



**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**  
*La Universidad Católica de Loja*

**ÁREA TÉCNICA**

**TÍTULO DE INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Diseño e implementación de una solución para el conteo de vehículos a partir de imágenes terrestres.**

TRABAJO DE TITULACIÓN

**AUTOR:** Sócola Escobar, Diego Andrés.

**DIRECTOR:** Aguirre Reyes, Daniel Fernando Ph.D.

LOJA - ECUADOR

2017



*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>*

*Septiembre, 2017*

## APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ph.D.

Daniel Fernando Aguirre Reyes.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Diseño e implementación de una solución para el conteo de vehículos a partir de imágenes terrestres, realizado por Sócola Escobar Diego Andrés, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, junio de 2017

f) .....

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo Sócola Escobar Diego Andrés declaro ser autor del presente trabajo de titulación: Diseño e implementación de una solución para el conteo de vehículos a partir de imágenes terrestres, de la Titulación de Electrónica y Telecomunicaciones, siendo Daniel Fernando Aguirre Reyes director 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: Sócola Escobar Diego Andrés

Cédula: 1104932072

## **DEDICATORIA**

A Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mi hijo Matías Andrés y Katherine su madre, que durante todos estos años de carrera han sido fuente de inspiración y han sabido apoyarme para continuar y nunca renunciar, gracias por su amor incondicional.

A mis padres Víctor y Dolores, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaron. Gracias por darme una carrera para mi futuro; todo esto se los debo a ustedes.

A Mis hermanos, Elizabeth, Byron, Fabricio y Daniel, por estar conmigo y apoyarme siempre, los quiero mucho.

**Diego Andrés Sócola Escobar**

## **AGRADECIMIENTO**

En primer lugar, doy infinitamente gracias a Dios, por haberme dado fuerza y valor para culminar esta etapa de mi vida.

El agradecimiento especial para Daniel Fernando Aguirre Reyes, director del presente trabajo, quien, con su sapiencia e incommensurable humildad, ha jugado un papel preponderante en el éxito del mismo.

Agradezco también la confianza y el apoyo brindado por parte de mi hijo Matías, Katherine, mis padres, mis hermanos, que sin duda alguna en el trayecto de mi vida me han demostrado su amor, corrigiendo mis faltas y celebrando mis triunfos.

Finalmente agradezco a todos aquellos familiares y amigos que ayudaron directa e indirectamente a la realización de este Trabajo de Titulación.

## ÍNDICE DE CONTENIDOS

CARÁTULA.....	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 FIGURAS.....	ix
ÍNDICE DE TABLAS.....	xii
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
CAPÍTULO I. GENERALIDADES.....	5
1.1. Planteamiento del problema.....	6
1.2. Justificación.....	6
1.3. Objetivos.....	6
1.3.1. General.....	6
1.3.2. Específicos.....	6
CAPÍTULO II. MARCO TEÓRICO.....	7
2.1. Estado del arte.....	8
2.2. Fundamentación teórica.....	16
2.2.1. Adquisición de la imagen.....	16
2.2.1.1. <i>Imagen digital</i> .....	16
2.2.2. Procesamiento de imágenes digitales.....	16
2.2.2.1. <i>Pre-procesamiento</i> .....	16
2.2.2.2. <i>Umbralización y binarización</i> .....	16
2.2.2.3. <i>Operaciones morfológicas</i> .....	17
2.2.2.3.1. <i>Operaciones de erosión-dilatación</i> .....	17

2.2.2.3.2.	<i>Operaciones compuestas de apertura-cierre.</i>	17
2.2.2.4.	<i>Proceso de filtrado de mediana.</i>	18
2.2.2.5.	<i>Segmentación de la imagen.</i>	18
2.2.3.	Técnicas de detección aplicadas al flujo vehicular.	18
2.2.3.1.	<i>Detección de vehículos basada en redes neuronales.</i>	18
2.2.3.2.	<i>Detección de vehículos basada en movimiento.</i>	18
2.2.3.3.	<i>Detección de vehículos basada en luminosidad.</i>	18
2.2.3.4.	<i>Detección de vehículos basada en contraste.</i>	19
2.3.	Programas para el procesamiento de imágenes.	19
2.3.1.	LabVIEW.	19
2.3.2.	MATLAB.	19
CAPÍTULO III. MATERIALES Y MÉTODOS		20
3.1.	Análisis de requerimientos.	21
3.1.1.	Programas.	21
3.1.2.	Equipos físicos.	21
3.1.2.1.	<i>Computador.</i>	21
3.1.2.2.	<i>Cámara y trípode.</i>	22
3.2.	Algoritmos seleccionados para la detección de vehículos	22
3.2.1.	Implementación del Algoritmo uno propuesto por Zhang (Zhang et al., 2013).	22
3.2.2.	Implementación del algoritmo dos propuesto por Bagyamani (Bagyamani, 2015).	29
3.2.3.	Implementación del algoritmo tres propuesto por Abinaya (Abinaya et al., 2013).	34
CAPÍTULO IV. DESARROLLO E IMPLEMENTACIÓN		42
4.1.	Descripción del algoritmo propuesto	43
4.2.	Diseño del algoritmo propuesto en MATLAB	43
4.2.1.	Pre-procesamiento.	44
4.2.2.	Segmentación.	46
4.2.3.	Detección y etiquetado de vehículos.	49
CAPÍTULO V. PRUEBAS Y RESULTADOS EXPERIMENTALES		51
5.1.	Descripción de la forma de evaluar las pruebas	52

5.2. Resultados obtenidos de la implementación del algoritmo uno propuesto por Zhang (Zhang et al., 2013).....	52
5.3. Resultados de la implementación del algoritmo dos propuesto por Bagyamani (Bagyamani, 2015).....	55
5.4. Resultados de la implementación del algoritmo tres propuesto por Abinaya (Abinaya et al., 2013).....	58
5.5. Resultados de la implementación del algoritmo MAT propuesto por el Autor .....	61
5.6. Comparación de los resultados de los cuatro algoritmos .....	64
CAPÍTULO VI. DISCUSIÓN .....	66
6.1. Análisis de costos e inversión .....	67
6.2. Limitaciones .....	70
CONCLUSIONES .....	78
RECOMENDACIONES.....	79
BIBLIOGRAFÍA .....	80
ANEXOS.....	84
ANEXO 1. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR ZHANG (Zhang et al., 2013).....	85
ANEXO 2. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR BAGYAMANI (Bagyamani, 2015).....	87
ANEXO 3. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR ABINAYA (Abinaya et al., 2013).....	89
ANEXO 4. CÓDIGO DEL ALGORITMO MAT .....	92

## ÍNDICE DE FIGURAS

Figura 1. Algoritmo propuesto por Zhang (Zhang et al., 2013). .....	23
Figura 2. Lectura de las imágenes y conversión a escala de grises en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	24
Figura 3. Filtrado de imágenes en el algoritmo propuesto por Zhang (Zhang et al., 2013). ..	25
Figura 4. Resta de imágenes en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	25
Figura 5. Imagen binarizada en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	26
Figura 6. Supresión de ruido en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	27
Figura 7. Operación de dilatación y llenado de agujeros en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	27
Figura 8. Operaciones de dilatación y apertura binaria en el algoritmo propuesto por Zhang (Zhang et al., 2013). .....	28
Figura 9. Etiquetado de objetos en el algoritmo propuesto por Zhang (Zhang et al., 2013). ..	28
Figura 10. Algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	29
Figura 11. Lectura de imágenes y conversión a escala de grises en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	30
Figura 12. Diferencia absoluta de las imágenes en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	31
Figura 13. Filtrado de imagen en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	31
Figura 14. Imagen binaria en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). ...	32
Figura 15. Aplicación de apertura y cierre binario en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	33
Figura 16. Objetos detectados en el algoritmo propuesto por Bagyamani (Bagyamani, 2015). .....	33
Figura 17. Algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	34
Figura 18. Lectura de imágenes y conversión a escala de grises en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	35
Figura 19. Resta de imágenes en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	36
Figura 20. Magnitud del gradiente en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	37
Figura 21. Funciones de apertura y cierre en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	37

Figura 22. Apertura-cierre por reconstrucción en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	38
Figura 23. Imagen binarizada en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	39
Figura 24. Segmentación de líneas divisorias en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	39
Figura 25. Resultado de la segmentación en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	40
Figura 26. Etiquetado de objetos en el algoritmo propuesto por Abinaya (Abinaya et al., 2013). .....	41
Figura 27. Algoritmo MAT. ....	43
Figura 28. Lectura de las imágenes y conversión a escala de grises en el algoritmo MAT. ....	44
Figura 29. Ajuste de intensidad en el algoritmo MAT. ....	45
Figura 30. Diferencia entre imágenes en el algoritmo MAT. ....	45
Figura 31. Magnitud del gradiente en el algoritmo MAT. ....	46
Figura 32. Aplicación de la función de apertura y cierre en el algoritmo MAT. ....	47
Figura 33. Reconstrucción por apertura-cierre en el algoritmo MAT. ....	47
Figura 34. Binarización de la imagen en el algoritmo MAT. ....	48
Figura 35. Aplicación de la operación de cierre en el algoritmo MAT. ....	48
Figura 36. Objetos detectados deseados y objetos por eliminar en el algoritmo MAT. ....	49
Figura 37. Resultado final con los vehículos detectados y contados en el algoritmo MAT. ....	50
Figura 38. Porcentajes del algoritmo propuesto por Zhang (Zhang et al., 2013) por escenario. ....	55
Figura 39. Porcentajes finales del algoritmo propuesto por Bagyamani (Bagyamani, 2015). ....	58
Figura 40. Porcentajes finales del algoritmo propuesto por Abinaya (Abinaya et al., 2013). ....	61
Figura 41. Porcentajes totales del algoritmo MAT. ....	64
Figura 42. Comparación de los porcentajes de los cuatro algoritmos. ....	65
Figura 43. Área de implementación. ....	67
Figura 44. Diagrama de cómo se obtuvieron las fotografías. ....	68
Figura 45. Diagrama de cómo se debería implementar las cámaras en la solución propuesta .....	68
Figura 46. Lectura y conversión a escala de grises de las imágenes adquiridas con la cámara del teléfono celular. ....	71
Figura 47. Diferencia entre las imágenes adquiridas con la cámara del teléfono celular. ....	72
Figura 48. Imagen binarizada de la prueba realizada con el teléfono celular. ....	72
Figura 49. Imagen calzada de la prueba realizada con el teléfono celular. ....	73
Figura 50. Resta de imágenes de la prueba realizada con el teléfono celular. ....	74

Figura 51. Cierre binario de la prueba realizada con el teléfono celular.....	74
Figura 52. Lectura y conversión a escala de grises de las imágenes adquiridas con la cámara y el trípode.....	75
Figura 53. Resta de las imágenes adquiridas con la cámara y el trípode. ....	76
Figura 54. Cierre binario de la prueba realizada con la cámara y el trípode. ....	76

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Comparación de los algoritmos descritos en el estado del arte. ....	14
<b>Tabla 2.</b> Detalle de características del computador utilizado.....	21
<b>Tabla 3.</b> Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado nublado.....	52
<b>Tabla 4.</b> Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado soleado. ....	53
<b>Tabla 5.</b> Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado de noche.....	54
<b>Tabla 6.</b> Resultados totales del algoritmo propuesto por Zhang (Zhang et al., 2013). ....	54
<b>Tabla 7.</b> Resultados experimentales del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado nublado.....	55
<b>Tabla 8.</b> Resultados del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado soleado. ....	56
<b>Tabla 9.</b> Resultados del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado de noche.....	57
<b>Tabla 10.</b> Resultados finales del algoritmo propuesto por Bagyamani (Bagyamani, 2015) ..	57
<b>Tabla 11.</b> Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado nublado. ....	58
<b>Tabla 12.</b> Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado soleado. ....	59
<b>Tabla 13.</b> Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado de noche.....	60
<b>Tabla 14.</b> Resultados finales del algoritmo propuesto por Abinaya (Abinaya et al., 2013)....	60
<b>Tabla 15.</b> Resultados del algoritmo MAT en estado nublado. ....	61
<b>Tabla 16.</b> Resultados del algoritmo MAT en estado soleado. ....	62
<b>Tabla 17.</b> Resultados del algoritmo MAT en estado de noche.....	63
<b>Tabla 18.</b> Resultados finales del algoritmo MAT.....	63
<b>Tabla 19.</b> Comparación de los resultados de los cuatro algoritmos.....	64
<b>Tabla 20.</b> Costos de equipos de la primera propuesta. ....	69
<b>Tabla 21.</b> Costos de equipos de la segunda propuesta.....	69
<b>Tabla 22.</b> Costos de equipos de la tercera propuesta. ....	69
<b>Tabla 23.</b> Costos de ingeniería.....	70
<b>Tabla 24.</b> Costos totales de las tres propuestas. ....	70
<b>Tabla 25.</b> Características del celular Huawei P9.....	70

## RESUMEN

El presente trabajo propone el desarrollo de un algoritmo que permita la identificación, reconocimiento y conteo de vehículos a partir de una imagen fija que puede provenir de una cámara fotográfica o de una cámara de un teléfono celular, mediante el procesamiento digital de las imágenes adquiridas en el casco urbano de la ciudad de Loja-Ecuador.

Para conseguir el objetivo principal se realizó el estado del arte para conocer los algoritmos que ya están implementados y de los cuales se puede partir para ajustarlo a la necesidad local tomando en cuenta que uno de los retos en la detección es la cantidad de variaciones del entorno.

El algoritmo será implementado en el programa MATLAB, al cual se le realizarán pruebas con las imágenes adquiridas dentro del casco urbano de la ciudad de Loja tomando en cuenta las distintas variaciones del clima y luz del día. Al final se describirán los resultados del algoritmo implementado y posibles mejoras a futuro.

**PALABRAS CLAVES:** procesamiento digital de imágenes, MATLAB, conteo de vehículos.

## **ABSTRACT**

The present work proposes the development of an algorithm that allows the identification, recognition and counting of vehicles from a fixed image that can come from a camera or from a cell phone camera, through the digital processing of the acquired images in the urban area of the city of Loja-Ecuador.

To achieve the main objective, the state of the art was performed in order to know the algorithms that are already implemented and from which it can be set to fit the local need taking into account that one of the challenges in the detection is the amount of variations of the environment.

The algorithm will be implemented in the MATLAB program, which will be tested with the images acquired within the urban area of the city of Loja taking into account the different variations of the climate and light of day. In the end, the results of the implemented algorithm and possible future improvements will be described.

**KEY WORDS:** digital image processing, MATLAB, vehicle counting.

## INTRODUCCIÓN

Hoy en día el tráfico vehicular es complicado en la mayoría de las áreas urbanas de casi todos los países en los que la congestión y el mantenimiento de las vías se han convertido en problemas cotidianos de difícil solución. Estos problemas son algunos de los causantes de efectos indeseados en la movilidad de los conductores y peatones.

Uno de los factores fundamentales a tomar en cuenta es la medición de la cantidad de vehículos presentes en las vías en determinados momentos para evitar la congestión vehicular. Entre los métodos usados para realizar estas mediciones se encuentran diversos sensores. Sin embargo, estos dispositivos físicos implican el montaje de aparatos en las propias vías terrestres, ocasionando perturbaciones en el tránsito, requiriendo gran cantidad de recursos y estando expuestos a la intemperie.

Por lo tanto, el presente trabajo describe el proceso de desarrollo de un algoritmo que permite el conteo de vehículos a partir de imágenes digitales, mediante fotografías tomadas desde un teléfono celular o una cámara fotográfica, a fin de conocer el flujo vehicular que existe en el casco urbano de la ciudad Loja-Ecuador.

El presente trabajo de titulación consta de seis capítulos, los mismos que se describen a continuación:

En el primer capítulo se realiza planteamiento del problema, la justificación donde se expone las razones que motivaron la realización del presente trabajo y los objetivos que se pretenden alcanzar.

En el segundo capítulo aborda el estado del arte, donde se presentan trabajos sobre algoritmos para el conteo de vehículos, publicados en revistas científicas, documentos técnicos y conferencias. Además, se resume los conceptos básicos necesarios para el desarrollo de dicho trabajo investigativo.

El tercer capítulo comprende los materiales y métodos, en los cuales, se describen los equipos físicos, programas utilizados y los algoritmos seleccionados que sirvieron de base para elaborar el algoritmo MAT.

Dentro del cuarto capítulo se muestra el desarrollo y funcionamiento del algoritmo propuesto.

En el quinto capítulo se incluyen los resultados obtenidos en las pruebas experimentales con su respectivo análisis.

En el sexto capítulo tenemos el análisis de los costos de implementación y las limitaciones que se presentaron en la realización del presente trabajo.

Finalmente se puntualizan las conclusiones y recomendaciones extraídas gracias al trabajo realizado.

**CAPÍTULO I.**  
**GENERALIDADES**

## **1.1. Planteamiento del problema**

El crecimiento de la densidad vehicular terrestre aumenta significativamente en los últimos años, provocando de forma rápida un alto grado de congestión en las vías de nuestra ciudad de Loja-Ecuador en horas pico, siendo este un problema cotidiano que afecta a conductores y peatones.

La inexistencia del monitoreo de vehículos que circulan por determinadas carreteras es otro de los inconvenientes que dificulta el tránsito. Los datos que se obtendrían de un sistema de monitoreo podrían tomarse en cuenta para evitar efectos adversos por la congestión vehicular.

Por esta razón, se desarrolla el presente trabajo de titulación, con la finalidad de proponer un algoritmo que permita realizar la detección y conteo de vehículos, para conocer el flujo vehicular en las calles del casco urbano de la ciudad de Loja-Ecuador

## **1.2. Justificación**

Se ha creído conveniente que, al existir población vehicular terrestre en continuo aumento en nuestra ciudad, existe la necesidad de crear un algoritmo que permita detectar el número de vehículos en determinadas vías a partir de imágenes digitales. Esto servirá como información útil para el control del tráfico vehicular y así evitar el congestionamiento en horas pico.

Los datos que se obtendrán serán importantes a futuro, ya que serán utilizados en estudios viales de los automotores que circulan en las vías principales del casco urbano de la ciudad de Loja, como, por ejemplo, por medio de la creación de una aplicación se puede informar a los conductores en tiempo real la situación del tráfico en el centro de la ciudad, con lo cual los conductores pueden tomar rutas alternativas a su lugar de destino; además ayudará a que la congestión vehicular disminuya.

## **1.3. Objetivos**

### **1.3.1. General.**

- Contar vehículos a partir de imágenes terrestres por medio del procesamiento de imágenes.

### **1.3.2. Específicos.**

- Revisar el estado del arte de los algoritmos existentes para la detección y conteo vehicular.
- Diseñar la arquitectura funcional de la solución.
- Implementar la solución.
- Evaluar el desempeño de la solución desde una perspectiva técnica y financiera.

**CAPÍTULO II.**  
**MARCO TEÓRICO**

## 2.1. Estado del arte

En este apartado, se presenta la investigación de trabajos relacionados con el tema.

En el documento de Ranga (Ranga, Kiran, Shekar, & Kumar, 2010) realizan la detección y clasificación de vehículos utilizando técnicas como la diferenciación de imagen, detección de bordes y procesos morfológicos binarios. El proceso de detección de vehículos lo dividen en dos fases: fase de pre-procesamiento y fase de detección de vehículos. En la primera fase se convierte a escala de grises y luego se binarizan dos imágenes, la imagen de referencia que es la imagen sin que haya vehículos y la imagen actual que sería la imagen con el tráfico vehicular existente. Seguidamente, se procede a realizar la diferencia de estas dos imágenes. En la fase dos se realiza la detección de bordes donde aplican la técnica de Sobel, dilatación binaria, llenado de agujeros y apertura morfológica. Con esto ya obtienen los vehículos de la imagen.

En el documento de Jain (Jain & Rani, 2010) para la detección de vehículos realiza dos etapas: el pretratamiento y detección de vehículos. En el pretratamiento se convierte a escala de grises la imagen de referencia y la imagen real. Posteriormente, se procede a filtrarlas utilizando un filtro de mediana para después convertirlas a imagen binaria para realizar la diferenciación. En la detección de vehículos se realiza la detección de bordes de Sobel, dilatación binaria y llenado de agujeros, para al final realizar la apertura binaria donde se obtiene los vehículos detectados.

En el trabajo de Parthasarathi (Parthasarathi, Surya, Akshay, Siva, & Vasudevan, 2015) para detectar el número de vehículos en la carretera, el marco actual y el marco de fondo se convierten a escala de grises y las imágenes se comparan y restan para obtener la presencia de objetos en la carretera. Operaciones morfológicas tales como dilatación y erosión se llevan a cabo para eliminar el ruido adicional en la imagen, y, esta imagen se mejora y se convierte en imagen binaria. Esta imagen es entonces filtrada usando un filtro gaussiano y para obtener solamente los vehículos en la carretera. El siguiente paso es contar el número de vehículos presentes en las carreteras. Para lograr estos conjuntos de píxeles de conexión se etiquetan que son los vehículos y se marcan con un cuadro delimitador.

En el trabajo de Sharma (Sharma, 2015) se utilizan algoritmos de reconocimiento morfológico para desarrollar un sistema automatizado. Se toman imágenes de satélite como imagen de entrada y se convierten en escala de grises para el preprocesamiento. Después estas imágenes se convierten en imágenes binarias. Luego el método de detección de bordes Canny (Canny, 1986) se ha realizado y pasado esta detección para el proceso de dilatación. La zona es seleccionada después de la filtración y la dilatación, donde cada uno de los

vehículos captados en la imagen procesada se establecen en forma de rectángulo. Se cuenta el número de vehículos por el análisis de Blob (Sharma, 2015), que es aquel que extrae las características de los objetos detectados, Además, los autores comparan dos algoritmos: el umbral de píxeles (Sharma, 2015) y el método de umbral de Otsu (Otsu, 1979). De acuerdo con los análisis de los resultados obtenidos por los autores, han determinado que el umbral de nivel de píxeles es mejor que el método de Otsu.

En el trabajo de Chen (Chen & Ke, 2014) primero convierten la imagen a escala de grises, luego aplican un filtro de mediana para eliminar el ruido, y a continuación, realizan la extracción del fondo que la realizan a partir de una secuencia de fotografías de la carretera de manera estática. Seguidamente, se realiza la diferencia entre el fondo y la imagen con vehículos y por último hacen el uso de la función de MATLAB *imclose* para obtener los vehículos en la imagen.

Los siguientes artículos toman como entrada videos de tráfico en carreteras, para realizar el procesamiento y, para obtener la detección y conteo de vehículos.

En el trabajo de Hashmi (Hashmi, Keskar, Reddy, & Kaushik, 2015) los autores efectúan un algoritmo para determinar el conteo y la clasificación de los vehículos en las intersecciones. En este algoritmo, primero el video se convierte en marcos desde donde se realiza la conversión a escala gris de cada trama. A continuación, las operaciones morfológicas y suavizado se realizan para eliminar cualquier ruido extraviado. La sombra del vehículo en movimiento se elimina mediante la técnica de eliminación de sombras propuesta. Luego esta secuencia de tramas se aplica al método de extracción de primer plano propuesto en el que se puede obtener el vehículo en movimiento con precisión. Entonces se aplica la detección de borde Canny, que entrega el esquema del vehículo. Se detectan contornos y se realiza una aproximación de polígono, entonces el vehículo en movimiento está limitado por una caja delimitadora. Los parámetros del cuadro delimitador también se obtienen como la posición inicial de largo, altura, anchura del cuadro delimitador. El seguimiento de los vehículos se realiza mediante este cuadro delimitador. Las regiones de interés "Region of Interest (ROI)" en este caso pueden ser establecidos por el usuario, es decir, un ROI dinámico. A continuación, utilizando estos parámetros, la clasificación de los vehículos se realiza correctamente comprobando el cuadrante en el que se encuentra el vehículo y su área correspondiente. El conteo de los vehículos que pasan a través de ROIs se hace comprobando si el centro de la caja de contorno ha pasado a través de ese ROI particular o no.

En el trabajo de Zhang (Zhang, Ge, Zhou, Jiang, & Wang, 2013) se utiliza el método de mezcla gaussiana para eliminar el impacto del vehículo y configura la imagen de fondo. El fondo en

tiempo real se segmenta completamente y con precisión, se actualiza el fondo de referencia dinámicamente haciendo coincidir con el fondo actual de la secuencia de imágenes y combinando las ventajas del método de diferencia de fondo con el método de diferencia entre tramas. Para asegurar la robustez de la detección de vehículos de seguimiento se aplica un filtro con plantilla de ventana tres por tres píxeles, para eliminar el punto de ruido aislado de la imagen de contorno del vehículo. Para obtener imágenes de los conjuntos de corrosión y conjuntos de expansión se utiliza el elemento estructural de plantilla y un filtro morfológico de erosión y dilatación. Para reducir el rango de búsqueda objetivo, mejorar la velocidad de cálculo y la precisión del algoritmo, los autores utilizan el modelo de filtrado de Kalman (Kalman, 1960) para realizar el seguimiento del vehículo de movimiento rápido.

