.  
Fuente: (Du et al., 2013)  
Elaboración: Juan Pablo Ordoñez Lopez.

Un sistema LPR (License plate recognition) que detecta los caracteres a partir de una imagen dada puede estar compuesto de cuatro etapas. La primera etapa es la adquisición de la imagen del auto utilizando una cámara. Los parámetros de la misma, tales como el tipo, la resolución, velocidad de obturación, la orientación y la luz, tienen que considerarse. La segunda etapa consiste en extraer la región de la placa de la imagen en función de algunas características, tales como el límite, el color, o la existencia de caracteres. La tercera etapa consiste en segmentar la placa y extraer los caracteres mediante la proyección de su información de color, etiquetarlos o por coincidencias mediante el uso de plantillas. La etapa final es reconocer los caracteres extraídos mediante coincidencia con plantillas o utilizando clasificadores, tales como las redes neuronales, máquinas de vectores de soporte y clasificadores difusos. La Figura 3 muestra la estructura del proceso de un sistema LPR. El rendimiento de un sistema LPR se basa en la solidez de cada etapa individual.

### Desafíos en la detección de la placa.

Las variaciones del tipo de placa o entornos causan desafíos en la detección y el reconocimiento de placas de vehículos. Los mismos se resumen a continuación.

- Las variaciones de la placa:
  - a. Localización: existen placas en diferentes puntos de una imagen.
  - b. La cantidad: una imagen puede contener ninguna o muchas placas.

- c. Tamaño: las placas pueden tener diferentes tamaños, esto debido a la distancia de la cámara y el factor de zoom.
- d. Color: las placas pueden tener varios caracteres y colores de fondo debido a los diferentes tipos de placas o dispositivos de captura.
- e. Fuente: las placas de diferentes naciones pueden ser escritos en diferentes tipos de letra y lenguaje.
- f. Estandarización: por ejemplo, la placa de licencia estándar en Alberta, Canadá, tiene tres y recientemente (en 2010) cuatro letras a la izquierda y tres números a la derecha, como se muestra en la Figura 4. (a). Al personalizar las placas pueden tener cualquier número de caracteres sin ningún tipo de regulaciones, como se muestra en la Figura 4. (b).
- g. La oclusión: Las placas pueden ser oscurecidas por la suciedad.
- h. Inclinación: Las placas pueden estar inclinadas en la imagen.
- i. Otros: Además de caracteres, una placa puede contener marcos y tornillos.



Figura 4. (a) Placa estándar Alberta. b) Placa Alberta extendida.  
Fuente: (Du et al., 2013)

- Variaciones de entorno:
  - a. Iluminación: Las imágenes de entrada pueden tener diferentes tipos de iluminación, principalmente debido a la iluminación y faros ambientales de los vehículos.
  - b. Fondo: El fondo de la imagen puede contener patrones similares a las placas, como números estampados en un vehículo, guardachoques con patrones verticales, y suelos con textura.

## **Detección de Vehículos.**



Figura 5. Imagen de salida en el proceso de detección de vehículos.  
Elaboración: Juan Pablo Ordoñez Lopez

Desde el punto de vista de la visión por ordenador, la detección de vehículos presenta innumerables retos en la carretera. Los mismos están típicamente en movimiento, presentando efectos de movimiento relativo. Existe una variabilidad en el tamaño, forma y color de los autos que se encuentran en el camino. Por lo tanto la detección de objetos a partir de una plataforma en movimiento requiere que el sistema pueda detectar, reconocer y localizar el objeto en el vídeo.

### **1.1.1 Métodos de detección de vehículos.**

En (Nguyen et al., 2013) podemos encontrar un resumen de métodos aplicados a la detección de vehículos, éste estudio se basa en métodos de visión estéreo, con sus configuraciones y sus distancias de detección mínimas y máximas, en la Tabla 1 podemos apreciar un resumen, en la misma se exponen métodos basados en la representación en el espacio, genéticos, filtros de Kalman, características de bordes, filtro de partículas, histograma polar y transformación de imágenes, movimiento simétrico de Imagen, características de esquina y correlación cruzada.

Tabla 1. Comparación de sistemas basados en visión estéreo para la detección y estimación de distancia de vehículos.

System	Stereo configuration	MDD	Stereo matching	Feature/Technique	Other
Nedevschi (2004)	N/A	90m	SAD	3-D points grouping	Four additional sensors are used. Processing time: 100ms/frame
Pauplim (2005)	N/A	N/A	N/A	Fly algorithm. Parisian approach	3-D point is randomly generated in ROI. Two CCD cameras are used. Processing time: 300ms/frame
Ruicheck (2004)	N/A	N/A	Genetic-based	Genetic algorithm	Multi-level searching and edge information. Processing time: 400ms/frame
Knoeppel (2000)	Focal length: 12mm.	150m	Cross correlation	3-D point extracting and clustering; Kalman Filter.	CMOS cameras
Bensrhair (2002)	Baseline: 95 cm	N/A	DP on edge points	3-D edge shapes of roads, symmetry	CCD cameras used to obtain disparity; Processing time:960 ms/frame
Toulminet (2006)	Focal length: 6mm.	100m	DP on edge points	3-D vertical features; symmetry; image correlation	Two stereo cameras used with monocular pattern analysis. Processing time: 95ms/frame
Jung (2007) [18]	Baseline: 30 cm	60m	Edge Feature	Lane Recognition; edge feature correlation	CMOS cameras
Huh (2008) [19]	Baseline: 15 cm	65m	Corner Feature	Corner features; epipolar constraint; kalman filter	Developed for highway circumstance
Southall (2009) [21]	Baseline: 15 inch (17.78 cm)	100m	SAD	Edge features; kalman filter	NTSC cameras
Kormann (2010) [22]	N/A	30m	SAD	Mean-Shift clustering of plane fitted segments; U/V disparity	3-D data from disparity map is used to detect vehicles as cuboid. Processign time 390ms/frame
Hwang (2009) [23]	Baseline: 150 cm	N/A	Edge feature based matching	Corner features; cross correlation; Kalman filter	Developed for highway circumstance. Processing time: 10ms/frame
Xu (2009) [24]	Baseline: 96 cm	N/A	SAD	Image symmetrical move	Symmetrical matching. Processing time: 155.5ms/frame. Detecction rate: 85%
Chiu (2010) [25]	Baseline: 20 cm	50 m	Horizontal and vertical line segments	Edge features; line segments	CMOS cameras; Asynchronus binocular system. Processing time: 30 ms/frame. Detection rate: 90 %
Franke (2011) [28]	Focal length: 830 pixeles, Baseline: 35 cm	N/A	SGM	Segmentation of the dynamic Stixel World	A Stixel is build by fusion of stereo and motion information. Processing time: 25 ms/frame

Broggi (2007) [31]	(baseline, focal length) = (0.8m,2.3mm),(0.5m, 2.3mm),(0.5m,2.2mm)	Short range detection	N/A	The polar histogram and image transformation are used	Support real-time segmentation without background knowledge. Stereobox CCD camera is used. Processing time: 80ms/frame
Oniga (2010) [24]	(Focal length/FOV:811 pixels/34 degrees) baseline 220mm	N/A	Census	Basased on Digital Elevation Map and RANSAC to detect obstacles, traffic isles	DeepSea G2 Embedded Vision System. Processing time 23 ms/frame
Nedevschi (2011) [36]	N/A	N/A	Block maching	The motion and diparaty are used	Monocular camera detects intersecting obstacles and stereo camera detects preceding obstacles. Processing time: 10 s/ frame
Our system	Focal length: 16 mm, 25 mm; baseline:35 cm, 60 cm (four different configurations)	140 m	Census	Genetic Algorithm	Combination of FPGA and CPU implemetations; CCD cameras. Processing time: 48 ms / frame. Detection rate: 96,08%

Fuente: (Du et al., 2013)

Elaboración: Juan Pablo Ordoñez Lopez.

### 1.1.1.1 Representación en el espacio.



Figura 6. Representación en el espacio de un vehículo.  
Fuente: (Nedevschi et al., 2004)

Este método permite calcular el movimiento del vehículo en el espacio, la fórmula a utilizar es la siguiente:

$$D(x, y, z) = 2 \left| |x - pX| - \left| \frac{v_x}{3} \right| \right| + 0.7 |y - pY| + 0.5 \left| |z - pZ| - \left| \frac{v_z}{3} \right| \right| \quad (1)$$

Donde,  $x$ ,  $y$  y  $z$  son las posiciones actuales del objeto en los ejes  $x$ ,  $y$  y  $z$  respectivamente;  $pX$ ,  $pY$  y  $pZ$  son las predicciones de las nuevas posiciones sobre los ejes  $x$ ,  $y$  y  $z$  respectivamente;  $V_x$  y  $V_z$  son las velocidades de desplazamiento sobre el eje  $x$  y  $z$  respectivamente.

### 1.1.1.2 Filtros de Kalman.

Este método funciona en un proceso de dos pasos por iteración. En una primera etapa de predicción, el filtro de Kalman produce estimaciones de las variables de estado actuales, junto con sus incertidumbres. En una segunda se observa el resultado de la siguiente medición (necesariamente corrupta con cierta cantidad de error, incluido un ruido aleatorio), estas estimaciones se actualizan utilizando un promedio ponderado, asignándose más peso a las que consiguen una mayor certeza. El algoritmo es recursivo. Puede funcionar en tiempo real, utilizando únicamente las presentes medidas

de entrada y el estado previamente calculado y su matriz de incertidumbre; no se requiere de información pasada adicional.

La fórmula utilizada en cada iteración es la siguiente:

Predicción:

$$\bullet \quad x_k = F_k x_{k-1} + B_k u_k + w_k \quad (2)$$

Donde  $F_k$  es el modelo de transición de estado que se aplica al estado anterior  $x_{k-1}$ .

$B_k$  es el modelo de control de entrada la cual se aplica al vector de control  $u_k$ .

$w_k$  es el ruido del proceso que se asume que puede extraerse de una media de distribución normal multivariante con covarianza  $Q_k$ .

### 1.1.1.3 Características de borde y segmentos de línea.

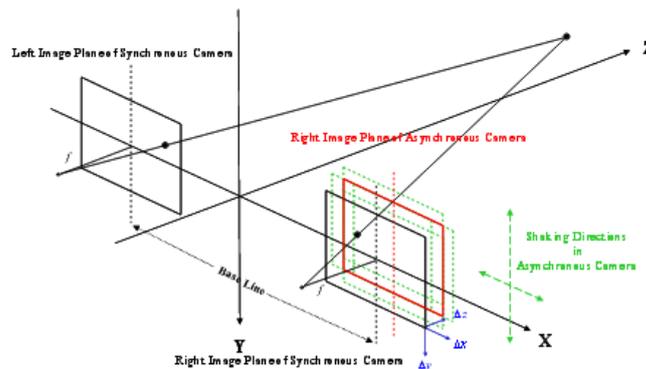


Figura 7. Esquema de un sistema estereoscópico.  
Fuente: (Chiu et al., 2010)

En este trabajo se describe un sistema de visión estereoscópica (binocular) multiresolución para detectar vehículos delante de autos con sistemas de seguridad avanzada (ASVs o advanced safety vehicles).

Se utilizan dos cámaras CMOS (complementary metal–oxide–semiconductor) asíncronas en el sistema propuesto, se montan sobre una plataforma que puede ser fácilmente fijada sobre el espejo retrovisor de un auto para detectar vehículos delante.

El algoritmo utiliza un módulo de adaptación de segmentos para encontrar coincidencia con los puntos extremos de las líneas de borde horizontal y vertical a diferentes resoluciones con el fin de disminuir el área de búsqueda y la complejidad computacional.

Como la distancia de cada segmento emparejado se puede calcular a partir del valor de disparidad, cada vehículo puede ser detectado por la agrupación de los segmentos que tienen distancias similares en la búsqueda, además con éste valor se puede realizar una estimación de distancia de cualquier objeto al sistema estéreo.

Las formulas resultantes de ésta investigación (Chiu et al., 2010) son:

$$D_{h_{avg}} = \frac{D_{h_{Left}} + D_{h_{Right}}}{2} \quad (3)$$

$$D_{v_{avg}} = \frac{D_{v_{Top}} + D_{v_{Bottom}}}{2} \quad (4)$$

Donde,  $D_{h_{Left}}$  y  $D_{h_{Right}}$ , son las distancias más a la izquierda y derecha respectivamente de un segmento de línea horizontal;  $D_{v_{Top}}$  y  $D_{v_{Bottom}}$  son las distancias más a la parte superior e inferior respectivamente de un segmento de línea vertical.

#### 1.1.1.4 Métodos Genéticos.

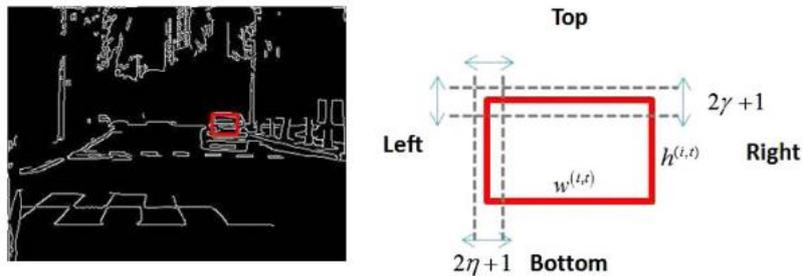


Figura 8. Representación de un cromosoma.  
Fuente: (Nguyen et al., 2013)

Según (Vericat, Stoico, Carlevaro, & Renzi, 2011) los algoritmos genéticos son una familia de modelos computacionales inspirados en la evolución natural. Estos algoritmos codifican una posible solución a un problema específico en una sencilla estructura de datos del cromosoma y se aplican los operadores (selección, cruzamiento, mutación) de recombinación a estas estructuras para preservar la información crítica. Los algoritmos genéticos son a menudo vistos como una función optimizador, aunque el rango de los problemas a los que se han aplicado los mismos es bastante amplio.

En el trabajo expuesto por (Nguyen et al., 2013) se centra en mejorar el rendimiento de los métodos basados en algoritmos evolutivos existentes para la detección de vehículos mediante la introducción de una función de adaptabilidad efectiva que puede capturar

con mayor precisión la información de un vehículo mediante la combinación del error de disparidad, la información de bordes, distancia focal representado por el valor de disparidad, la posición (distancia euclidiana) y la predicción del movimiento de un vehículo. El método propuesto puede detectar varios vehículos mediante el uso de un algoritmo genético turn-back (GA) y puede evitar la detección falsa mediante el uso de la detección de movimiento.

El método genético consta de las siguientes etapas:

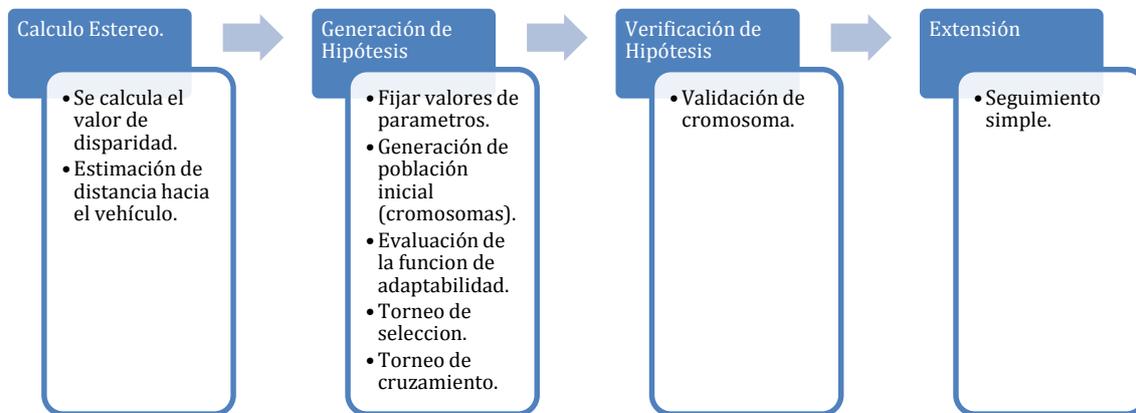


Figura 9. Proceso para detectar y seguir vehículos.  
Elaboración: Juan Pablo Ordoñez Lopez.

**Cálculo Estéreo.-** Este proceso es detallado en el estado del arte numeral 1.12.

**Generación de Hipótesis.-** En ésta etapa se realiza los siguientes pasos:

- **Fijar valores de parámetros.-** En éste subproceso se considera tamaños iniciales para los cromosomas, la distancia máxima en la cual fueron detectados los vehículos, el número de cromosomas en la población inicial y el número de generaciones, según el estudio realizado en (Nguyen et al., 2013) se define un tamaño inicial de cromosoma de 20 y 15 pixeles para el ancho y alto respectivamente, 120 metros para la distancia máxima detectada, 1000 cromosomas para la población inicial y 5 generaciones.
- **Generación de población inicial (cromosomas).-** En ésta fase se crean los cromosomas de la población inicial, esto se lo hace mediante un proceso

gaussiano (Lizotte, Wang, Bowling, & Schuurmans, 2007) con el fin de asegurar la distribución uniforme con respecto al centro de la imagen.

- **Evaluación de la función de adaptabilidad (Fitness Function) (Vericat et al., 2011).**- El valor de ésta función nos indica que tan adaptable es un cromosoma al problema, para ello se considera las siguientes 5 características en los cromosomas:

- **Error de disparidad.**- Es la razón que existe entre la disparidad dominante y el área de la región del cromosoma, la ecuación es la siguiente para esta característica:

$$G_{de}^{(i,t)} = \frac{1}{w^{(i,t)}h^{(i,t)}} \left( w^{(i,t)}h^{(i,t)} - \sum_{j=0}^{h^{(i,t)}} \sum_{k=0}^{w^{(i,t)}} d(j, t, k) \right) \quad (5)$$

Donde,  $w^{(i,t)}$  y  $h^{(i,t)}$  son el ancho y el alto de cromosoma  $i$  en el tiempo  $t$ , respectivamente;  $d(j, t, k)$  es definida como la función auxiliar como sigue:

$$d(j, t, k) = \begin{cases} 1, & \text{Si } D_v(j, k, t) = D_v^{(i,t)} \\ 0, & \text{Si } D_v(j, k, t) \neq D_v^{(i,t)} \end{cases} \quad (6)$$

Donde,  $D_v(j, k, t)$  es el valor de disparidad en la posición  $(j,k)$  en el tiempo  $t$  y  $D_v^{(i,t)}$  es el valor dominante de disparidad en la región del cromosoma.

- **Densidad en los bordes de los cromosomas.**- Esta característica consiste en la forma que se observan los vehículos en las carreteras ya sea por la parte frontal o trasera, el mismo consiste de cuatro esquinas, de aquí que se consideran las mismas en el cálculo de la densidad en los bordes del cromosoma sobre la imagen izquierda del sistema estéreo, basados en parámetros que definen los anchos de bordes tanto verticalmente y horizontalmente en el estudio realizado por (Nguyen et al., 2013) se considera estos anchos con el valor de 3, la ecuación es la siguiente para esta característica:

$$G_l^{(i,t)} = \frac{1}{(2n+1)h^{(i,t)}} \sum_{j=0}^{h^{(i,t)}} \sum_{k=-n}^n I_{eg}(G_{uy}^{(i,t)} + j, G_{ux}^{(i,t)} + k) \quad (7)$$

$$G_r^{(i,t)} = \frac{1}{(2n+1)h^{(i,t)}} \sum_{j=0}^{h^{(i,t)}} \sum_{k=-n}^n I_{eg}(G_{uy}^{(i,t)} + j, G_{lx}^{(i,t)} + k) \quad (8)$$

$$G_{to}^{(i,t)} = \frac{1}{(2\gamma+1)w^{(i,t)}} \sum_{j=-\gamma}^{\gamma} \sum_{k=0}^{w^{(i,t)}} I_{eg}(G_{uy}^{(i,t)} + j, G_{ux}^{(i,t)} + k) \quad (9)$$

$$G_b^{(i,t)} = \frac{1}{(2\gamma+1)w^{(i,t)}} \sum_{j=-\gamma}^{\gamma} \sum_{k=0}^{w^{(i,t)}} I_{eg}(G_{ly}^{(i,t)} + j, G_{ux}^{(i,t)} + k) \quad (10)$$

Donde,  $G_l^{(i,t)}$ ,  $G_r^{(i,t)}$ ,  $G_{to}^{(i,t)}$ ,  $G_b^{(i,t)}$  son las densidades de pixeles a lo largo de los lados izquierdo, derecho, superior e inferior del cromosoma  $i$ ,

respectivamente en el tiempo  $t$ .  $I_{eg}(i,j)$  es el valor de intensidad en el pixel  $(i,j)$  en la imagen y  $n$  es el valor del ancho para los lados izquierdo y derecho en pixeles e  $\gamma$  es el valor del ancho para los lados superior e inferior en pixeles.

- **Distancia del sistema estéreo hacia el cromosoma.**- Esta distancia es obtenida en base a la información de disparidad, revisar el estado del arte numeral 1.12, ésta característica aumenta el valor de la función de adaptabilidad para vehículos muy cercanos y decrementa el mismo para obstáculos lejanos como construcciones, luces de tráfico, señales de tránsito las cuales tienen cuatro bordes que son considerados en las formas de los vehículos, la medición se la hace con la siguiente formula:

$$G_d^{(i,t)} = \frac{1}{\lambda_{max}} D_v^{(i,t)} f b \quad (11)$$

Donde,  $\lambda_{max}$  es la distancia máxima en la que se detecta un vehículo por el sistema propuesto,  $f$  es la distancia focal de la cámara en pixeles y  $b$  es la distancia entre las cámaras en metros.

- **Objetos cercanos al centro de la imagen.**- Para descartar objetos lejanos al centro de la imagen se utiliza la distancia euclidiana, el modelo matemático utilizado es el siguiente:

$$G_{rd}^{(i,t)} = \frac{1 - \sqrt{(G_{ux}^{(i,t)} - x)^2 + (G_{uy}^{(i,t)} - y)^2}}{\sqrt{x^2 + y^2}} \quad (12)$$

Donde,  $(x,y)$  es el centro de la imagen izquierda.

- **Detección de movimiento.**- Para situaciones de movimiento los cromosomas localizados en objetos que se mueven aportan mayor valor a la función de adaptabilidad que objetos estáticos, se calcula mediante la siguiente formula:

$$G_m^{(i,t)} = \frac{1}{w^{(i,t)} h^{(i,t)}} \sum_{j=0}^{h(i)} \sum_{k=0}^{w(i)} B(|I^{(t)}(j,k) - I^{(t-1)}(j,k)|) \quad (13)$$

$$B(I) = \begin{cases} 1, & Si I \geq w \\ 0, & Si I < w \end{cases} \quad (14)$$

Donde,  $I^{(t)}$  e  $I^{(t-1)}$  son los valores de intensidad para la imagen original izquierda en el tiempo  $t$  y  $t-1$  respectivamente; y  $w$  es un umbral binario.

Finalmente todas estas características se combinan en la función de adaptabilidad de la siguiente manera:

$$F^{(i,t)} = -C_{de}G_{de}^{(i,t)} + C_lG_l^{(i,t)} + C_rG_r^{(i,t)} + C_{to}G_{to}^{(i,t)} + C_bG_b^{(i,t)} + C_dG_d^{(i,t)} + C_{rd}G_{rd}^{(i,t)} + C_mG_m^{(i,t)} \quad (15)$$

Donde,  $G_{de}^{(i,t)}$  es el valor del error de disparidad;  $G_l^{(i,t)}$ ,  $G_r^{(i,t)}$ ,  $G_{to}^{(i,t)}$  y  $G_b^{(i,t)}$  son los valores de densidad en los bordes del cromosoma en los lados izquierdo, derecho, arriba y abajo, respectivamente;  $G_d^{(i,t)}$  es el valor de disparidad del cromosoma;  $G_{rd}^{(i,t)}$  es la distancia euclidiana al centro de la imagen;  $G_m^{(i,t)}$  es el desplazamiento del cromosoma y  $C_x$  son constantes de coeficientes para cada variable.