En el trabajo de Padmavathi (Padmavathi, Naveen, & Kumari, 2016) se realiza un algoritmo de detección de vehículos basado en faros. En este artículo los faros se identifican en base a puntos brillantes elípticos mediante el análisis de blob, que extrae las características de los objetos detectados. Se propone un algoritmo de emparejamiento que segrega los faros de cuatro ruedas de los de dos ruedas. El tamaño del faro varía en función de la distancia de las cámaras. Se sugiere un factor de escala adecuado basado en la proyección en perspectiva. Las luces cercanas entre sí se identifican como una región en forma de gota única y se cuentan con precisión. Sin embargo, según los autores, las luces de niebla y los faros con suficiente espacio entre ellos se identifican como cuatro ruedas separadas que no son efectivas en el cálculo del número de vehículos durante la noche.

En el trabajo de Bhaskar (Bhaskar & Yong, 2014) los autores proponen desarrollar un algoritmo único para el reconocimiento y el seguimiento de datos de vehículos mediante el método de mezcla gaussiana y los métodos de detección de blob, el cual extrae las características de los objetos detectados. Primero, diferencian el primer plano del fondo en los marcos, aprendiendo el fondo. Aquí, el detector de primer plano detecta el objeto y se realiza un cálculo binario para definir regiones rectangulares alrededor de cada objeto detectado. Para detectar correctamente el objeto en movimiento y para eliminar el ruido se han aplicado algunas operaciones morfológicas. A continuación, el conteo final se realiza mediante el seguimiento de los objetos detectados y sus regiones.

En el trabajo de Bhaskar (Bhaskar, Yong, & Jung, 2015) se propone desarrollar un algoritmo único para el reconocimiento y el seguimiento de datos del vehículo utilizando el método de flujo óptico paralelo basado en el algoritmo Lucas-Kanade. Aquí, la detección de movimiento se determina mediante diferenciación temporal y la coincidencia de plantillas se realiza sólo en las ubicaciones guiadas por la etapa de detección de movimiento para proporcionar un método robusto de seguimiento de objetivos. El detector de flujo óptico en primer plano detecta

el objeto y se realiza un cálculo binario para definir regiones rectangulares alrededor de cada objeto detectado. Para detectar correctamente el objeto en movimiento y para eliminar el ruido se han aplicado algunas operaciones morfológicas, específicamente, la erosión. El conteo final se realiza mediante el seguimiento de los objetos detectados y sus regiones en una secuencia en tiempo real.

En el sistema de detección de flujo de tráfico propuesto por Pan (Pan, Guo, & Men, 2010); primero, ingresa el video de tráfico real en la computadora. A continuación, el método de detección de borde se utiliza para extraer el borde del marco completo. Basándose en la información de los bordes, usan el método de fondo medio, que mediante la comparación de las tramas del video de entrada se obtiene el fondo de la imagen. Con la resta de fondo obtienen el ROI. En segundo lugar, han calculado varios carriles de la carretera utilizando un algoritmo de división de carril. En tercer lugar, la ventana de conteo se redimensiona automáticamente con referencia al carril dividido. Finalmente, el flujo de tráfico se puede detectar contando el número de vehículos que pasan por las ventanas.

En el trabajo de Panda (Panda, Naik, & Patel, 2015) la investigación se divide en cuatro partes. La primera parte es procesar la señal de vídeo y la adquisición de imágenes de una cámara fija donde inicialmente el sistema captura la imagen de un camino vacío cuando no hay vehículo presente. La segunda parte es seleccionar el área objetivo donde los vehículos podrían estar presentes usando la técnica de recorte de imágenes, el propósito es identificar la región de la carretera donde los vehículos están presentes y excluir la información de fondo innecesaria. La tercera parte es la detección de objetos que se realiza mediante la mejora de las características de la imagen. Para realizar la detección del objeto primero se extrae el fotograma de la secuencia de vídeo en tiempo real. El siguiente paso es convertir ambas imágenes: la de referencia y la de tiempo real en escala de grises, y luego se determina la diferencia absoluta de estas dos imágenes. Dado que las dimensiones de la carretera se fijan, la imagen de diferencia sólo resalta la presencia de vehículos en el área objetivo deseada. Con el fin de mejorar la visibilidad de los vehículos, la imagen de diferencia se convierte en una imagen binaria basada en un valor de umbral. Con el fin de determinar los vehículos en el área deseada, la multiplicación de la imagen recortada con la imagen de diferencia se realiza para filtrar la información innecesaria y resaltar la presencia de vehículos en el área deseada. Finalmente, la última parte es el conteo de densidad, donde se cuenta el número de vehículos.

En el trabajo de Vibha (Vibha, Hegde, Shenoy, Venugopal, & Patnaik, 2008) el flujo del algoritmo es el siguiente: en la primera etapa se calcula la diferencia entre tramas, procediendo así a comparar estas diferencias y en la segunda etapa se eliminan los píxeles

que tienen los mismos valores en la diferencia de fotogramas, con el propósito de obtener la imagen en primer plano. La tercera fase es la etapa de post-procesamiento ejecutada sobre la imagen obtenida en la segunda etapa, donde utiliza un filtro de mediana. Después de aplicar el filtro de mediana, la imagen resultante se convierte en una imagen binaria. La técnica de apertura morfológica se aplica en esta imagen binaria. La fase final es el conteo del objeto, donde se realiza la fase de sintonización de objetos donde la salida es una imagen binaria de los objetos detectados denominados mask1. La imagen binaria rastreada mask1 forma la imagen de entrada para el conteo. Esta imagen se escanea de arriba hacia abajo para detectar la presencia de un objeto. Se mantienen dos variables: "count" que registra el número de vehículos y "countreg", que contiene la información del objeto registrado. Este concepto se aplica para toda la imagen y el conteo final de objetos está presente en el conteo de variables.

En el trabajo de Bagyamani (Bagyamani, 2015) para detectar y contar el vehículo se toman dos imágenes: la primera se mantiene como imagen de referencia (carretera sin vehículo) y la segunda imagen es con presencia de vehículos. Primero, las dos imágenes RGB (red, green, blue) se convierten en imagen de escala de grises y luego estas imágenes se restan usando la operación de diferencia absoluta. La imagen resultante se filtra usando un filtro mediano para reducir el ruido. La imagen filtrada se convierte entonces en imagen binaria utilizando el umbral de escala de grises de Otsu. Las operaciones morfológicas se utilizan para segmentar la forma de los vehículos. El algoritmo propuesto detecta automáticamente la forma de los vehículos estableciendo algunos parámetros iniciales como una serie de operaciones de apertura y cierre de estructura en forma de disco para implementar el perfil morfológico. Finalmente se calcula el centroide de las formas segmentadas y luego se utiliza la longitud del centroide para contar el número de vehículos en la imagen.

En el trabajo de Abinaya (Abinaya, Ravichandran, & Santhi, 2013) los videos se convierten en marcos y extraen los marcos para procesos adicionales como detección de vehículos, clasificación y conteo. Posteriormente, se substraen el fondo, donde la segmentación de la división de cuencas ("watershed") controlada por marcadores se utiliza para la extracción de primer plano de imágenes. La segmentación de imágenes es el objeto de interés. El propósito principal de la segmentación de la imagen es extraer las características del objeto y también para reducir los datos que varían del modelo de entrada. Para la detección de vehículos se encuentran los bordes de la imagen actual y la imagen de fondo, después se substraen el fondo de la imagen actual para encontrar el vehículo en movimiento; luego realizan el rastreo y clasificación del vehículo y por último se escanean las imágenes binarias para contar. El proceso aquí para el conteo hace uso de dos variables: "count" y "count reg".

El sistema propuesto por Juneja (Juneja & Grover, 2013) ayuda a la detección eficiente de objetos en movimiento en el área de vigilancia. Este modelo utiliza la técnica de sustracción de trama sucesiva para detectar objetos en movimiento, aplica operaciones morfológicas e implementa algoritmo de líneas divisorias sobre los resultados obtenidos. Los autores implementan un algoritmo con segmentación normal y otro con segmentación de líneas divisorias (“watershed”), donde comparan ambos resultados y concluyen que la segmentación de cuenca obtiene mejores resultados que la normal.

En el trabajo de Liu (Liu, Zhao, Yuan, Luo, & Liu, 2011) se presenta un nuevo enfoque de extracción de borde para detectar un vehículo en movimiento en una secuencia de video. La nueva extracción de borde mejora la tolerancia del proceso de detección hacia las variaciones de las condiciones de iluminación y disminuye la influencia de las sombras del vehículo. El método de predicción de gradiente de vecindad (neighborhood gradient prediction NGP), como lo denominan los autores, también permite detectar el vehículo basándose en la variedad de la información de borde de vehículo que es simple y eficaz.

En el trabajo de Moutakki (Moutakki, Ayaou, Afdel, & Amghar, 2014) se propone el siguiente algoritmo: en la primera fase, se lee el clip de vídeo y se convierte en marcos. La segunda fase del sistema es aislar la región (llamada primer plano) del resto del marco (llamado fondo). La detección de vehículos es el siguiente paso del sistema que es el proceso de identificar el vehículo que se extrae. Para identificar cada vehículo se debe considerar cada vehículo dibujando rectángulos. En general se utilizan tres etapas en esta fase: el primer paso es la segmentación de las regiones objeto de interés. En este paso las regiones que pueden contener objetos desconocidos tienen que ser detectados. El segundo paso se centra en la extracción de características adecuadas y luego en la extracción de objetos. El propósito principal de la extracción de características es reducir los datos mediante la medición de ciertas características que distinguen los patrones de entrada. El siguiente paso es la clasificación donde se asigna una etiqueta a un objeto basada en la información proporcionada por sus descriptores, entonces, cada objeto segmentado está limitado en un rectángulo y el centro y el área de tal rectángulo son considerados como características de ese vehículo. Basado en esas características, cada vehículo es contado y rastreado.

En el trabajo de Raghtate (Raghtate & Tiwari, 2014) los fotogramas se seleccionan aleatoriamente de la entrada de video y se utilizan para la estimación de fondo. Desde el marco de vídeo de entrada actualizado, el marco de fondo se resta para obtener la imagen del objeto en primer plano. A continuación, se realiza el postprocesamiento, donde se hace uso del filtro mediano y operación de cierre morfológico, así se obtiene la imagen clara del

objeto en primer plano. A continuación, se realiza la sintonización de objeto para la identificación del objeto. Luego realizan el conteo mediante una línea virtual.

El algoritmo propuesto por Aslani (Aslani & Mahdavi-Nasab, 2013) se emplea varios métodos para detectar, filtrar, segmentar y rastrear objetos. Utilizan el algoritmo de Horn-Schunck (Horn & Schunck, 1981), como el método más adecuado de estimación de flujo óptico, para detectar objetos en movimiento por los cambios de intensidad de los marcos. También se hace uso del filtro de mediana por su buen rendimiento. El cierre morfológico extrae características significativas de las formas de los objetos a partir de imágenes binarias y, a continuación, hacen uso del análisis de blob que encuentra las propiedades de los objetos, ya que su gran ventaja es el bajo coste computacional. El sistema es capaz de eliminar objetos de movimiento no deseados que no son vehículos en la secuencia de imágenes, usando restricciones en áreas de blob.

De acuerdo al estado del arte y los resultados obtenidos por los autores que se comparan en la Tabla 1, se realiza la selección de tres algoritmos con mayor precisión en la detección de vehículos los cuales son los propuestos por los siguientes autores: Zhang (Zhang et al., 2013), Bagyamani (Bagyamani, 2015) y Abinaya (Abinaya et al., 2013).

**Tabla 1.** Comparación de los algoritmos descritos en el estado del arte.

<b>Método / Técnica</b>	<b>Autor</b>	<b>Porcentaje de Eficiencia</b>	<b>Observaciones importantes</b>
Diferenciación de imagen, detección de bordes y procesos morfológicos binarios.	(Ranga et al., 2010)	-	Poco complejo
Procesos morfológicos	(Jain & Rani, 2010)	-	Poco complejo
Procesos morfológicos, conectividad de píxeles.	(Parthasarathi et al., 2015)	-	No realizan pruebas reales.
Procesos morfológicos, detección de bordes, método umbral de píxeles.	(Sharma, 2015)	-	Usan imágenes satelitales.
Diferenciación de imagen, Operaciones morfológicas.	(Chen & Ke, 2014)	-	Excelentes resultados en buenas condiciones climáticas.
Operaciones morfológicas detección de bordes, ROI.	(Hashmi et al., 2015)	93%	Fiable y altamente preciso en todas las condiciones posibles.
Diferenciación de imagen, Análisis de blob.	(Zhang et al., 2013)	96%	Probado en diferentes escenarios ambientales y de tráfico.

Detección basada en faros, análisis de blob	(Padmavathi et al., 2016)	93,3%	-
Método de mezcla gaussiana, detección blob.	(Bhaskar & Yong, 2014)	91%	-
Método de flujo óptico, procesos morfológicos.	(Bhaskar et al., 2015)	98%	-
Diferenciación de imagen, detección de borde, operaciones morfológicas, ROI.	(Pan et al., 2010)	95%	Pruebas realizadas en los diferentes estados de iluminación.
Diferenciación de imagen. Procesos binarios.	(Panda et al., 2015)	-	-
Diferenciación de imagen, filtro mediano, apertura morfológica.	(Vibha et al., 2008)	94%	Los vehículos cuya intensidad es similar a la superficie no se pierden.
Diferenciación de imagen, método de umbral de Otsu, Operaciones morfológicas	(Bagyamani, 2015)	97,9%	Algoritmo poco complejo.
Diferenciación de imagen, segmentación.	(Abinaya et al., 2013)	80%	La segmentación tiene un buen desempeño.
Operaciones morfológicas, segmentación watershed	(Juneja & Grover, 2013)	91%	-
Método de predicción de gradiente de vecindad (NGP)	(Liu et al., 2011)	95%	Robusto frente a las variaciones de iluminación y presencia de sombras, pero inestable en la congestión.
Diferenciación de imagen, segmentación y extracción de características.	(Moutakki et al., 2014)	-	-
Diferenciación de imagen, filtrado y operaciones morfológicas.	(Raghtate & Tiwari, 2014)	-	Alta precisión en la detección.
Algoritmo Horn-Shunk, operaciones morfológicas, análisis de blob.	(Aslani & Mahdavi-Nasab, 2013)	-	-

**Fuente:** (Ranga et al., 2010), (Jain & Rani, 2010), (Parthasarathi et al., 2015), (Sharma, 2015), (Chen & Ke, 2014), (Hashmi et al., 2015), (Zhang et al., 2013), (Padmavathi et al., 2016), (Bhaskar & Yong, 2014), (Bhaskar et al., 2015), (Pan et al., 2010), (Panda et al., 2015), (Vibha et al., 2008), (Bagyamani, 2015), (Abinaya et al., 2013), (Juneja & Grover, 2013), (Liu et al., 2011), (Moutakki et al., 2014), (Raghtate & Tiwari, 2014), (Aslani & Mahdavi-Nasab, 2013).

**Elaboración:** Autor.

## **2.2. Fundamentación teórica**

### **2.2.1. Adquisición de la imagen.**

La adquisición de la imagen está a cargo de algún transductor o conjunto de transductores (sensores) que captan la luz emitida por una escena y forman una representación del objeto, o formato de imagen válido. Este proceso se realiza mediante una cámara fotográfica, de video o convencional (Quiceno Gutiérrez, 2012).

#### **2.2.1.1. Imagen digital.**

Las imágenes se generan en píxeles donde cada uno tiene la información del color que está representada en dos dimensiones (x,y) que son coordenadas espaciales y los valores que toman estas variables son proporcionales a la intensidad. La imagen digital es un documento electrónico que contiene la representación digital de una imagen. Cada elemento que representa esta imagen es llamado píxel (Ruíz, 2014).

### **2.2.2. Procesamiento de imágenes digitales.**

El procesamiento digital de imágenes es una rama de la electrónica que permite mejorar la calidad de la imagen para la percepción del ojo humano, y permite extraer información de la misma para posteriores aplicaciones. Para procesar una imagen es necesario que la misma al ser capturada como una señal física (luz) sea transformada a una señal eléctrica, es decir, como una imagen digital (Chávez Fuentes, 2015).

El procesamiento digital de imágenes está compuesto por una serie de etapas en las cuales se busca mejorar la información contenida en una imagen para la interpretación y el tratamiento autónomo de una máquina. Este proceso permite dar un formato modificado a la toma adquirida (Prieto Valdés, 2010).

#### **2.2.2.1. Pre-procesamiento.**

La etapa de pre-procesamiento permite identificar el tipo de imágenes adquiridas y son clasificadas de acuerdo a lo que se necesita, en este caso, clasificar el tipo de vehículo para posteriormente mejorar las imágenes que fueron adquiridas con la finalidad de aumentar las posibilidades de éxito en etapas posteriores y destacar características para una aplicación específica. En el pre-procesamiento se mejora la imagen que al momento de la adquisición tuvo alteraciones en la calidad producidas por factores externos (Hincapié Cifuentes, 2014).

#### **2.2.2.2. Umbralización y binarización.**

La umbralización se acompaña de un proceso de binarización, lo cual permite utilizar posteriormente operadores morfológicos para la individualización en los automóviles

(Hincapié Cifuentes, 2014). La binarización se realiza únicamente sobre la imagen de frente generada. Como la escena es cambiante en términos de iluminación y color de los objetos (vehículos) con respecto al fondo, no es factible utilizar un valor fijo de umbral (Prieto Valdés, 2010).

### **2.2.2.3. Operaciones morfológicas.**

Permiten extraer los componentes útiles de una imagen, entre ellos bordes, esqueletos, operaciones de llenado y operadores lógicos, filtros, entre otros (Quiceno Gutiérrez, 2012). Las operaciones morfológicas en imágenes se definen como procedimientos en los cuales cada nuevo píxel de la imagen resultante es obtenido de una operación no lineal entre un conjunto de puntos de la imagen original y un conjunto de puntos conocido como elemento estructurante. Este elemento estructurante recorre toda la imagen para obtener todos los puntos de la nueva imagen, sin cambiar el tamaño de la imagen de salida (Prieto Valdés, 2010). Estas operaciones son realizadas sobre imágenes binarias basadas en forma, que toman como entrada una imagen binaria regresando como resultado una imagen también binaria (Hincapié Cifuentes, 2014).

#### **2.2.2.3.1. Operaciones de erosión-dilatación.**

La dilatación adiciona píxeles en las fronteras de los objetos, mientras que la erosión los remueve; todo gracias al elemento estructurante (Chávez Fuentes, 2015). La dilatación es un proceso binario en el cual se toma el elemento estructurante u origen y se lo superpone en la imagen, se aplica para rellenar entrantes de objetos y unir pequeñas separaciones en las imágenes. La erosión se aplica para reducir salientes de objetos, eliminar partes de tamaño menor que el elemento estructurante o eliminar conexiones insignificantes entre objetos mayores (Prieto Valdés, 2010).

#### **2.2.2.3.2. Operaciones compuestas de apertura-cierre.**

La apertura es una operación en la que primero la imagen es sometida a una erosión y luego la dilatación, la imagen resultante no será la original debido a que estas operaciones no son inversas (Chávez Fuentes, 2015). Suaviza contornos de los objetos, elimina protuberancias y rompe conexiones débiles (Hincapié Cifuentes, 2014).

El cierre es la realización de una dilatación seguida de una erosión, rellena detalles conectando objetos que están próximos entre sí, suaviza contornos, rellena de vacíos en el contorno y elimina pequeños huecos (Hincapié Cifuentes, 2014).

#### **2.2.2.4. Proceso de filtrado de mediana.**

El filtrado aplicado al procesamiento de imágenes en este documento es el de mediana. Los filtros morfológicos, pueden detectar bordes en las imágenes, filtrar objetos de tamaños menores a uno determinado, suavizar fondos de texturizados, detectar fallos en patrones de textura (Prieto Valdés, 2010). El filtrado de mediana es un método de filtrado espacial que calcula la mediana del conjunto de píxeles vecinos al píxel que se está procesando.

#### **2.2.2.5. Segmentación de la imagen.**

El proceso de segmentación consiste en extraer de una imagen un objeto u objetos que son importantes o se requieren para el proceso de representación y descripción. Se escribe generalmente como extraer los elementos de una escena (también conocida como fondo). Este proceso también es conocido como binarización de una imagen (Quiceno Gutiérrez, 2012). La segmentación parte una imagen de entrada en sus partes constituyentes, fundamentalmente se busca el reconocimiento de caracteres. Al finalizar este proceso se tienen los datos que constituyen bien el contorno de una región o todos los puntos de una región determinada. En cada caso es necesario convertir los datos a una forma adecuada para el procesamiento y toma de decisiones respecto a los datos, como un contorno o como una región completa (Hincapié Cifuentes, 2014).

### **2.2.3. Técnicas de detección aplicadas al flujo vehicular.**

#### **2.2.3.1. Detección de vehículos basada en redes neuronales.**

Esta técnica se basa en la recolección de patrones de vehículos. El correcto funcionamiento de este sistema depende del número de patrones ingresados en el mismo. En este sistema se encuentra la desventaja de que el número de patrones que siguen los vehículos es muy amplio y deben ser todos introducidos en un procedimiento de aprendizaje. Esto implica una gran capacidad de memoria. Esta técnica no muestra resultados favorables (GÓMEZ, 2009).

#### **2.2.3.2. Detección de vehículos basada en movimiento.**

Esta técnica se basa en el estudio de una secuencia de imágenes separadas entre sí por muy poco tiempo, del orden de milisegundos, con el fin de detectar si ha habido cambios entre unas y otras capturas (GÓMEZ, 2009).

#### **2.2.3.3. Detección de vehículos basada en luminosidad.**

Este procedimiento se basa en el estudio de los componentes que superan un cierto umbral de luminosidad en una imagen. Una vez discrepados estos elementos, se estudian sus características principales, sobre todo su morfología para diferenciar entre vehículos y no vehículos (GÓMEZ, 2009).

#### **2.2.3.4. Detección de vehículos basada en contraste.**

El método de la detección de vehículos mediante contraste se basa en la técnica de detección de bordes. Esta forma de detectar bordes está constituida por el cálculo de la entropía de los elementos que aparecen en una imagen. A grandes rasgos, la entropía es el contraste que existe entre dos o más elementos de una imagen. La detección de los elementos que se encuentran en la calzada se realiza fácilmente mediante esta técnica ya que existe un alto contraste entre el pavimento de la calzada con los vehículos que circulan por ella (GÓMEZ, 2009).

### **2.3. Programas para el procesamiento de imágenes.**

#### **2.3.1. LabVIEW.**

LabVIEW es una plataforma de programación gráfica desarrollada por la empresa National Instruments que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Este programa emplea una programación gráfica o también denominada lenguaje G, donde la G simboliza que es lenguaje Gráfico. Es un lenguaje fácil de aprender e intuitivo, ya que se basa en símbolos. LabVIEW posee extensas librerías de funciones y subrutinas. Adicionalmente de las funciones básicas de cualquier lenguaje de programación, LabView cuenta con librerías específicas para: adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos (Chávez Fuentes, 2015).

LabVIEW se puede conectar con todo tipo de dispositivos como: tarjetas insertables, controladores de movimiento y controladores lógicos programables (PLCs), cabe mencionar que National Instruments cuenta con sus propias tarjetas de Adquisición denominadas DAQ, LabVIEW responde de manera más óptima en cuanto a tiempo de respuesta con el uso de estas tarjetas (Chávez Fuentes, 2015).

#### **2.3.2. MATLAB.**

Matlab (MATrix LABoratory) es un lenguaje de alto nivel desarrollado por la empresa Mathworks que permite analizar datos, desarrollar algoritmos, modelos, aplicaciones y realizar cualquier tipo de cálculo con vectores y matrices. Adicionalmente permite mostrar gráficos en dos y tres dimensiones. Es uno de los lenguajes más empleados y está disponible para las plataformas de Unix, Windows y Apple (Chávez Fuentes, 2015).

**CAPÍTULO III.**  
**MATERIALES Y MÉTODOS**

### 3.1. Análisis de requerimientos

En este apartado se toma en cuenta los requerimientos necesarios para llevar a cabo el diseño del sistema de detección de flujo vehicular en el casco urbano de la ciudad de Loja-Ecuador. Mediante las herramientas detalladas se logrará dar forma al sistema en general, desde la adquisición de imágenes, así como el algoritmo de detección y conteo.

#### 3.1.1. Programas.

Para el desarrollo del algoritmo que permitirá la detección de vehículos se necesita un programa que sea eficiente en el procesamiento de imágenes. MATLAB, es un lenguaje de programación que cuenta con una librería que proporciona un conjunto completo de algoritmos estándar de referencia y apps de flujo de trabajo para el procesamiento, el análisis y la visualización de imágenes, así como para el desarrollo de algoritmos ("Image Processing Toolbox - MATLAB - MATLAB," 2017).

Para facilitar el tratamiento de imágenes MATLAB ha desarrollado una caja de herramientas denominado Procesamiento de Imágenes, que permite utilizar funciones y herramientas probadas y robustas, entre las más importantes: análisis de imágenes y estadística, diseño de filtros y recuperación de imágenes, mejora de imágenes, operaciones morfológicas, definición de mapas de colores, modificación gráfica, operaciones geométricas y transformación de imágenes, soporta casi todos los formatos y tamaños de imagen (Chávez Fuentes, 2015).

#### 3.1.2. Equipos físicos.

##### 3.1.2.1. Computador.

Es importante contar con un ordenador en el cual instalar el software escogido, este debe contar con un buen desempeño y para eso hay que tener en cuenta sus características (Tabla 2).

**Tabla 2.** Detalle de características del computador utilizado.

CARACTERÍSTICAS DEL ORDENADOR	
<b>Modelo</b>	DELL XPS- L502X
<b>Procesador</b>	Intel Core i7-2670QM-2.20 GHz
<b>Memoria RAM</b>	8 GB
<b>Sistema operativo</b>	Windows 10
<b>Tarjeta de Video</b>	Nvidia GT 525M (1GB)

Fuente: Autor.

Elaboración: Autor.

### **3.1.2.2. Cámara y trípode.**

Estos equipos permitirán realizar la toma de las imágenes en cada una de las calles de la ciudad que serán procesadas por el algoritmo.

Las características del equipo que se utilizó para realizar las fotografías fueron: Cámara Nikon coolpix s6800 de 16.0 Megapíxeles, wide 12xzoom Full HD, con un lente de 4.5-54.0mm 1:3.3-6.3, con una tarjeta de memoria sandisck extreme pro de 32 GB. Esta cámara y lente se utilizó tanto para el día como para la noche.

El trípode ayudará a estabilizar la cámara para obtener la toma más clara y que al realizarse dos capturas una en estado transitado y otra intransitado se obtenga la máxima precisión en posición de la imagen. Se usó un trípode Induro AKB= trípode de 1,2 Kg de color negro que llega a una altura máxima de 1,60m.

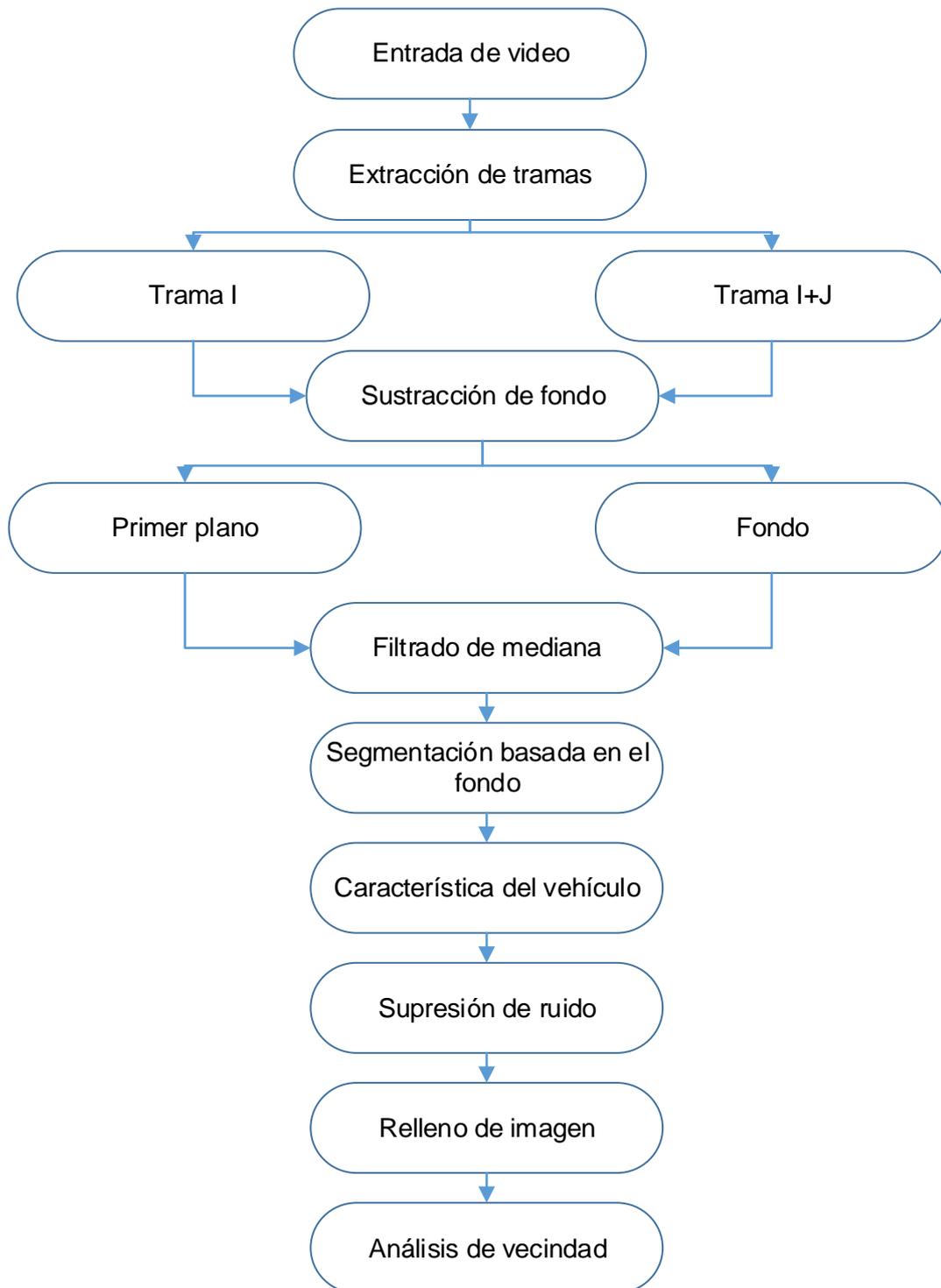
Se tomaron un total de 120 fotografías que se realizaron en tres ambientes diferentes que fueron nublado, soleado y noche.

## **3.2. Algoritmos seleccionados para la detección de vehículos**

Mediante la investigación realizada se escoge tres algoritmos obtenidos de diferentes autores Zhang (Zhang et al., 2013), Bagyamani (Bagyamani, 2015) y Abinaya (Abinaya et al., 2013). A través de estos algoritmos se establece los códigos que sirven como referencia para diseñar un código más eficiente que representará el trabajo realizado. Cabe recalcar que no se encontró los códigos base dentro de los documentos y por esta razón se procedió a implementar cada uno de estos en base a los algoritmos.

### **3.2.1. Implementación del Algoritmo uno propuesto por Zhang (Zhang et al., 2013).**

El algoritmo propuesto por (Zhang et al., 2013) (ver Figura 1) tiene como entrada de datos video a diferencia de imagen. Es por esto que al momento de establecer el código fuente se ignora ciertos pasos, ya que el video de entrada es transformado a imagen en el proceso de sustracción de fondo, el primer plano representa la toma cuando existe presencia de vehículos y el fondo es una toma en vacío. Cuando existe una entrada de video estos dos pasos el fondo y primer plano se obtienen del video debido a la captura automática de fondo, a diferencia de la fotografía que necesita de dos tomas.



**Figura 1.** Algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Tomado de (Zhang et al., 2013)

**Elaboración:** Autor.

Las dos imágenes se leen con el comando *imread*, en el cual se especifica el nombre de los archivos y su formato. Luego se las convierte a escala de grises haciendo uso del comando *rgb2gray* (ver Figura 2).



**Figura 2.** Lectura de las imágenes y conversión a escala de grises en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

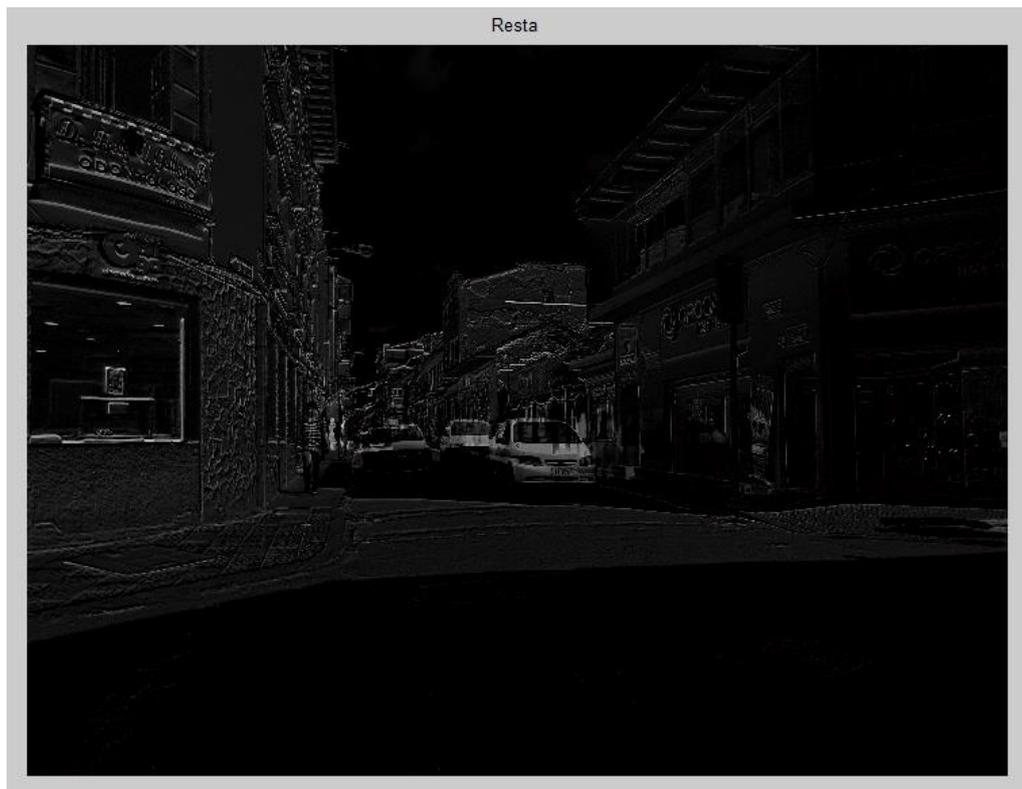
Con las imágenes en escala de grises se procede a realizar la fase de filtrado en el cual se aplica el filtro de mediana que se obtiene con el comando **medfilt2** (ver Figura3), con esto se elimina el ruido presente en las imágenes para proceder a realizar la diferencia entre las dos imágenes, donde se hace uso de la función **imsubtract** el resultado se observa en la Figura 4.



**Figura 3.** Filtrado de imágenes en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

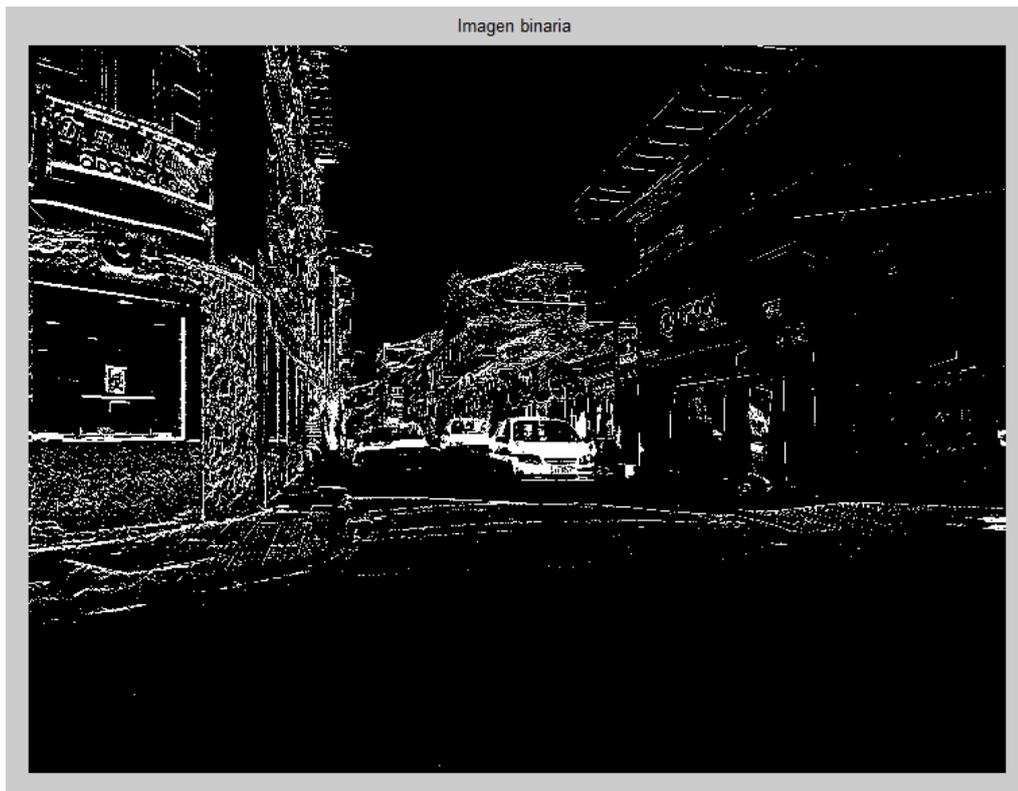


**Figura 4.** Resta de imágenes en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

Para el proceso de segmentación se transforma la imagen en escala de grises a imagen binaria, primero se calcula el nivel de umbral óptimo con el comando **graythresh**, este valor se introduce en la función **im2bw** que como resultado se obtendrá la imagen binarizada (ver Figura 5).

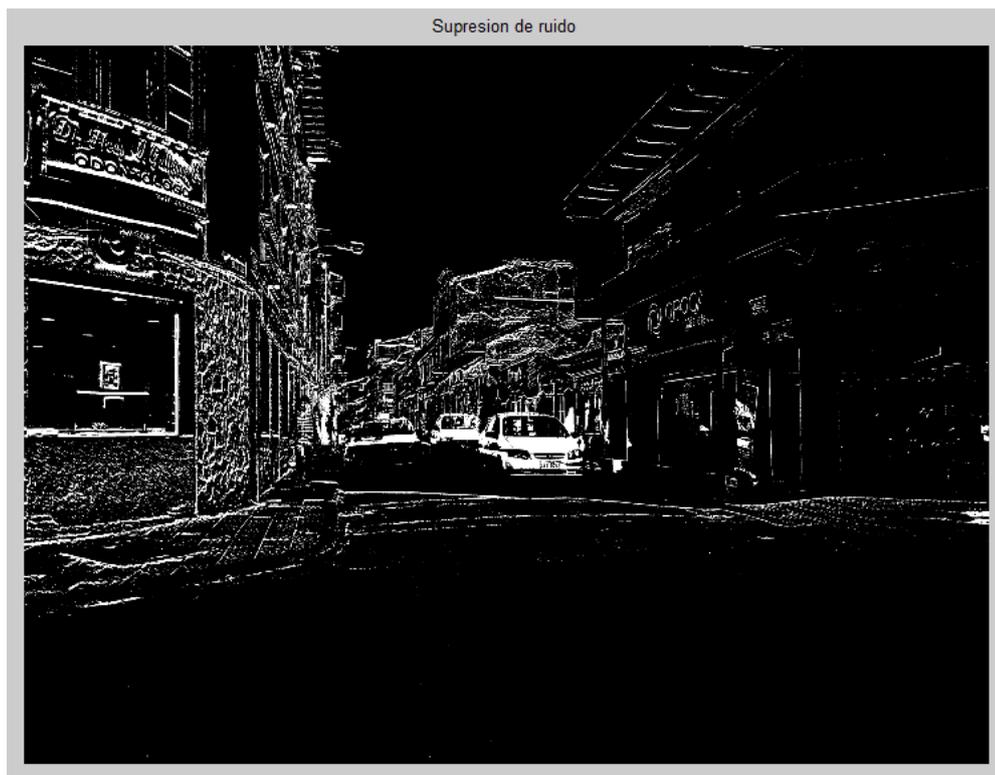


**Figura 5.** Imagen binarizada en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

Con la imagen binarizada se procede a realizar la supresión de ruido que se obtiene al aplicar la función *filter2*, para obtener la imagen filtrada se debe crear una estructura de tres por tres píxeles (ver Figura 6).



**Figura 6.** Supresión de ruido en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Colaboración:** Autor.

A continuación, se procede a realizar operaciones morfológicas: primero se realiza el proceso de erosión para que el ruido pueda ser eliminado y la imagen de corrosión de los vehículos en movimiento se pueda obtener, usando el elemento estructurante de diamante. Esto se realiza con la función *imerode* (ver Figura 7). Después se procede con el llenado de agujeros para que los orificios dentro de los vehículos sean llenados, se hace uso de la función *imfill* (ver Figura 7).



**Figura 7.** Operación de dilatación y llenado de agujeros en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Colaboración:** Autor.

Luego se aplica el proceso de dilatación para buscar regiones conectadas y marcarlas, se aplica el elemento estructural de diamante usando la función *imdilate* (ver Figura8), seguido a esto se emplea la apertura binaria con la estructura de disco para eliminar pequeños píxeles que se encuentren aislados, esto se realiza con la función *imopen* (ver Figura8).



**Figura 8.** Operaciones de dilatación y apertura binaria en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Colaboración:** Autor.

Por último, se procede al etiquetado de los objetos detectados con la función *bwlabel*, luego para extraer las propiedades de los objetos se usa la función *regionprops* que mediante la propiedad *BoundingBox* envuelve en un rectángulo todos los objetos detectados, que es el resultado de todo el algoritmo (ver Figura 9).



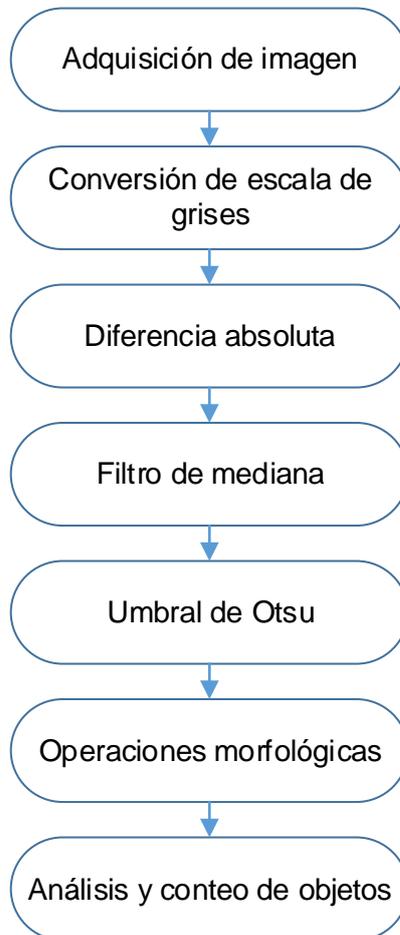
**Figura 9.** Etiquetado de objetos en el algoritmo propuesto por Zhang (Zhang et al., 2013).

**Fuente:** Autor.

**Colaboración:** Autor.

### 3.2.2. Implementación del algoritmo dos propuesto por Bagyamani (Bagyamani, 2015).

El algoritmo propuesto por Bagyamani (Bagyamani, 2015) (ver figura 10) empieza con la adquisición de dos imágenes una con vehículos y otra sin vehículos, luego se procede hacer la lectura de imágenes con la función *imread*, para luego convertirlas a escala de grises con la función *rgb2gray* (ver figura 11).



**Figura 10.** Algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Tomado de (Bagyamani, 2015).

**Colaboración:** Autor.

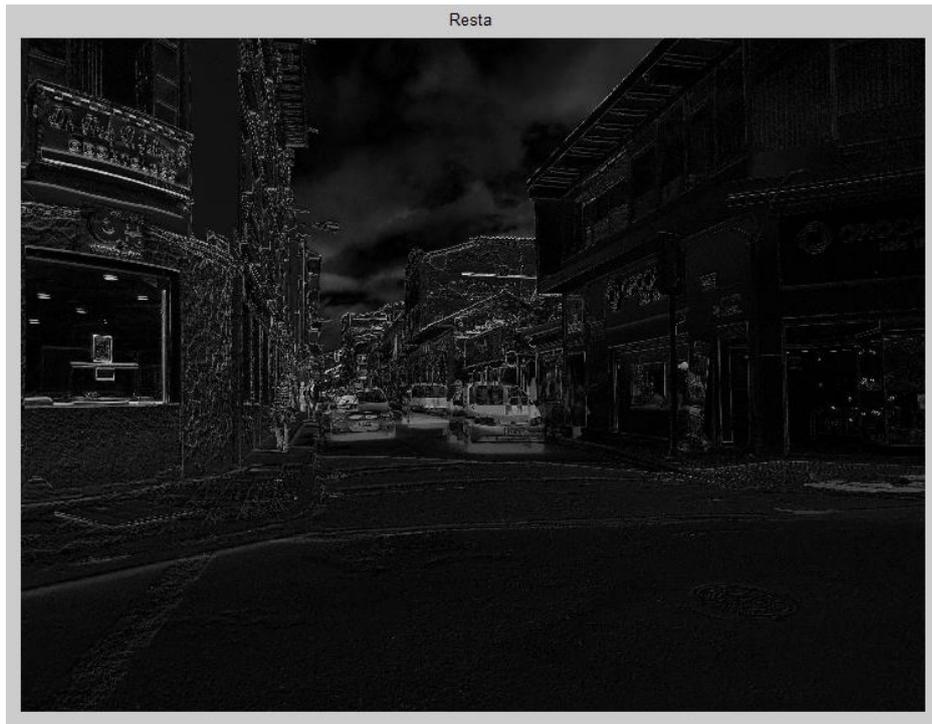


**Figura 11.** Lectura de imágenes y conversión a escala de grises en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Colaboración:** Autor.

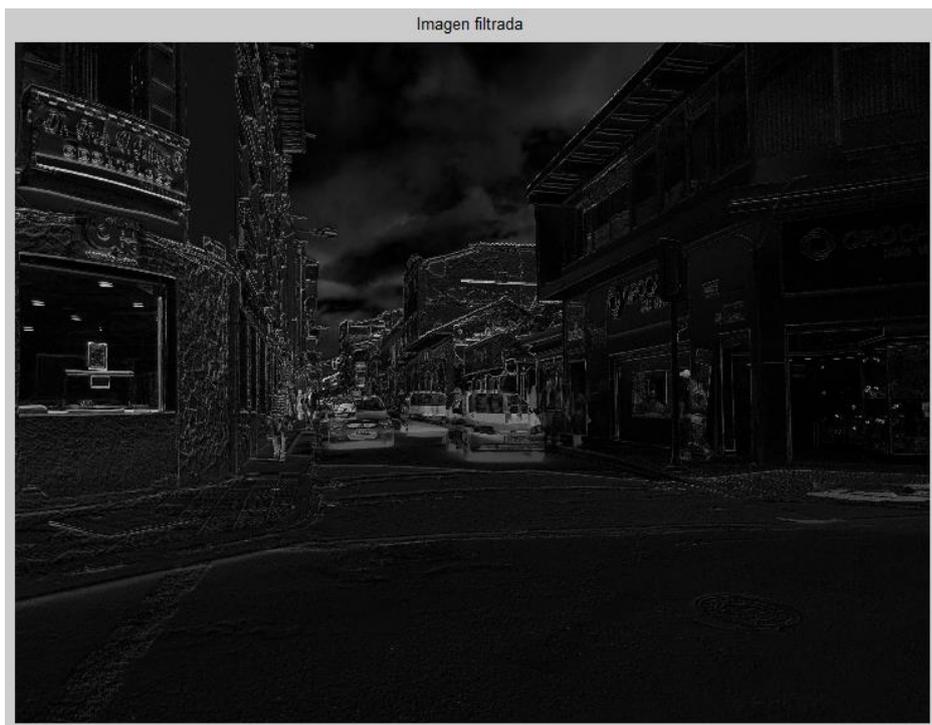
Se procede a realizar la diferencia absoluta que es una resta píxel a píxel de las imágenes con la función ***imabsdiff*** (ver Figura 12), una vez obtenida la imagen se procede a filtrar para mejorar calidad de la imagen usando un filtro de mediana porque ayuda mejorar el contraste de la imagen y las características de brillo, reducir su contenido de ruido o agudizar sus detalles, se aplica la función ***medfilt2*** (ver Figura 13).



**Figura 12.** Diferencia absoluta de las imágenes en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Elaboración:** Autor.