En resumen según (Nguyen et al., 2013), se ha diseñado una función de adaptabilidad que aumenta el valor de la misma en los cromosomas que se encuentran cerca del centro de la imagen, tienen cuatro esquinas, están más cerca del sistema propuesto, estaban en movimiento, tienen una región de disparidad homogénea, y finalmente disminuyen los valores de adaptabilidad en otras posiciones.

- **Torneo de selección.-** En este se eligen los mejores individuos para utilizarlos en el torneo de cruzamiento en base al valor que se obtiene en el cálculo de la función de adaptabilidad.
- **Torneo de cruzamiento.-** Este proceso se lleva a cabo mediante la comparación del valor de disparidad dominante en las regiones de dos cromosomas y sus coordenadas. Las coordenadas de los nuevos cromosomas son calculadas con las funciones de máximos y mínimos con las siguientes formulas:

$$G_{ux}^{(hijo)} = \min(G_{ux}^{(i)}, G_{ux}^{(j)}), G_{uy}^{(hijo)} = \min(G_{uy}^{(i)}, G_{uy}^{(j)}) \quad (16)$$

$$G_{lx}^{(hijo)} = \max(G_{lx}^{(i)}, G_{lx}^{(j)}), G_{ly}^{(hijo)} = \max(G_{ly}^{(i)}, G_{ly}^{(j)}) \quad (17)$$

**Verificación de Hipótesis.-** En ésta etapa de verificación de cromosomas se remueve candidatos inválidos basados en el movimiento e información del mapa V-disparity (contiene información de obstáculos en la carretera). La verificación se realiza utilizando la siguiente formula:

$$V(i, t) = \begin{cases} Verdadero, & \text{Si } f_m(i, t) \geq \alpha_m \text{ y } f_{v-d}(i, t) \geq \alpha_{v-d} \\ Falso, & \text{Otro caso} \end{cases} \quad (18)$$

Donde,  $\alpha_m$  y  $\alpha_{v-d}$  son valores predefinidos en este método propuesto se utilizaron con un valor de 0,5 cada uno, las funciones auxiliares  $f_m(i, t)$  y  $f_{v-d}(i, t)$  son utilizadas para medir la razon de movimiento y la información de disparidad, respectivamente.

$$f_m(i, t) = \frac{1}{wh} \sum_{j=0}^h \sum_{k=0}^w B(|I(j, k, t) - I(j, k, t - 1)|) \quad (19)$$

$$b(j, k) = \begin{cases} 1, & \text{Si } d(j, k) > 0 \\ 0, & \text{Otro caso} \end{cases} \quad (20)$$

Donde,  $B(I)$  es calculado en la detección del movimiento,  $w$  y  $h$  es el ancho y alto de los cromosomas respectivamente y  $d(i, j)$  es el valor de disparidad en el mapa V-Disparity.

- **Seguimiento de Vehículos Simple.**- El proceso se realiza mediante una reducción de la población a la mitad de la fase de detección y se limita el espacio de búsqueda para obtener buenos resultados en cuanto al tiempo de ejecución, la reducción del espacio de búsqueda  $(x, y)$  se lo hace basado en la posición de la previa detección con las siguientes formulas.

$$x_u^l - w \leq x \leq x_{lo}^r + w, w = x_{lo}^r - x_u^l \quad (21)$$

$$y_u^l - h \leq y \leq y_{lo}^r + h, h = y_{lo}^r - y_u^l \quad (22)$$

Donde,  $(x_u^l, y_u^l)$  y  $(x_{lo}^r, y_{lo}^r)$  son la posición de la esquina superior izquierda y la esquina inferior derecha del cromosoma previamente detectado respectivamente.

### 1.1.2 Selección del método para la detección de vehículos.

En ésta tarea se utilizaran las siguientes métricas para proceder a elegir un método para la detección de vehículos:

- **Tiempo de procesamiento.**- Es el tiempo que toma procesar una imagen para detectar un vehículo.
- **Máxima distancia detectada.**- Es la distancia máxima a la cual un método puede detectar un auto.
- **Direcciones de detección.**- Son las vistas tanto frontal o trasera en las cuales el algoritmo puede detectar los vehículos.
- **Tasa de detección de vehículos.**- Es la relación que existe entre el número total de vehículos detectados y la cantidad de vehículos reales.

- Combinación de técnicas de cada método.- Son todas las consideraciones tomadas para hacer más preciso el método al realizar la detección.
- Condiciones de iluminación.- Son las condiciones en las cuales se exponen los vehículos en las imágenes con bajas, normales o altas cantidades de iluminación.
- Número de imágenes en la pruebas.- Es el número total de imágenes utilizadas para probar un método.
- Porcentaje de falsas alarmas.- Es la relación que existe entre las regiones detectadas incorrectamente como vehículos y el total de imágenes que no contienen vehículos que intervienen en las pruebas.

Tabla 2. Comparación de métodos para la detección de vehículos.

Author	DR %	Detection direction	Processing time (ms)	MDD (m)
Nedevschi	94	Front and rear	100	90
Yi-Min Tsai	97.1	Rear	970	140
Toulminet	90	Rear	95	100
Yaqian Li	91	Front and rear	50	105
Xu	85	Front and rear	155.5	70.58
Nedevschi	96.69	Front and rear	23	N/A
S. Sivaraman	93.75	Rear	57	88
R. Wang	95.6	Rear	510	89
Y. M. Chan	85	Front and rear	47	105
Bin-dai	93	Rear	96.9	90
B. Leibe	98.5	Rear	2150.5	110
Proposed method	96.08	Front and rear	48	140

Fuente: (Nguyen et al., 2013).

La selección del método para la detección de vehículos se basó en los resultados obtenidos en (Nguyen et al., 2013), en el cual podemos apreciar que el algoritmo genético propuesto en esta investigación con sus cinco características, proporciona una exactitud promedio de detección de vehículos de 96.08%, los reconocimientos de vehículos los realiza por visualización posterior y frontal, además cuenta con una gran distancia máxima de detección de autos de 140 m, así también se tiene un tiempo de procesamiento promedio de 48 ms.

Tabla 3. Tabla de escenarios con configuraciones diferentes para la detección de vehículos.

Seq	Configuration	Illumination	# images	Feature system		Genetic system		Motion system		Stixel system		Our system	
				DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)
SQ01	b35,f16	Normal	110	92	8	92	8	93	3	95	1	100	0
SQ02	b60,f25	Normal	310	91	5	89	7	92	5	94	2	97	3
SQ03	b60,f25	Low, changed	76	78	7	75	5	82	6	85	6	90	3
SQ04	b60,f25	Normal	115	89	3	80	7	95	7	98	2	100	0
SQ05	b35,f25	Low	225	80	4	85	3	94	3	95	3	96	4
SQ06	b60,f16	High, changed	85	82	8	79	6	85	4	88	4	91	4
SQ07	b35,f16	Normal	197	88	4	84	4	96	2	98	2	98	2
SQ08	b35,f25	Normal	126	90	6	86	9	92	3	100	3	97	5
SQ09	b35,f16	Low, changed	597	79	7	89	11	92	5	93	2	98	2
SQ10	b35,f16	High	754	85	5	90	8	96	2	100	4	95	3
SQ11	b35,f16	High	800	87	11	85	9	92	6	98	5	94	4
SQ12	b35,f16	High	505	82	15	79	16	92	5	97	3	97	3
Average				85.25	6.92	84.42	7.75	91.75	4.25	95.58	3.08	96.08	2.75

Fuente: (Nguyen et al., 2013)

Además en la Tabla 3 se puede observar un conjunto de 12 escenarios, en los cuales el porcentaje promedio de falsas alarmas en la detección de vehículos es del 2.75% para el algoritmo genético propuesto en (Nguyen et al., 2013), la menor en todos los métodos estudiados, así mismo es apreciable que el algoritmo responde de buena manera en diferentes condiciones de luz. Por todos estos datos favorables elegimos el algoritmo genético expuesto en (Nguyen et al., 2013) para utilizarlo e implementarlo en nuestro proyecto para lograr el objetivo específico de detectar vehículos.

### **Extracción de la región de la placa.**

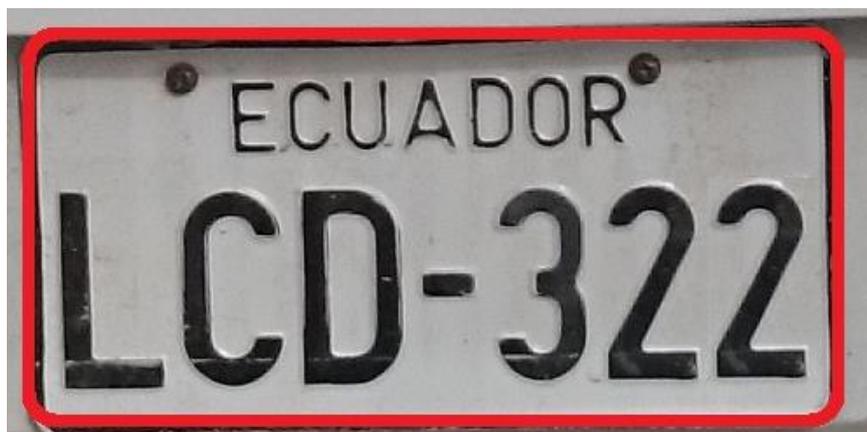


Figura 10. Imagen resultante de la detección de la región de la placa.  
Elaboración: Juan Pablo Ordoñez Lopez

La etapa de extracción de placa influye en la precisión de un sistema LPR (License plate recognition). La entrada a ésta etapa es una imagen de un auto, y la salida es una porción de la imagen que contiene la región de placa. La región de placa puede existir en cualquier parte de la imagen. En lugar de procesar cada píxel en la imagen, lo que aumenta el tiempo de procesamiento, la placa se puede distinguir por sus características, y por lo tanto el sistema procesa sólo los píxeles que tienen estas características. Las características se derivan del formato de la placa y los caracteres que la constituyen. El color de la misma es una de las características ya que algunas jurisdicciones (es decir, países, estados o provincias) tienen ciertos colores para sus placas de matrícula. La forma rectangular de los límites de la placa es otra característica que se utiliza para extraer la misma. La diferencia de color entre los caracteres y el fondo de la placa, conocida como la textura, se utiliza para extraer la región de la placa en la imagen. La existencia de caracteres se puede usar como una característica para identificar la región de la placa. Dos o más características se pueden combinar para identificar la placa.

### 1.1.3 Métodos para la detección de placas en vehículos.

#### 1.1.3.1 Filtro de Sobel.

Dado que la placa de matrícula normalmente tiene una forma rectangular con una relación de aspecto conocido, se puede extraer mediante la búsqueda de todos los rectángulos posibles en la imagen. Los métodos de detección de borde se utilizan comúnmente para encontrar estos rectángulos. Un método conocido es el filtro de Sobel usado para detectar bordes. Debido a la transición de color entre la placa y la carrocería del vehículo, el límite de la placa está representado por bordes en la imagen. Los bordes son dos líneas horizontales al realizar la detección de bordes horizontales, dos líneas verticales cuando se realiza la detección de bordes verticales y un rectángulo completo cuando se realiza ambas tareas al mismo tiempo.

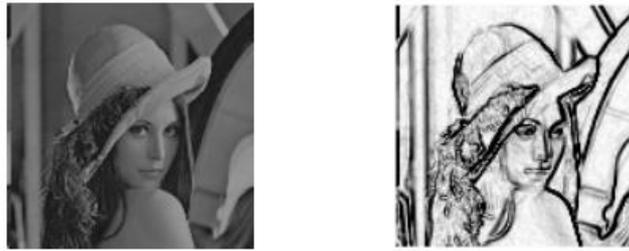


Figura 11. Aplicando el método de la gradiente.  
Fuente: (Vincent & Folorunso, 2009)

Los modelos matemáticos utilizados en el filtro de Sobel son los siguientes (Vincent & Folorunso, 2009):

$$\Delta x = f(i + 1, j) - f(i, j) \quad (23)$$

$$\Delta y = f(i, j + 1) - f(i, j) \quad (24)$$

$$M = \sqrt{\Delta x^2 + \Delta y^2} \quad (25)$$

$$\theta = \tan^{-1} \left[ \frac{\Delta y}{\Delta x} \right] \quad (26)$$

Donde (i,j) son las coordenadas de cada pixel en la imagen, f es la función de intensidad de cada pixel,  $\Delta x$  es el cambio de intensidad en dos pixeles consecutivos en el eje x,  $\Delta y$  es el cambio de intensidad en dos pixeles consecutivos en el eje y, M es valor de la gradiente y  $\theta$  es la dirección en la que se dirige la gradiente.

Para la extracción de la región de placa se toma en consideración el atributo geométrico para la localización de las líneas que forman un rectángulo. Regiones candidatas son generadas mediante la coincidencia sólo entre sus bordes verticales. La magnitud de los bordes verticales del área de placa se considera una característica de extracción robusta, durante el uso sólo de los bordes horizontales puede dar lugar a errores debidos a topes o formas horizontales del vehículo, que normalmente son más comunes. Los bordes verticales se hacen coincidir para obtener algunos rectángulos candidatos. Luego estos son clasificados como candidatos al tener la misma relación de aspecto que la región de la placa de matrícula.

### 1.1.3.2 Basado en bloques.

Otro método es el basado en bloques, se maneja mediante la búsqueda de magnitudes elevadas de borde, las cuales son identificadas como posibles áreas de placa.

El procesamiento de bloque no depende de los bordes de los límites de la región de placa, éste se puede aplicar a una imagen con un área de placa con ruido en los límites.

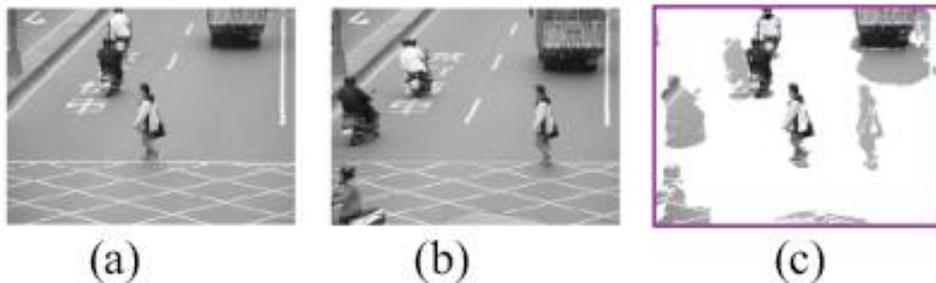


Figura 12. Imagen a y b son dos imágenes de secuencia y c elimina los estáticos.  
Fuente: (Lee et al., 2004)

La ecuación que mide la diferencia de bloques es la siguiente (Lee et al., 2004):

$$diff = \frac{\left[ \left( \frac{S_P + S_R}{2} \right) - \left( \frac{M_P - M_R}{2} \right)^2 \right]^2}{(S_P * S_R)} \quad (27)$$

Donde,  $M$  y  $S$  denotan la media y la varianza para el bloque en el procesamiento (índice  $P$ ) y la referencia de imágenes (índice  $R$ ), respectivamente. Si la medida  $diff$  es menor que un umbral, entonces el bloque se clasifica como estacionario.

### 1.1.3.3 La transformada de Hough.

Otro método es la extracción basada en límites para ello se utiliza la transformada de Hough (HT o Hough transform). Esta consiste en detectar líneas rectas en la imagen para localizar la placa de matrícula. La transformada de Hough tiene la ventaja de la detección de líneas rectas con un máximo de 30° de inclinación. Sin embargo, la transformada de Hough es un proceso que consume tiempo y memoria.

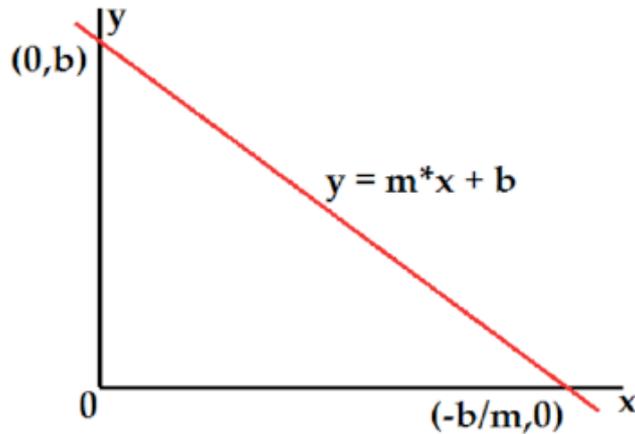


Figura 13. Línea recta en el sistema cartesiano.  
Fuente: (Barba, 2015).

El modelo matemático a utilizar es la ecuación de la recta según se define en (Barba, 2015).

$$y = m * x + b \quad (28)$$

Donde,  $m$  es la pendiente de la recta y  $b$  es una constante de desplazamiento en el eje  $y$ .

### 1.1.3.4 Análisis de componentes Conectados.

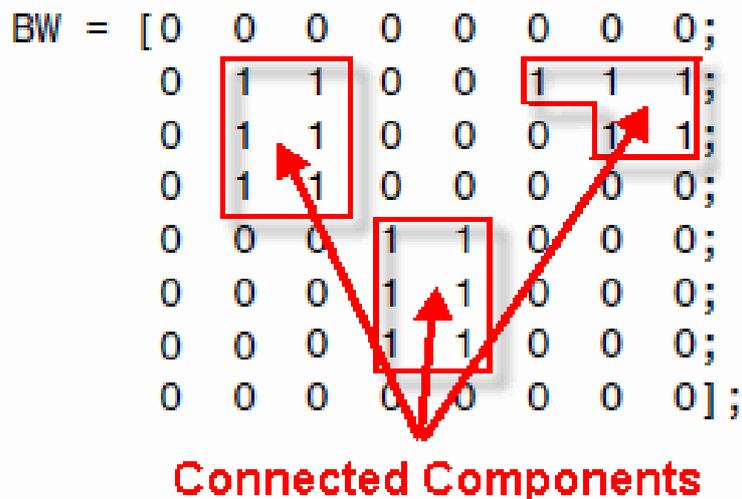


Figura 14. Región de la placa representada como rectángulos.  
Fuente: ("Matlab binary image," 2015).

Análisis de componentes Conectados (CCA o Connected component analysis) es una técnica importante en el procesamiento de imágenes binarias. Escanea una imagen binaria y etiqueta sus píxeles en componentes basados en la conectividad de los mismos. Mediciones espaciales, como el área y la relación de aspecto, se utilizan comúnmente para la extracción de la placa. Se usan algoritmos de detección de contorno sobre la imagen binaria para detectar objetos conectados. Los objetos conectados que tienen las mismas características geométricas como la placa se eligen como candidatos. Este algoritmo puede fallar en el caso de imágenes de mala calidad, lo que resulta en contornos distorsionados.

El modelo matemático a utilizar está basado en la relación de aspecto según la siguiente igualdad:

$$r(i) = \frac{H(i)}{W(i)} \quad (29)$$

Donde,  $W(i)$  es el ancho y  $H(i)$  es el alto de cada componente conectado respectivamente y  $r(i)$  es la relación de longitud entre ambos, siendo el valor de  $r(i)$  que mas se aproxime a la relación de longitud de la placa  $R=(\text{alto de la placa})/(\text{ancho de la placa})$  un posible candidato a región de placa de vehículo.

#### 1.1.3.5 Métodos 2-D de correlación cruzada.

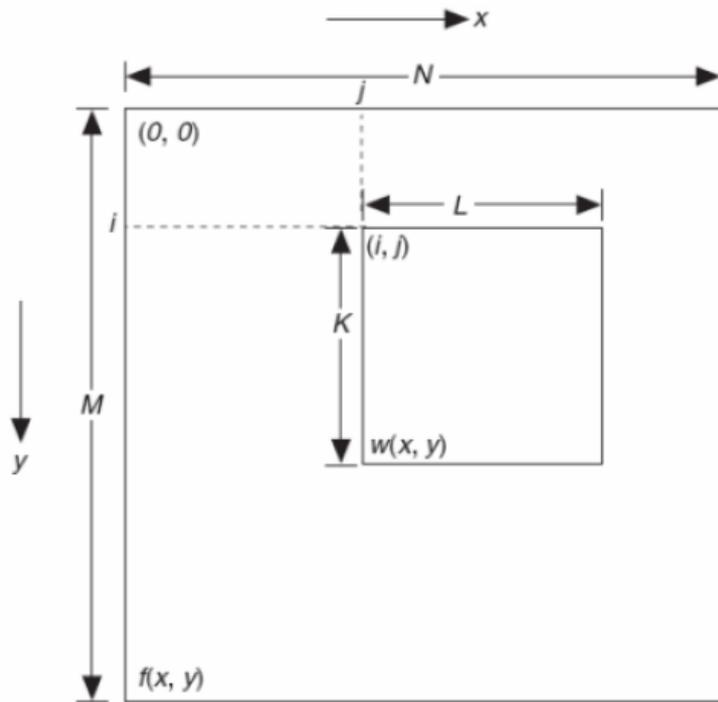


Figura 15. Coincidencia de plantillas en la detección de la placa.  
Fuente: (Corporation, 2010).

También se utilizan métodos 2-D de correlación cruzada. Este método usa una plantilla de región de placa pre almacenado mediante la cual se realiza una búsqueda a través de toda la imagen para localizar el área de placa más probable. La extracción de las áreas de placas utilizando correlación con una plantilla es independiente de la posición de la placa en la imagen. Sin embargo, la correlación cruzada 2-D toma mucho tiempo en procesamiento. Es del orden de  $n^4$  para una imagen de  $n \times n$  pixeles.

Este método se basa en la suma de productos:

$$C_{(i,j)} = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) f(x + i, y + j) \quad (30)$$

Donde  $w(x, y)$  es la imagen patrón y  $f(x+i, y+j)$  son todos los posibles bloques de la imagen original con ancho y alto  $L$  y  $K$  respectivamente.

Indicando el máximo valor para  $C(i, j)$  la región de imagen que más se ajusta a la imagen patrón, en este caso se considera una región de placa candidata.

### 1.1.3.6 Cascada de clasificadores.

Otro método es la cascada de clasificadores, éste algoritmo introducido por Viola y Jones en (Viola & Jones, 2001) nos permite saber dónde se encuentra cualquier objeto dentro de una imagen en el espacio y escala. Para la clasificación se utiliza un conjunto combinado de clasificadores los cuales se encargan de desechar las regiones consideradas como fondo de una imagen para poder emplear la mayor cantidad de tiempo y recursos sobre regiones prometedoras que son posibles candidatas a los objetos buscados.

Este método está basado en una cascada de clasificadores de tres procesos principales:

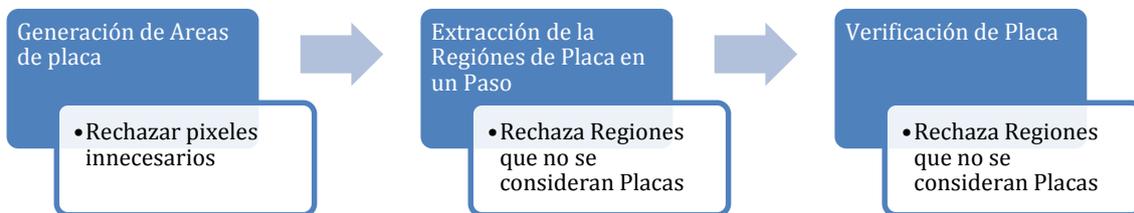


Figura 16. Proceso Detección de la Región de la Placa.

Fuente: (Wang et al., 2007).

Elaboración: Juan Pablo Ordoñez Lopez

- Generación de áreas de placa.- En ésta etapa se aplica la gradiente vertical, con el fin de obtener solo los bordes verticales, como resultado del proceso mencionado las regiones de placas aparecen más claras que al aplicarles otros métodos según (Wang et al., 2007), además se aplica el método de OTSU para obtener los píxeles más relevantes como subproceso adicional a la obtención de los bordes verticales.
- Extracción de la Región de la Placa en un solo paso.- En ésta etapa se consideran las siguientes reglas para eliminar las regiones de la imagen que no se consideran áreas de placa: Tramos verticales de Placa que sean menores a la altura de la placa  $H_p$ , tramos horizontales de Placa que sean mayores a la anchura de la placa  $W_p$ , tramos de placa localizados entre dos regiones de fondo que son mayores al máximo ancho de los caracteres de la placa.  $H_p$  y  $W_p$  se consideran con valores de 15 y 60 píxeles, respectivamente en (Wang et al., 2007).
- Verificación de Placa.- La verificación de las posibles placas se lo hace mediante un algoritmo de impulso adaptativo (AdaBoost) para más detalles ver (Wang et al., 2007), para ello se debe entrenar el mismo con una muestra de imágenes que se consideran regiones de placa y con otra que no se consideran regiones de placa,

con el fin de obtener las características más relevantes que representan una región de placa.

Este método utiliza la imagen integral para la extracción de características que es una versión transformada de la imagen original que se crea previamente para su evaluación con las características Haar-Like, este concepto fue creado con el fin de reducir los altos costos de procesamiento sobre la imagen original (Barba, 2015).

Para realizar el cálculo de la imagen integral se debe sumar todos los valores de las celdas (píxeles) que se encuentren a la izquierda y sobre un punto (x,y) en el plano (imagen original) según se puede observar en la Figura 17 a). A partir de este cálculo se obtiene la segunda matriz en la Figura 17 b).



Original					Integral				
5	2	3	4	1	5	7	10	14	15
1	5	4	2	3	6	13	20	26	30
2	2	1	3	4	8	17	25	34	42
3	5	6	4	5	11	25	39	52	65
4	1	3	2	6	15	30	47	62	81

$$5 + 2 + 3 + 1 + 5 + 4 = 20$$

Figura 17. a) Suma valores al punto (x,y)  
Fuente: ("Haar-feature Object Detection in C#," 2014).

b) Calculo Imagen Integral

Mediante la matriz integral se puede calcular la intensidad de cualquier bloque de la imagen original definido por los puntos:  $p(x_a, y_a)$ ,  $p(x_a, y_b)$ ,  $p(x_b, y_a)$ ,  $p(x_b, y_b)$  con la siguiente ecuación según (Gao & Lu, 2008), una descripción grafica se puede apreciar en la Figura 18.

$$\sum_{x=x_a}^{x_b} \sum_{y=y_a}^{y_b} i(x, y) = ii(x_b, y_b) - ii(x_{a-1}, y_b) - ii(x_b, y_{a-1}) + ii(x_{a-1}, y_{a-1}) \quad (31)$$

Donde,  $x_a, x_b, y_a$  y  $y_b$ , son los intervalos en x e y para obtener la sumatoria o valor de un area específico en la imagen,  $ii(x_b, y_b)$  es el valor de la imagen integral en la posición  $(x_b, y_b)$ ,  $ii(x_{a-1}, y_b)$  es el valor de la imagen integral en la posición  $(x_{a-1}, y_b)$ ,  $ii(x_b, y_{a-1})$  es el valor de la imagen integral en la posición  $(x_b, y_{a-1})$  y  $ii(x_{a-1}, y_{a-1})$  es el valor de la imagen integral en la posición  $(x_{a-1}, y_{a-1})$ .

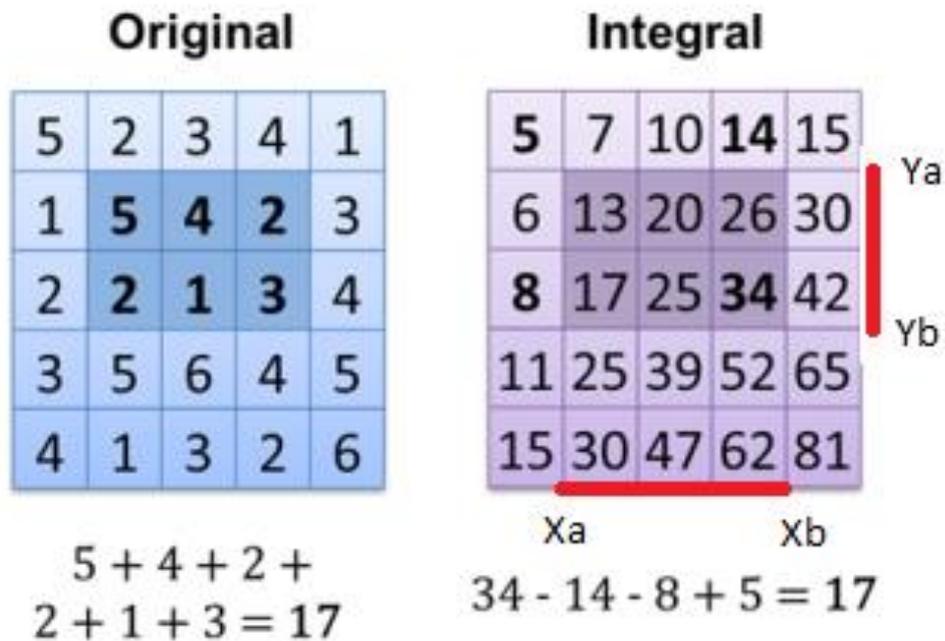


Figura 18. Cálculo del área para una región de la imagen original, basados en la imagen integral.

Fuente: ("Haar-feature Object Detection in C#," 2014).

Como es observable el cálculo para la suma de intensidad de cualquier bloque de la dimensión que sea, siempre será constante a partir de la imagen integrada, pues se requiere de operaciones simples sobre los valores de cuatro puntos de esta matriz.

#### 1.1.4 Selección del método para la detección de la región de placa.

Para la tarea en cuestión se utiliza las siguientes métricas, para proceder a elegir un método para la detección de la región de la placa en vehículos:

- Tiempo de procesamiento.- Es el tiempo que toma procesar una imagen para detectar la región de la placa en vehículos.
- Tasa de aciertos en la detección de regiones de placa.- Es la relación que existe entre el número total de regiones de placa detectadas y la cantidad real de placas a detectar.
- Combinación de técnicas de cada método.- Son las técnicas consideradas en los procesos para realizar con mayor exactitud la detección de regiones de placa.
- Condiciones de iluminación.- Son las condiciones en las cuales se exponen los vehículos en las imágenes con bajas, normales o altas cantidades de iluminación.

- Número de imágenes en la pruebas.- Es el número total de imágenes utilizadas en la experimentación de un método.
- Porcentaje de falsas alarmas.- Es la relación que existe entre la detección de las regiones que no son placa y el total de áreas de placa reales que intervienen en las pruebas.
- Similitud de formatos de placas de otros países con el formato estándar para Ecuador.

La selección del algoritmo para la detección de la región de la placa se basa en las investigaciones realizadas en (Du et al., 2013), en la Tabla 3 se presenta un resumen de la comparación de métodos tanto para la extracción de la región de la placa, la segmentación de los caracteres y el reconocimiento de los mismos, según los datos presentados se elige el algoritmo planteado en (Wang et al., 2007) por proporcionar una tasa alta de detección de la región de placa del 99,6% en un sistema de tiempo real que tiene la capacidad para resolver el reconocimiento de caracteres de una placa en un tiempo de 30ms para imágenes de 640x480 píxeles, además el sistema reconoce placas taiwanesas, las cuales son muy similares a las placas ecuatorianas como se puede apreciar en la Figura 19.



Figura 19. Placas Taiwanesa y Ecuatoriana.  
Fuente: ("Vehicle registration plates of Taiwan," 2007).  
Elaboración: Juan Pablo Ordoñez Lopez

Es considerable la cantidad de imágenes utilizadas para el proceso de pruebas del sistema según se indica en (Wang et al., 2007) un total de 11896 imágenes fueron empleadas, detectando el algoritmo 11598, son detectadas 516 imágenes como regiones falsas de placas, así mismo se indica que la eficacia del método se basa en la adopción de un framework en cascada, la utilización de las características de la placa, y el desarrollo de algoritmos de paso rápido.

Tabla 4. Comparación de métodos para la extracción de la región de la placa, segmentación de caracteres y reconocimiento de los mismos.

Método	Procedimientos principales			Tamaño de la base de datos	Condiciones de la imagen	Tasa de extracción de placas	Tasa de segmentación de caracteres	Tasa de Reconocimiento de caracteres	Tasa total del método	Tiempo de procesamiento	Formato de Placa
	Extracción de la región de la placa	Segmentación de caracteres	Reconocimiento de caracteres								
[8]	Bordes verticales	–	–	1165 imágenes	384 × 288 píxeles	~ 100%	–	–	–	47.9 ms	China
[18]	Procesamiento basado en bloques	–	Plantilla de coincidencia	180 pares de imágenes	Multiplicas con oclusión y tamaños diferentes	94.4%	–	95.7%	–	75ms para extracción de placa	Taiwanese
[22]	Detección de borde y Proyecciones Verticales y Horizontales	Proyecciones Verticales y Horizontales	retropropagación redes neuronales	12 s video	320 × 240 píxeles	–	–	–	85.5% y ~ 100% después de reentrenamiento	100 ms	Taiwanese
[5]	Estadísticas de borde y morfología	–	–	9825 imágenes	768 × 534 píxeles, diferentes condiciones de iluminación	99.6%	–	–	–	100 ms	China
[26]	Análisis de componentes conectados	–	–	4 hrs + video	320 × 240 píxeles, baja resolución	96.6%	–	–	–	30 ms	Taiwanese

					degradada de vídeo						
[50]	Ventanas concéntricas sobre corredores	Ventanas concéntricas sobre corredores	Redes neuronales probabilísticas de dos capas	1334 imágenes	Fondos e iluminación diferentes	96.5%	–	89.1%	86%	276 ms (111 ms para extracción de la región de las placas, 37 ms para segmentación de caracteres, and 128 ms para reconocimiento de caracteres)	Griega
[53]	Características Haar-Like locales e impulso adaptativo	–	–	160 imágenes	648 x 486 pixeles, varias condiciones y ángulos de vista	93.5%	–	–	–	80 ms	Australia
[58]	Características Haar-Like e impulso adaptativo en cascada	Análisis de picos y valles	SVM basado en CCs, DCs, y PBA	11 896 imágenes	640x480 pixeles, diferentes formatos, tamaños y condiciones de luz	99.6%	–	–	98.3%	30 ms	Taiwanese

[14]	-	Escaneo de líneas y proyección vertical	-	30000+ imágenes	-	-	99.2%	-	-	10-20 ms para segmentación de caracteres	China
------	---	---	---	-----------------	---	---	-------	---	---	--	-------

Fuente: (Du et al., 2013).

## Segmentación de los caracteres de la región de la placa.



Figura 20. Caracteres segmentados de la región de la placa.  
Elaboración: Juan Pablo Ordoñez Lopez

Una vez aislada la región de placa de los vehículos, se procede a realizar el segmentado de los caracteres para extraer y poderlos reconocer. En la etapa anterior en la extracción del área de la placa se puede tener algunos problemas, tales como la inclinación y el brillo no uniforme. Entonces los algoritmos de segmentación de caracteres deben superar todos estos problemas en una etapa de pre procesamiento.

Para las tareas correctivas se utilizan métodos como la transformación bilineal para mapear el número de placa inclinada extraída a un rectángulo recto. Otro método es el de los mínimos cuadrados que también se usa para tratar inclinaciones horizontales y verticales en imágenes de placas de matrícula.

### 1.1.5 Métodos para la segmentación de los caracteres de la placa.

#### 1.1.5.1 Análisis de componentes Conectados.

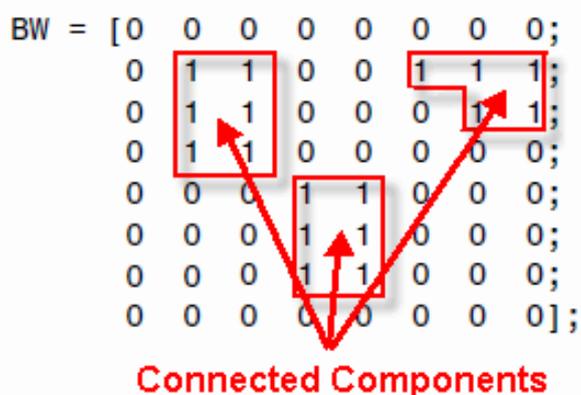


Figura 21. Caracteres como rectángulos.  
Fuente: ("Matlab binary image," 2015).

En (Vorobioff & G, 2011) mediante un algoritmo de componentes conectados se etiqueta de forma única y se construye un conjunto de componentes que se considerará para la identificación de una línea de texto.

Para el análisis de componentes identificados se realiza el siguiente proceso, primero se busca el  $H_{cavg}$  y  $W_{cavg}$  o altura y ancho promedio de los componentes, respectivamente, se calcula de la siguiente manera:

Sea  $N$  el número total de componentes, entonces la altura del componente toma valores del conjunto de datos finito  $x_1, x_2, \dots, x_N$ .

De los mismos se obtiene que la media aritmética de las alturas de los componentes es  $\mu$ . Entonces,

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (32)$$

Sólo los componentes con una altura que oscila entre  $\mu - \sigma$  a  $\mu + \sigma$  se consideran para el cálculo de la media  $H_{cavg}$  de los componentes, que es la media aritmética de todos esos valores. El promedio  $W_{cavg}$  de los componentes se calcula de una manera similar. Se ha observado en algunos casos de texto escrito a mano que un carácter que pertenece a una determinada línea puede tocar a otro carácter que pertenece a una línea superior o inferior. Estos dos caracteres se identifican como un solo componente por el algoritmo y tienen una altura mucho mayor que la  $H_{cavg}$  altura promedio que se calcula utilizando el método anteriormente mencionado.

Por lo tanto se hace necesario que este componente se divida en sus caracteres constituyentes de modo que dos o más líneas no se fusionen en la etapa de identificación de la línea. Según el método expuesto en (Vorobioff & G, 2011), los componentes que tienen una altura superior a  $\delta * H_{cavg}$  se han dividido horizontalmente en dos mitades iguales; donde  $\delta$  es un parámetro de ajuste heurísticamente elegido. En el mismo, se ha supuesto que no más de dos líneas de texto consecutivos se toquen entre sí.

#### **1.1.5.2 Proyección horizontal y vertical.**

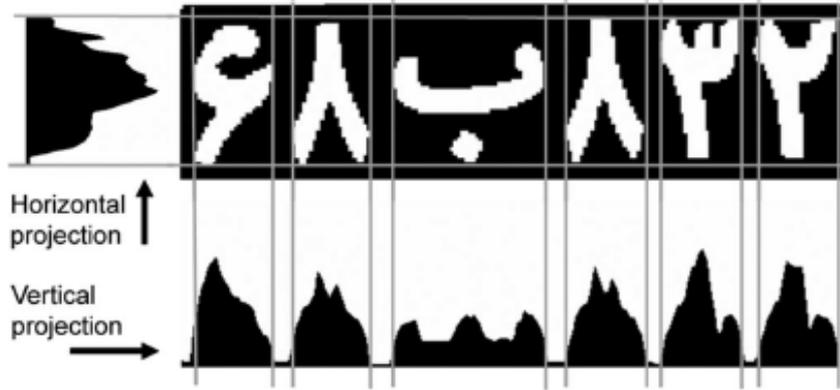


Figura 22. Extracción de caracteres utilizando el método de proyección horizontal y vertical.

Fuente: (Anagnostopoulos, Anagnostopoulos, Psoroulas, Loumos, & Kayafas, 2008).

En un proceso anterior se obtiene una imagen binaria, la idea es sumar los píxeles de columnas o filas de la imagen y obtener un vector (o proyección), cuyos valores mínimos nos permiten la segmentación de caracteres Figura 22. El análisis de componentes conectados o CCA (Connected component analysis) también está fuertemente involucrado en la segmentación de caracteres, conjuntamente con las mediciones o características geométricas de objetos binarios tales como altura, anchura, área y la orientación.

Por lo general, el método CCA etiqueta un conjunto de píxeles en componentes basados en conectividad de vecindad 4 u 8.

Las desigualdades utilizadas para seleccionar las líneas de un posible carácter según (Shi, Zhao, & Shen, 2005) se escriben a continuación:

$$\lambda_h < P_h(i) \quad (33)$$

$$\lambda_v < P_v(i) \quad (34)$$

y

$$whr + \rho > \frac{w(k)}{h(k)} > whr - \rho \quad (35)$$

Donde,  $\lambda_h$ ,  $\lambda_v$  es la mínima cantidad de píxeles para seleccionar las líneas horizontales y verticales, respectivamente;  $P_h(i)$ ,  $P_v(i)$  es la función cantidad de píxeles en una línea en orientación horizontal o vertical, respectivamente;  $whr$  es una constante de la relación entre el ancho y el alto de un carácter;  $w(k)$ ,  $h(k)$  son el ancho y el alto de un carácter encontrado  $k$  y  $\rho$  es una holgura utilizada en la desigualdad (35) para descartar falsas regiones de caracteres.

### 1.1.5.3 Análisis de picos y valles.

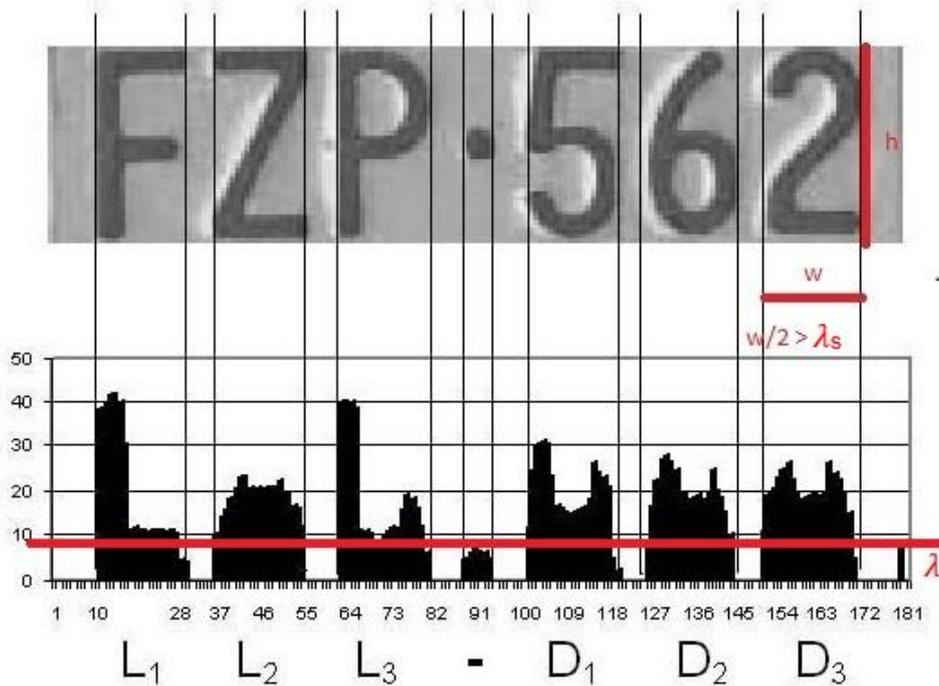


Figura 23. Picos y Valles basados en el histograma de la imagen.

Fuente: ("OpenCV Color Concentration Histogram," 2013).

Elaborado: Juan Pablo Ordoñez Lopez

En el trabajo expuesto en (Wang et al., 2007) se propone un método basado en una combinación de técnicas para resolver la segmentación de caracteres mediante dos procesos, el primer proceso se encarga de segmentar las imágenes mediante la técnica de componentes conectados resultados Figura 24. b) y c), luego un segundo proceso segmenta regiones de caracteres que contienen más de un carácter Figura 24. c) como una sola región de carácter detectado en el proceso anterior, ésta tarea se la realiza mediante proyecciones verticales de las imágenes de los caracteres indistintos obtenidas en el primer proceso. Para el cálculo de las proyecciones se utiliza tablas de aceleración (imagen integral) para la Sumarización de Áreas.



Figura 24. a) Placas compactas b) Caracteres distintos c) Caracteres indistintos

Fuente: (Wang et al., 2007).

En un perfil de proyección o histograma, un pico es un punto con el valor de la altura máxima entre dos mínimos locales.

Para considerar un punto como un pico, el valor debe ser el máximo local y al menos  $\lambda$  mayor que el mínimo local.

Del mismo modo, un punto de valle se define como un punto con el valor mínimo que es menor que  $\lambda$  entre dos máximos locales.

Para detectar todos los picos y valles, se aplican dos modos de decisión, llamados "modo de decisión de pico" y "modo de decisión de valle", están diseñados para determinar un pico y un valle, respectivamente. Por otra parte, se proponen dos procedimientos en cascada para detectar todos los picos y valles prometedores en un solo paso de exploración o escaneo.

El primer procedimiento se llama "modo de inicialización" (Wang et al., 2007), donde  $f(i)$  denota el valor del perfil en la posición  $i$ . El modo de encontrar picos o valles se determina de acuerdo con el aumento progresivo inicial o disminución en los valores de perfil. El segundo procedimiento se llama decisión pico-valle (Wang et al., 2007). Un punto se selecciona como un pico cuando su valor es superior a la de su vecino y el valle anterior  $\lambda$  en el modo de hallazgo de picos, mientras que un punto se selecciona como un valle cuando su valor es menor que la de su vecino y el pico anterior por  $\lambda$  en el modo de hallazgo de valle. Los modos de búsqueda de pico y valle se realizan alternativamente con el fin de determinar todos los picos y valles.

Los modelos matemáticos a utilizar son los siguientes:

Para decisión de picos,

$$f(Pcand) \geq f(i) + \lambda \quad (36)$$

Para decisión de valles,

$$f(i) \geq f(Vcand) + \lambda \quad (37)$$

Para relación de aspecto ancho y altura región del carácter,

$$\frac{w}{h} = \frac{1}{2} \quad (38)$$

Para descartar posibles regiones de caracteres candidatos que no cumplen con un valor de distancia mínima  $\lambda_s$ ,

$$\frac{w}{2} > \lambda_s \quad (39)$$

Donde,  $f(Pcand)$ ,  $f(Vcand)$  son las funciones de perfil para los candidatos temporales a picos y valles, respectivamente;  $f(i)$  es la función de perfil para la posición  $i$ ;  $w$  es el ancho,  $h$  la altura de la región que representa una posible región de carácter en la placa,  $\lambda_s$  representa un valor de referencia que tiene que ser menor a la mitad de  $w$  y  $\lambda$  el umbral considerado para la separación de caracteres.

### **1.1.6 Selección del método para la segmentación de caracteres en la región de la placa.**

En ésta tarea se utilizó las siguientes métricas para proceder a elegir un método para segmentar los caracteres de la región de la placa:

- Tiempo de procesamiento.- Es el tiempo que toma procesar una imagen para lograr la segmentación de los caracteres de la placa de un vehículo.
- Inclinación de la placa.- Es la rotación que tiene una región de placa con respecto a eje x.
- Tasa de segmentación correcta de caracteres.- Es la relación que existe entre el número total de segmentos de caracteres detectados correctamente y los segmentos que deberían detectarse.
- Combinación de técnicas de cada método.- Son todas las consideraciones tomadas para hacer más exacto el método al realizar la segmentación de caracteres.
- Condiciones de iluminación.- Son las condiciones en las cuales se exponen las regiones de placa en las imágenes con bajas, normales o altas cantidades de iluminación.
- Número de imágenes en la pruebas.- Es el número total de imágenes utilizadas para probar un método.
- Porcentaje de falsas alarmas.- Es la relación que existe entre el número de segmentaciones incorrectas de caracteres y el total de segmentaciones que debería realizar satisfactoriamente.
- Similitud de formatos de placas de otros países con el formato estándar para ecuador.