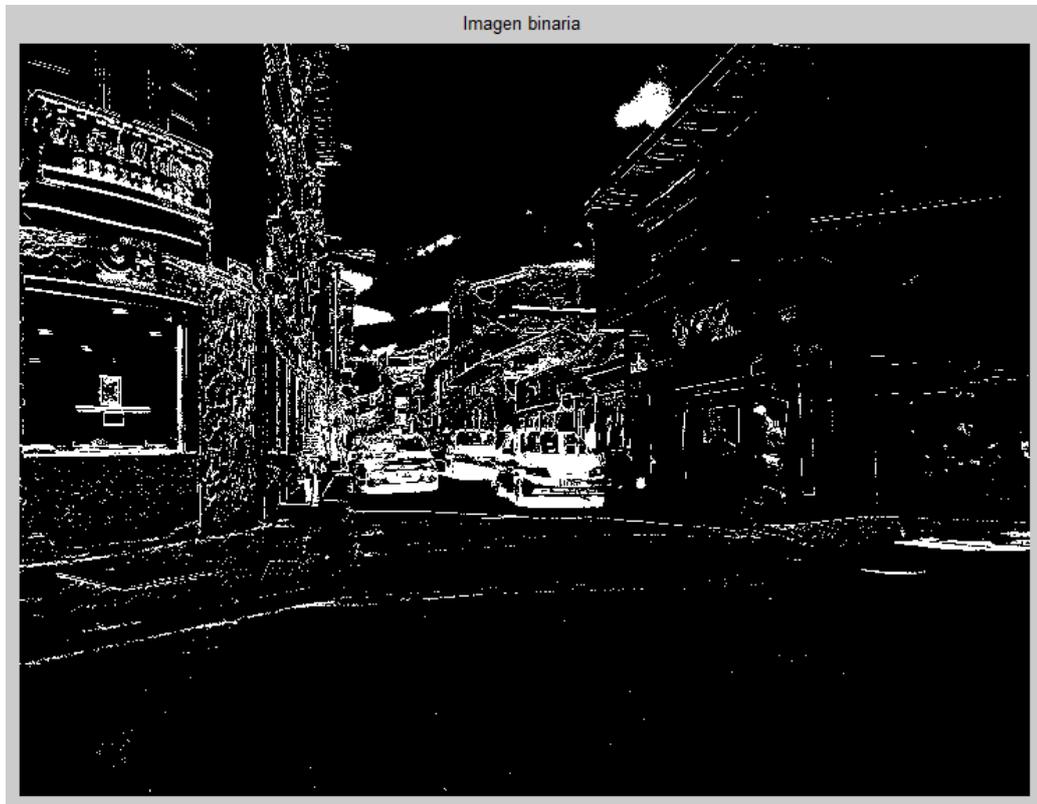


**Figura 13.** Filtrado de imagen en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Elaboración:** Autor.

Con la imagen filtrada se procede a convertir a una imagen binaria con el objeto de segmentarla, para ello primero se encuentra el valor de umbral óptimo automáticamente basándose en la forma del histograma, con la función **graythresh**, seguido a esto se aplica la función **im2bw** para binarizar la imagen (ver Figura 14) con el uso del valor encontrado anteriormente.

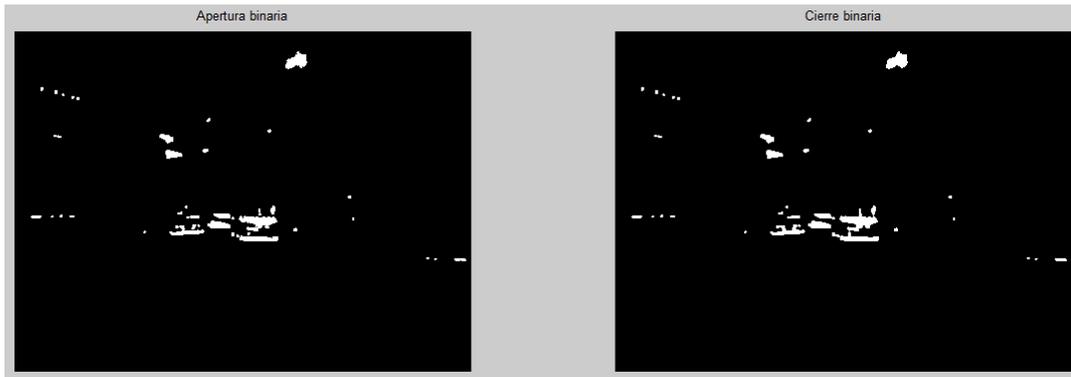


**Figura 14.** Imagen binaria en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Elaboración:** Autor.

Obtenida la imagen binaria se continua con la aplicación de las operaciones morfológicas: primero se aplica la apertura binaria con la función **imopen** con la estructura de disco para desaparecer los elementos que son más pequeños que el elemento estructural (ver Figura 15). Luego se aplica el cierre binario con la función **imclose** usando también un elemento estructural de disco para rellenar los huecos que se encuentran dentro de los vehículos (ver Figura 15).



**Figura 15.** Aplicación de apertura y cierre binario en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Elaboración:** Autor.

Para finalizar se procede al etiquetado de objetos que se realiza con la función ***bwlabel*** luego con la función ***regionprops*** se obtiene las propiedades de los objetos y los encierra en un rectángulo (ver Figura 16).



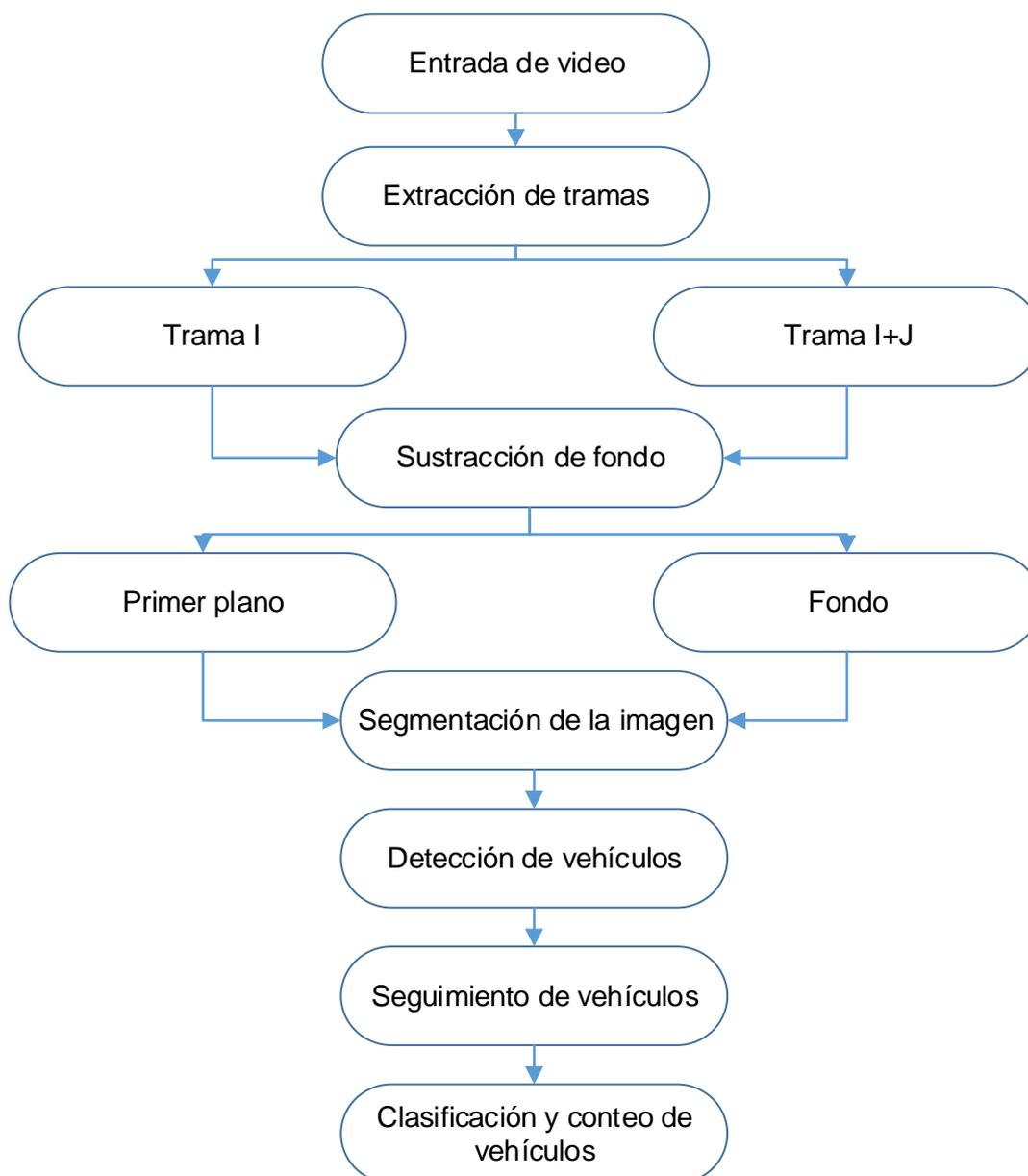
**Figura 16.** Objetos detectados en el algoritmo propuesto por Bagyamani (Bagyamani, 2015).

**Fuente:** Autor.

**Elaboración:** Autor.

### 3.2.3. Implementación del algoritmo tres propuesto por Abinaya (Abinaya et al., 2013).

El algoritmo propuesto por Abinaya (Abinaya et al., 2013) (ver Figura 17) tiene como entrada de datos video al igual que el algoritmo uno y como se explicó anteriormente al momento de establecer el código fuente se ignora ciertos pasos, ya que en este Trabajo de Titulación se toma como entrada imágenes. Una vez adquiridas las imágenes se procede a leer las imágenes con la función *imread*, para luego convertirlas a escala de grises con la función *rgb2gray* (ver Figura 18).



**Figura 17.** Algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Tomado de (Abinaya et al., 2013)

**Elaboración:** Autor.

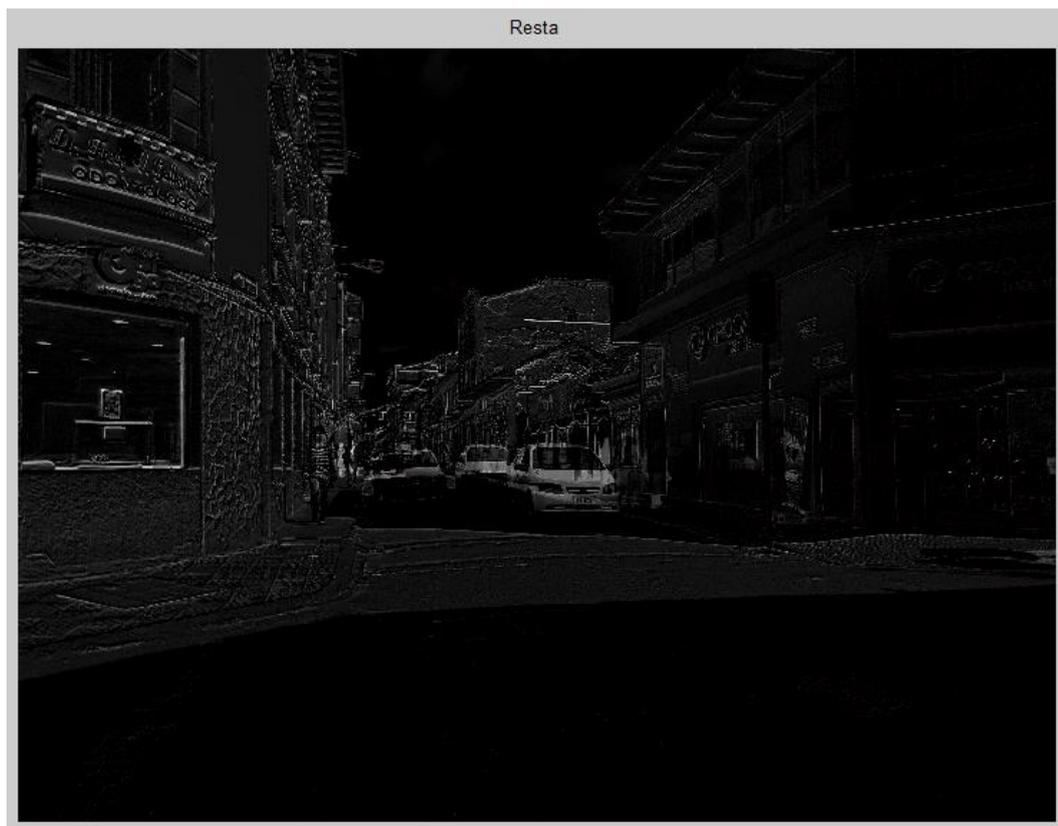


**Figura 18.** Lectura de imágenes y conversión a escala de grises en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

Luego, se procede a realizar la resta de la imagen con vehículos de la imagen sin vehículos con la función ***imsubtract***, que realiza una resta píxel a píxel entre las imágenes cuyo resultado se muestra en la Figura 19.

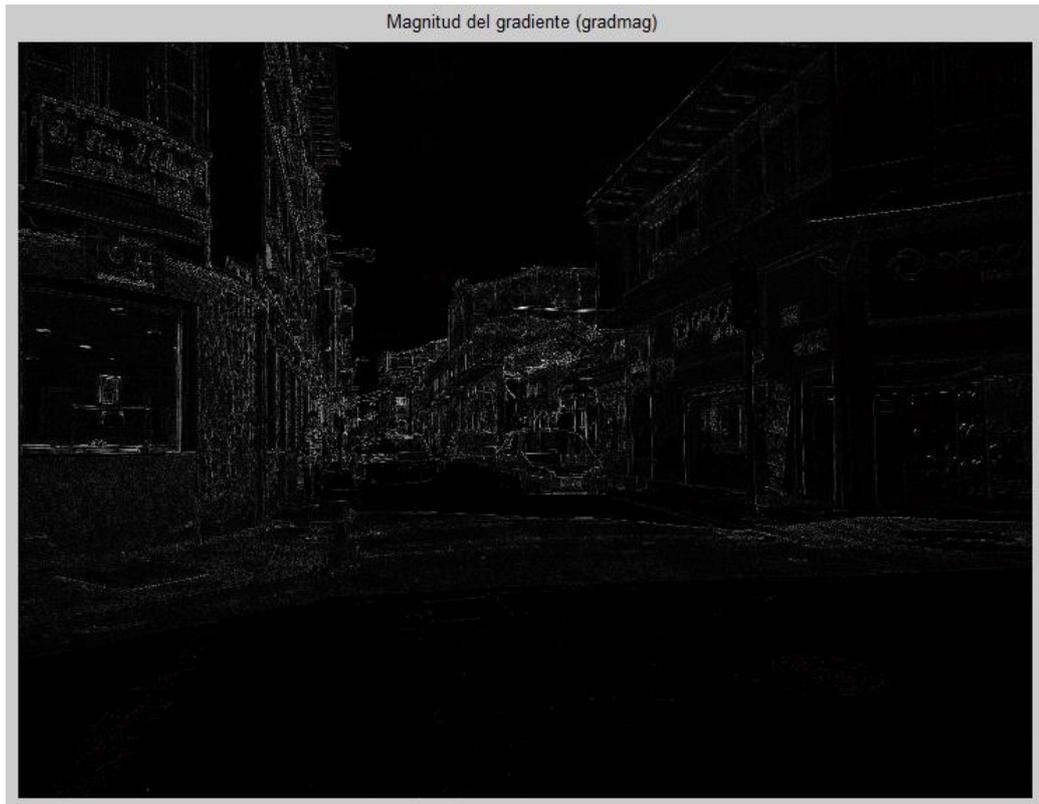


**Figura 19.** Resta de imágenes en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

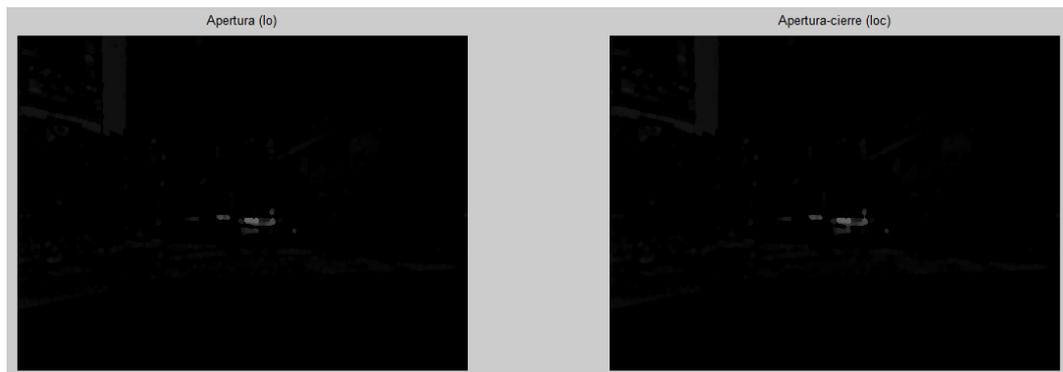
Posteriormente, se procede con la segmentación de líneas divisorias “watersheed”: primero se encuentra la magnitud del gradiente con la función **gradmag** (ver Figura 20), luego se procede a marcar los objetos en primer plano se realizan los procesos morfológicos de apertura con la función **imopen** para eliminar los elementos pequeños que se encuentren aislados (ver Figura 21) y de cierre con **imclose** para llenar los huecos que se encuentran dentro de los vehículos (ver Figura 21). A continuación, se realiza una reconstrucción de la imagen con la función **imreconstruct** que como resultado nos presenta la eliminación de manchas pequeñas sin que afecte a los objetos detectados en el primer plano (ver Figura 22).



**Figura 20.** Magnitud del gradiente en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

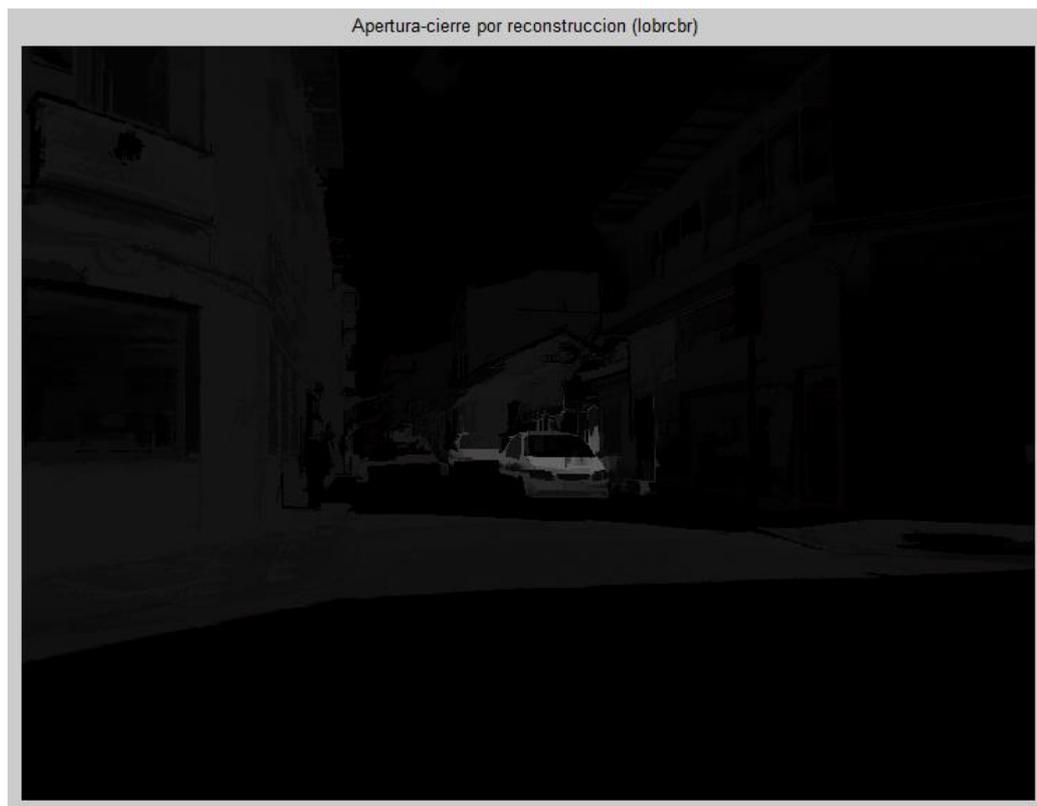
**Elaboración:** Autor.



**Figura 21.** Funciones de apertura y cierre en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

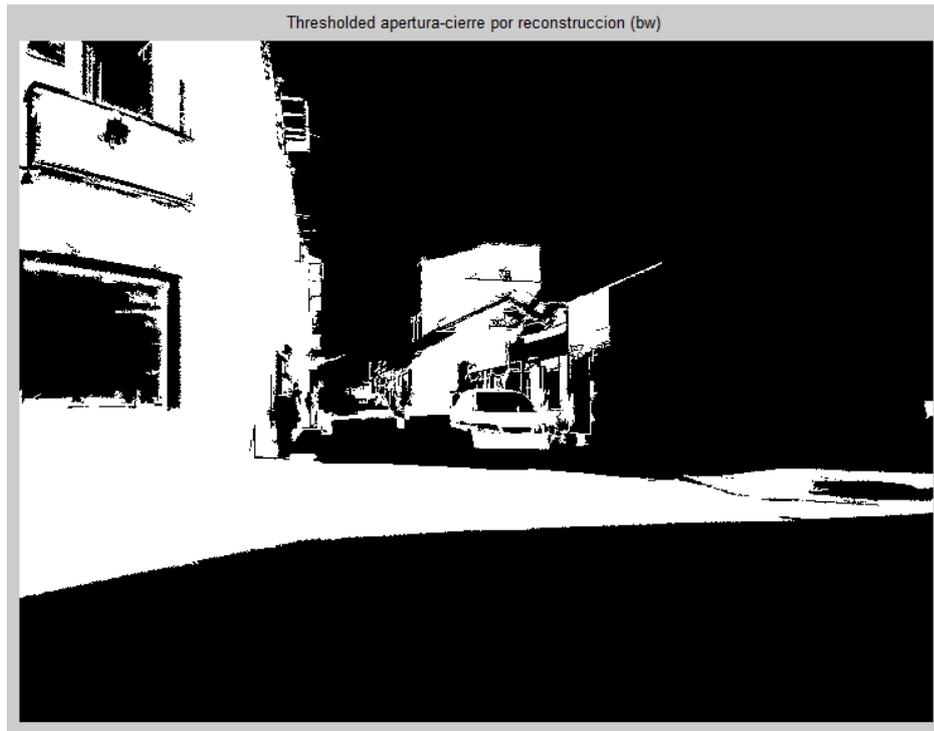


**Figura 22.** Apertura-cierre por reconstrucción en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

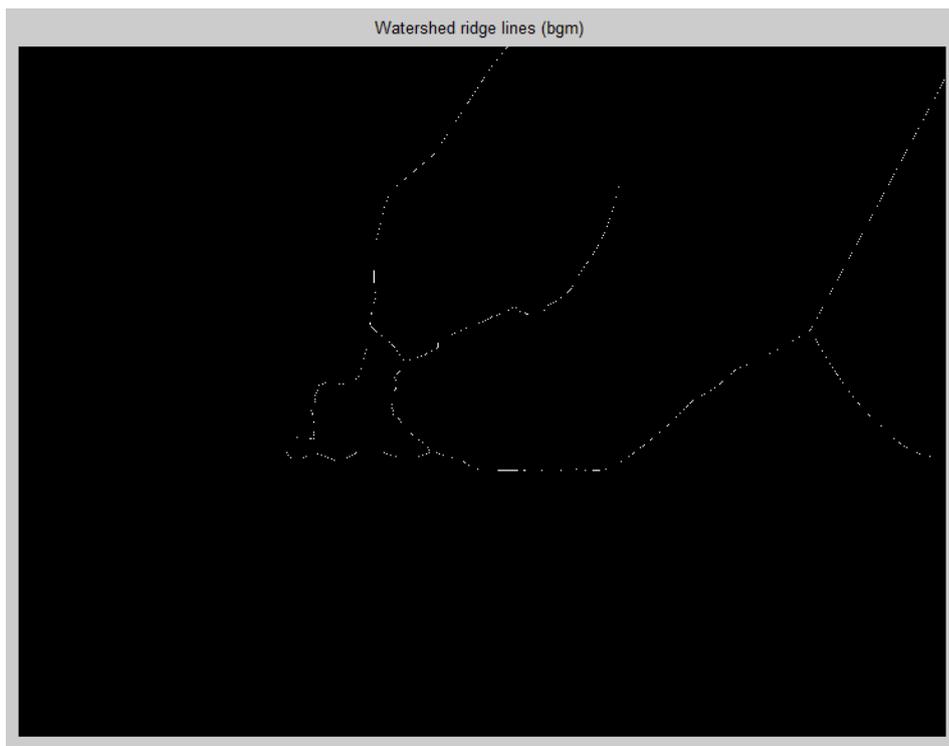
Finalmente, se procede a binarizar la imagen con la función *im2bw* (ver Figura 23) para poder realizar la segmentación de líneas divisorias con la función *watershed* con el objeto de marcar las líneas divisorias entre los objetos detectados (ver Figura 24).