La selección del algoritmo para la segmentación de los caracteres se la enfoca en las investigaciones realizadas en (Du et al., 2013) en base a la Tabla 3 se presenta un resumen de la comparación de métodos tanto para la extracción de la región de la placa, la segmentación de los caracteres y el reconocimiento de los mismos, según los datos presentados se elige el algoritmo planteado en (Wang et al., 2007) por proporcionar una tasa alta de segmentación de caracteres correcta de la región de la placa del 98,3% en un sistema de tiempo real que tiene la capacidad para resolver el reconocimiento de caracteres de una placa en un tiempo de 30ms para imágenes de 640x480 pixeles,

además el método reconoce placas taiwanesas, las cuales son muy similares a las placas ecuatorianas como se puede apreciar en la Figura 25, en la Tabla 3 el método [14] tiene una buena tasa de aciertos y una gran cantidad de imágenes (30000+) utilizadas en las pruebas, pero se elige el método mencionado porque el método [14] trabaja con placas chinas, las cuales tienen un formato diferente a las placas que poseen los vehículos en Ecuador según Figura 25, además este algoritmo emplea un tiempo de segmentación de caracteres en el intervalo de 10 a 20 milisegundos lo que en promedio nos resulta en 15 ms, utilizando aproximadamente la mitad del tiempo que emplea el método en (Wang et al., 2007) para realizar tres tareas en el proceso de la identificación de vehículos: la detección de la región de la placa, la segmentación de caracteres y el reconocimiento de los mismos.

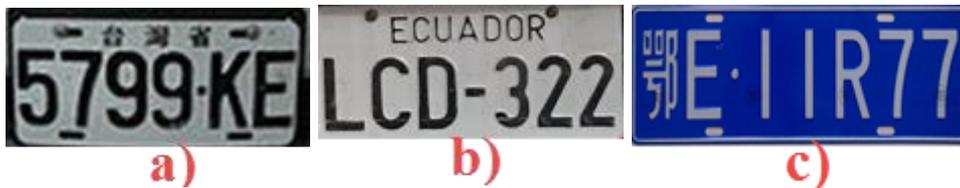


Figura 25. a) Placa Taiwanesa, b) Placa Ecuatoriana y c) Placa China.  
Fuente: ("Vehicle registration plates of Taiwan," 2007), ("Chinese License Plate," 2012)  
Elaboración: Juan Pablo Ordoñez Lopez.

### Reconocimiento de los caracteres obtenidos en la segmentación de la región de la placa.



Figura 26. Caracteres obtenidos para identificar un vehículo.  
Elaboración: Juan Pablo Ordoñez Lopez.

Los caracteres extraídos son luego reconocidos y la salida es el número de la placa. El reconocimiento de caracteres en los sistemas LPR puede tener algunas dificultades. Debido al factor de zoom de la cámara, los caracteres extraídos no tienen el mismo

tamaño y el mismo espesor. Cambiar el tamaño de los caracteres antes del reconocimiento ayuda a superar este problema. Otro gran problema es la fuente de los caracteres que no es la misma todo el tiempo, además a esto cada país tiene diferentes fuentes de letras. Los caracteres extraídos pueden tener algo de ruido, cortados o cierto grado de inclinación.

### 1.1.7 Métodos para el reconocimiento de caracteres.

#### 1.1.7.1 Redes Neuronales Artificiales.

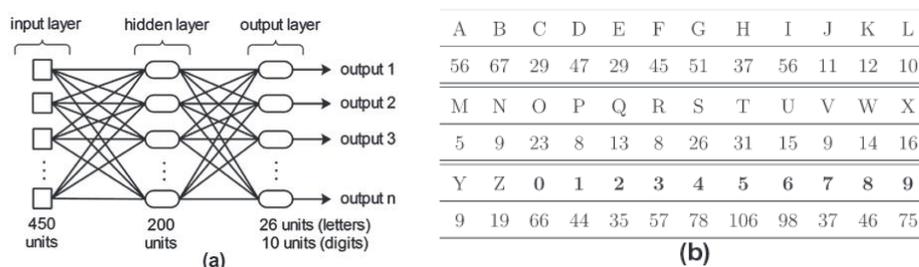


Figura 27. a) Capas de la ANN y b) Número de ejemplos de entrenamiento usado. Fuente: (Gazcón et al., 2012).

El uso de las Redes Neuronales Artificiales o sus siglas en ingles ANN para la solución de Reconocimiento Óptico de Caracteres o sus siglas en ingles OCR es ampliamente aceptado como una alternativa en el reconocimiento de patrones, como ANN presenta una fuerte tolerancia al ruido en los datos de entrada, obtenemos como resultando un comportamiento robusto en sistema para la tarea mencionada. Esto favorece la realización de reconocimiento de caracteres en condiciones que están muy alejados de las situaciones ideales. Implica un tiempo de entrenamiento y el complejo refinamiento de los parámetros asociados, requiriendo así tener un gran número de ejemplos de entrenamiento, correspondientes a diferentes situaciones (por ejemplo, imágenes en diferentes condiciones climáticas, ángulos, etc.).

En (Gazcón et al., 2012), el reconocimiento de caracteres se llevó a cabo utilizando dos ANN, que corresponden a las letras y dígitos, respectivamente. Este trabajo utiliza Redes neuronales Artificiales compuestas de tres capas: una capa de entrada, una capa oculta y una capa de salida Figura 27.

- Capas de entrada

La entrada de ambas redes se definió por los píxeles del carácter segmentado. En este caso, se partió de una imagen de píxeles 15x30, lo que resulta en 450 neuronas de entrada para cada red.

- Capas de salida

Había 26 unidades de salida de la ANN para el reconocimiento de letras, y 10 unidades de salida para el reconocimiento de dígitos.

- Capas ocultas

Las neuronas correspondientes a la capa oculta se aproximan de manera experimental. Como regla general, se supuso que el número de neuronas en la capa oculta estarían en un rango entre el número de unidades de entrada y el número de unidades de salida. En ambas ANN se considera 50-450 unidades (con un paso de 50), en busca de un error de menos de 0.001. Los mejores resultados se obtuvieron con 200 unidades.

- La función de activación

Tanto para las capas ocultas, así como para las capas de salida se empleó la clásica función sigmoide.

$$f(x) = \frac{1}{1+e^{-x}} \quad (40)$$

Donde,  $e$  es una constante conocida como número de Euler o constante de Napier.

#### **1.1.7.2 Plantillas de coincidencia.**

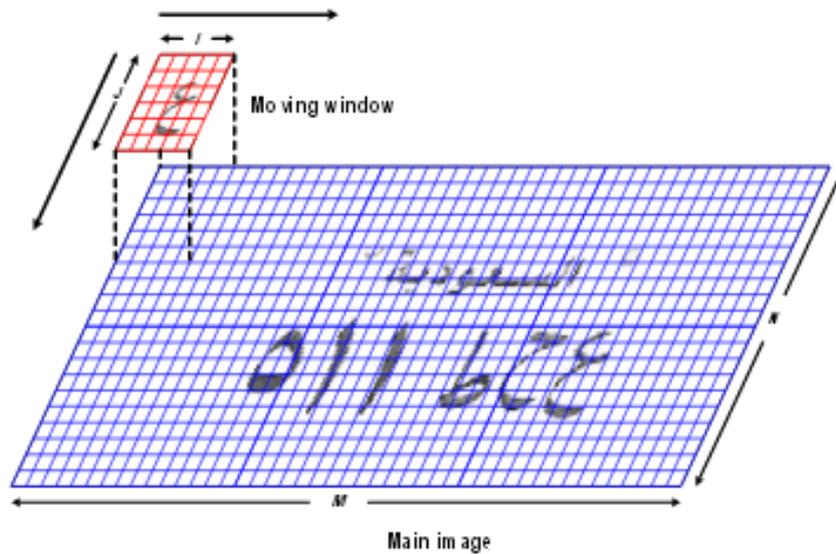


Figura 28. Esquema de movimiento para realizar coincidencia mediante plantilla.  
Fuente: (Khalil, 2010).

Una ventana móvil es utilizada en el método de emparejamiento de plantillas (suma de las diferencias al cuadrado) es una técnica común utilizada en muchas aplicaciones de reconocimiento de patrones. El método de emparejamiento de plantillas proporciona una alta precisión de reconocimiento y reduce el tiempo de procesamiento en comparación con otros métodos tales como la correlación cruzada. El método aplicado calcula la suma de las diferencias al cuadrado en cada posición, mientras que la imagen de la palabra que se quiere reconocer se mueve sobre la plantilla de fondo. El punto donde la suma de la diferencia al cuadrado es menor que un umbral predeterminado será considerado como el punto de coincidencia. El esquema de comparación de plantillas mediante una ventana móvil propuesta se ilustra en la Figura 28. En primer lugar se define una ventana que contiene un objeto con un tamaño más pequeño que el de la imagen principal. Sólo una parte de la imagen es visible a través de esta ventana. La función de emparejamiento de plantillas se realiza entre el objeto en la ventana y el área correspondiente de la imagen. A continuación, la ventana se desplaza y la función de comparación de plantillas se lleva a cabo entre el objeto en la ventana y la nueva parte de la imagen visible a través de la ventana. Por lo tanto, la ventana se mueve de izquierda a derecha y de arriba abajo el desplazamiento se lo hace por cada píxel individual hasta que toda la imagen sea cubierta y la comparación de plantillas se lleva a cabo para todas las diferentes posiciones de la ventana. Matemáticamente, la medida de distancia es una medida de las similitudes o propiedades compartidas entre dos señales. La distancia métrica comúnmente utilizada es la métrica de Minkowski

$$d(X, Y) = \left( \sum_{i=1}^N |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (41)$$

Donde, donde  $X, Y$  son dos vectores de características dimensionales  $N$ , y  $r$  es un factor de Minkowski. Cuando  $r$  vale 2, en realidad es la distancia euclidiana.

Para el método propuesto en (Khalil, 2010) hay dos señales discretas  $f$  y  $t$ , las mismas representan dos imágenes que denotan el objeto a ser buscado y la plantilla respectivamente. El objeto es de dimensión  $I \times J$  píxeles y la plantilla es de dimensión  $M \times N$  como en la Figura 28.

$$d^2(X, Y) = \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} [f(i, j) - t(x + i, y + j)]^2 \quad (42)$$

$$x = 0, 1, \dots, M - I, y = 0, 1, \dots, N - J \quad (43)$$

Donde la suma es sobre  $i$  y  $j$  debajo de la ventana que contiene la característica posicionado en  $x, y$ . Para reducir el tiempo de cálculo, la ecuación anterior se puede simplificar a la distancia Manhattan mediante la asignación  $r = 1$ :

$$d(X, Y) = \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} |f(i, j) - t(x + i, y + j)| \quad (44)$$

### 1.1.7.3 Reconocimiento de caracteres basados en CCs, DCs y PBA.

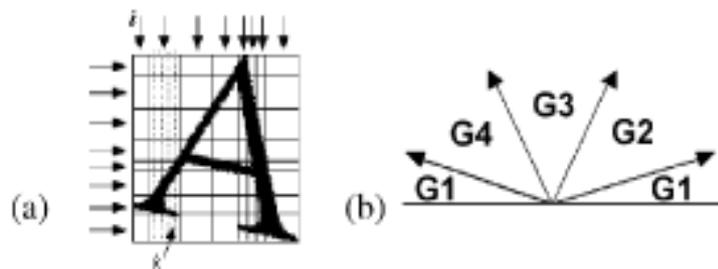


Figura 29. Características prototipos OCR a) CC y b) DC.  
Fuente: (Wang et al., 2007).

Las variaciones en los caracteres y no caracteres extraídos de los procedimientos de segmentación precedentes se aprende automáticamente utilizando el mismo escenario de la verificación de la región de la placa que se ha descrito en la Sección III- C en (Wang et al., 2007). La fase de clasificación de la verificación de un carácter se representa en la Figura 30.

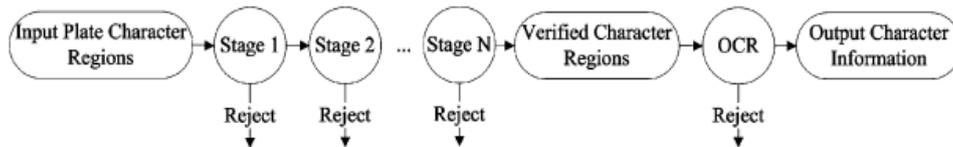


Figura 30. Esquema para la verificación de un carácter.  
Fuente: (Wang et al., 2007).

Durante la etapa de extracción de características, las tablas de aceleración SAT y SSAT (Wang et al., 2007) se vuelven a utilizar para calcular las características Haar-Like para el rechazo de caracteres. En comparación con la verificación de placa, se adoptó un procedimiento más de tipo OCR para reconocer cada carácter de forma explícita. Se ha investigado el desarrollo de un método para el reconocimiento de caracteres de placa con diferentes apariencias. En éste proceso, el paso de extracción de características se adoptó como primer paso para transformar las imágenes de caracteres en vectores de características.

- Extracción de características OCR: Tres rasgos robustos de los caracteres se extraen para representarlos:
  - Conteos contorno-cruce (CC).
  - Conteos direccionales (DC).
  - Área de fondo periférica (PBA).

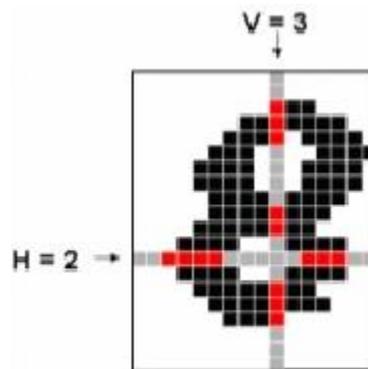


Figura 31. Extracción característica CC.  
Fuente: (Vamvakas, 2014).

Para la característica de CC, cada subimagen se segmenta no uniformemente en ocho tiras en las direcciones horizontal y vertical. Estas tiras tienen el mismo número de píxeles negros. Con el fin de aumentar la velocidad de la extracción, cuatro líneas de exploración son seleccionadas para extraer las características de cada tira. La

dimensión de la característica de CC de una muestra de entrada es 16 (=8x2). Cada valor de característica CC  $f_i$  en cada tira  $i$  se define como sigue:

$$f_i = \sum_{k=1}^4 CC_{ik} \quad (45)$$

Donde, un valor del conteo de contorno cruce  $CC_{ik}$  es el número de toques de intersección de cada línea de exploración  $k$  representados en la Figura 29. (a) por líneas entrecortadas e  $i=1,2,3,\dots,8$ .

La característica DC representa el número de puntos de contorno en cuatro direcciones principales. Como se muestra en la Figura 29. (b), que dividen las direcciones en cuatro grupos:  $G1$ ,  $G2$ ,  $G3$  y  $G4$ , que corresponden a los trazos horizontal, vertical, diagonal y diagonal inversa, respectivamente. Los patrones solamente se miden desde los píxeles situados en los límites de los trazos. El ángulo de un toque o intersección, ocurre en el intervalo 0 a 180 grados y se determina por el operador Sobel. La dimensión de la característica DC de una muestra de entrada se estima en 64 ( $4 \times 4 \times 4$ ) al dividir la imagen del carácter de entrada en  $4 \times 4$  sub-bloques.

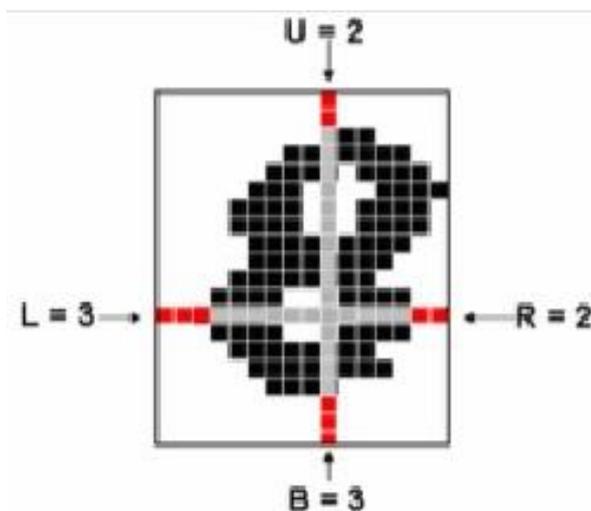


Figura 32. Extracción característica PBA.  
Fuente: (Vamvakas, 2014).

La característica PBA es la longitud de los segmentos de línea que comienzan desde el límite de la imagen hacia el contorno de los caracteres. La imagen del carácter de entrada se segmenta primero en ocho tiras en ambas direcciones horizontal y vertical por el método descrito en la técnica de procesamiento anteriormente. En cada tira, dos valores de características se miden desde las longitudes entre los límites de la tira y el contorno de carácter más cercano. Cada valor de característica PBA se divide por la longitud de la tira para la normalización. La dimensión característica PBA de una

muestra de entrada es de 32 (8x2x2). La dimensión característica definitiva de cada carácter es de 112 (16+64+32).

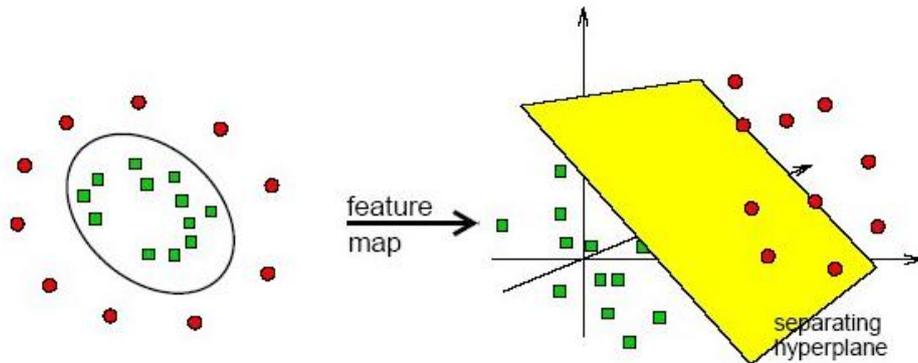


Figura 33. a) Características de imagen b) Características de imagen mapeadas al espacio.  
Fuente: (DTREG, 2014).

- Clasificación OCR.- La etapa de clasificación se realiza mediante una máquina de vectores de soporte (SVM o Support Vector Machine) para clasificar cada carácter de la placa. El método SVM busca un hiperplano de separación lineal como una combinación lineal de vectores de soporte (además de una constante). La distancia de todos los vectores de soporte al hiperplano es 1. Según las investigaciones en (Wang et al., 2007) con el fin de determinar la mejor configuración para la SVM, se probó diferentes ajustes de los parámetros. La función de base radial (RBF) es el kernel utilizado. La precisión de OCR más alta (con sólo un error de reconocimiento de carácter en el conjunto de entrenamiento) se obtiene mediante el establecimiento de gamma (en relación con el radio RBF) a 0,0078125.

### 1.1.8 Selección del método para el reconocimiento de los caracteres segmentados.

En ésta tarea se utilizó las siguientes métricas para elegir un método para el reconocimiento de los caracteres segmentados:

- Tiempo de procesamiento.- Es el tiempo que toma procesar una imagen para realizar el proceso de reconocimiento de un carácter segmentado.
- Inclinación del carácter segmentado.- Es la rotación que tiene un carácter en la imagen.
- Tasa de reconocimiento correcta de caracteres.- Es la relación que existe entre el número total de caracteres detectados correctamente y los caracteres que deberían detectarse.

- Combinación de técnicas que utiliza el método.- Son todas las consideraciones tomadas para hacer más preciso el método al realizar el reconocimiento de caracteres.
- Condiciones de iluminación.- Son las condiciones en las cuales se exponen los caracteres en las imágenes con bajas, normales o altas cantidades de iluminación.
- Número de imágenes en la pruebas.- Es el número total de imágenes utilizadas para probar un método.
- Similitud de formatos de placas de otros países con el formato estándar para Ecuador.

Para la selección del método para el reconocimiento de los caracteres de la región de la placa se consideró las investigaciones realizadas en (Du et al., 2013) en base a la Tabla 3 se presenta un resumen de la comparación de métodos tanto para la extracción de la región de la placa, la segmentación de los caracteres y el reconocimiento de los mismos, según los datos presentados se elige el algoritmo planteado en (Wang et al., 2007) por proporcionar una tasa alta de reconocimiento de caracteres sobre la región de la placa segmentada del 98,3% en un sistema de tiempo real, el mismo tiene la capacidad para resolver el reconocimiento de caracteres de una placa en un tiempo inferior a los 30ms, éste método soporta segmentos de carácter con rotaciones, así también trabaja con diferentes condiciones de luminosidad como se puede apreciar en la Figura 34.



Figura 34. a) Regiones compactas de placa, b) Caracteres distintos y c) Caracteres Indistintos. Fuente: (Wang et al., 2007).

### **Sistemas de estimación de distancia de vehículos basados en visión estéreo.**

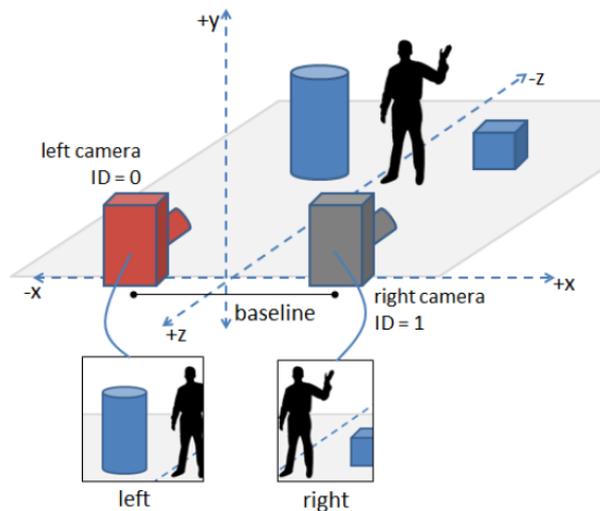


Figura 35. Cámaras en escena.  
Fuente: (Davison, 2013).

Los relativamente pocos estudios que se han realizado en el ámbito del transporte inteligente han incluido excelentes informes sobre la detección de vehículos, estimaciones de carril, y mediciones de la velocidad de vehículos. Los primeros estudios realizados en el campo de la VDDE (Vehicle detection and distance estimation) comenzaron en el año 2000, y gran parte de la información sigue siendo vigente y relevante debido al crecimiento de las técnicas eficientes de visión por computador. Algunos de los nuevos sistemas comparten modelos comunes, métodos, configuraciones y características. Otros sistemas se han originado a partir de diversos enfoques para VDDE. Los sistemas basados en VDDE estéreo existentes se han presentado en términos de sus características, incluyendo sus configuraciones estéreo, distancia máxima detectada (MDD), coincidencia estéreo, extracción de características, y la evaluación de los resultados. La mayoría de los sistemas estéreo han sido evaluados por la comparación de su distancia de salida a la distancia real. Los sistemas VDDE han seguido cuatro etapas principales: pre procesamiento, detección de vehículos, seguimiento de vehículos, y la estimación de la distancia al vehículo. En general, las tareas necesarias para estimar la distancia al vehículo son las siguientes:

- La creación y la calibración del sistema del par de cámaras.
- Obtener imágenes sucesivas izquierda y derecha.
- La detección y seguimiento de vehículos predecesores.
- La estimación de distancia del vehículo basándose en la información estéreo obtenida.

### **Cálculo de disparidad en un sistema de visión estéreo.**

**Rectificación de Imágenes.-** Una etapa de pre procesamiento que se utiliza generalmente en la mayoría de los algoritmos de correspondencia estéreo es la rectificación de imágenes. Ésta proyecta las mismas en un plano de referencia común, de modo que los puntos de correspondencia tienen la misma coordenada de fila. Esto transforma esencialmente el problema de correspondencia estéreo 2D a 1D. El resultado del proceso se lo puede observar en la Figura 36. Sin embargo, el propio proceso de rectificación se sumará a la complejidad computacional del algoritmo según (Athreyas, Lai, Gupta, & Gupta, 2014).



Figura 36. Imágenes Rectificadas.  
Fuente (Davison, 2013).

La correspondencia estéreo es el cálculo de las disparidades (compensaciones) entre los puntos comunes que aparecen tanto en la imagen izquierda y derecha.

Utilizamos los cálculos de disparidad todo el tiempo cuando juzgamos distancias. Los objetos que están cerca de nosotros tienen una gran disparidad mientras que aquellos más alejados tienen una disparidad más pequeña. Se puede observar esto mirando algo cerca de nosotros, primero con el ojo izquierdo, y luego con el derecho. El objeto parece saltar a la izquierda cuando cambia a su ojo derecho o viceversa. El salto es mucho menos notable si se concentra en algo en la media distancia, y la disparidad se reduce o tiende a 0 para los objetos lejanos.

La misma técnica se puede aplicar a las imágenes rectificadas izquierda y derecha de una manera eficiente debido a su alineación.

Esto significa que un punto en  $(x, y)$  en la imagen de la izquierda se encuentra en algún lugar en la imagen de la derecha. La idea se ilustra en la Figura 36.

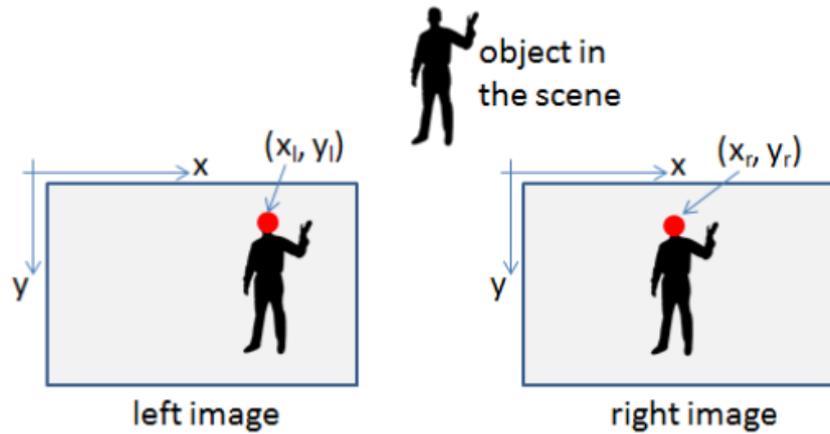


Figura 37. Correspondencia de puntos entre las imágenes izquierda y derecha.  
Fuente: (Davison, 2013).

$y_l$  y  $y_r$  será el mismo (debido a la rectificación), por lo que la disparidad está dada por la diferencia  $(x_l - x_r)$ . Cuando las diferencias se calculan para todos los puntos en la escena, la más grande ocurrirá para los objetos cercanos a las cámaras, y las disparidades más pequeñas serán evidentes para las cosas que están más lejos, como se muestra en la Figura 38.

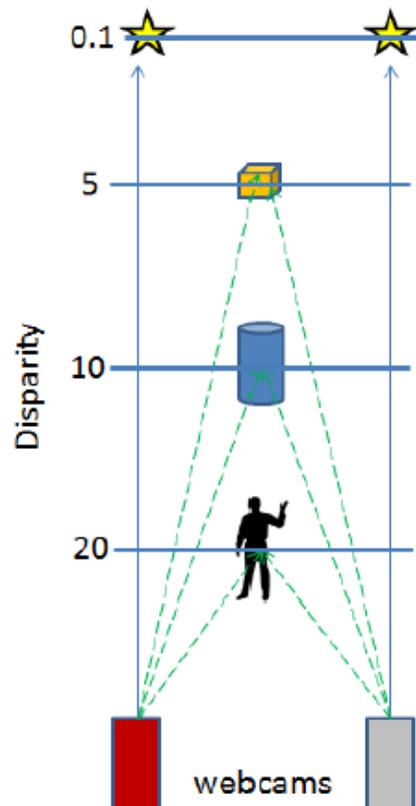


Figura 38. Valores de disparidad relativos para las cámaras.  
Fuente: (Davison, 2013).

## Selección de cámaras para la visión estéreo.



Figura 39. Webcams en una barra trípode.  
Fuente: (Davison, 2013).

Para la selección de cámaras para el sistema de visión estéreo se considera el tamaño de las imágenes, utilizado tanto en la detección de vehículos como en la detección de la región de la placa, así como configuraciones realizadas.

En el trabajo (Nguyen et al., 2013) para ejecutar la detección de vehículos (método seleccionado) se consideran las siguientes configuraciones de cámara y tamaños de lente focal según se aprecia en la siguiente Figura 40.

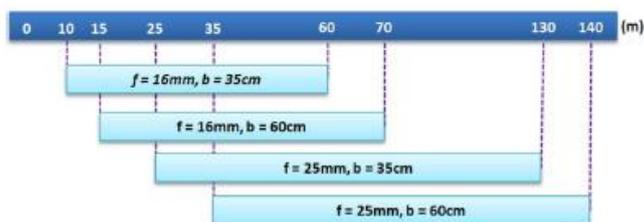


Figura 40. Posibles rangos de detección de vehículos del método respecto a cuatro diferentes configuraciones.  
Fuente: (Nguyen et al., 2013).

Basados en estas configuraciones longitud de lente focal 16mm a 25mm, distancia máxima detectada (140m), distancia de línea base entre 35 a 60 cm y tamaños de las imágenes en pixeles utilizadas en las pruebas (320 x 240) y (640x480) para detección de vehículos y detección de la región de la placa (Du et al., 2013), respectivamente, se ha seleccionado las cámaras con las siguientes características:

HIKVISION DS-2CD2632F-I(S) (3MP) VF IR Bullet Network Camera

Tabla 5. Principales características cámaras seleccionadas.

Image sensor	1/3" progressive scan CMOS
Min. Illumination	0.07 lux @F1.2, AGC ON, 0 lux with IR
Shutter time	1/25s (1/30s) ~ 1/100,000s
Lens	2.8 - 12 mm @ F1.4, Angle of view: 105.2°-31.9°
Lens mount	Φ14
Day & night	ICR
Digital noise reduction	3D DNR
Wide dynamic range	Digital WDR
Backlight compensation	Yes, zone configurable
Max. Image Resolution	2048 × 1536
Frame rate	50Hz: 20fps (2048 x 1536), 25fps (1920 × 1080), 25fps (1280 x 720) 60Hz: 20fps (2048 x 1536), 30fps (1920 × 1080), 30fps (1280 x 720)

Fuente: (Hikvision, 2014).

Se elige las cámaras de 3 megapíxeles en base a las resoluciones de las imágenes utilizadas en los métodos, sabiendo que para la detección de vehículos se utilizó una resolución de (320 x 240) lo suficiente para la tarea en cuestión, el problema resulta cuando se detecta un vehículo, obteniendo como resultando una imagen en un área aproximadamente de 1/20 del espacio de la imagen original como se puede apreciar en la Figura 41, ésta imagen parcial según nuestro propósito debería tener gran cantidad de información para las tareas posteriores de detección de la placa, segmentación de caracteres y reconocimiento de los mismos.



Figura 41. Detección de vehículo.

Fuente: (Du et al., 2013).

Así, si dividimos la imagen original que nos proporciona la cámara seleccionada de tamaño (2048 × 1536) píxeles en una matriz de 5 x 4 (20 partes) para filas y columnas

respectivamente, tendríamos que cada celda conservaría un tamaño aproximado de (409x384) pixeles, estas subimágenes resultan en las entradas para los procesos de detección de las placas, segmentación de caracteres y reconocimiento de los mismos.

### **Configuración y Calibración de las cámaras del sistema estéreo.**



Figura 42. Par de imágenes para calibración estéreo.  
Fuente: (Davison, 2013).

La calibración de las cámaras desempeña un papel crucial en los sistemas basados en visión estéreo, en particular los sistemas de transporte. La precisión de la calibración de las cámaras afecta directamente la distancia estimada al objeto, cuando se aplica una técnica de calibración de cámaras precisa, el problema de la determinación de la información de profundidad y disparidad a partir de dos imágenes estéreo se simplifica sustancialmente. La mayoría de los sistemas utilizan cámaras con distancias focales de 6 a 12 mm, y la distancia de la línea base varía de 15 a 150 cm. La Figura 40 muestra los posibles rangos de detección del sistema propuesto en (Nguyen et al., 2013) con respecto a cuatro configuraciones de línea base y diferentes longitudes focales, éste trabajo utiliza la técnica de tablero de ajedrez para la calibración de las cámaras.

Con el fin de recuperar las coordenadas 3D para un objeto arbitrario, las cámaras deben calibrarse primero para obtener las matrices de parámetros intrínsecos y externos de la cámara. El enfoque ampliamente aceptado hasta ahora para la calibración de cámaras se lleva a cabo con un objeto preparado cuya forma geométrica 3D ya se conoce, es decir, dado un modelo de cámara fija, basada en condición de experimento especial (como el uso del tablero de ajedrez cuya forma y tamaño son conocidos), las imágenes capturadas se procesan. Los parámetros intrínsecos y externos de la cámara se calculan utilizando una serie de transformaciones matemáticas y principios básicos. Este tipo de método de calibración puede alcanzar resultados satisfactorios con mayor precisión. Un método típico de este tipo es el método de Zhang Zhengyou para la calibración.

Por lo general se necesita un par de cámaras en el sistema de visión estéreo binocular. En consecuencia, debemos resolver la posición relativa entre dos cámaras además de los parámetros intrínsecos de cada cámara en el proceso de calibración estéreo binocular. En este método, el parámetro de estructura que incluye la matriz de rotación  $R$  y el vector de traslación  $T$  se resuelve durante el proceso de calibración. Utilizamos el método de Zhang que tiene la ventaja de la flexibilidad, simplicidad y alta precisión. En primer lugar, el tablero de ajedrez se utiliza como objeto de calibración capturado por dos cámaras al mismo tiempo. Los parámetros intrínsecos de las dos cámaras se pueden resolver por separado. En segundo lugar, los parámetros externos de dos cámaras se asumen como  $R_1$ ,  $T_1$  y  $R_2$ ,  $T_2$  (incluyendo matriz de rotación y vector de traslación) respectivamente. Estas matrices reflejan la posición relativa de las dos cámaras en el mundo del sistema de coordenadas. Para un punto arbitrario  $P$  en el sistema de coordenadas en el mundo, los puntos correspondientes  $P_1$  y  $P_2$  en los sistemas de coordenadas de las cámaras izquierda y derecha se representan como:

$$P_1 = R_1 P + T_1 \quad (46)$$

$$P_2 = R_2 P + T_2 \quad (47)$$

La proyección del punto  $P$  hacia las dos cámaras pueden ser transformados por la siguiente fórmula.

$$P_1 = R^T (P_2 - T) \quad (48)$$

En esta fórmula,  $R$  y  $T$  representan la matriz de rotación y el vector de traslación entre dos cámaras. Resolvemos la rotación y traslación sobre la base de las fórmulas (46) y (47). Entonces podemos obtener nuevas fórmulas descritas a continuación (Zou & Li, 2010):

$$R = R_2 (R_1)^T \quad (49)$$

$$T = T_2 - R T_1 \quad (50)$$

La calibración estéreo es completada por el proceso mencionado anteriormente. Mientras se realiza la calibración de cada cámara, la posición relativa de las dos cámaras se calcula, lo que muestra la eficiencia y la simplicidad del enfoque.

## **METODOLOGIA**

## Arquitectura del Sistema.

Para la operación del sistema se han considerado dos procesos básicos:

- Proceso Aislado.- En este se considera la calibración de las cámaras mediante el método del tablero de ajedrez, se toma un número de par de imágenes como el de la Figura. 42, mediante este conjunto de pares de imágenes se procede a calcular parámetros intrínsecos y extrínsecos de las cámaras, éste se lo hace con la invocación al método stereoCalibrate de la librería opencv.

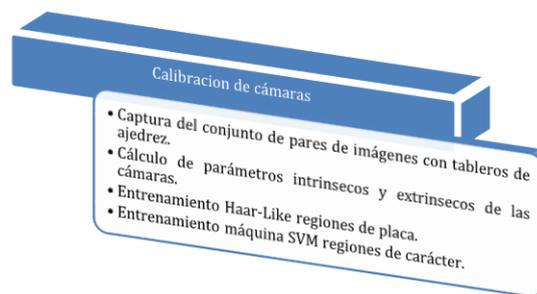


Figura 43. Módulos proceso aislado – Calibración de cámaras.  
Elaboración: Juan Pablo Ordoñez Lopez.

- Proceso en tiempo real.- Este proceso empieza con la captura del par de imágenes, luego pasan por un subproceso de rectificación, en este paso se toman en cuenta los parámetros intrínsecos y extrínsecos de las cámaras, el siguiente paso realiza cálculos de profundidad y disparidad de las imágenes, enseguida se procede con la detección de vehículos mediante un algoritmo genético, se lo hace mediante la creación de la población inicial de cromosomas que representan los posibles vehículos a detectar, se procede con la evaluación de la función de adaptación y la ejecución de los torneos de selección y cruzamiento para obtener los mejores candidatos, las áreas de imagen de los vehículos detectados son pasadas al proceso de detección de la región de la placa, en éste se obtiene la región de la placa, se segmenta los caracteres y se reconoce los mismos.

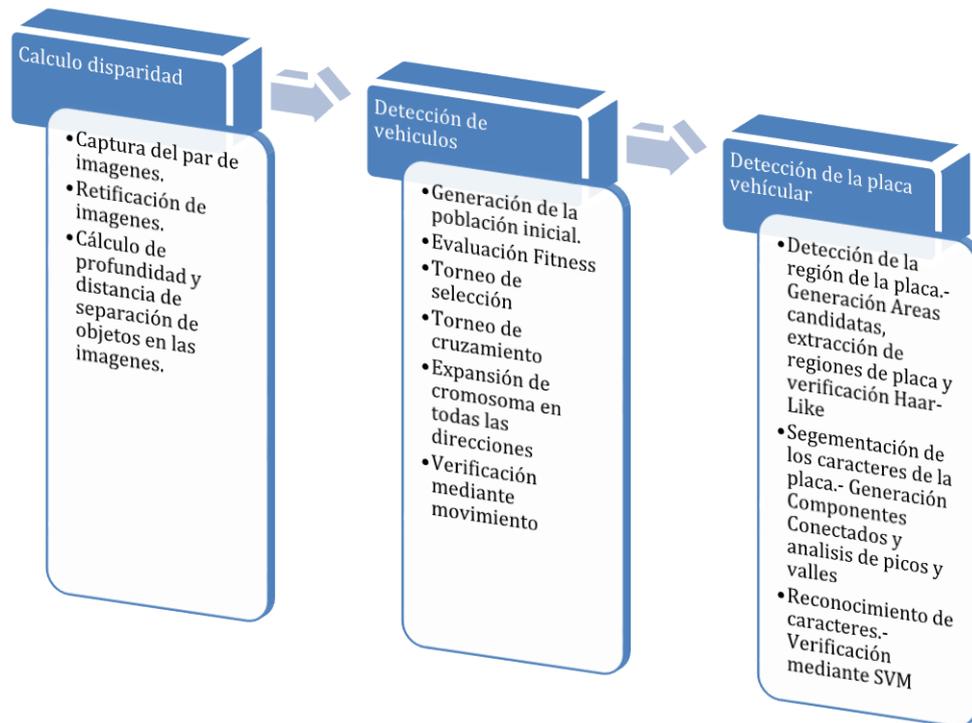


Figura 44. Módulos proceso en tiempo real – Detección de placas.  
Elaboración: Juan Pablo Ordoñez Lopez.

### Metodología XP como metodología de desarrollo.

La selección de una metodología de desarrollo se basa en la información en la Tabla 5 expuesta en el trabajo (Cadavid, Daniel, Martínez, & Vélez, 2013), la misma hace una comparación de Metodologías tradicionales y Metodologías ágiles.

Tabla 6. Metodologías tradiciones vs metodologías Ágiles.

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Fuente: (Cadavid et al., 2013)

Según las características se ha elegido una metodología ágil, por ser orientadas al cliente final, la flexibilidad que permiten y la constante comunicación con el cliente, finalmente se elige XP (Extreme Programming) como metodología de desarrollo, por la siguiente razón, según (Cadavid et al., 2013) está enfocada a equipos pequeños y medianos de desarrollo de software, entre dos y diez desarrolladores, en ambientes de requerimientos imprecisos o cambiantes y no requiere extensa documentación.

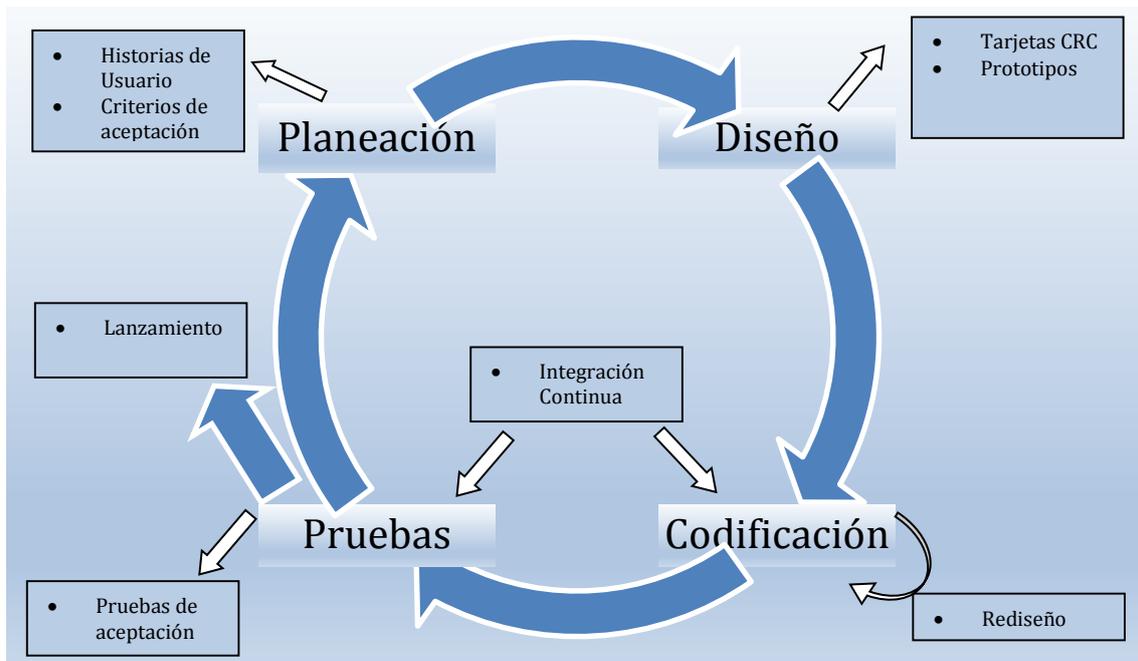


Figura 45. Fases Programación extrema.  
 Fuente: (Abdías Gómez D., 2014).  
 Elaboración: Juan Pablo Ordoñez Lopez.

### Planeación - Historias de usuario.

Para la descripción del sistema se utiliza el artefacto de XP Historias de Usuario para resumir las funcionalidades y su funcionamiento.

#### 2.1.1 Historia de usuario Capturar par de Imágenes.

Historia de usuario	
Número: 001	Usuario: Administrador Sistema
Nombre Historia: Capturar par de Imágenes	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Juan Pablo Ordóñez López	

<b>Descripción:</b> La calibración de las cámaras depende de un conjunto de pares de imágenes que contienen en las mismas un tablero de ajedrez que es utilizado para realizar el proceso antes mencionado. Con esta funcionalidad se facilita la captura del par de imágenes para ejecutar el proceso de calibración de las cámaras.
<b>Observaciones:</b>

### 2.1.2 Historia de usuario Calibrar Cámaras.

Historia de usuario	
<b>Número:</b> 002	<b>Usuario:</b> Administrador Sistema
<b>Nombre Historia:</b> Calibrar Cámaras	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Pablo Ordóñez López	
<b>Descripción:</b> La calibración de las cámaras es un proceso importante para el cálculo de la distancia de un objeto en las imágenes con respecto a las cámaras. En este se deben obtener los parámetros intrínsecos y extrínsecos de las cámaras, como las matrices de cada cámara, longitud focal, la distancia de la línea base y las matrices de rotación y traslación.	
<b>Observaciones:</b>	

### 2.1.3 Historia de usuario Detectar vehículos.

Historia de usuario	
<b>Número:</b> 003	<b>Usuario:</b> Proceso Automático
<b>Nombre Historia:</b> Detectar vehículos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Pablo Ordóñez López	
<b>Descripción:</b> En el proceso para la detección de placas vehiculares es indispensable segmentar u obtener las áreas de la imagen que contengan los vehículos.	
<b>Observaciones:</b>	

#### 2.1.4 Historia de usuario Detectar área de la placa.

Historia de usuario	
Número: 005	Usuario: Proceso Automático
Nombre Historia: Detectar área de la Placa	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Programador responsable: Juan Pablo Ordóñez López	
Descripción: Una vez identificada el área del vehículo se debe realizar la detección de la región de la placa que contiene los caracteres a reconocer.	
Observaciones:	

#### 2.1.5 Historia de usuario Segmentar los caracteres de la placa.

Historia de usuario	
Número: 006	Usuario: Proceso Automático
Nombre Historia: Segmentar los caracteres de la Placa	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Programador responsable: Juan Pablo Ordóñez López	
Descripción: Una vez identificada el área de la placa se debe realizar la segmentación de los caracteres de la placa, aquí se obtiene un conjunto de imágenes que contienen los caracteres de la misma.	
Observaciones:	

#### 2.1.6 Historia de usuario Reconocer los caracteres de la Placa.

Historia de usuario	
Número: 007	Usuario: Proceso Automático
Nombre Historia: Reconocer los caracteres de la Placa	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Programador responsable: Juan Pablo Ordóñez López	
Descripción:	

Una vez que se tiene todas las imágenes de los caracteres de la placa se debe proceder a reconocer los mismos mediante su equivalente con una letra o número.
<b>Observaciones:</b>

### Planeación – Criterios de aceptación.

Para la aceptación del sistema se toman los siguientes criterios bajo los cuales el software se considerará que cumple con las especificaciones exigidas.

Tabla 7. Requerimientos y sus criterios de aceptación.

Requerimientos	Criterios de Aceptación
Capturar par de Imágenes	Permite la captura de un par de imágenes desde dos cámaras de manera síncrona.
Calibrar Cámaras	Permite la calibración de un par de cámaras mediante un conjunto de pares de imágenes capturadas de forma síncrona con una configuración establecida.
Detectar Vehículos	Permite la detección de vehículos con una tasa de aciertos de al menos el 80%.
Detectar Caracteres de la Placa	Permite la detección de los caracteres de la Placa con una tasa de aciertos de al menos el 80%.

Elaboración: Juan Pablo Ordoñez Lopez

### Diseño mediante tarjetas CRC.

El diseño se lo hace mediante las tarjetas de Cargo o Clase, Responsabilidad y Colaboración.