**Figura 23.** Imagen binarizada en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

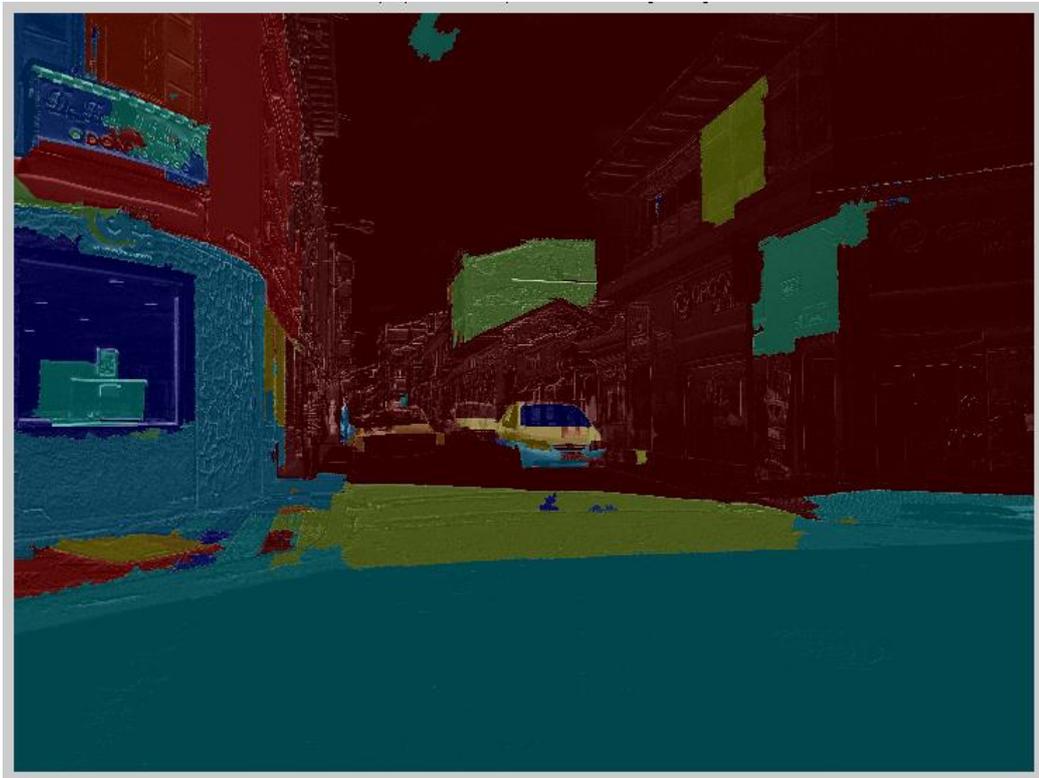


**Figura 24.** Segmentación de líneas divisorias en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

**Elaboración:** Autor.

Una vez hecha la segmentación se presenta el resultado donde se usa una técnica de visualización que puede convertir a las imágenes en color verdadero para fines de visualización mediante el uso de *label2rgb* (ver figura 25) y por último se muestra los objetos detectados haciendo uso de las funciones *bwlabel* y *regionprops*, la primera ayuda a etiquetar los objetos detectados y la segunda encierra los objetos detectados en un rectángulo (ver Figura 26).



**Figura 25.** Resultado de la segmentación en el algoritmo propuesto por Abinaya (Abinaya et al., 2013).

**Fuente:** Autor.

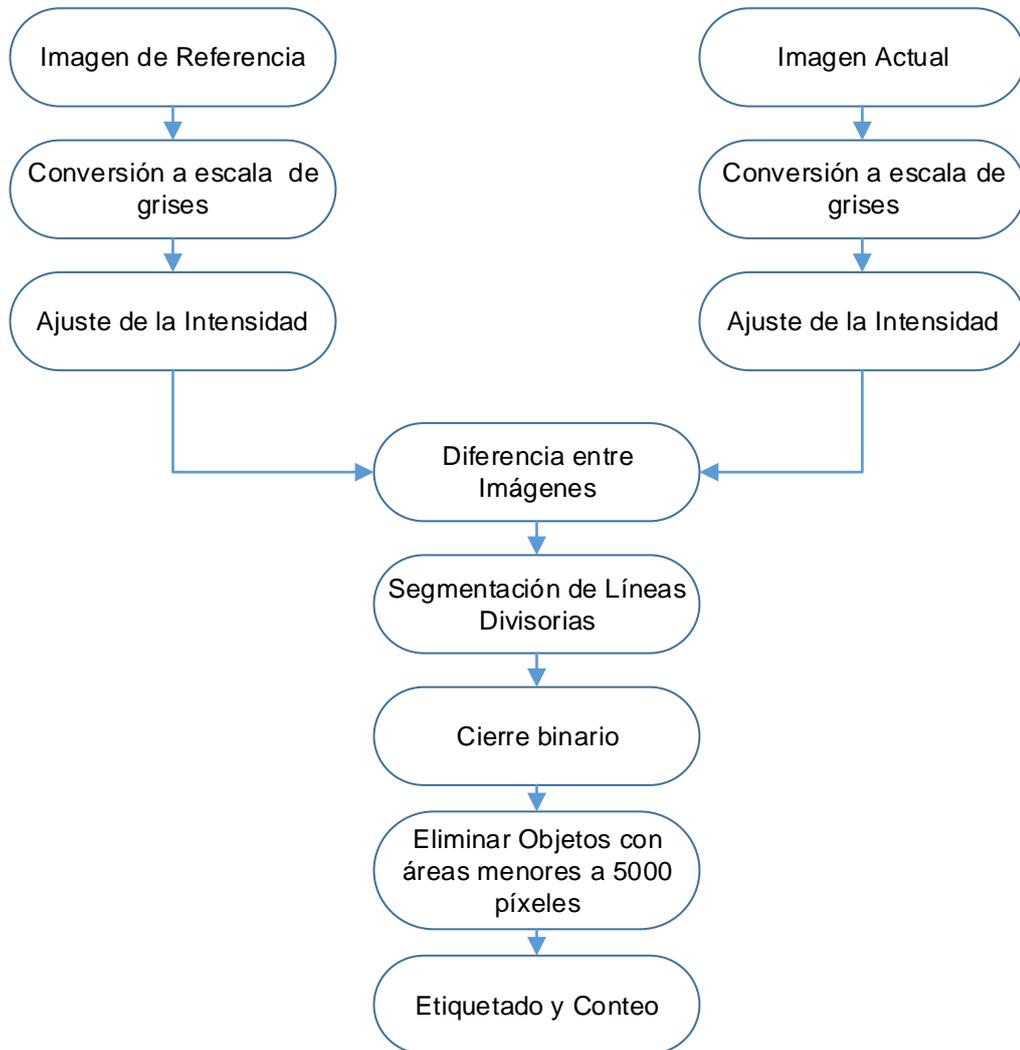
**Elaboración:** Autor.



**CAPÍTULO IV.**  
**DESARROLLO E IMPLEMENTACIÓN**

#### 4.1. Descripción del algoritmo propuesto

En este apartado se presenta el cuarto algoritmo el cual se lo ha denominado MAT (ver Figura 27) que para la realización del mismo se ha basado en los tres algoritmos descritos anteriormente, con el objetivo de alcanzar un mejor resultado en la detección de vehículos.



**Figura 27.** Algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

#### 4.2. Diseño del algoritmo propuesto en MATLAB

El algoritmo MAT se diferencia de los demás algoritmos en los siguientes aspectos:

- Toma como entrada dos imágenes una que es el primer plano que es la imagen con vehículos y la otra el fondo que es la imagen sin vehículos.
- En el preprocesamiento se ha agregado la etapa de ajuste de intensidad.

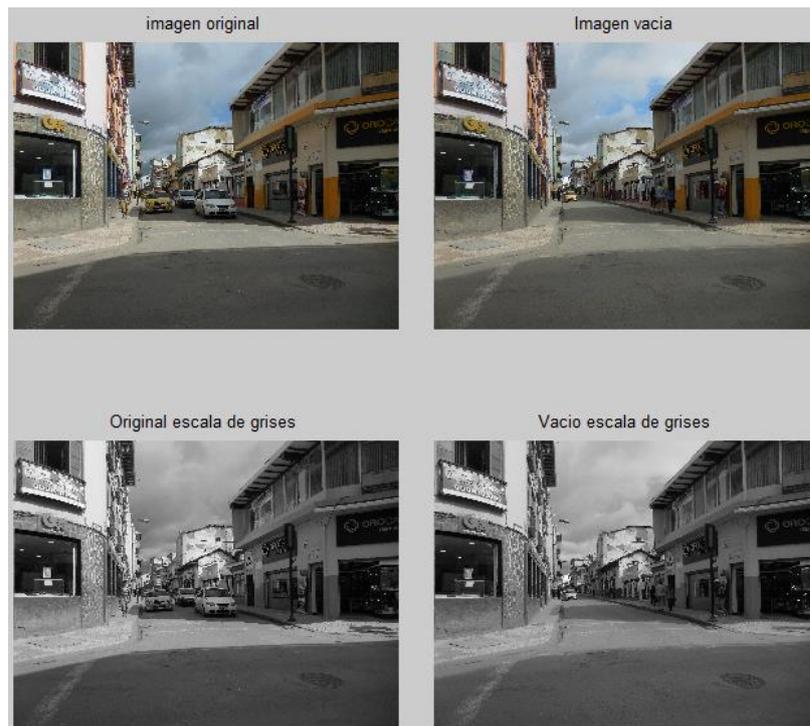
- Se usa la segmentación “watershed” con la diferencia que en el proceso de reconstrucción de la imagen se omite la etapa de dilatación.
- Se agrega el proceso de eliminación de objetos con áreas menores al rango establecido que es de 5000 píxeles.

Con los cambios realizados en el algoritmo MAT se obtendrá las siguientes ventajas:

- Mayor sensibilidad, ya que se obtendrá un mejor desempeño en el reconocimiento de los objetos deseados que son los vehículos.
- Mayor precisión, ya que se tendrá una mejor exactitud frente a los errores en la detección, es decir, que tendrá mayor capacidad al descartar aquellos objetos que no son vehículos.
- Robustez, ya que presentará un buen desempeño frente a cambios climáticos y de luz.

#### 4.2.1. Pre-procesamiento.

En esta etapa se prepara las imágenes para la segmentación y detección de los vehículos. Se empieza por leer las dos imágenes que son la imagen de referencia y la actual, esto se hace con la función *imread*, seguido a esto se las convierte a escala de grises haciendo uso de la función *rgb2gray* (ver Figura 28).



**Figura 28.** Lectura de las imágenes y conversión a escala de grises en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

Luego, se aplica la función ***imadjust*** que puede ampliar, reducir y en general, cambiar los rangos de intensidad de las imágenes de entrada a unos nuevos rangos en la imagen de salida (ver Figura 29).

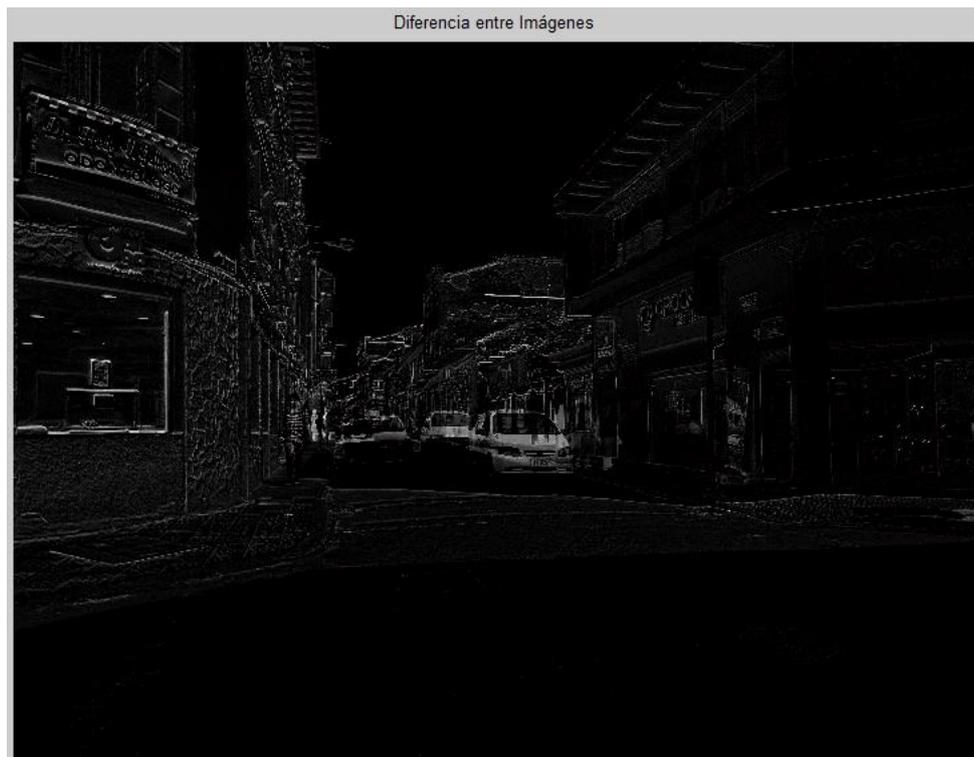


**Figura 29.** Ajuste de intensidad en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

Para finalizar la etapa de pre-procesamiento se realiza la diferencia entre la imagen actual y la imagen de referencia que es la imagen sin vehículos, esto se realiza con la función ***imsubtract*** que realiza una resta píxel a píxel entre ambas imágenes cuyo resultado se muestra en la Figura 30.



**Figura 30.** Diferencia entre imágenes en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

#### 4.2.2. Segmentación.

Se aplica la segmentación de líneas divisorias “watersheed”. Primero se obtiene la magnitud del gradiente, para ello se usa la detección de borde de Sobel para obtener los bordes ya que el gradiente es alto en las fronteras y bajo en el interior de los objetos. Esto se realiza con la función *fspecial*, también se usa la función *imfilter* para eliminar el ruido existente en la imagen y por último se aplica aritmética, la raíz cuadrada de la suma de los cuadrados de los resultados de las dos funciones anteriores, para obtener el gradiente (ver Figura 31).



**Figura 31.** Magnitud del gradiente en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

Luego, se procede a marcar los objetos en primer plano, para lo cual se usa la función de apertura *imopen* para eliminar aquellos objetos pequeños que se encuentran aislados y la de cierre *imclose* para llenar aquellos huecos que se encuentran dentro de los objetos, todo este proceso se realiza con el elemento estructural de disco (ver Figura 32); seguido a esto se realiza la reconstrucción de la imagen a partir de las funciones anteriores con la función *imreconstruct* que elimina las manchas pequeñas y se obtiene una imagen limpia de los objetos en primer plano (ver Figura 33).



**Figura 32.** Aplicación de la función de apertura y cierre en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

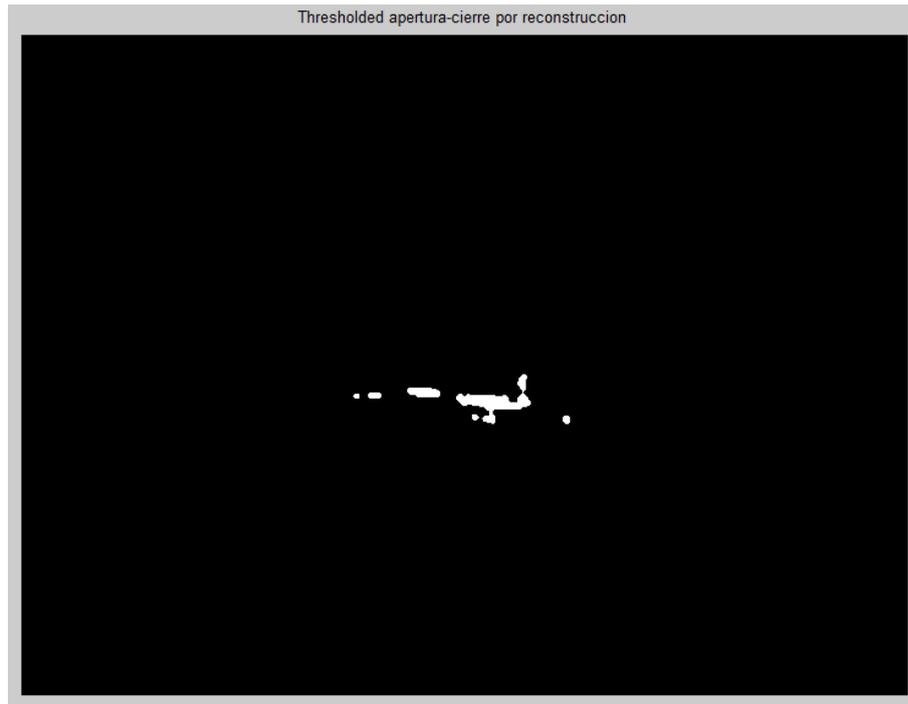


**Figura 33.** Reconstrucción por apertura-cierre en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

A continuación, sigue el proceso de umbralización y binarización para preparar la imagen para la aplicación de las operaciones morfológicas, para ello se procede a calcular el umbral óptimo automáticamente basándose en la forma del histograma con la función ***graythresh***, para luego, binarizar la imagen haciendo uso de este umbral con la función ***im2bw*** (ver Figura 34).

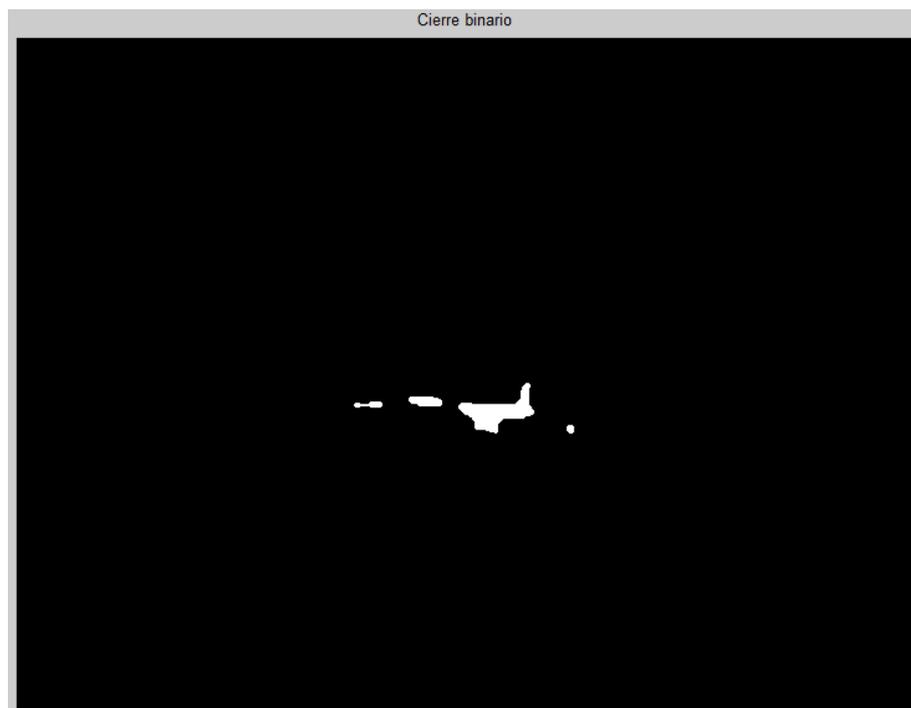


**Figura 34.** Binarización de la imagen en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

Para terminar con esta etapa se realiza la operación de cierre con la función *imclose*, la cual ayudará a rellenar aquellos pequeños agujeros que existan en la imagen (ver Figura 35).



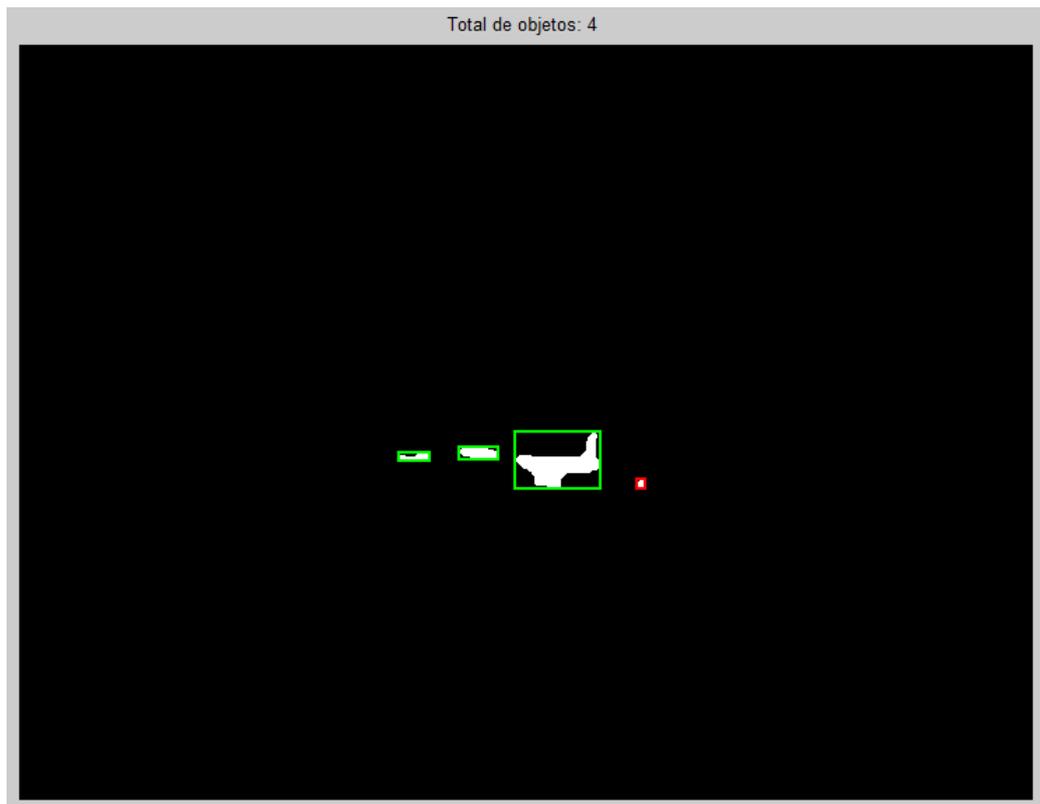
**Figura 35.** Aplicación de la operación de cierre en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

### 4.2.3. Detección y etiquetado de vehículos.

En esta etapa final para el etiquetado se hace uso de la función *bwlabel* luego, para encontrar las propiedades de los objetos de la imagen se usa la función *regionprops*, además, esta función tiene la propiedad de *BoundingBox* que permite enmarcar a los objetos con un rectángulo. También por medio de la propiedad *area* y haciendo uso de la función *find*, permite encontrar el área en píxeles de los objetos que se encuentran en la imagen, con esto se puede establecer un rango y eliminar aquellas áreas que están por debajo de nuestro rango que en este caso es de 3000 píxeles, los cuales se encierran en un rectángulo de color rojo (ver figura 36). Por último, se presenta el resultado final donde se muestra la detección y el conteo de los vehículos que se encuentran en la imagen (ver Figura 37).



**Figura 36.** Objetos detectados deseados y objetos por eliminar en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

Total de vehículos: 3



**Figura 37.** Resultado final con los vehículos detectados y contados en el algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

**CAPÍTULO V.**  
**PRUEBAS Y RESULTADOS EXPERIMENTALES**

## 5.1. Descripción de la forma de evaluar las pruebas

Para la validación de las pruebas realizadas se ha utilizado los conceptos estadísticos de sensibilidad y especificidad, para ello, se definirá varios conceptos que ayudarán a comprender de mejor manera estos conceptos estadísticos.

- **Verdaderos positivos (VP):** El vehículo está presente y es detectado.
- **Verdaderos negativos (VN):** El vehículo no está presente y no es detectado.
- **Falsos positivos (FP):** El vehículo no está presente, pero es detectado como uno.
- **Falsos negativos (FN):** El vehículo está presente pero no es detectado.
- **Sensibilidad:** Es la probabilidad de detectar correctamente a un vehículo. La fórmula (Pita Fernández & Pértegas Díaz, 2003) que se usa se la presenta a continuación:

$$\text{Sensibilidad} = \frac{VP}{VP+FN} \quad (1)$$

- **Especificidad:** Es la probabilidad de no detectar objetos que no son vehículos. La ecuación (Pita Fernández & Pértegas Díaz, 2003) es la siguiente:

$$\text{Especificidad} = \frac{VN}{VN+FP} \quad (2)$$

## 5.2. Resultados obtenidos de la implementación del algoritmo uno propuesto por Zhang (Zhang et al., 2013)

Para realizar las pruebas se tomaron como muestra 120 fotografías tomadas en veinte calles del centro de la ciudad de Loja en tres diferentes ambientes que fueron nublado, soleado y noche. A continuación, se presenta la Tabla 3 con los resultados experimentales del algoritmo propuesto por (Zhang et al., 2013) en estado nublado.