#### 2.1.7 Tarjeta CRC sistema detector de placas vehiculares.

<b>Nombre de la clase:</b> Principal	
<b>Responsabilidad</b>	<b>Colaboración</b>
Seleccionar Ruta Estéreo	Gestor de archivos
Capturar Par de Imágenes	Gestor de cámaras (izquierda y derecha)
Calibrar Cámaras	Cálculo Estéreo
Iniciar Detección de Placas	Gestor de cámaras (izquierda y derecha), Cálculo Estéreo, Detector de Vehículos,

	Detector de la región de la placa, Segmentador de caracteres de la placa, Identificador de Caracteres y Proceso Automático (encargado de la captura automática del par de imágenes y orquestar el proceso de detección de placas)
Detener Detección de Placas	

Prototipo:

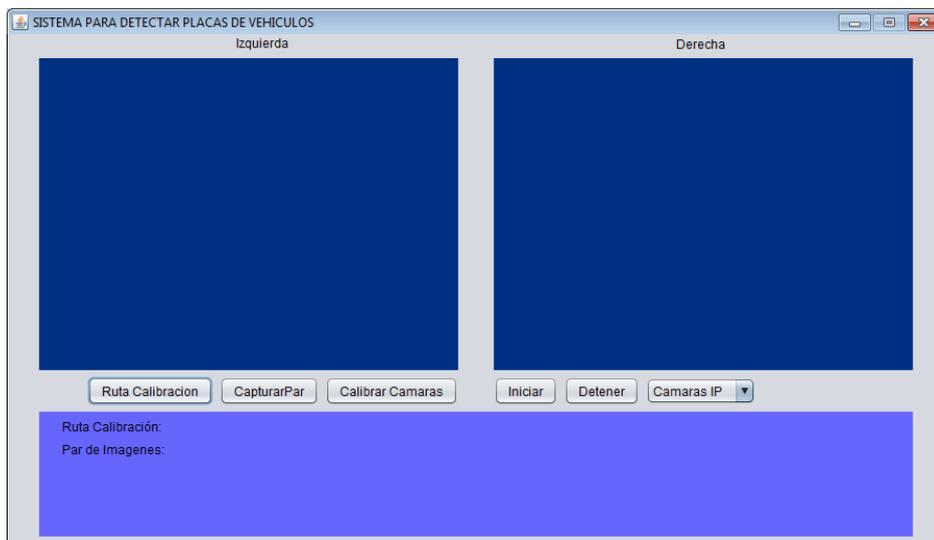


Figura 46. Prototipo Detector de Placas Vehiculares.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 2.1.8 Tarjeta CRC calibración de cámaras.

<b>Nombre de la clase:</b> CalculoEstereo	
<b>Responsabilidad</b>	<b>Colaboración</b>
Cargar Conjunto de par de Imágenes	
Calibrar Con el Conjunto de par de Imágenes	
Guardar Información de Calibración	
Cargar Información de Calibración	
Rectificar Imagen	
Calcular Mapa de Disparidad	

Prototipo: Figura 46.

### 2.1.9 Tarjeta CRC detector de vehículos.

<b>Nombre de la clase:</b> DetectorVehiculos	
<b>Responsabilidad</b>	<b>Colaboración</b>
Rectificar Imagen Izquierda	Cálculo Estéreo
Calcular mapa de disparidad del par de Imágenes que contienen los vehículos.	Cálculo Estéreo
Ecualizar Histograma de la imagen	
Detectar bordes en la imagen	
Crear Población Inicial de cromosomas (posibles vehículos)	
Evaluación de la función de Adaptabilidad	
Realizar Torneo de Selección	
Realizar Torneo de Cruzamiento	
Validar Cromosoma	

Prototipo:

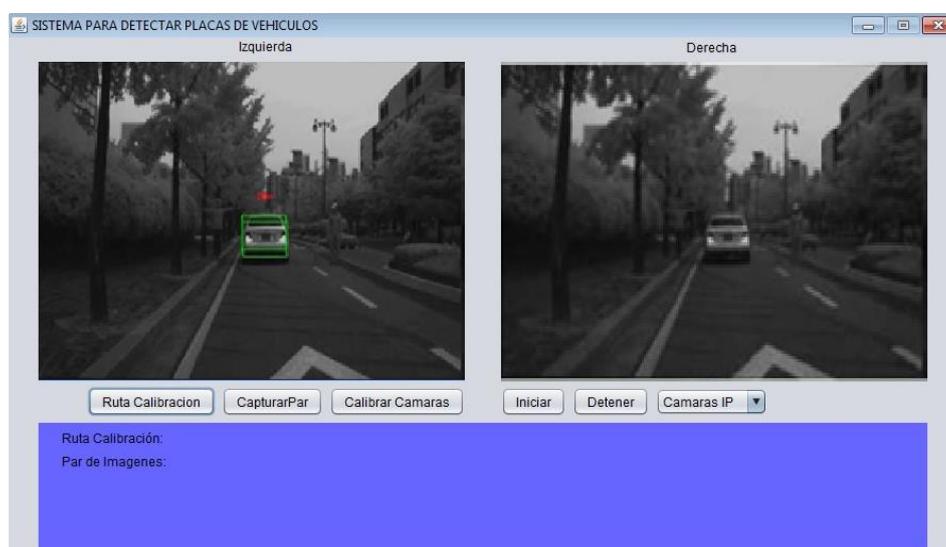


Figura 47. Prototipo Detector de vehículos.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 2.1.10 Tarjeta CRC detector de región de la placa.

<b>Nombre de la clase:</b> DetectorRegionPlaca	
<b>Responsabilidad</b>	<b>Colaboración</b>

Generar Regiones de placa Candidatas	
Extraer Región de Placa	
Verificar Región de Placa	

Prototipo: No Aplica

### **2.1.11 Tarjeta CRC segmentador e identificador de los caracteres de la placa.**

<b>Nombre de la clase:</b> SegmentadorCaracteresPlaca	
<b>Responsabilidad</b>	<b>Colaboración</b>
Segmentar Basado en el histograma	
Segmentar Basado en la proyección	
Verificar Caracteres	
Guardar Imagen Región de la Placa y los Caracteres asociados a la misma.	

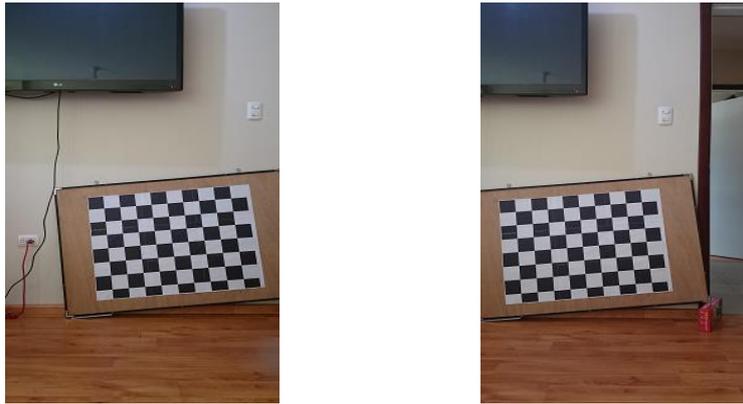
Prototipo: No Aplica

### **Codificación mediante lenguaje de programación java.**

Para la codificación se utiliza el lenguaje de programación java y las librerías Opencv versión 3.0.0 integrada con su Interfaz para java Javacv versión 1.1, ésta permite realizar procesamiento artificial y cuenta con una gran cantidad de métodos implementados y extensa documentación. Se elige el lenguaje de programación Java por estar familiarizado con el mismo aproximadamente por 3 años.

#### **2.1.1 Implementación Calibración de cámaras.**

Para implementar la calibración de cámaras se utilizó un conjunto de pares de imágenes que fueron tomadas con tableros de ajedrez para proceder a realizar la calibración de las mismas mediante el método mencionado en (Nguyen et al., 2013). Las imágenes que forman parte del conjunto de calibración se pueden observar a continuación en la Figura 48.



a) Imagen izquierda

b) Imagen derecha

Figura 48. Par de imágenes del conjunto de calibración de cámaras.

Elaborado: Juan Pablo Ordoñez Lopez.

Para realizar la calibración se utilizó un grupo integrado por 40 pares de imágenes, estas se emplearon en el siguiente método implementado `calibrarCamaras(int numeroParImágenes)`, como resultado el método genera los siguientes archivos `mx1.txt`, `mx2.txt`, `my1.txt`, `my2.txt` y `q.txt`, en donde los primeros cuatro son las matrices de distorsión de las cámaras, estas son utilizadas para la rectificación de un par de imágenes y `q.txt` contiene información externa relacionada a la distancia entre las cámaras y longitudes focales, necesarias para el cálculo del mapa de disparidad.

### 2.1.2 Implementación Detección de Vehículos.

La implementación de éste método se lo realizó con la siguiente secuencia de pasos:

- Sistema estéreo.- Para implementar el sistema estéreo (Nguyen et al., 2013) para la generación del mapa de disparidad se utilizó un par de cámaras como las que se observa en la Figura 49.



Figura 49. Par de cámaras para simular el sistema estéreo.

Elaborado: Juan Pablo Ordoñez Lopez.

- Capturar par de imágenes.- En este subproceso se realiza la captura del par de imágenes, las mismas deben ser tomadas al mismo tiempo, el resultado de este paso se lo puede observar en la Figura 50.



a) Imagen izquierda

b) Imagen derecha

Figura 50. Par de imágenes para generar el mapa de disparidad.  
Elaborado: Juan Pablo Ordoñez Lopez.

- Rectificación de imágenes.- El par de imágenes capturado es suministrado como entrada para la rectificación de imágenes al método *remap* de la librería *opencv*, éste método es llamado por cada imagen con los parámetros de distorsión de cada cámara, se retorna como resultado la imagen rectificada, para la imagen izquierda se utilizan las matrices *mx1.txt*, y *my1.txt*, de manera similar se lo hace con la imagen derecha mediante las matrices *mx2.txt* y *my2.txt*.



Figura 51. Par de imágenes rectificadas.  
Elaborado: Juan Pablo Ordoñez Lopez.

- Creación de la Población inicial de cromosomas.- Para crear la población inicial según (Nguyen et al., 2013) se utiliza un proceso gaussiano, el mismo fue implementado mediante el siguiente método *crearGenesProcesoGausiano(int Xi, int Yi)* definido en la clase *CromosomaVehiculo* el cual se encarga de la distribución uniforme de cromosomas con respecto a los parámetros de entrada *Xi* e *Yi* que representan el centro de la imagen, el resultado del proceso se lo puede observar en la Figura 52.

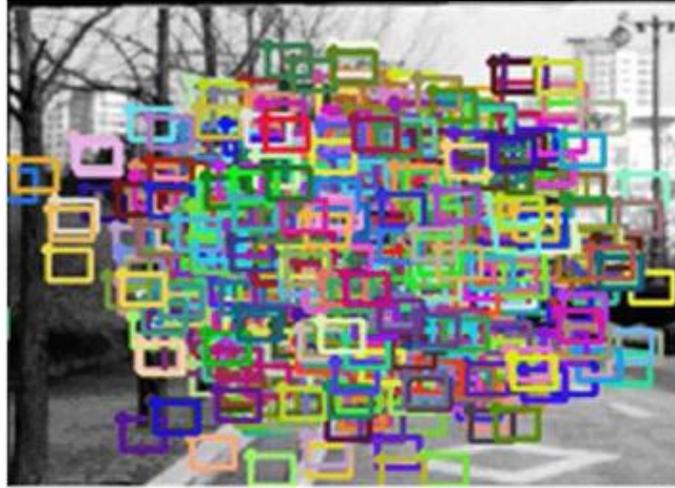


Figura 52. Creación de la población inicial de cromosomas.  
Fuente: (Nguyen et al., 2013).

- Evaluación de la función de adaptabilidad.- Para evaluar las cinco características de cada cromosoma definidas en (Nguyen et al., 2013) se utiliza la función *evaluar(Mat imagen, Mat mapaProfundidadCromosoma, Mat imagenBordes, Mat diferencia)* implementada en la clase *CromosomaVehiculo*, la misma recibe como parámetros de entrada la imagen que contiene los posibles vehículos, el mapa de disparidad de la región del cromosoma, la imagen completa de los bordes y la imagen completa de la diferencia entre una secuencia de imágenes de la cámara izquierda utilizada en la detección de movimiento, al final la función almacena y devuelve el valor de la evaluación.
- Torneo de selección.- Para la selección de los mejores cromosomas se utilizó el método indicado en (Nguyen et al., 2013) “Torneo por parejas”, el mismo fue implementado mediante la siguiente función *seleccionarCromosomasSiguieteGeneracion()* en la clase *Poblacion*, éste proceso se ejecuta 1000 veces el torneo de selección para obtener los cromosomas más fuertes.
- Torneo de cruzamiento.- En este se combinan dos cromosomas con valores de disparidad dominante similares mediante los modelos matemáticos que se indican en el método seleccionado para la detección de vehículos, se lo implementó con el siguiente método *cruzarMejoresCromosomasVehiculos()* en la clase *Poblacion*.

### 2.1.3 Implementación Detección de la Región de la Placa.

Para la implementación de éste proceso se considera los siguientes pasos:

- Imagen de entrada.- Como entrada del método se recibe la imagen que resulta del proceso de la detección de vehículos.



Figura 53. Vehículo Detectado.  
Elaboración: Juan Pablo Ordoñez Lopez.

- Obtener los bordes verticales.- Esto se lo hizo mediante la función que nos proporciona opencv "Sobel(vehiculo, bordesVerticales, CV\_8UC1, 1, 0)" para obtener solo los bordes verticales de la imagen el resultado se muestra en la siguiente imagen.

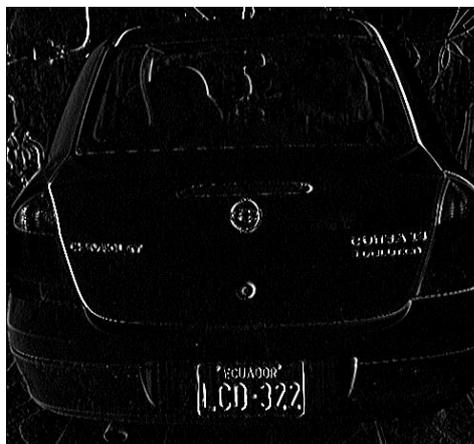


Figura 54. Bordes Verticales.  
Elaboración: Juan Pablo Ordoñez Lopez.

- Generación de áreas candidatas.- Este se lo hace mediante el método de OTSU que obtiene los pixeles más relevantes en la imagen, en opencv es el siguiente "threshold(bordesVerticales, Gp, 0, 255, THRESH\_BINARY\_INV + THRESH\_OTSU)", se le proporciona como entrada la imagen de los bordes verticales como resultado de aplicarle el método tenemos la siguiente imagen.

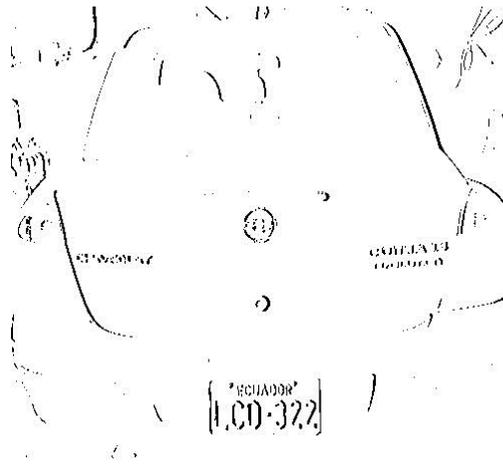


Figura 55. Generación de áreas candidatas.  
Elaboración: Juan Pablo Ordoñez Lopez.

- Extracción de las áreas de placa.- Aquí se consideran las reglas mencionadas en el método seleccionado para detectar la Región de la Placa, todo este proceso se lleva a cabo por el algoritmo “One-pass scanning algorithm for extracting compact plate regions” mencionado en (Wang et al., 2007) el mismo ha sido implementado por la siguiente función `extraerRegionPlaca(areasCandidatas)`, el método recibe como entrada la imagen de la Generación de las áreas de placa, los resultados del mismo se muestra en la siguiente figura.

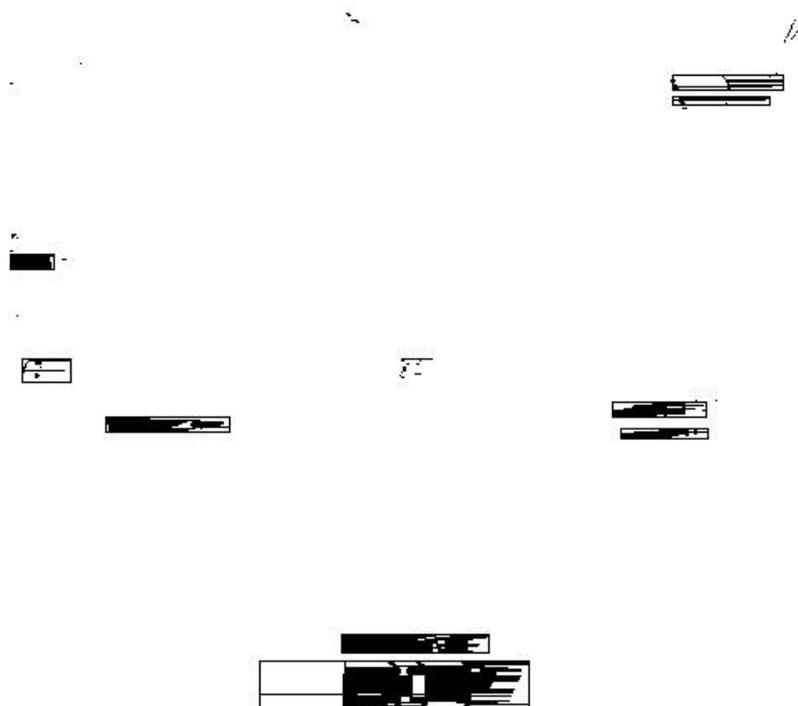


Figura 56. Extracción de áreas candidatas – Algoritmo un solo paso.  
Elaboración: Juan Pablo Ordoñez Lopez.

- Verificación del área de placa.- Para la verificación de las regiones de placas candidatas se utiliza el Algoritmo CascadeClassifier que proporciona la librería OpenCV, se debe proporcionar el archivo que se genera en el entrenamiento para las regiones de placas compactas, para realizar el entrenamiento se debe seguir la siguiente secuencia de pasos:
  - Disponer de un conjunto de imágenes que contengan las regiones de placa.
  - Disponer de un conjunto de imágenes que no contengan las regiones de placa.
  - Crear el archivo de ejemplos positivos el mismo debe tener el siguiente formato por línea.  
C:\data\placas\positivo\9.jpg 2 141 226 54 29 809 1009 287 87, en donde C:\data\placas\positivo\9.jpg representa la imagen positiva o la que contiene la región de placa, 2 es la cantidad de regiones de placa y 141 226 54 29 809 1009 287 87 representas las áreas que contienen la misma.
  - Crear el archivo de ejemplos negativos el formato por línea debe ser el siguiente.  
C:\data\placas\negativo\1.jpg, el mismo es la ruta de la imagen que no contiene la región de placa.
  - Crear el archivo con extensión .vec, para generar el mismo utilizamos la herramienta de OpenCv “opencv\_createsamples”, lo hacemos de la siguiente manera “opencv\_createsamples.exe -vec regiones\_placa.vec -info positivo.txt -bg negativo.txt -num 62 -h 8 -w 32”, donde regiones\_placa.vec es el archivo a crear, positivo.txt es el archivo de ejemplos positivos, negativo.txt es el archivo de ejemplos negativos, 62 es el número de ejemplos positivos, 8 y 32 es el alto y ancho base respectivamente para las regiones de placa.
  - Crear el archivo cascada de clasificadores xml, esto se lo hace mediante la herramienta de opencv “opencv\_traincascade.exe” se utiliza el siguiente comando “opencv\_traincascade.exe -data data -vec regiones\_placa.vec -bg negativo.txt - numPos 500 -numNeg 500 - numStages 10 -w 32 -h 8 -featureType LBP”, donde data es el directorio donde se guarda el archivo xml de la cascada generada, regiones\_placa.vec es el archivo generado en la creación de ejemplos positivos con la herramienta de OpenCv, negativo.txt es el archivo de

ejemplos negativos, numPos es el número de ejemplos positivos, numNeg es el número de ejemplos negativos, numStages es el número de etapas que implementará la cascada, featureType es el tipo de característica a utilizar para la generación de la cascada, 8 y 32 es el alto y ancho base respectivamente para las regiones de placa.

#### 2.1.4 Implementación Segmentación de los Caracteres de la Placa.

Una vez que se ha logrado aislar la placa de matrícula de los vehículos se procede a realizar entonces el segmentado de los caracteres para extraer y reconocerlos, se ejecuta las siguientes etapas:



Figura 57. Región de placa detectada.  
Elaboración: Juan Pablo Ordoñez Lopez.

- *Etiquetado de componentes conectados.*- En éste subproceso se buscan todos los componentes conectados en la región de la placa detectada, el resultado de ésta etapa genera la imagen de la Figura 58.



Figura 58. Etiquetado de componentes conectados.  
Elaboración: Juan Pablo Ordoñez Lopez.

- *Análisis de picos y valles.*- Este método es utilizado para separar caracteres enlazados como un solo componente conectado, el resultado al aplicar el algoritmo se puede observar en la Figura 59.



Figura 59. Caracteres de placa segmentados.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 2.1.5 Implementación Reconocimiento de Caracteres de la Placa.

Para obtener los caracteres que representan cada segmento de imagen extraído de la región de la placa se utiliza una máquina de vectores de soporte (SVM), considerándose tres características principales, la dimensión característica definitiva de cada carácter es de 112 (16+64+32) para el método seleccionado (Wang et al., 2007), para implementar se utilizó la clase SVM que proporciona la librería opencv:

- *Conteos contorno-cruce (CC)*.- Consiste en el conteo vertical y horizontal de las intersecciones entre las líneas y la región del carácter, el tamaño de característica es 8\*2 (Wang et al., 2007).

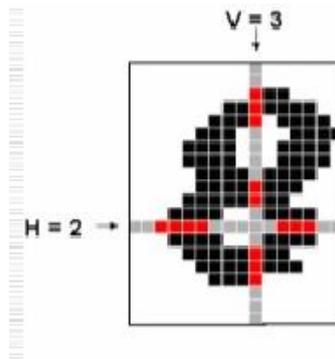


Figura 60. Extracción característica CC.  
Fuente: (Vamvakas, 2014)

- *Conteos direccionales (DC)*.- La característica DC representa el número de puntos de contorno en cuatro direcciones principales. Como se muestra en la Figura 61. (b), que dividen las direcciones en cuatro grupos: G1, G2, G3 y G4, que corresponden a los trazos horizontal, vertical, diagonal y diagonal inversa, respectivamente, el tamaño de característica es 4\*4\*4 (Wang et al., 2007). Los rangos de los ángulos de cada grupo se indican a continuación con una parte de la codificación.

```

if (0 <= faseEnGrados && faseEnGrados < 22.5 || 157.5 <= faseEnGrados && faseEnGrados <= 180) {
    G1++;
} else {
    if (22.5 <= faseEnGrados && faseEnGrados < 67.5) {
        G2++;
    } else {
        if (67.5 <= faseEnGrados && faseEnGrados < 112.5) {
            G3++;
        } else {
            if (112.5 <= faseEnGrados && faseEnGrados < 157.5) {
                G4++;
            }
        }
    }
}

```

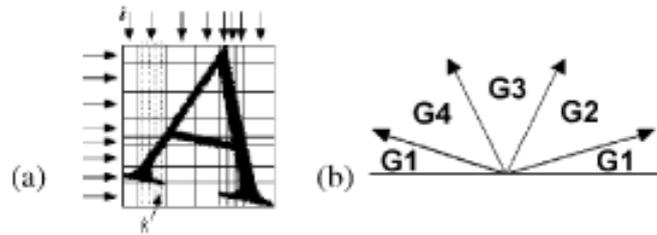


Figura 61. Características prototipos OCR a) CC y b) DC.  
Fuente: (Wang et al., 2007).

- *Área de fondo periférica (PBA)*.- La característica PBA es la longitud de los segmentos de línea que comienzan desde el límite de la imagen hacia el contorno de los caracteres como se puede observar en la Figura 62, el tamaño de característica es  $8*2*2$  (Wang et al., 2007).

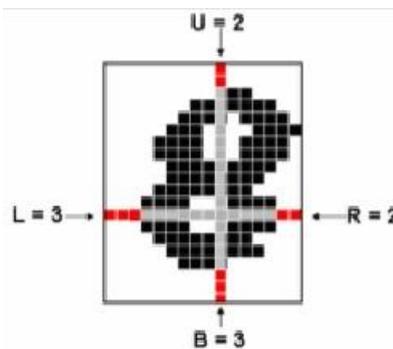


Figura 62. Extracción característica PBA.  
Fuente: (Vamvakas, 2014).