**Tabla 3.** Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado nublado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	6	2	3	4	0	2	100,00 %	42,86 %
2	10 de Agosto y Sucre	20	2	3	18	0	2	100,00 %	14,29 %
3	Bolívar y José Antonio	10	3	2	7	0	3	100,00 %	22,22 %
4	José Antonio y Bolívar	22	2	2	20	2	2	50,00 %	9,09 %
5	10 de Agosto y Bolívar	12	1	4	11	1	1	50,00 %	26,67 %
6	Bolívar y 10 de Agosto	21	5	3	9	0	5	100,00 %	25,00 %
7	Bolívar y Rocafuerte	22	4	4	18	0	4	100,00 %	18,18 %

8	Rocafuerte y Bolívar	34	3	3	31	1	3	75,00 %	8,82 %
9	Bernardo y Rocafuerte	5	2	0	3	1	2	66,67 %	0,00 %
10	Rocafuerte y Bernardo	29	1	2	28	3	1	25,00 %	6,67 %
11	10 de Agosto y Bernardo	7	3	4	4	1	3	75,00 %	50,00 %
12	Bernardo y 10 de Agosto	20	3	2	17	1	3	75,00 %	10,53 %
13	Miguel Riofrío y Bernardo	11	1	2	10	1	1	50,00 %	16,67 %
14	Bernardo y Miguel Riofrío	6	1	2	5	2	1	33,33 %	28,57 %
15	Bernardo y Azuay	6	1	2	5	2	1	33,33 %	28,57 %
16	Bernardo y José Antonio	32	3	1	29	0	3	100,00 %	3,33 %
17	José Antonio y Bernardo	20	3	1	17	2	3	60,00 %	5,56 %
18	Bolívar y Azuay	30	4	1	26	1	4	80,00 %	3,70 %
19	Azuay y Bolívar	12	1	1	11	0	1	100,00 %	8,33 %
20	Bolívar y Miguel Riofrío	13	1	0	12	1	1	50,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 4.** Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado soleado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	12	3	1	9	0	3	100,00 %	10,00 %
2	10 de Agosto y Sucre	17	2	1	15	1	2	66,67 %	6,25 %
3	Bolívar y José Antonio	14	2	2	12	1	2	66,67 %	14,29 %
4	José Antonio y Bolívar	5	0	1	5	1	0	0,00 %	16,67 %
5	10 de Agosto y Bolívar	20	1	1	19	1	1	50,00 %	5,00 %
6	Bolívar y 10 de Agosto	124	1	0	123	1	1	50,00 %	0,00 %
7	Bolívar y Rocafuerte	43	2	0	42	3	2	40,00 %	0,00 %
8	Rocafuerte y Bolívar	18	2	1	16	2	2	50,00 %	5,88 %
9	Bernardo y Rocafuerte	20	1	2	19	0	1	100,00 %	9,52 %
10	Rocafuerte y Bernardo	53	2	1	51	3	2	40,00 %	1,92 %
11	10 de Agosto y Bernardo	14	3	1	11	0	3	100,00 %	8,33 %
12	Bernardo y 10 de Agosto	9	1	1	8	2	1	33,33 %	11,11 %
13	Miguel Riofrío y Bernardo	61	1	1	60	2	1	33,33 %	1,64 %
14	Bernardo y Miguel Riofrío	9	3	0	6	0	3	100,00 %	0,00 %
15	Bernardo y Azuay	21	4	0	17	0	4	100,00 %	0,00 %
16	Bernardo y José Antonio	31	4	1	27	0	4	100,00 %	3,57 %
17	José Antonio y Bernardo	8	1	0	7	0	1	100,00 %	0,00 %
18	Bolívar y Azuay	8	2	1	6	1	2	66,67 %	14,29 %
19	Azuay y Bolívar	10	2	1	8	0	2	100,00 %	11,11 %
20	Bolívar y Miguel Riofrío	34	3	1	31	1	3	75,00 %	3,13 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 5.** Resultados experimentales del algoritmo propuesto por Zhang (Zhang et al., 2013) en estado de noche.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	8	2	0	6	0	2	100,00 %	0,00 %
2	10 de Agosto y Sucre	9	2	0	7	1	2	66,67 %	0,00 %
3	Bolívar y José Antonio	8	1	3	7	2	1	33,33 %	30,00 %
4	José Antonio y Bolívar	10	1	1	9	1	1	50,00 %	10,00 %
5	10 de Agosto y Bolívar	14	2	0	12	1	2	66,67 %	0,00 %
6	Bolívar y 10 de Agosto	8	2	0	6	0	2	100,00 %	0,00 %
7	Bolívar y Rocafuerte	11	2	1	9	1	2	66,67 %	10,00 %
8	Rocafuerte y Bolívar	8	2	0	6	0	2	100,00 %	0,00 %
9	Bernardo y Rocafuerte	15	3	0	12	1	3	75,00 %	0,00 %
10	Rocafuerte y Bernardo	8	3	2	5	0	3	100,00 %	28,57 %
11	10 de Agosto y Bernardo	3	1	0	2	0	1	100,00 %	0,00 %
12	Bernardo y 10 de Agosto	10	2	1	8	0	2	100,00 %	11,11 %
13	Miguel Riofrío y Bernardo	18	2	1	16	0	2	100,00 %	5,88 %
14	Bernardo y Miguel Riofrío	11	1	0	10	0	1	100,00 %	0,00 %
15	Bernardo y Azuay	2	1	0	1	0	1	100,00 %	0,00 %
16	Bernardo y José Antonio	10	1	1	9	1	1	50,00 %	10,00 %
17	José Antonio y Bernardo	9	2	1	7	0	2	100,00 %	12,50 %
18	Bolívar y Azuay	13	2	1	11	1	2	66,67 %	8,33 %
19	Azuay y Bolívar	8	1	1	7	0	1	100,00 %	12,50 %
20	Bolívar y Miguel Riofrío	9	2	0	7	0	2	100,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

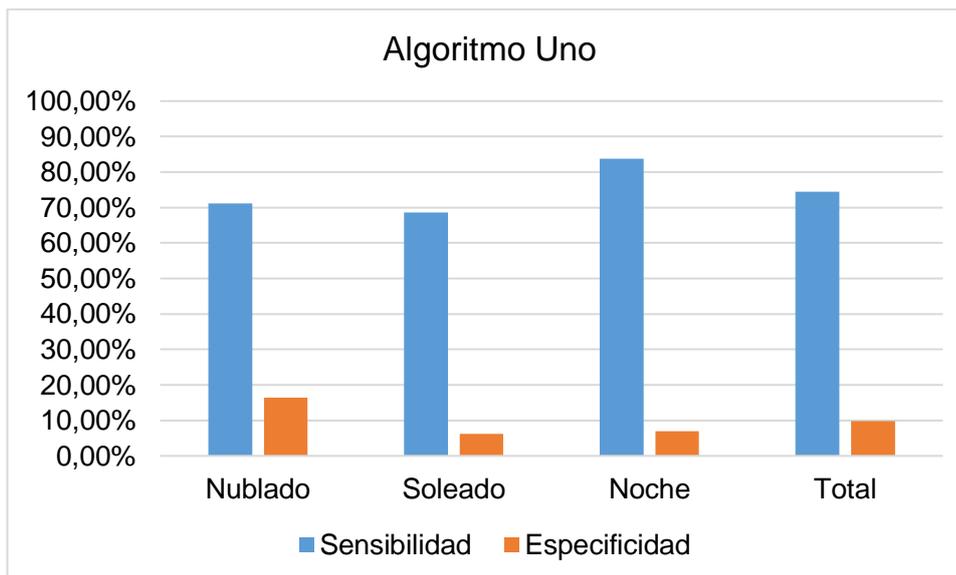
En la Tabla 6 se presenta los resultados totales de sensibilidad y especificidad en cada uno de los tres escenarios.

**Tabla 6.** Resultados totales del algoritmo propuesto por Zhang (Zhang et al., 2013).

	Sensibilidad	Especificidad
<b>Nublado</b>	71,17 %	16,45 %
<b>Soleado</b>	68,58 %	6,14 %
<b>Noche</b>	83,75 %	6,94 %
<b>Total</b>	74,50 %	9,84 %

Fuente: Autor.

Elaboración: Autor.



**Figura 38.** Porcentajes del algoritmo propuesto por Zhang (Zhang et al., 2013) por escenario.

**Fuente:** Autor.

**Elaboración:** Autor.

En los resultados totales del algoritmo uno, se puede observar que en los diferentes escenarios obtiene un buen porcentaje en sensibilidad, pero un valor muy bajo en especificidad. Esto da a entender que este algoritmo tiene buena capacidad en la detección de los vehículos, pero poca precisión en no detectar objetos que no son vehículos.

### 5.3. Resultados de la implementación del algoritmo dos propuesto por Bagyamani (Bagyamani, 2015)

A continuación, se presentan los resultados de las pruebas de las 120 imágenes en los tres escenarios distintos que son nublado, soleado y noche.

**Tabla 7.** Resultados experimentales del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado nublado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	4	2	2	2	0	2	100,00 %	50,00 %
2	10 de Agosto y Sucre	12	2	3	10	0	2	100,00 %	23,08 %
3	Bolívar y José Antonio	5	2	2	3	1	2	66,67 %	40,00 %
4	José Antonio y Bolívar	17	2	2	15	2	2	50,00 %	11,76 %
5	10 de Agosto y Bolívar	11	2	3	8	0	2	100,00 %	27,27 %
6	Bolívar y 10 de Agosto	7	4	3	3	1	4	80,00 %	50,00 %
7	Bolívar y Rocafuerte	12	4	4	8	0	4	100,00 %	33,33 %

8	Rocafuerte y Bolívar	18	3	3	15	1	3	75,00 %	16,67 %
9	Bernardo y Rocafuerte	4	2	0	2	1	2	66,67 %	0,00 %
10	Rocafuerte y Bernardo	22	1	2	21	3	1	25,00 %	8,70 %
11	10 de Agosto y Bernardo	5	3	4	2	1	3	75,00 %	66,67 %
12	Bernardo y 10 de Agosto	11	2	2	9	2	2	50,00 %	18,18 %
13	Miguel Riofrío y Bernardo	7	1	2	6	1	1	50,00 %	25,00 %
14	Bernardo y Miguel Riofrío	5	1	2	4	2	1	33,33 %	33,33 %
15	Bernardo y Azuay	3	1	2	2	2	1	33,33 %	50,00 %
16	Bernardo y José Antonio	33	3	1	30	0	3	100,00 %	3,23 %
17	José Antonio y Bernardo	13	3	1	10	2	3	60,00 %	9,09 %
18	Bolívar y Azuay	22	4	1	18	1	4	80,00 %	5,26 %
19	Azuay y Bolívar	8	1	1	7	0	1	100,00 %	12,50 %
20	Bolívar y Miguel Riofrío	9	1	0	8	1	1	50,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 8.** Resultados del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado soleado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	7	3	1	4	0	3	100,00 %	20,00 %
2	10 de Agosto y Sucre	13	2	1	11	1	2	66,67 %	8,33 %
3	Bolívar y José Antonio	6	2	2	4	1	2	66,67 %	33,33 %
4	José Antonio y Bolívar	2	0	1	2	1	0	0,00 %	33,33 %
5	10 de Agosto y Bolívar	15	1	1	14	1	1	50,00 %	6,67 %
6	Bolívar y 10 de Agosto	61	1	0	60	1	1	50,00 %	0,00 %
7	Bolívar y Rocafuerte	24	2	0	22	3	2	40,00 %	0,00 %
8	Rocafuerte y Bolívar	11	2	1	9	2	2	50,00 %	10,00 %
9	Bernardo y Rocafuerte	13	1	2	12	0	1	100,00 %	14,29 %
10	Rocafuerte y Bernardo	31	2	2	29	3	2	40,00 %	6,45 %
11	10 de Agosto y Bernardo	10	2	1	8	1	2	66,67 %	11,11 %
12	Bernardo y 10 de Agosto	6	1	1	5	2	1	33,33 %	16,67 %
13	Miguel Riofrío y Bernardo	35	1	1	34	2	1	33,33 %	2,86 %
14	Bernardo y Miguel Riofrío	7	3	0	4	0	3	100,00 %	0,00 %
15	Bernardo y Azuay	13	3	0	10	1	3	75,00 %	0,00 %
16	Bernardo y José Antonio	30	4	1	26	0	4	100,00 %	3,70 %
17	José Antonio y Bernardo	5	1	0	4	0	1	100,00 %	0,00 %
18	Bolívar y Azuay	7	2	1	5	1	2	66,67 %	16,67 %
19	Azuay y Bolívar	6	2	1	4	0	2	100,00 %	20,00 %
20	Bolívar y Miguel Riofrío	26	3	1	23	1	3	75,00 %	4,17 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 9.** Resultados del algoritmo propuesto por Bagyamani (Bagyamani, 2015) en estado de noche.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	8	2	0	6	0	2	100,00 %	0,00 %
2	10 de Agosto y Sucre	5	2	0	3	1	2	66,67 %	0,00 %
3	Bolívar y José Antonio	7	1	1	6	2	1	33,33 %	14,29 %
4	José Antonio y Bolívar	8	1	1	7	1	1	50,00 %	12,50 %
5	10 de Agosto y Bolívar	9	2	0	7	1	2	66,67 %	0,00 %
6	Bolívar y 10 de Agosto	6	2	0	4	0	2	100,00 %	0,00 %
7	Bolívar y Rocafuerte	10	2	1	8	1	2	66,67 %	11,11 %
8	Rocafuerte y Bolívar	6	2	0	4	0	2	100,00 %	0,00 %
9	Bernardo y Rocafuerte	11	3	0	8	1	3	75,00 %	0,00 %
10	Rocafuerte y Bernardo	7	3	2	4	0	3	100,00 %	33,33 %
11	10 de Agosto y Bernardo	4	1	0	3	0	1	100,00 %	0,00 %
12	Bernardo y 10 de Agosto	5	2	1	3	0	2	100,00 %	25,00 %
13	Miguel Riofrío y Bernardo	11	2	1	9	0	2	100,00 %	10,00 %
14	Bernardo y Miguel Riofrío	10	1	0	9	0	1	100,00 %	0,00 %
15	Bernardo y Azuay	2	1	0	1	0	1	100,00 %	0,00 %
16	Bernardo y José Antonio	6	1	1	5	1	1	50,00 %	16,67 %
17	José Antonio y Bernardo	7	2	1	5	0	2	100,00 %	16,67 %
18	Bolívar y Azuay	12	2	1	10	1	2	66,67 %	9,09 %
19	Azuay y Bolívar	6	1	1	5	0	1	100,00 %	16,67 %
20	Bolívar y Miguel Riofrío	8	2	0	6	0	2	100,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

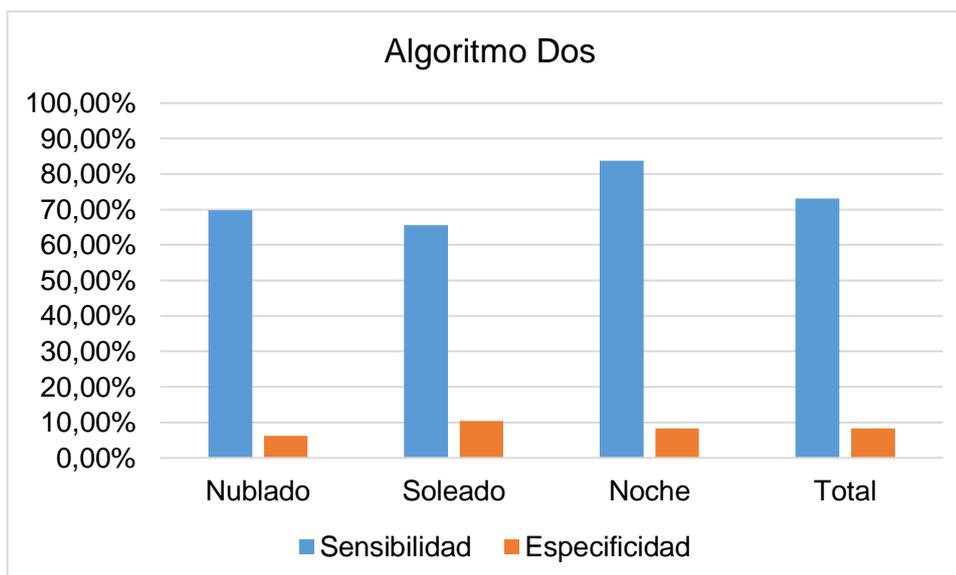
A continuación, se presentan los resultados finales en cada uno de los estados y los resultados totales obtenidos de este algoritmo.

**Tabla 10.** Resultados finales del algoritmo propuesto por Bagyamani (Bagyamani, 2015)

	Sensibilidad	Especificidad
<b>Nublado</b>	69,75 %	6,14 %
<b>Soleado</b>	65,67 %	10,38 %
<b>Noche</b>	83,75 %	8,27 %
<b>Total</b>	73,06 %	8,26 %

Fuente: Autor.

Elaboración: Autor.



**Figura 39.** Porcentajes finales del algoritmo propuesto por Bagyamani (Bagyamani, 2015)

**Fuente:** Autor.

**Elaboración:** Autor.

En los resultados totales del algoritmo dos al igual que el algoritmo uno se puede observar que en los diferentes escenarios obtiene un buen porcentaje en sensibilidad, pero un valor muy bajo en especificidad. Esto da a entender que este algoritmo tiene buena capacidad en la detección de los vehículos, pero poca precisión en no detectar objetos que no son vehículos.

#### 5.4. Resultados de la implementación del algoritmo tres propuesto por Abinaya (Abinaya et al., 2013)

A continuación, se presentan los resultados de las pruebas de las 120 imágenes en los tres escenarios distintos que son nublado, soleado y noche.

**Tabla 11.** Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado nublado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	1	1	3	0	1	1	50,00 %	100,00 %
2	10 de Agosto y Sucre	7	2	3	5	0	2	100,00 %	37,50 %
3	Bolívar y José Antonio	2	2	2	0	1	2	66,67 %	100,00 %
4	José Antonio y Bolívar	6	3	2	3	1	3	75,00 %	40,00 %
5	10 de Agosto y Bolívar	8	1	3	7	1	1	50,00 %	30,00 %
6	Bolívar y 10 de Agosto	3	2	3	1	3	2	40,00 %	75,00 %
7	Bolívar y Rocafuerte	10	3	4	7	1	3	75,00 %	36,36 %
8	Rocafuerte y Bolívar	11	2	3	9	2	2	50,00 %	25,00 %

9	Bernardo y Rocafuerte	3	2	0	1	1	2	66,67 %	0,00 %
10	Rocafuerte y Bernardo	14	1	2	13	3	1	25,00 %	13,33 %
11	10 de Agosto y Bernardo	9	3	4	6	1	3	75,00 %	40,00 %
12	Bernardo y 10 de Agosto	6	2	1	4	2	2	50,00 %	20,00 %
13	Miguel Riofrío y Bernardo	8	0	1	8	2	0	0,00 %	11,11 %
14	Bernardo y Miguel Riofrío	5	1	2	4	2	1	33,33 %	33,33 %
15	Bernardo y Azuay	1	1	2	0	2	1	33,33 %	100,00 %
16	Bernardo y José Antonio	13	3	1	10	0	3	100,00 %	9,09 %
17	José Antonio y Bernardo	8	3	1	5	2	3	60,00 %	16,67 %
18	Bolívar y Azuay	9	3	1	6	2	3	60,00 %	14,29 %
19	Azuay y Bolívar	5	1	1	4	0	1	100,00 %	20,00 %
20	Bolívar y Miguel Riofrío	6	1	0	5	1	1	50,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 12.** Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado soleado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	5	2	0	3	1	2	66,67 %	0,00 %
2	10 de Agosto y Sucre	5	1	1	4	2	1	33,33 %	20,00 %
3	Bolívar y José Antonio	6	1	2	5	2	1	33,33 %	28,57 %
4	José Antonio y Bolívar	4	0	1	4	1	0	0,00 %	20,00 %
5	10 de Agosto y Bolívar	7	1	2	6	1	1	50,00 %	25,00 %
6	Bolívar y 10 de Agosto	22	0	0	22	2	0	0,00 %	0,00 %
7	Bolívar y Rocafuerte	13	1	0	12	4	1	20,00 %	0,00 %
8	Rocafuerte y Bolívar	5	2	1	3	2	2	50,00 %	25,00 %
9	Bernardo y Rocafuerte	5	1	2	4	0	1	100,00 %	33,33 %
10	Rocafuerte y Bernardo	18	2	2	16	3	2	40,00 %	11,11 %
11	10 de Agosto y Bernardo	11	2	1	9	1	2	66,67 %	10,00 %
12	Bernardo y 10 de Agosto	6	1	1	5	2	1	33,33 %	16,67 %
13	Miguel Riofrío y Bernardo	10	1	1	9	2	1	33,33 %	10,00 %
14	Bernardo y Miguel Riofrío	5	2	0	3	1	2	66,67 %	0,00 %
15	Bernardo y Azuay	7	2	0	5	2	2	50,00 %	0,00 %
16	Bernardo y José Antonio	20	2	1	18	2	2	50,00 %	5,26 %
17	José Antonio y Bernardo	2	1	0	1	0	1	100,00 %	0,00 %
18	Bolívar y Azuay	4	2	1	2	1	2	66,67 %	33,33 %
19	Azuay y Bolívar	3	1	1	2	1	1	50,00 %	33,33 %
20	Bolívar y Miguel Riofrío	16	3	1	13	1	3	75,00 %	7,14 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 13.** Resultados del algoritmo propuesto por Abinaya (Abinaya et al., 2013) en estado de noche.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	6	2	0	4	0	2	100,00 %	0,00 %
2	10 de Agosto y Sucre	4	2	0	2	1	2	66,67 %	0,00 %
3	Bolívar y José Antonio	3	1	1	2	1	1	50,00 %	33,33 %
4	José Antonio y Bolívar	8	1	1	7	1	1	50,00 %	12,50 %
5	10 de Agosto y Bolívar	6	2	1	4	1	2	66,67 %	20,00 %
6	Bolívar y 10 de Agosto	4	2	1	2	0	2	100,00 %	33,33 %
7	Bolívar y Rocafuerte	8	2	1	6	1	2	66,67 %	14,29 %
8	Rocafuerte y Bolívar	6	2	0	4	0	2	100,00 %	0,00 %
9	Bernardo y Rocafuerte	6	3	0	3	1	3	75,00 %	0,00 %
10	Rocafuerte y Bernardo	4	2	2	2	1	2	66,67 %	50,00 %
11	10 de Agosto y Bernardo	3	1	0	2	0	1	100,00 %	0,00 %
12	Bernardo y 10 de Agosto	3	2	1	1	0	2	100,00 %	50,00 %
13	Miguel Riofrío y Bernardo	8	2	1	6	0	2	100,00 %	14,29 %
14	Bernardo y Miguel Riofrío	3	1	0	2	0	1	100,00 %	0,00 %
15	Bernardo y Azuay	2	1	0	1	0	1	100,00 %	0,00 %
16	Bernardo y José Antonio	4	1	2	2	1	1	50,00 %	50,00 %
17	José Antonio y Bernardo	3	2	1	1	0	2	100,00 %	50,00 %
18	Bolívar y Azuay	9	2	1	7	1	2	66,67 %	12,50 %
19	Azuay y Bolívar	5	1	1	4	0	1	100,00 %	20,00 %
20	Bolívar y Miguel Riofrío	6	2	0	4	0	2	100,00 %	0,00 %

Fuente: Autor.

Elaboración: Autor.

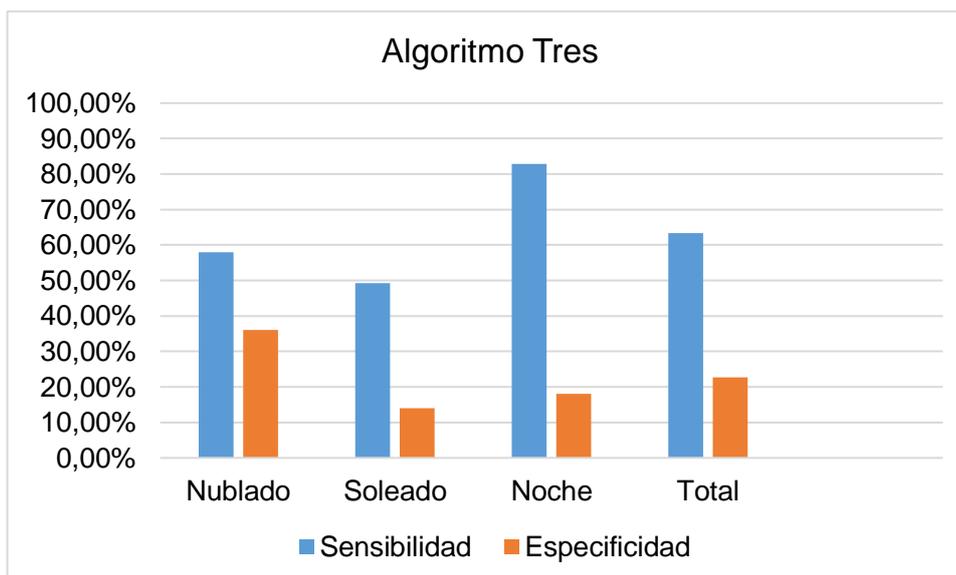
A continuación, se presentan los resultados finales en cada uno de los estados y los resultados totales obtenidos de este algoritmo.

**Tabla 14.** Resultados finales del algoritmo propuesto por Abinaya (Abinaya et al., 2013).

	Sensibilidad	Especificidad
<b>Nublado</b>	58,00 %	36,08 %
<b>Soleado</b>	49,25 %	13,94 %
<b>Noche</b>	82,92 %	18,01 %
<b>Total</b>	63,39 %	22,68 %

Fuente: Autor.

Elaboración: Autor.



**Figura 40.** Porcentajes finales del algoritmo propuesto por Abinaya (Abinaya et al., 2013)

**Fuente:** Autor.

**Elaboración:** Autor.

En los resultados totales del tercer algoritmo, se puede observar que en los diferentes escenarios obtiene un buen porcentaje en sensibilidad, pero un valor muy bajo en especificidad, aunque los valores no sean muy buenos, se observa que este algoritmo está mejor equilibrado de los anteriores. Esto significa que este algoritmo tiene buena capacidad en la detección de los vehículos, pero con relación a los algoritmos anteriores tiene una mejor precisión en no detectar objetos que no son vehículos.

### 5.5. Resultados de la implementación del algoritmo MAT propuesto por el Autor

A continuación, se presentan los resultados de las pruebas de las 120 imágenes en los tres escenarios distintos que son nublado, soleado y noche.

**Tabla 15.** Resultados del algoritmo MAT en estado nublado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	2	2	3	0	0	2	100,00 %	100,00 %
2	10 de Agosto y Sucre	6	2	4	2	0	2	100,00 %	66,67 %
3	Bolívar y José Antonio	2	2	2	0	1	2	66,67 %	100,00 %
4	José Antonio y Bolívar	8	3	2	5	1	3	75,00 %	28,57 %
5	10 de Agosto y Bolívar	8	2	4	6	0	2	100,00 %	40,00 %
6	Bolívar y 10 de Agosto	2	2	3	0	3	2	40,00 %	100,00 %

7	Bolívar y Rocafuerte	7	3	5	4	1	3	75,00 %	55,56 %
8	Rocafuerte y Bolívar	11	3	3	8	1	3	75,00 %	27,27 %
9	Bernardo y Rocafuerte	3	2	1	1	1	2	66,67 %	50,00 %
10	Rocafuerte y Bernardo	11	1	3	10	3	1	25,00 %	23,08 %
11	10 de Agosto y Bernardo	6	3	4	3	1	3	75,00 %	57,14 %
12	Bernardo y 10 de Agosto	7	3	4	4	1	3	75,00 %	50,00 %
13	Miguel Riofrío y Bernardo	7	1	4	6	1	1	50,00 %	40,00 %
14	Bernardo y Miguel Riofrío	6	2	3	4	1	2	66,67 %	42,86 %
15	Bernardo y Azuay	3	2	2	1	1	2	66,67 %	66,67 %
16	Bernardo y José Antonio	8	3	4	5	0	3	100,00 %	44,44 %
17	José Antonio y Bernardo	7	3	3	4	2	3	60,00 %	42,86 %
18	Bolívar y Azuay	9	4	3	5	1	4	80,00 %	37,50 %
19	Azuay y Bolívar	2	1	2	1	0	1	100,00 %	66,67 %
20	Bolívar y Miguel Riofrío	3	1	1	2	1	1	50,00 %	33,33 %

Fuente: Autor.

Elaboración: Autor.

Tabla 16. Resultados del algoritmo MAT en estado soleado.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	3	3	2	0	0	3	100,00 %	100,00 %
2	10 de Agosto y Sucre	5	2	3	3	1	2	66,67 %	50,00 %
3	Bolívar y José Antonio	4	2	3	3	1	2	66,67 %	50,00 %
4	José Antonio y Bolívar	3	1	2	2	0	1	100,00 %	50,00 %
5	10 de Agosto y Bolívar	3	1	3	2	1	1	50,00 %	60,00 %
6	Bolívar y 10 de Agosto	16	1	3	15	1	1	50,00 %	16,67 %
7	Bolívar y Rocafuerte	8	2	1	6	3	2	40,00 %	14,29 %
8	Rocafuerte y Bolívar	5	3	2	2	1	3	75,00 %	50,00 %
9	Bernardo y Rocafuerte	4	1	6	3	0	1	100,00 %	66,67 %
10	Rocafuerte y Bernardo	12	2	3	10	3	2	40,00 %	23,08 %
11	10 de Agosto y Bernardo	11	3	3	8	0	3	100,00 %	27,27 %
12	Bernardo y 10 de Agosto	5	2	2	3	1	2	66,67 %	40,00 %
13	Miguel Riofrío y Bernardo	4	2	2	1	1	2	66,67 %	66,67 %
14	Bernardo y Miguel Riofrío	5	3	2	2	0	3	100,00 %	50,00 %
15	Bernardo y Azuay	6	4	1	2	0	4	100,00 %	33,33 %
16	Bernardo y José Antonio	14	4	3	10	0	4	100,00 %	23,08 %
17	José Antonio y Bernardo	1	1	1	0	0	1	100,00 %	100,00 %
18	Bolívar y Azuay	2	2	3	0	1	2	66,67 %	100,00 %
19	Azuay y Bolívar	3	2	3	1	0	2	100,00 %	75,00 %
20	Bolívar y Miguel Riofrío	10	3	4	7	1	3	75,00 %	36,36 %

Fuente: Autor.

Elaboración: Autor.

**Tabla 17.** Resultados del algoritmo MAT en estado de noche.

Número de calle	Nombre de calle	Vehículos detectados	Verdaderos positivos	Verdaderos negativos	Falsos positivos	Falsos Negativos	Vehículos totales	Sensibilidad	Especificidad
1	Sucre y 10 de Agosto	4	2	1	2	0	2	100,00 %	33,33 %
2	10 de Agosto y Sucre	4	3	1	1	0	3	100,00 %	50,00 %
3	Bolívar y José Antonio	3	2	3	1	1	2	66,67 %	75,00 %
4	José Antonio y Bolívar	5	2	2	3	0	2	100,00 %	40,00 %
5	10 de Agosto y Bolívar	5	3	1	2	0	3	100,00 %	33,33 %
6	Bolívar y 10 de Agosto	3	2	2	1	0	2	100,00 %	66,67 %
7	Bolívar y Rocafuerte	6	3	2	3	0	3	100,00 %	40,00 %
8	Rocafuerte y Bolívar	3	2	1	1	0	2	100,00 %	50,00 %
9	Bernardo y Rocafuerte	5	4	0	1	0	4	100,00 %	0,00 %
10	Rocafuerte y Bernardo	4	3	4	1	0	3	100,00 %	80,00 %
11	10 de Agosto y Bernardo	2	1	1	1	0	1	100,00 %	50,00 %
12	Bernardo y 10 de Agosto	3	2	2	1	0	2	100,00 %	66,67 %
13	Miguel Riofrío y Bernardo	5	2	3	3	0	2	100,00 %	50,00 %
14	Bernardo y Miguel Riofrío	2	1	0	1	0	1	100,00 %	0,00 %
15	Bernardo y Azuay	2	1	0	1	0	1	100,00 %	0,00 %
16	Bernardo y José Antonio	3	2	4	1	0	2	100,00 %	80,00 %
17	José Antonio y Bernardo	2	1	2	1	0	1	100,00 %	66,67 %
18	Bolívar y Azuay	7	3	2	4	0	3	100,00 %	33,33 %
19	Azuay y Bolívar	2	1	2	1	0	1	100,00 %	66,67 %
20	Bolívar y Miguel Riofrío	4	2	1	2	0	2	100,00 %	33,33 %

Fuente: Autor.

Elaboración: Autor.

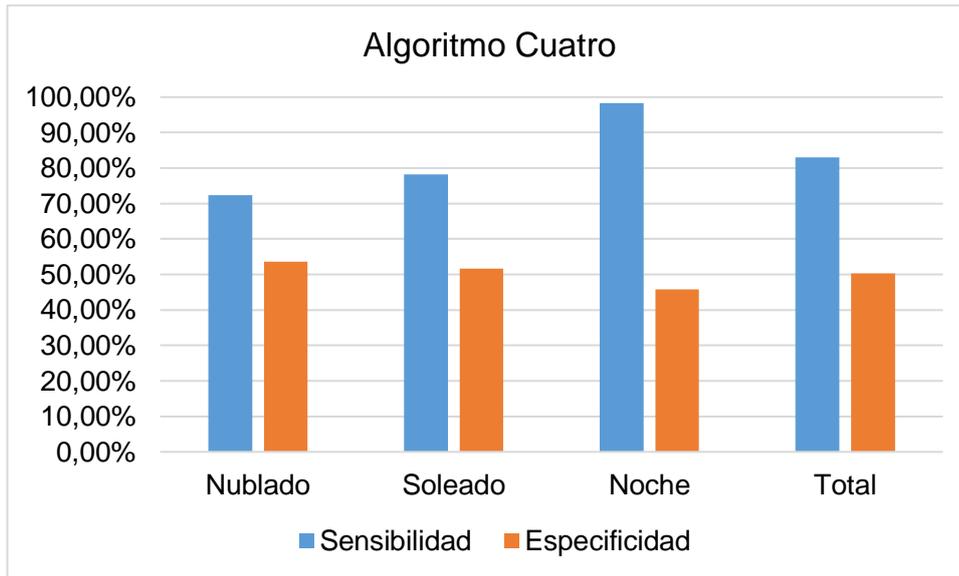
A continuación, se presentan los resultados finales en cada uno de los estados y los resultados totales obtenidos de este algoritmo.

**Tabla 18.** Resultados finales del algoritmo MAT.

	Sensibilidad	Especificidad
<b>Nublado</b>	72,33 %	53,63 %
<b>Soleado</b>	78,17 %	51,62 %
<b>Noche</b>	98,33 %	45,75 %
<b>Total</b>	82,94 %	50,33 %

Fuente: Autor.

Elaboración: Autor.



**Figura 41.** Porcentajes totales del algoritmo MAT.

**Fuente:** Autor.

**Elaboración:** Autor.

En los resultados totales de este algoritmo se puede observar claramente que aumenta el porcentaje en la precisión de detectar vehículos, pero lo más importante es que se logra un porcentaje que está alrededor del 50% en la no detección de objetos que no son vehículos.

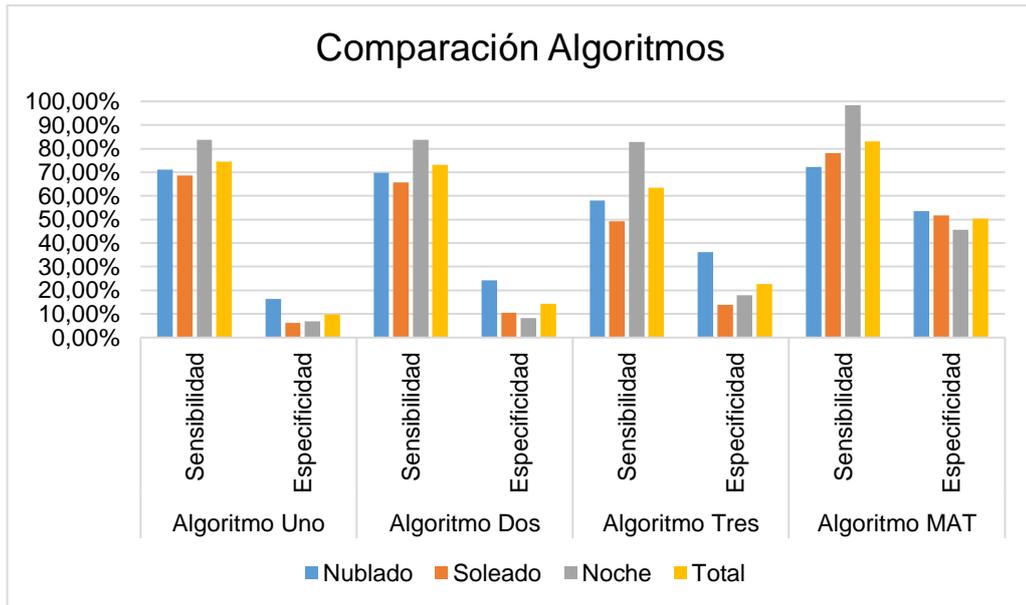
### 5.6. Comparación de los resultados de los cuatro algoritmos

**Tabla 19.** Comparación de los resultados de los cuatro algoritmos.

	Algoritmo Uno		Algoritmo Dos		Algoritmo Tres		Algoritmo MAT	
	Sensibilidad	Especificidad	Sensibilidad	Especificidad	Sensibilidad	Especificidad	Sensibilidad	Especificidad
<b>Nublado</b>	71,17 %	16,45 %	69,75 %	24,20 %	58,00 %	36,08 %	72,33 %	53,63 %
<b>Soleado</b>	68,58 %	6,14 %	65,67 %	10,38 %	49,25 %	13,94 %	78,17 %	51,62 %
<b>Noche</b>	83,75 %	6,94 %	83,75 %	8,27 %	82,92 %	18,01 %	98,33 %	45,75 %
<b>Total</b>	74,50 %	9,84 %	73,06 %	14,28 %	63,39 %	22,68 %	82,94 %	50,33 %

**Fuente:** Autor.

**Elaboración:** Autor.



**Figura 42.** Comparación de los porcentajes de los cuatro algoritmos.

**Fuente:** Autor.

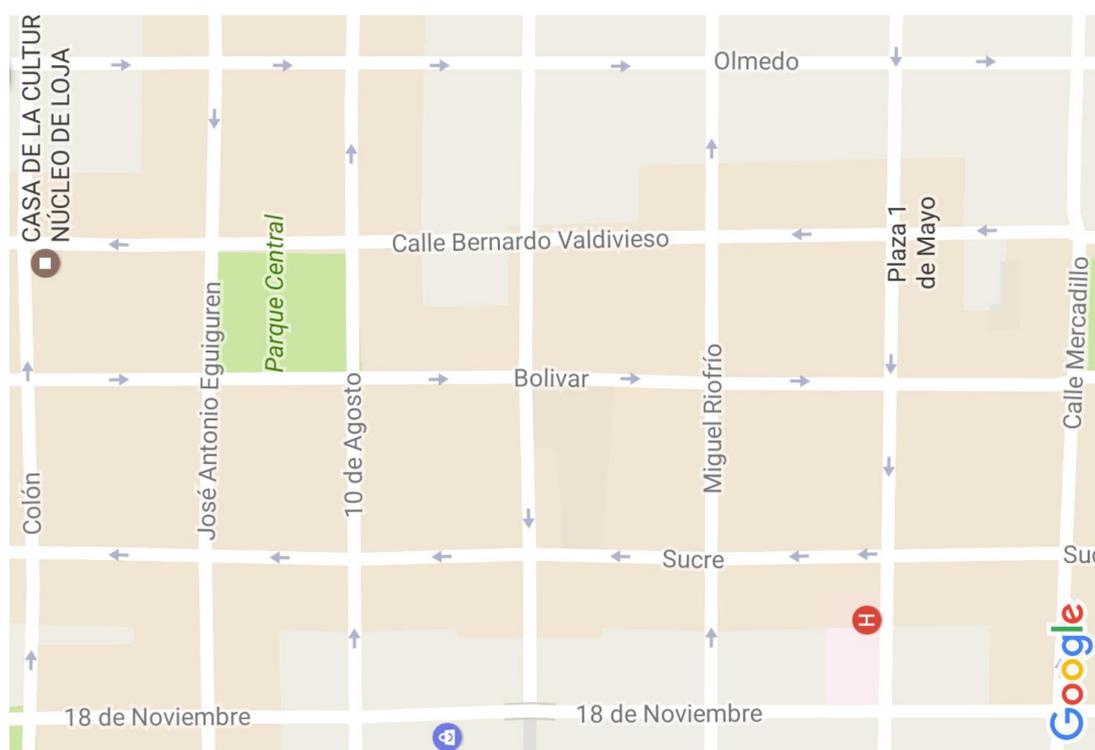
**Elaboración:** Autor.

**CAPÍTULO VI.**  
**DISCUSIÓN**

## 6.1. Análisis de costos e inversión

Para la implementación de la solución propuesta para el conteo de vehículos se necesitan cámaras y computadores. No se hará uso del trípode, las cámaras se las puede ubicar en los postes donde se encuentran ubicados los semáforos.

La implementación se debe realizar en el casco urbano de la ciudad de Loja-Ecuador, para lo cual se ha tomado en cuenta el área con mayor aforo vehicular; por cuanto, al momento de tomar las fotografías se pudo observar que en dicha área hay mayor presencia de vehículos. Esta área va horizontalmente desde la calle Mercadillo hasta la calle Colón y verticalmente desde la calle 18 de Noviembre hasta la calle Olmedo, esta área es presentada en la Figura 43.



**Figura 43.** Área de implementación.

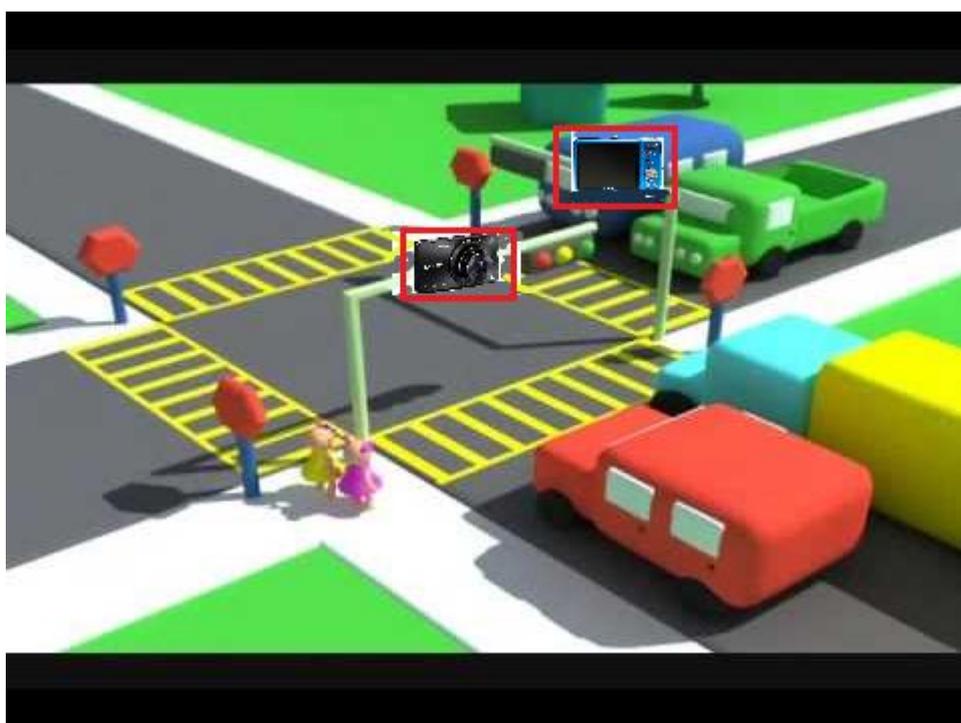
**Fuente:** Google Maps.

**Elaboración:** Autor.

A continuación, se presentan tres propuestas con tres cámaras digitales diferentes (ver Tabla 20, 21, 22). Vale recalcar que para la selección de las cámaras se ha tomado en cuenta su robustez frente a cambios climático, así como también que posean conexión inalámbrica. En la Figura 44 se muestra el diagrama de cómo se realizaron las fotografías para el desarrollo de este trabajo, pero para su implementación se tendrá que ubicar dos cámaras por cada esquina (ver Figura 45) con lo que se tiene un total de 70 cámaras y se colocará un computador por cada calle comprendida por dos esquinas, es decir, que cada computador recibirá fotografías de cuatro cámaras por lo que se tendrá un total de 18 computadoras.



**Figura 44.** Diagrama de cómo se obtuvieron las fotografías  
**Fuente:** Tomado de ("Seguridad vial calle," 2017), (Dreamstime, 2017).  
**Elaboración:** Autor.



**Figura 45.** Diagrama de cómo se debería implementar las cámaras en la solución propuesta  
**Fuente:** Tomado de ("Seguridad vial calle," 2017), (Interactivo, 2017b), (Interactivo, 2017a)  
**Elaboración:** Autor.

**Tabla 20.** Costos de equipos de la primera propuesta.

Equipo	Especificaciones	Cantidad	Costo Unitario	Costo Total
<b>Panasonic LUMIX DMC-TS6A</b>	<ul style="list-style-type: none"> <li>• 16.1 Megapíxeles</li> <li>• Resistencia al agua</li> <li>• Wi-fi</li> </ul>	70	636,26	44.538,20
<b>Dell Inspiron 5000 Series</b>	<ul style="list-style-type: none"> <li>• Procesador: Intel Core i7-7500U 2.7 GHz Up to 3.5 GHz</li> <li>• RAM: 16 GB DDR4, 2400MHz</li> <li>• Video: AMD Radeon R7 M445 con 4GB GDDR5</li> </ul>	18	1.252,82	22.550,76
<b>Imprevistos 10 %</b>				6.708,89
<b>TOTAL</b>				<b>73.797,85</b>

Fuente: Autor.

Elaboración: Autor.

**Tabla 21.** Costos de equipos de la segunda propuesta.

Equipo	Especificaciones	Cantidad	Costo Unitario	Costo Total
<b>Nikon COOLPIX W100</b>	<ul style="list-style-type: none"> <li>• 13,2 megapíxeles</li> <li>• Resistencia al agua</li> <li>• Wi-fi</li> </ul>	70	234,42	16.409,40
<b>Dell Inspiron 5000 Series</b>	<ul style="list-style-type: none"> <li>• Procesador: Intel Core i7-7500U 2.7 GHz Up to 3.5 GHz</li> <li>• RAM: 16 GB DDR4, 2400MHz</li> <li>• Video: AMD Radeon R7 M445 con 4G GDDR5</li> </ul>	18	1.252,82	22.550,76
<b>Imprevistos 10 %</b>				3.896,01
<b>TOTAL</b>				<b>42.856,17</b>

Fuente: Autor.

Elaboración: Autor.

**Tabla 22.** Costos de equipos de la tercera propuesta.

Equipo	Especificaciones	Cantidad	Costo Unitario	Costo Total
<b>Olympu s TG-870</b>	<ul style="list-style-type: none"> <li>• 16 megapíxeles</li> <li>• Resistencia al agua</li> <li>• Wi-fi</li> </ul>	70	549,66	38.476,20
<b>Dell Inspiron 5000 Series</b>	<ul style="list-style-type: none"> <li>• Procesador: Intel Core i7-7500U 2.7 GHz Up to 3.5 GHz</li> <li>• RAM: 16 GB DDR4, 2400MHz</li> <li>• Video: AMD Radeon R7 M445 con 4G B GDDR5</li> </ul>	18	1.252,82	22.550,76
<b>Imprevistos 10 %</b>				6.102,69
<b>TOTAL</b>				<b>67.129,65</b>

Fuente: Autor.

Elaboración: Autor.

Obtenidos los costos de los equipos, se procede a realizar los costos de ingeniería que son los costos de desarrollo de programa y licencias del mismo; luego se presentará los costos totales del sistema.

**Tabla 23.** Costos de ingeniería.

<b>Cantidad</b>	<b>Descripción</b>	<b>Costo Unitario</b>	<b>Costo total</b>
1	Desarrollo del programa	8.000,00	8.000,00
1	Licencia anual del sistema	1.500,00	1.500,00
<b>TOTAL</b>			9.500,00

**Fuente:** Autor.

**Elaboración:** Autor.

**Tabla 24.** Costos totales de las tres propuestas.

<b>Descripción</b>	<b>Primera Solución</b>	<b>Segunda Solución</b>	<b>Tercera Solución</b>
Costo de equipos	73.797,85	42.856,17	67.129,65
Costos de ingeniería	9.500,00	9.500,00	9.500,00
<b>COSTO TOTAL</b>	83.297,85	52.356,17	76.629,65

**Fuente:** Autor.

**Elaboración:** Autor.

Las tres propuestas con los costos de implementación del sistema son técnica y económicamente factibles por las ventajas que se ha mencionado a lo largo de este Trabajo de Titulación que son de mucha ayuda para la movilidad en el casco urbano de la ciudad de Loja.

## 6.2. Limitaciones

En un principio el objetivo era adquirir las imágenes con un teléfono celular, pero al realizarlo se obtuvo problemas ya que en los algoritmos en el pre-procesamiento se realiza la resta entre la imagen de fondo (imagen sin vehículos) y la imagen actual (imagen con vehículos). Al realizarlo con el celular no se puede tomar ambas imágenes desde la misma posición, por lo que existe el problema de la imagen fuera de foco y esto permite que varios objetos que no son de interés sean detectados.

A continuación, se presenta los resultados que se obtuvieron con las fotografías adquiridas con la cámara celular de un Huawei P9 cuyas características se presentan en la Tabla 25.