Las tres características se la obtuvo mediante la implementación del método *generarCaracteristicasOCR(Mat regionCaracterBinario, Mat imagenIntegral, Rect areaRegionCaracter)* en la clase GestorSVM el mismo recibe como parámetros de entrada la región del carácter en binario, la imagen integral de la región de la placa y la región del carácter representado por un rectángulo, como resultado se devuelve un vector de dimensión 112 que contiene las características de un carácter.

- *Entrenamiento de la máquina de vectores de soporte*.- Para entrenar la SVM se considera los segmentos de imagen de los caracteres extraídos de las regiones de placas utilizadas para el entrenamiento de las mismas, para ello se usa una aplicación diseñada para asociar una imagen que representa a una letra o dígito como se puede observar en la Figura 63.

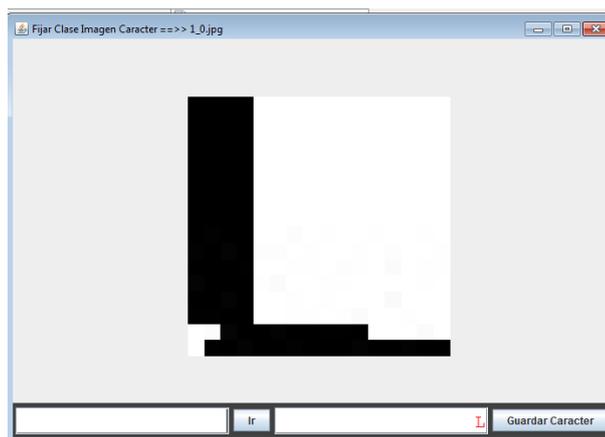


Figura 63. Aplicación para asociar una imagen a un carácter o dígito.  
Elaboración: Juan Pablo Ordoñez Lopez.

Al finalizar la asociación de los caracteres seleccionados a una letra o dígito se genera un archivo con el nombre `caracterClase.txt` el mismo contiene líneas de texto con el siguiente formato `1_0.jpg|L`, donde `1_0.jpg` es la imagen del carácter y `L` es la letra o dígito que representa la misma.

Una vez creado el archivo `caracterClase.txt` para entrenar la máquina de vectores de soporte se procede a ejecutar el siguiente método `generarVectoresCaracteristicasConjuntoEntrenamiento(String rutaConjuntoEntrenamiento, Map<String, String> nombreImagenClase)` para extraer el vector de características de cada imagen, los mismos que son suministrados a la clase SVM con sus respectiva clase para proceder a realizar el entrenamiento.

Ahora la máquina de vectores de soporte está entrenada y lista para reconocer la letra o dígito a la que representa una imagen esto se lo hace por llamar a ejecutar el método `reconocer(Mat regionCaracterBinario, Mat imagenIntegral, Rect areaRegionCaracter)`, éste recibe como parámetros de entrada la región del carácter en binario, la imagen integral de la región de la placa y el área del carácter representado por un rectángulo.

### Pruebas de aceptación.

Para realizar la aceptación del sistema se utiliza el artefacto “Prueba de aceptación” de la metodología XP.

#### 2.1.1 Validar Captura del par de imágenes y Guardado.

<b>Caso de Prueba de Aceptación</b>	
<b>Número de Caso de Prueba:</b> 1	<b>Historia de Usuario #:</b> 001
<b>Nombre:</b> Validar Captura del par de imágenes y Guardado.	
<b>Descripción:</b> Se validará que se Capture el par de imágenes de manera síncrona mediante la verificación de la hora de creación que indique las propiedades del archivo de imagen en el sistema operativo.	

<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La prueba se realizará en el sistema operativo Windows 7.</li> <li>• Se debe tener acceso al par de cámaras ip o de pc.</li> </ul>
<p><b>Entrada / Pasos de Ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Alistar el par de cámaras, revisar que estén lo más paralelo posible.</li> <li>2. Verificar que se tiene acceso a las cámaras.</li> <li>3. Ejecutar el programa detector de placas.</li> <li>4. Seleccionar el tipo de cámaras en el combo ip o de pc.</li> <li>5. Seleccionar la ruta estéreo donde se desea guardar los pares de imágenes capturados.</li> <li>6. Presionar en el botón Capturar Par para Guardar las capturas.</li> <li>7. Verificar que las imágenes fueron guardadas en la ruta estéreo seleccionada.</li> <li>8. Verificar la hora, minutos y segundos de la creación del par de imágenes.</li> </ol>
<p><b>Resultado Esperado:</b> La hora, minutos y segundos debe coincidir en las propiedades de los archivos del par de imágenes capturadas.</p>
<p><b>Evaluación de la prueba:</b></p> <p>Versión 1: Positiva</p> <p>Versión 2: Positiva</p>

### 2.1.2 Validar Calibración de cámaras.

<b>Caso de Prueba de Aceptación</b>	
<b>Número de Caso de Prueba:</b> 2	<b>Historia de Usuario #:</b> 002
<b>Nombre:</b> Validar Calibración de cámaras.	
<b>Descripción:</b> Se validará la calibración de cámaras mediante el método del tablero de ajedrez.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La prueba se realizará en el sistema operativo Windows 7.</li> <li>• Se debe tener acceso al par de cámaras ip o de pc.</li> <li>• Contar con un tablero de ajedrez impreso, longitud en mm de los cuadros del tablero impreso, número de esquinas interiores tanto horizontalmente como verticalmente.</li> </ul>	
<p><b>Entrada / Pasos de Ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Alistar el par de cámaras, revisar que estén lo más paralelo posible.</li> <li>2. Verificar que se tiene acceso a las cámaras.</li> <li>3. Ejecutar el programa detector de placas.</li> <li>4. Seleccionar el tipo de cámaras en el combo ip o de pc.</li> <li>5. Seleccionar la ruta estéreo donde se desea guardar los pares de imágenes capturados.</li> </ol>	

<p>6. Ubicar el tablero de ajedrez de tal manera que se lo pueda visualizar en las dos cámaras completamente.</p> <p>7. Presionar en el botón Capturar Par para Guardar las capturas.</p> <p>8. Repetir los pasos 6 y 7 para capturar los pares de imágenes que se deseen.</p> <p>9. Una vez que se tenga el conjunto de par de imágenes deseado presionar el botón Calibrar cámaras para realizar el proceso de Calibrado.</p>
<p><b>Resultado Esperado:</b> Verificar que se han creado las matrices de las cámaras o parámetros intrínsecos mx1.txt, mx2.txt, my1.txt, my2.txt y la matriz de parámetros extrínsecos q.txt en la ruta estéreo seleccionada.</p>
<p><b>Evaluación de la prueba:</b></p> <p>Versión 1: Positiva</p> <p>Versión 2: Positiva</p>

### 2.1.3 Validar tasa de exactitud detección de vehículos.

<b>Caso de Prueba de Aceptación</b>	
<b>Número de Caso de Prueba:</b> 3	<b>Historia de Usuario #:</b> 003
<b>Nombre:</b> Validar Tasa de Detección de Vehículos.	
<b>Descripción:</b> Se validará la Tasa de aciertos en la detección de vehículos en imágenes que contienen vehículos.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La prueba se realizará en el sistema operativo Windows 7.</li> <li>• Se debe tener acceso al par de cámaras ip o de pc.</li> <li>• El proceso de Calibración de cámaras debe estar realizado, las cámaras no se deben mover una vez que se ha realizado el proceso antes mencionado, con el fin de obtener los mejores resultados.</li> </ul>	
<p><b>Entrada / Pasos de Ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Verificar que se tiene acceso a las cámaras.</li> <li>2. Ejecutar el programa detector de placas.</li> <li>3. Seleccionar el tipo de cámaras en el combo ip o de pc.</li> <li>4. Presionar en el botón Iniciar para empezar con el proceso de detección de vehículos y sus placas.</li> <li>5. Verificar en una muestra de N imágenes en el video que contengan vehículos cuantos son detectados por el sistema.</li> </ol>	

6. Calcular la Tasa de aciertos con la siguiente formula $TA = \frac{\text{Vehículos Detectados} * 100}{\text{Vehículos Reales}}$ .
<b>Resultado Esperado:</b> La tasa de aciertos debe ser aproximada o mayor al 80%.
<b>Evaluación de la prueba:</b> Versión 1: 67% Negativa Versión 2: No se realizó

#### 2.1.4 Validar tasa de exactitud de la región de la placa, segmentación y reconocimiento de caracteres en vehículos detectados.

<b>Caso de Prueba de Aceptación</b>	
<b>Número de Caso de Prueba:</b> 4	<b>Historia de Usuario #:</b> 004,005,006
<b>Nombre:</b> Validar Tasa de Detección de los Caracteres de la Placa.	
<b>Descripción:</b> Se validará la Tasa de aciertos en la detección de la región de la placa, segmentación y reconocimiento de caracteres en vehículos detectados.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>• La prueba se realizará en el sistema operativo Windows 7.</li> <li>• Se debe tener acceso al par de cámaras ip o de pc.</li> <li>• El proceso de Calibración de cámaras debe estar realizado, las cámaras no se deben mover una vez que se ha realizado el proceso antes mencionado, con el fin de obtener los mejores resultados.</li> <li>• Se debe haber detectado un vehículo.</li> </ul>	
<b>Entrada / Pasos de Ejecución:</b> <ol style="list-style-type: none"> <li>1. Verificar que se tiene acceso a las cámaras.</li> <li>2. Ejecutar el programa detector de placas.</li> <li>3. Seleccionar el tipo de cámaras en el combo ip o de pc.</li> <li>4. Presionar en el botón Iniciar para empezar con el proceso de detección de vehículos y sus placas.</li> <li>5. Verificar en una muestra de N imágenes que contengan vehículos detectados cuantas placas, su segmentación y sus caracteres son detectados correctamente por el sistema.</li> <li>6. Calcular la Tasa de aciertos con la siguiente formula <math>TA = \frac{\text{Objetos detectados Correctamente} * 100}{\text{Objetos reales}}</math>.</li> </ol>	
<b>Resultado Esperado:</b> La Tasa de aciertos debe ser aproximada o mayor al 80%.	
<b>Evaluación de la prueba:</b> Versión 1: Detección región placa 76,15% Negativa	

Segmentación de caracteres 72,37% Negativa

Reconocimiento de caracteres 65,16% Negativa

Versión 2:

Detección región placa 84,95% Positiva

Segmentación de caracteres 79,45% Positiva

Reconocimiento de caracteres 68,5% Negativa

## RESULTADOS

Para evaluar el sistema se desarrolló dos versiones del aplicativo, en la primera versión se buscan los principales problemas de los cuatro métodos implementados y en la segunda versión se realizan mejoras para lograr una mejor precisión y exactitud de los mismos.

### Versión 1.

Al evaluar el aplicativo 1 se obtiene indicadores con respecto a cada fase principal del sistema LPR. Las pruebas se realizaron con un conjunto de 200 (1.jpg a 200.jpg) imágenes que contienen vehículos y sus placas.

Para calcular las tasas de exactitud del sistema se utilizó las siguientes ecuaciones (Powers, 2011):

$$TPR = \frac{TP}{P} \quad (51)$$

$$FPR = \frac{FP}{N} \quad (52)$$

Donde:

*TP* son los verdaderos positivos (objetos reales reconocidos)

*FP* los falsos positivos (escenas que no contienen los objetos y son reconocidos)

*P* la cantidad de escenas que contienen el objeto.

*N* la cantidad de escenas que no contienen el objeto.

Los resultados se resumen en la Tabla 8.

Tabla 8. Tasas de exactitud del sistema propuesto.

TDCRP	TFARP	TSCRCP	TRCC
76%	6%	72,37%	65,16%

Donde:

TDCRP: Tasa de detecciones correctas en regiones de placa.- Para el cálculo se utilizó (51) con los siguientes valores TP=152, P=200.

TFARP: Tasa de falsas alarmas en regiones de placa.- Para el cálculo se utilizó (52) con los siguientes valores FP=12, N=200.

TSCRCP: Tasa de segmentación de caracteres en la región de la placa.- Para el cálculo se utilizó (51) con los siguientes valores TP=110, P=152.

TRCC: Tasa de reconocimiento correcto de caracteres.- Para el cálculo se utilizó (51) con los siguientes valores TP=576, P=884.

Las pruebas de cada método se las describe a continuación:

#### 3.1.1 Detección de vehículos.

Al validar el método para la detección de vehículos se observa que este trabaja con una distancia mayor a los 3.64 metros, esto ocurre porque no se generan regiones homogéneas para los mapas de disparidad en imágenes que contienen vehículos como se observa en la Figura 64, como se puede ver en la imagen en la que el vehículo está ubicado a una distancia de 1.92 metros no se puede detectar.



Figura 64. Mapas de disparidad para un vehículo ubicado a diferentes distancias.  
Elaboración: Juan Pablo Ordoñez Lopez.

En la Figura 65 se observa por que el mapa de disparidad no es de buena calidad, esto se debe a que la calibración utilizada no realiza las correcciones necesarias en la rectificación de imágenes, ocasionando correspondencia incorrecta de puntos comunes entre imágenes, lo que genera el problema.



Figura 65. a) Rectificación imagen izquierda, b) Rectificación imagen derecha y c) Correspondencia de imágenes.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.2 Detección de la región de la placa.

Al verificar el sistema con respecto a la detección de la región de la placa, se observa que el método falla en imágenes donde el área de placa tiene una inclinación mayor o igual a los 5 grados, esto se puede apreciar en la Figura 66.



Figura 66. Problema en la extracción de las regiones de placa inclinada.  
Elaboración: Juan Pablo Ordoñez Lopez.

En la Figura 67 se observa imágenes de baja calidad estas generan problemas al extraer la región candidata de la placa.



Figura 67. Problema al generar las regiones candidatas a placa en imagen con brillos.  
Elaboración: Juan Pablo Ordoñez Lopez.

Otro escenario donde hay problemas se produce cuando las imágenes son tomadas de forma lateral con respecto al vehículo, lo que ocasiona que el clasificador en cascada rechace las regiones extraídas, los casos se observan en la Figura 68.



Figura 68. Problema en regiones de placa laterales no reconocidas por el clasificador en cascada.

Elaboración: Juan Pablo Ordoñez Lopez.

En la etapa de verificación mediante la cascada de clasificadores Haar-Like, existe ciertas regiones de placa que no las considera como tal, se puede observar en la Figura 69.



Figura 69. Problema en regiones de placa en condiciones ideales no reconocidas por el clasificador en cascada.

Elaboración: Juan Pablo Ordoñez Lopez.

Finalmente, al verificar el método implementado para la detección de la región de la placa, se observó que la mayoría de falsas alarmas ocurre dentro de la región de interés, donde existen palabras similares a la forma de la región de la placa, especialmente en buses públicos, escolares y taxis como el de la Figura 70.



Figura 70. Detección de falsas alarmas en vehículos con textos.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.3 Segmentación de los caracteres de la placa.

En el proceso de segmentación de los caracteres, se observó que el método tiene fallas al segmentar las placas que tienen 6 caracteres (placas antiguas), según el formato actual en Ecuador se permiten hasta 7 caracteres, lo que causa que en una región donde hay 6 caracteres el borde derecho sea considerado como un carácter adicional, el resultado se puede observar en la Figura 71.

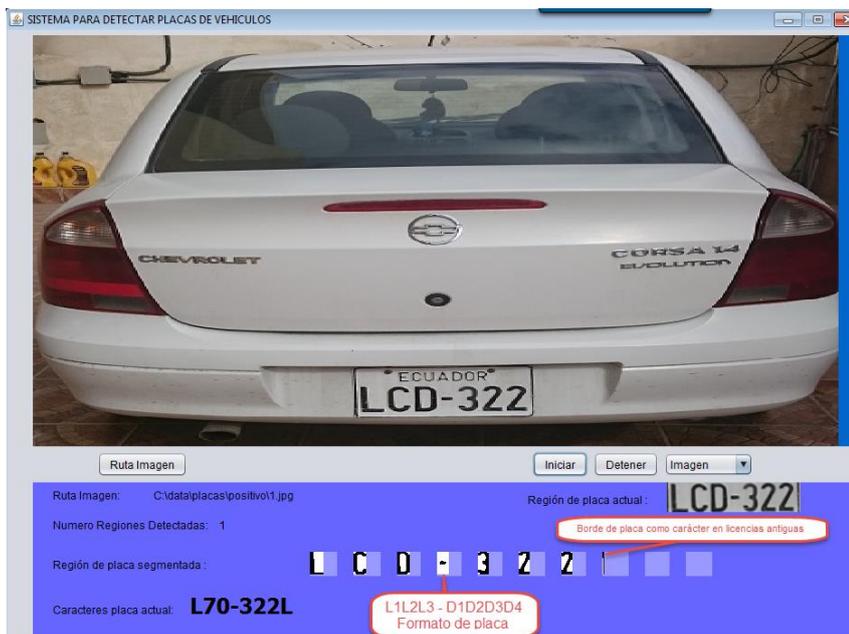


Figura 71. Problema en la segmentación de caracteres en placas antiguas.  
Elaboración: Juan Pablo Ordoñez Lopez.

Otro problema al segmentar ocurre en los caracteres enlazados mediante un marco adicional en la placa como en la Figura 72.

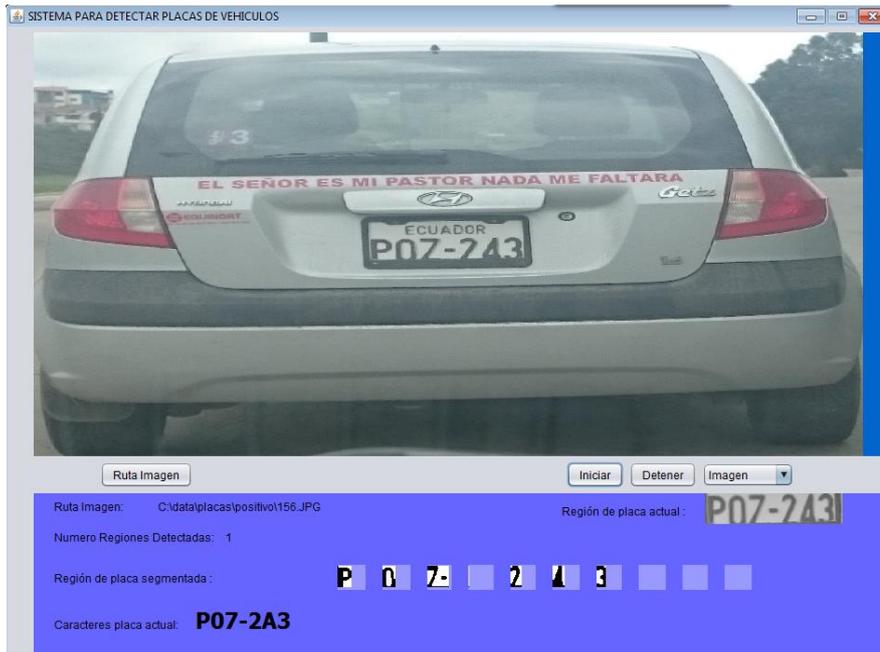


Figura 72. Problema con los caracteres enlazados entre sí mediante borde de la región de la placa.

Elaboración: Juan Pablo Ordoñez Lopez.

Otro problema en el proceso de segmentación ocurre cuando en la imagen hay irregularidades en los brillos, lo que produce caracteres cortados, el resultado se lo puede ver en la Figura 73.

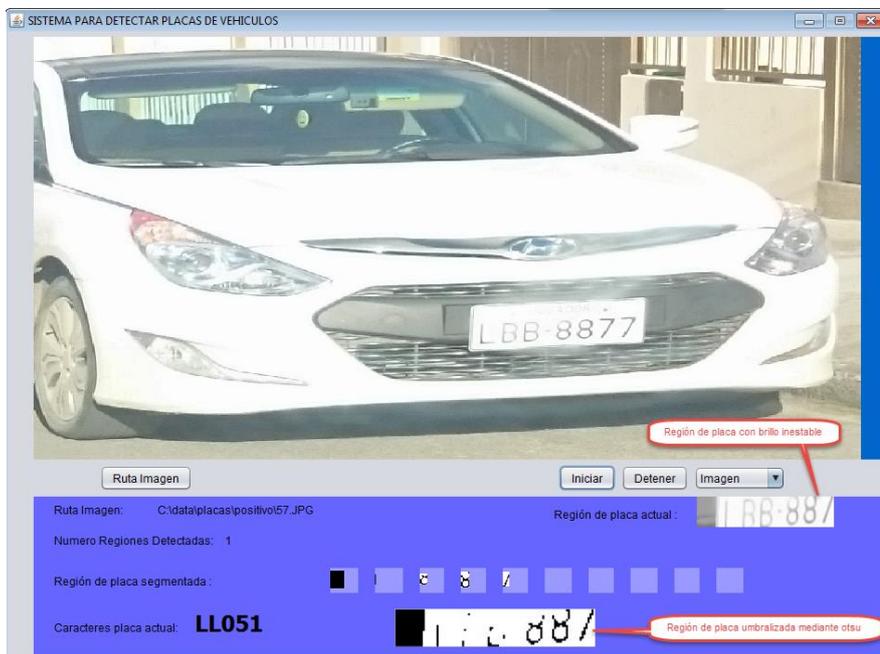


Figura 73. Problema en la segmentación de caracteres para imágenes con brillos.

Elaboración: Juan Pablo Ordoñez Lopez.

Finalmente, se presenta el problema donde hay un color de fondo fuerte lo que produce una mezcla entre los caracteres y el fondo de la placa, el resultado se observa en la Figura 74.

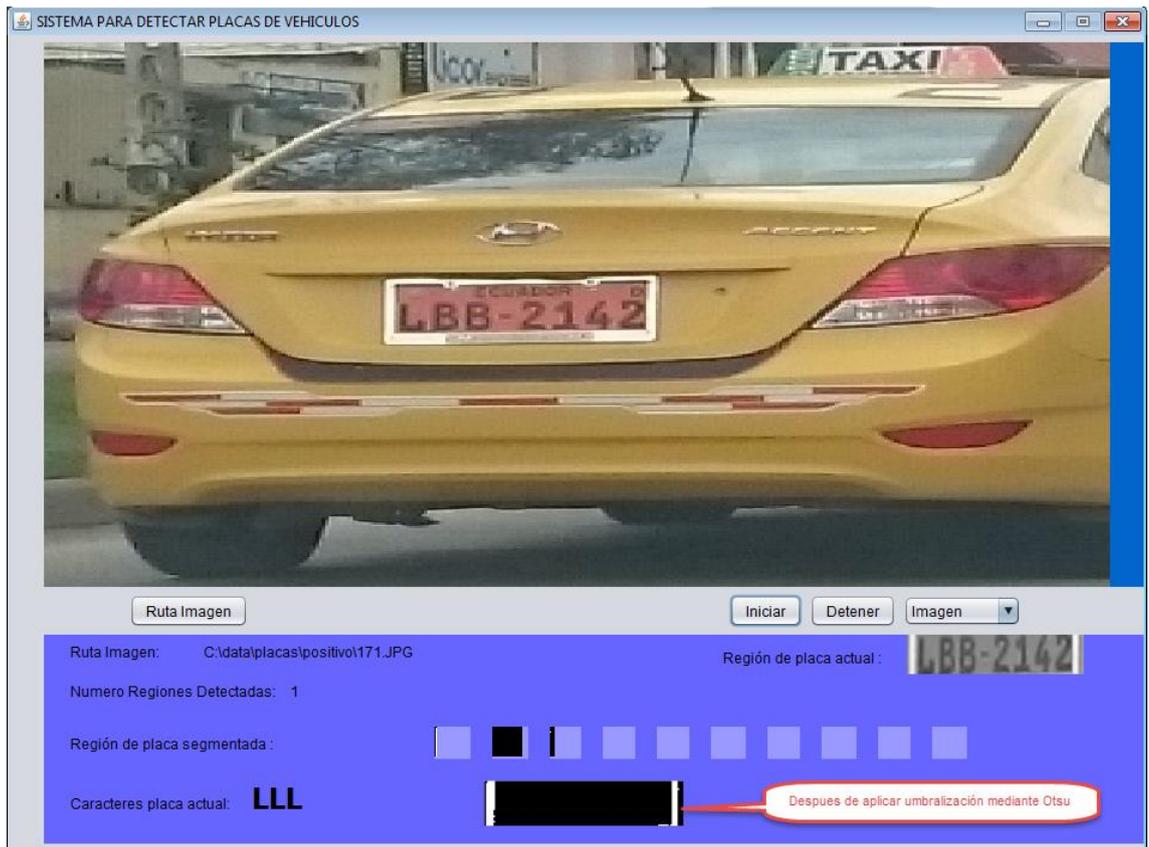


Figura 74. Problema en la segmentación de caracteres en regiones de placa con fondo de color fuerte.

Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.4 Reconocimiento de caracteres.

En el proceso de reconocimiento de caracteres se observa que los principales problemas ocurren cuando las imágenes de los caracteres contienen grietas, ruidos y manchas en el exterior, algunos de los caracteres se enlazan con los bordes de la placa a través de tornillos y suciedad, estos factores hacen variar sus características que son la entrada para el reconocimiento mediante el clasificador SVM (máquina de vectores de soporte), el resultado se lo puede observar en la Figura 75.

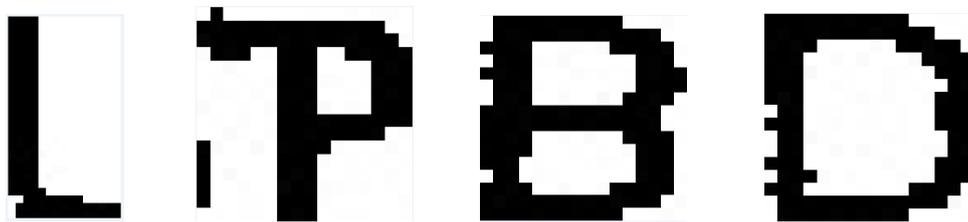


Figura 75. Problema en regiones de carácter con grietas y ruido como tornillos.

Elaboración: Juan Pablo Ordoñez Lopez.

## Versión 2.

En la versión 1 se presentaron muchos problemas en los cuatro procesos de reconocimiento de los caracteres de la placa, en esta versión se utilizó varias técnicas para mejorar la exactitud de cada uno de los métodos.

Se encontraron los siguientes indicadores con respecto a cada fase principal del sistema LPR. Las pruebas se realizaron con un conjunto de 186 imágenes (250 imágenes fueron seleccionados de manera aleatoria resultando 186 diferentes) que contienen vehículos y sus placas.

Los resultados se resumen en la Tabla 9, se usó las ecuaciones (51) y (52).

Tabla 9. Tasas de exactitud del sistema propuesto.

TDCRP	TFARP	TSCRCP	TRCC
84,95%	2,69%	79,45%	68,5%

Donde:

*TDCRP*: Tasa de detecciones correctas en regiones de placa.- Para el cálculo se utilizó (51) con los siguientes valores  $TP=158$ ,  $P=186$ .

*TFARP*: Tasa de falsas alarmas en regiones de placa.- Para el cálculo se utilizó (52) con los siguientes valores  $FP=5$ ,  $N=186$ .

*TSCRCP*: Tasa de segmentación de caracteres en la región de la placa.- Para el cálculo se utilizó (51) con los siguientes valores  $TP=116$ ,  $P=146$ .

*TRCC*: Tasa de reconocimiento correcto de caracteres.- Para el cálculo se utilizó (51) con los siguientes valores  $TP=137$ ,  $P=200$ .

Encontrados los principales problemas por cada uno de los métodos implicados en la identificación de caracteres de placa, se procedió a implementar las correcciones necesarias, los resultados se describen a continuación:

### 3.1.5 Detección de vehículos.

En esta versión se intentó mejorar la calidad de los mapas de disparidad, los resultados obtenidos no fueron exitosos, las imágenes de correspondencia se puede observar en la Figura 76.

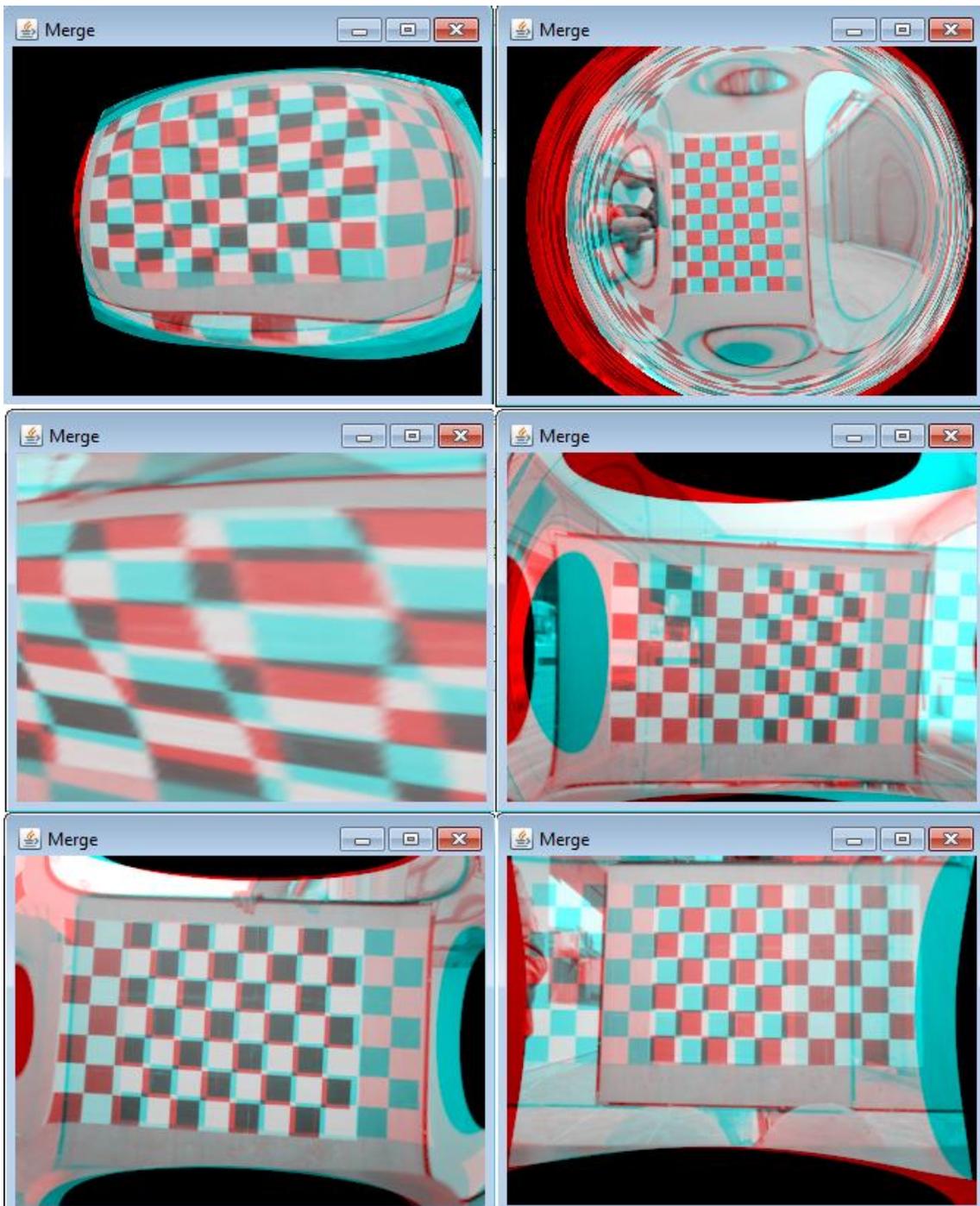


Figura 76. Merge o correspondencia del par de imágenes izquierda y derecha para diferentes pruebas de calibración.

Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.6 Detección de la región de la placa.

En esta versión para mejorar las detecciones positivas de las regiones de placa se calculó el histograma de la imagen que contiene el vehículo, esto para estabilizar el brillo. Así también se utilizó la operación morfológica *dilatación*, para engrosar los caracteres con mayor fuerza.

La imagen dilatada resulta en la entrada al proceso de detección de la región de la placa, se observa que al aplicar la gradiente vertical resaltan mejor la región de los caracteres que es muy importante para detectar la misma completamente sin cortes, el resultado se lo puede ver en la Figura 77.



Figura 77. a) Imagen ecualizada, b) Imagen dilatada y c) Generación de áreas candidatas.  
Elaboración: Juan Pablo Ordoñez Lopez.

A pesar de tener una mejor precisión en la detección de la región de la placa con los cambios implementados, se obtuvieron casos de detecciones de regiones parciales de placa, el resultado se lo puede observar en la Figura 78.



Figura 78. Vehículo y detección de la región de placa parcial.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.7 Segmentación de la región de la placa.

Para mejorar la tasa de exactitud en el proceso de segmentación de caracteres se implementó un subproceso para remover inclinaciones y sobrantes que no pertenecen a los caracteres de la placa, este último se hizo mediante proyección horizontal y vertical, los resultados se los puede observar en la Figura 79.



Figura 79. a) Región de placa detectada, b) Placa sin inclinación y c) Región sin sobrantes.  
Elaboración: Juan Pablo Ordoñez Lopez.

### 3.1.8 Reconocimiento de caracteres.

Finalmente, para mejorar la tasa de exactitud en el reconocimiento de caracteres se implementó el uso de operaciones morfológicas, en este caso se utilizó la operación Opening cuya características hacen que se remueva el ruido exterior, los resultados se observan en la Figura 80.



Figura 80. a) Región de carácter con grietas y b) Carácter después de aplicar opening. Elaboración: Juan Pablo Ordoñez Lopez.

### Exactitud del sistema en las versiones implementadas.

Según se observa en la Figura 81 la etapa de detección de la placa y la segmentación de caracteres tiene la mayor tasa de exactitud con un porcentaje aproximado al 80%, mientras que el reconocimiento de caracteres tiene una tasa aproximada del 68%.

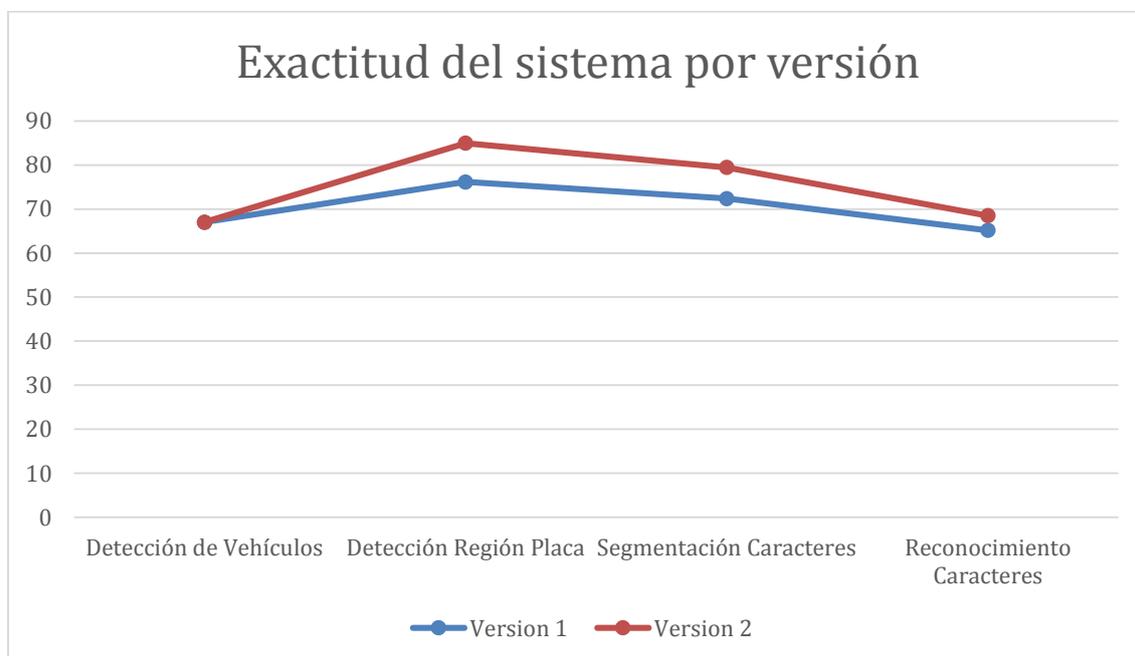


Figura 81. Exactitud del sistema por versión. Elaboración: Juan Pablo Ordoñez Lopez.

A continuación en la Tabla 10 se presenta un resumen de los problemas encontrados y sus respectivas soluciones y las técnicas empleadas.

Tabla 10. Comparación de versiones implementadas.

Problema	Versión 1	Versión 2	Solución
Mapas de disparidad de mala calidad.	Se presenta el problema	Se presenta el problema	Obtención de nuevas matrices de distorsión de las cámaras (calibración), sin resultado exitoso.
Perdida de información en la región de la placa.	Se presenta el problema	Mejorado	Se ecualizó y dilató la imagen.
No generación de regiones de placa en imágenes con brillos.	Se presenta el problema	Mejorado	Se entrenó el método Haar-Like con 30 imágenes más.
Rechazo de regiones de placa en tomas laterales y frontales de vehículos.	Se presenta el problema	Mejorado	Se entrenó el método Haar-Like con 30 imágenes más.
Segmentación incorrecta en placas antiguas de seis caracteres.	Se presenta el problema	Corregido	Se removió inclinaciones y sobrantes a los caracteres.
Segmentación incorrecta en imágenes de placa con brillos.	Se presenta el problema	Mejorado	Se ecualizó la imagen.
Caracteres con grietas	Se presenta el problema	Mejorado	Aplicación de operación morfológica opening.

## CONCLUSIONES

Al culminar el presente trabajo de fin de titulación, se concluyó que:

- El sistema implementado tiene un tiempo promedio de 434,51 ms en la ejecución de las cuatro fases, mientras los métodos seleccionados suman un total de 78 ms, el mismo nos indica que nuestro sistema resuelve la identificación de caracteres en 5,57 veces el tiempo que emplean los métodos seleccionados, el tiempo es razonable sabiendo que el sistema está desarrollado con el lenguaje java que se ejecuta sobre una máquina virtual, además el método seleccionado en la detección de vehículos tiene hardware especializado para generar los mapas de disparidad, los mismos en nuestro sistema implementado son generados mediante librerías que consumen en promedio 60 ms.
- Se desarrolló un aplicativo para la detección de vehículos mediante un sistema estéreo (par de cámaras), el mismo detecta y asigna una distancia del vehículo al sistema de cámaras.
- Se desarrolló un aplicativo que permite la detección de la región de la placa, segmentación y reconocimiento de caracteres, los resultados fueron positivos en las fases de detección de la placa y segmentación de caracteres, la tasa de exactitud aproximada es del 80%, así también el reconocimiento de caracteres presenta una tasa de exactitud aproximada al 70%.
- El módulo implementado para la detección de vehículos no pudo ser desarrollado al 100%, se presentaron problemas al calibrar las cámaras y no se lograron crear mapas de disparidad satisfactorios, lo que bloqueó las mejoras en la exactitud de detección.
- Es posible utilizar técnicas de reconocimiento de objetos como la cascada de clasificadores (Haar-Like) y la máquina de vectores de soporte (SVM) para realizar la verificación de regiones de placa y caracteres, respectivamente, mediante la extracción de características de los mismos.
- Es posible obtener la distancia aproximada a un objeto a partir de una par de cámaras.
- Las regiones de placa pueden ser detectadas con su ancho ubicado en el rango de 60 pixeles hasta los 140 pixeles.

## RECOMENDACIONES

Al culminar el presente trabajo de fin de titulación, se recomienda lo siguiente:

- Utilizar un sistema estéreo con hardware especializado, con este se gana precisión en la generación de los mapas de disparidad y tiempos de procesamiento, debido a que no se tienen problemas de movimientos en las cámaras, lo que afecta la generación correcta de los mapas mencionados.
- A partir de la presente investigación se puede realizar seguimiento de autos, control de velocidad y conteo de vehículos.
- Para integrar la detección de vehículos con el reconocimiento de caracteres se recomienda mejorar los tiempos de ejecución de los procesos implicados, evitando el uso en lo posible de librerías que tienen que leer toda la imagen cada vez que se invoca un método.
- Mejorar el proceso de segmentación incorrecta en imágenes de placa con colores de fondo fuertes (ejemplo color rojo).
- Corregir problemas en el reconocimiento de caracteres que tienen en su exterior ruido como tornillos.

## REFERENCIAS

- Abdías Gómez D., L. P. y O. O. (2014). SMART-TEST : una aplicación de las TIC al proceso de evaluación de estudiantes universitarios, 42–46.
- Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V., & Kayafas, E. (2008). License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 9(3), 377–391. <http://doi.org/10.1109/TITS.2008.922938>
- Athreyas, N., Lai, Z., Gupta, J., & Gupta, D. (2014). A NALOG SIGNAL PROCESSING SOLUTION FOR.
- Barba, R. (2015). Utilización de métodos de visión artificial para PC como apoyo en la automoción.
- Cadavid, A. N., Daniel, J., Martínez, F., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva* 11.2, 11, 30–39. Retrieved from <http://ojs.uac.edu.co/index.php/prospectiva/article/viewFile/36/21>
- Chinese License Plate. (2012). Retrieved from [https://commons.wikimedia.org/wiki/File:Chinese\\_License\\_Plate.jpg](https://commons.wikimedia.org/wiki/File:Chinese_License_Plate.jpg)
- Chiu, C. C., Chung, M. L., & Chen, W. C. (2010). Real-time front vehicle detection algorithm for an asynchronous binocular system. *Journal of Information Science and Engineering*, 26(3), 735–752. Retrieved from [http://www.iis.sinica.edu.tw/page/jise/2010/201005\\_01.pdf](http://www.iis.sinica.edu.tw/page/jise/2010/201005_01.pdf)
- Corporation, N. I. (2010). Reconocimiento de imágenes, 41–54.
- Davison, A. (2013). Chapter NUI-13 . Depth Processing, 1(June), 1–44.
- DTREG. (2014). SVM - Support Vector Machines.
- Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013). Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2), 311–325. <http://doi.org/10.1109/TCSVT.2012.2203741>
- Duan, T. D. D. T. D., Duc, D. A. D. D. A., & Du, T. L. H. D. T. L. H. (2004). Combining Hough transform and contour algorithm for detecting vehicles' license-plates. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, 747–750. <http://doi.org/10.1109/ISIMP.2004.1434172>
- Gao, C., & Lu, S. L. (2008). Novel FPGA based haar classifier face detection algorithm acceleration. *Proceedings - 2008 International Conference on Field Programmable Logic and Applications, FPL*, 373–378. <http://doi.org/10.1109/FPL.2008.4629966>
- Gazcón, N. F., Chesñevar, C. I., & Castro, S. M. (2012). Automatic vehicle identification for Argentinean license plates using intelligent template matching. *Pattern*

- Recognition Letters*, 33(9), 1066–1074. <http://doi.org/10.1016/j.patrec.2012.02.004>
- Haar-feature Object Detection in C#. (2014). Retrieved from <http://www.codeproject.com/Articles/441226/Haar-feature-Object-Detection-in-Csharp>
- Hikvision. (2014). VF IR Bullet Network Camera, 2632.
- Khalil, M. (2010). Car plate recognition using the template matching method. *International Journal of Computer Theory and ...*, 2(5), 3–7. Retrieved from <http://ijcte.org/papers/224-G290.pdf>
- Lee, H. J., Chen, S. Y., & Wang, S. Z. (2004). Extraction and recognition of license plates of motorcycles and vehicles on highways. *Proceedings - International Conference on Pattern Recognition*, 4, 356–359. <http://doi.org/10.1109/ICPR.2004.1333776>
- Lizotte, D., Wang, T., Bowling, M., & Schuurmans, D. (2007). Automatic gait optimization with Gaussian process regression. *IJCAI International Joint Conference on Artificial Intelligence*, 944–949.
- Matlab binary image. (2015). Retrieved from <http://navneetsau.com/matlab-binary-image/>
- Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., ... Graf, T. (2004). High accuracy stereo vision system for far distance obstacle detection. *IEEE Intelligent Vehicles Symposium, 2004*, 292–297. <http://doi.org/10.1109/IVS.2004.1336397>
- Negri, P., Clady, X., Hanif, S., & Prevost, L. (2008). A Cascade of Boosted Generative and Discriminative Classifiers for Vehicle Detection. *EURASIP Journal on Advances in Signal Processing*, 2008(1), 782432. <http://doi.org/10.1155/2008/782432>
- Nguyen, V. D., Nguyen, T. T., Nguyen, D. D., Lee, S. J., & Jeon, J. W. (2013). A Fast Evolutionary Algorithm for Real-Time Vehicle Detection. *IEEE Transactions on Vehicular Technology*, 62(6), 2453–2468. <http://doi.org/10.1109/TVT.2013.2242910>
- OpenCV Color Concentration Histogram. (2013). Retrieved from <http://stackoverflow.com/questions/9351278/opencv-color-concentration-histogram>
- Ozbay, S., & Ercelebi, E. (2005). Automatic vehicle identification by plate recognition. *World Academy of Science, Engineering and ...*, 222–225. Retrieved from [http://pdf.aminer.org/000/349/486/gray\\_scale\\_character\\_recognition\\_by\\_gabor\\_jet\\_s\\_projection.pdf](http://pdf.aminer.org/000/349/486/gray_scale_character_recognition_by_gabor_jet_s_projection.pdf)
- Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation, 2(1), 24. <http://doi.org/10.1.1.214.9232>

- Shi, X., Zhao, W., & Shen, Y. (2005). Automatic License Plate Recognition System Based on Color Image Processing. *Computational Science and Its Applications – ICCSA 2005 International Conference, Singapore, May 9-12, 2005, Proceedings, Part IV*, 1159–1168.
- Valencia, U. P. D. E. (2014). Reconocimiento De Matrículas De Coche ”.
- Vamvakas, G. (2014). Off line handwritten optical character recognition OCR. Retrieved from <http://www.slideshare.net/lovebot/off-line-handwritten-optical-character-recognition-ocr>
- Vehicle registration plates of Taiwan. (2007). Retrieved from [https://en.wikipedia.org/wiki/Vehicle\\_registration\\_plates\\_of\\_Taiwan](https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Taiwan)
- Vericat, F., Stoico, C. O., Carlevaro, C. M., & Renzi, D. G. (2011). Genetic algorithm for the pair distribution function of the electron gas. *Interdisciplinary Sciences: Computational Life Sciences*, 3(4), 283–289. <http://doi.org/10.1007/s12539-011-0108-3>
- Vincent, O. R., & Folorunso, O. (2009). A Descriptive Algorithm for Sobel Image Edge Detection. *Proceedings of Informing Science & IT Education Conference (InSITE) 2009*, 1–11.
- Viola, P., & Jones, M. J. (2001). Robust Real-time Object Detection. *International Journal of Computer Vision*, (February), 1–30. <http://doi.org/10.1.1.23.2751>
- Vivotek. (2015). Retrieved from <http://www.vivotek.com/es/vivotek-lanca-la-ip816a-lpc-su-m%C3%A1s-moderna-soluci%C3%B3n-de-monitoreo-de-tr%C3%A1fico-en-tiempo-real-disen%C3%B1ada-para-la-captura-de-placas-de-autom%C3%B3viles/>
- Vorobioff, J., & G, R. C. A. B. N. (2011). Pattern Recognition and Machine Intelligence, 6744(November 2015), 3–4. <http://doi.org/10.1007/978-3-642-21786-9>
- Wang, S., Member, S., & Lee, H. (2007). A Cascade Framework for a Real-Time Statistical Plate Recognition System. *Ieee Transactions on Information Forensics and Security*, 2(2), 267–282. <http://doi.org/10.1109/TIFS.2007.897251>
- Zou, L. Z. L., & Li, Y. L. Y. (2010). A method of stereo vision matching based on OpenCV. *2010 International Conference on Audio Language and Image Processing*, 185–190. <http://doi.org/10.1109/ICALIP.2010.5684978>