**Tabla 25.** Características del celular Huawei P9

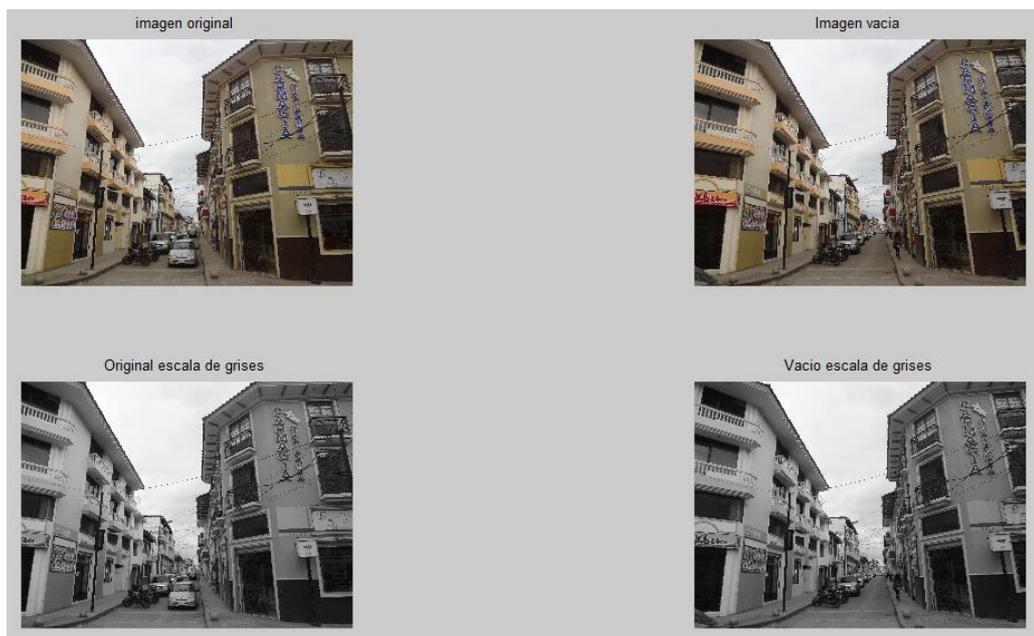
<b>PROCESADOR</b>	HiSilicon Kirin 955, 4 Cortex-A53 a 1,8 GHz y 4 Cortex-A72 a 2,5 GHz
<b>RAM</b>	3 / 4 GB

<b>MEMORIA</b>	32 / 64 GB (ampliables a otros 128 GB vía microSD)
<b>CONECTIVIDAD</b>	LTE Cat 6, NFC, Bluetooth 4.1, Wi-Fi 802.11ac
<b>CÁMARAS</b>	2 x 12 Mpíxeles (RGB + monocromo) sin estabilización f/2.2 Leica, frontal 8 MP f/2.0

**Fuente:** Tomado de (Puerto, 2016).

**Elaboración:** Autor.

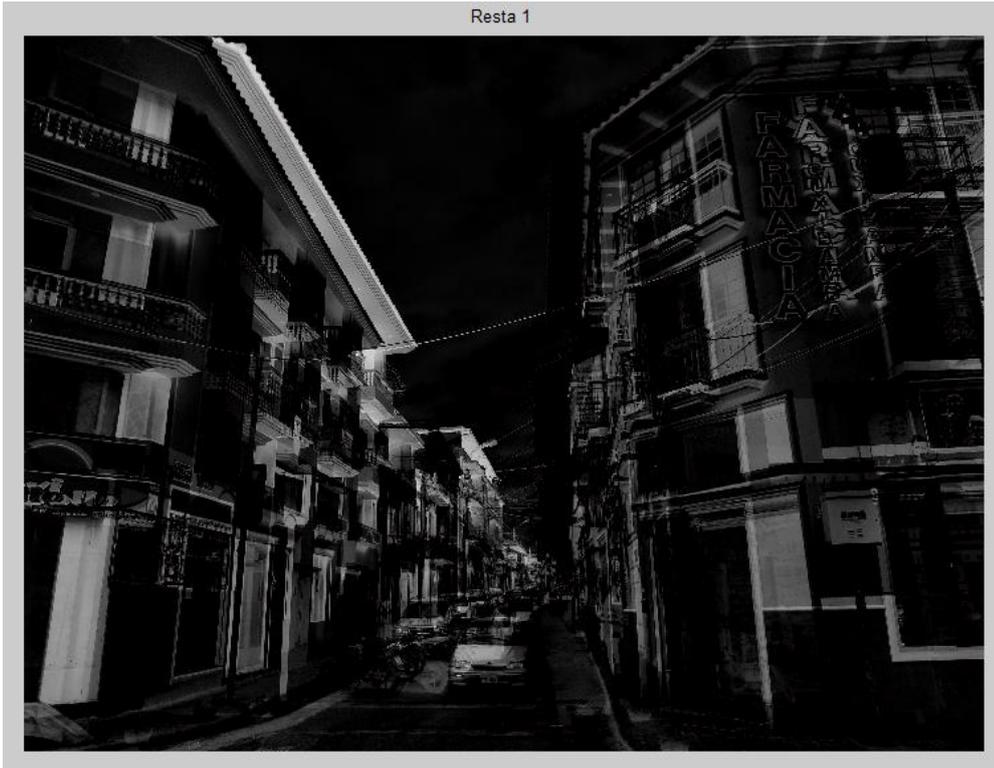
En la Figura 46 se presenta la lectura y la conversión a escala de grises de las imágenes adquiridas con el teléfono celular. Luego en la Figura 47 se muestra la resta de las imágenes de primer plano y la de fondo.



**Figura 46.** Lectura y conversión a escala de grises de las imágenes adquiridas con la cámara del teléfono celular.

**Fuente:** Autor.

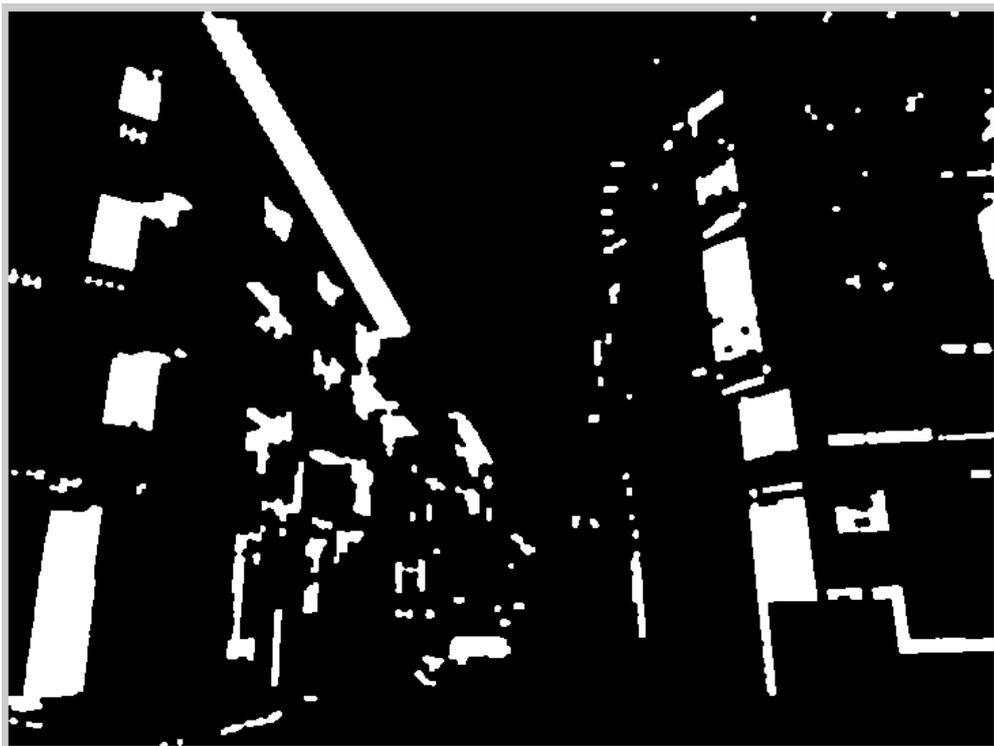
**Elaboración:** Autor.



**Figura 47.** Diferencia entre las imágenes adquiridas con la cámara del teléfono celular.

**Fuente:** Autor.

**Elaboración:** Autor.



**Figura 48.** Imagen binarizada de la prueba realizada con el teléfono celular.

**Fuente:** Autor.

**Elaboración:** Autor.

Como se puede observar en la Figura 48 no se puede identificar los vehículos debido a la gran cantidad de objetos que resultaron de la diferencia de imágenes. Para intentar solucionar este problema se hizo uso de la función *imregister*, dicha función toma una de las imágenes de referencia para realizar una medida cuantitativa de similitud entre ambas imágenes con el objeto optimizarlas para que queden alineadas. A continuación, en la Figura 49 se presentan los resultados obtenidos.



**Figura 49.** Imagen calzada de la prueba realizada con el teléfono celular

**Fuente:** Autor.

**Elaboración:** Autor.

Para la imagen calzada, se tomó como referencia la imagen con vehículos, por lo que la imagen sin vehículos es la que se alinea de acuerdo a la imagen de referencia, luego se procede a la resta entre la imagen calzada y la imagen con vehículos (ver Figura 50).



**Figura 50.** Resta de imágenes de la prueba realizada con el teléfono celular.

**Fuente:** Autor.

**Elaboración:** Autor.



**Figura 51.** Cierre binario de la prueba realizada con el teléfono celular.

**Fuente:** Autor.

**Elaboración:** Autor.

Claramente se puede ver que este resultado (ver Figura 51) con relación al primero se disminuye la cantidad de objetos no deseados, pero aun así el resultado no es el esperado por lo que no se detecta con claridad los vehículos.

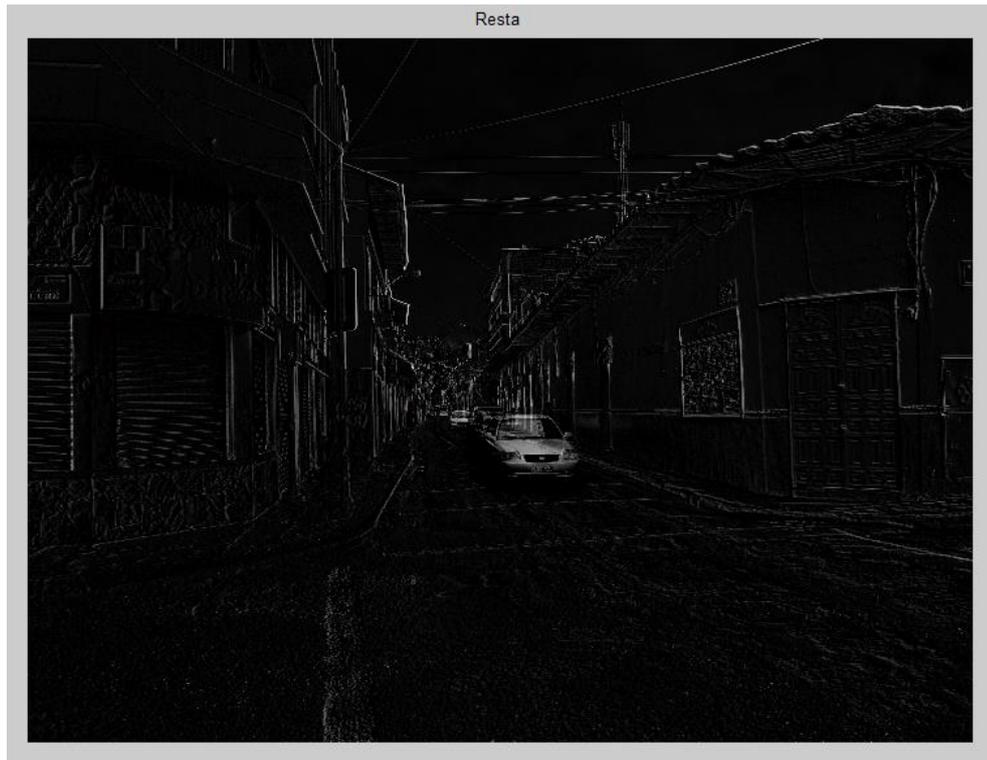
Por tal motivo, se ha optado por ocupar un trípode y una cámara fotográfica para realizar la adquisición de las imágenes, ya que utilizando esto se puede obtener las imágenes con mayor precisión y el resultado se notará al realizar la diferencia entre la imagen de fondo y la imagen actual. A continuación, se presenta una prueba realizada con la solución propuesta. En la Figura 52, se presenta la lectura de las imágenes y la conversión de las mismas a escala de grises, luego en la Figura 53, se presenta la resta de la imagen actual de la imagen de fondo.



**Figura 52.** Lectura y conversión a escala de grises de las imágenes adquiridas con la cámara y el trípode.

**Fuente:** Autor.

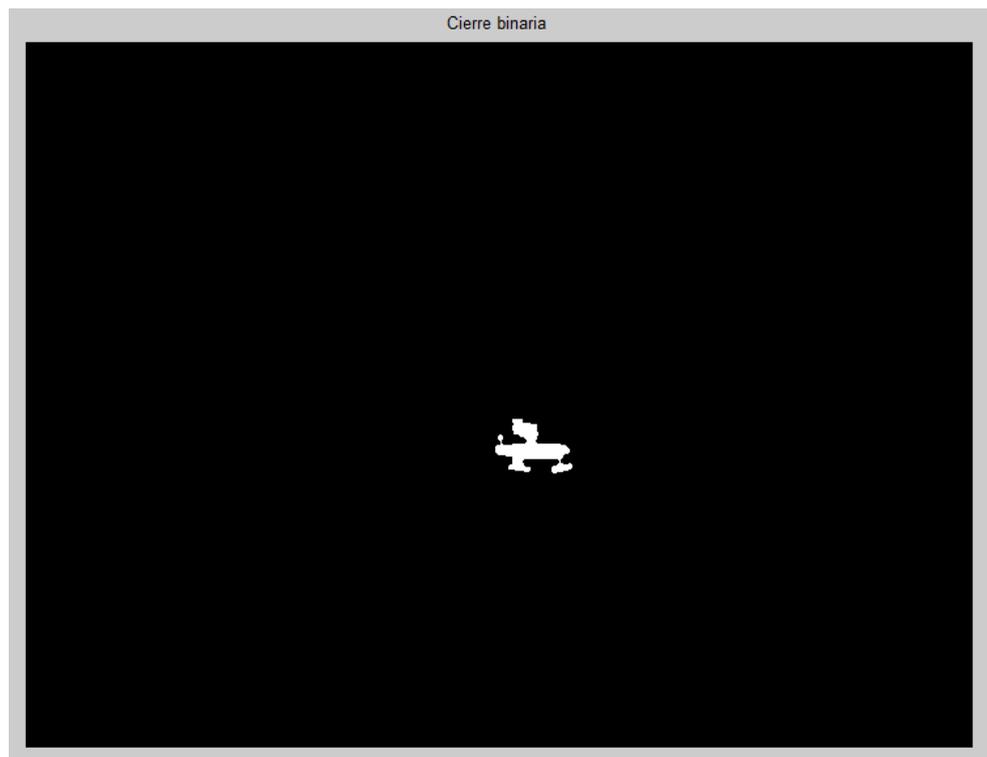
**Elaboración:** Autor.



**Figura 53.** Resta de las imágenes adquiridas con la cámara y el trípode.

**Fuente:** Autor.

**Elaboración:** Autor.



**Figura 54.** Cierre binario de la prueba realizada con la cámara y el trípode.

**Fuente:** Autor.

**Elaboración:** Autor.

De acuerdo a los resultados la solución propuesta (ver Figura 54) a los dos primeros resultados es la adecuada, por cuanto, se puede observar claramente cómo se detecta los objetos de interés que son los vehículos.

La mejor forma de trabajar con cámaras fijas es ubicándolas a una altura considerable para que así se pueda tener una mejor visión de los vehículos, ya que a la altura del trípode que se usa para adquirir las imágenes, los vehículos más cercanos a la cámara ocultan a los vehículos que se encuentran detrás.

## CONCLUSIONES

Se logró diseñar e implementar una solución para el conteo de vehículos a partir de imágenes terrestres.

Con la ayuda de los valores estadísticos de sensibilidad y especificidad se llegó a dar una validez a todas las pruebas realizadas por los cuatro algoritmos, ya que mediante esos valores se pudo saber con qué precisión se detectan los vehículos y la capacidad de no detectar objetos que no son vehículos. Los valores de sensibilidad y especificidad obtenidos son de 82,94 % y 50,33 % respectivamente.

La técnica de segmentación por líneas divisorias “watersheed” presentó mejores resultados por lo que mediante su proceso de reconstrucción de la imagen haciendo uso de operaciones morfológicas redujo en un 8,4 % los errores en la detección de los vehículos.

El algoritmo presentado presenta un aumento del 8,39 % en la sensibilidad con relación al primer algoritmo que es el que obtuvo mayor sensibilidad con respecto a los demás. lo que indica que el programa propuesto tiene mayor precisión en la detección de vehículos.

El resultado más importante es que se ha mejorado el valor de especificidad en un 27,65 % con relación al tercer algoritmo que es el que obtuvo mayor valor con respecto a los otros dos algoritmos, lo cual es muy alentador por cuanto, el algoritmo propuesto como solución tendrá menos errores en la detección porque tendrá un mejor funcionamiento al no detectar objetos que no son vehículos.

La noche es el escenario en donde se alcanza el valor más alto de precisión en la detección de los vehículos (98,33 %), pero se observa el porcentaje más bajo en la especificidad (45,75 %), ya que lo que se detecta son los faros de los vehículos, debido que a un vehículo lo reconoce como si fueran dos.

La solución propuesta también es técnica y económicamente factible lo cual se pudo comprobar en el análisis de los costos de inversión de equipos e ingeniería para ser aplicado por empresas inversionistas ya sean públicas o privadas.

## RECOMENDACIONES

Para un mejor desempeño en los resultados se recomienda que las fotografías adquiridas se las realice a una altura de 3 metros y fijas, donde se pueda observar con mayor claridad los vehículos presentes en las vías.

También es muy indispensable que la cámara se encuentre en una posición fija y estable para que las fotografías de entrada al sistema propuesto sean las correctas y no existan errores en la detección de los vehículos.

De acuerdo a la altura con que se toman las fotografías se puede realizar un análisis para determinar el valor mínimo y máximo de área de píxeles, con el cual se puede borrar o eliminar los objetos detectados que estén fuera del rango.

En el escenario de la noche lo que se detecta son los faros del vehículo, por lo que se sugiere que para este escenario se diseñe un nuevo algoritmo para que los dos faros del vehículo sean detectados como uno solo y no como dos, y así se puede reducir el margen de error en la detección.

## BIBLIOGRAFÍA

- Abinaya, P., Ravichandran, K., & Santhi, B. (2013). Watershed segmentation for vehicle classification and counting. *International journal of engineering and technology*, 5(2), 770-775.
- Aslani, S., & Mahdavi-Nasab, H. (2013). Optical flow based moving object detection and tracking for traffic surveillance. *International Journal of Electrical, Electronics, Communication, Energy Science and Engineering*, 7(9), 789-793.
- Bagyamani, J. (2015). Traffic Analysis on Highways based on Image Processing. *International Journal of Computational Intelligence and Informatics*, 5(1).
- Bhaskar, P. K., & Yong, S.-P. (2014). *Image processing based vehicle detection and tracking method*. Paper presented at the Computer and Information Sciences (ICCOINS), 2014 International Conference on.
- Bhaskar, P. K., Yong, S.-P., & Jung, L. T. (2015). *Enhanced and effective parallel optical flow method for vehicle detection and tracking*. Paper presented at the Mathematical Sciences and Computing Research (iSMSC), International Symposium on.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*(6), 679-698.
- Chávez Fuentes, C. P. (2015). *Sistema de semaforización inteligente para el control de flujo vehicular mediante el procesamiento digital de imágenes*. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones.
- Chen, L., & Ke, K. (2014). *The Research of Image Processing In Intelligent Transportation System*. Paper presented at the Applied Mechanics and Materials.
- Diego, B. (2017). *.:MATLAB - USO DE REGIONPROPS*. Retrieved from [http://www.matpic.com/esp/matlab/funcion\\_regionprops.html](http://www.matpic.com/esp/matlab/funcion_regionprops.html)
- Dreamstime. (2017). Silueta del hombre que toma imágenes con la cámara en el trípode. Retrieved from <https://es.dreamstime.com/imagen-de-archivo-silueta-del-hombre-que-toma-imagenes-con-la-cámara-en-el-trípode-image33402691>
- GÓMEZ, E. D. L. R. (2009). Mejora del tráfico en un cruce regulado por semáforos, mediante un sistema basado en Visión Artificial: Universidad Pontificia Comillas.[En línea] link: <http://www.iit.upcomillas.es/TT/resumenes/4a40baa5a5a7c.pdf>.

- Hashmi, M. F., Keskar, A. G., Reddy, R. S. K., & Kaushik, A. U. (2015). Ghost Vehicle and Shadow Removal Approach for Traffic Surveillance and Monitoring at Various Intersections Using Computer Vision. *International Journal of Multimedia and Ubiquitous Engineering*, 10(3), 375-388.
- Hincapié Cifuentes, L. (2014). *Desarrollo de una herramienta sistémica basada en procesamiento digital de video para el conteo de flujo vehicular, velocidad y tipo de automotor*. Universidad Autónoma de Occidente.
- Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3), 185-203.
- Image Processing Toolbox - MATLAB - MATLAB. (2017). Retrieved from <https://es.mathworks.com/products/image.html>
- Interactivo, C. (2017a). Cámara Digital Nikon Coolpix (S2600).
- Interactivo, C. (2017b). Cámara digital Olympus VG-140.
- Jain, I., & Rani, B. (2010). Vehicle detection using image processing and fuzzy logic. *International Journal of Computer Science & Communication*, 1(2), 255-257.
- Juneja, M., & Grover, P. (2013). *Improved segmentation of occluded and adjoining vehicles in traffic surveillance videos*. Paper presented at the Sixth International Conference on Machine Vision (ICMV 13).
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35-45.
- Liu, J., Zhao, Y., Yuan, Y., Luo, W., & Liu, K. (2011). *Vehicle capturing and counting using a new edge extraction approach*. Paper presented at the Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.
- Marker-Controlled Watershed Segmentation - MATLAB & Simulink Example. (2017). Retrieved from <https://www.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html>
- Moutakki, Z., Ayaou, T., Afdel, K., & Amghar, A. (2014). *Prototype of an embedded system using Stratix III FPGA for vehicle detection and traffic management*. Paper presented at the Multimedia Computing and Systems (ICMCS), 2014 International Conference on.

- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- Padmavathi, S., Naveen, C., & Kumari, V. A. (2016). Vision based Vehicle Counting for Traffic Congestion Analysis during Night Time. *Indian Journal of Science and Technology*, 9(20).
- Pan, X., Guo, Y., & Men, A. (2010). *Traffic surveillance system for vehicle flow detection*. Paper presented at the Computer Modeling and Simulation, 2010. ICCMS'10. Second International Conference on.
- Panda, A., Naik, A., & Patel, P. (2015). Application of Image Processing In Road Traffic Control. *Image*, 2(4).
- Parthasarathi, V., Surya, M., Akshay, B., Siva, K. M., & Vasudevan, S. K. (2015). Smart control of traffic signal system using image processing. *Indian Journal of Science and Technology*, 8(16), 1.
- Pita Fernández, S., & Pértegas Díaz, S. (2003). Pruebas diagnósticas: Sensibilidad y especificidad. *Cad Aten Primaria*, 10, 120-124.
- Prieto Valdés, M. A. (2010). *Algoritmo para conteo vehicular en tiempo real con base en franjas de interés*.
- Puerto, K. (2016). Huawei P9, Análisis. Review con características, precio y especificaciones - Xataka. *Xataka*.
- Quiceno Gutiérrez, C. A. (2012). *Metodología para la detección de incidentes de tráfico a partir de imágenes digitales empleando técnicas de inteligencia artificial*. Universidad Nacional de Colombia, Medellín.
- Raghtate, G., & Tiwari, A. K. (2014). Moving object counting in video signals. *International Journal of Engineering Research and General Science*, 2(3), 2091-2730.
- Ranga, H. P., Kiran, M. R., Shekar, S. R., & Kumar, S. N. (2010). *Vehicle detection and classification based on morphological technique*. Paper presented at the 2010 International Conference on Signal and Image Processing.
- Ruíz, M. O. (2014). *MATLAB Aplicado a Telecomunicaciones* (F. J. R. Cruz Ed. Primera ed.).
- Seguridad vial calle. (2017). Retrieved from <http://www.hmongbuy.com/related.php?uploader=NKMm1Avjeuc>

- Sharma, A. (2015). Automatic Vehicle Detection Using Various Object Detecting Algorithm and Thresholding Methods. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 1(1).
- Vibha, L., Hegde, C., Shenoy, P. D., Venugopal, K., & Patnaik, L. (2008). Dynamic object detection, tracking and counting in video streams for multimedia mining. *IAENG International journal of computer science*, 35(3), 16-21.
- Zhang, R., Ge, P., Zhou, X., Jiang, T., & Wang, R. (2013). An method for vehicle-flow detection and tracking in real-time based on Gaussian mixture distribution. *Advances in Mechanical Engineering*.

## **ANEXOS**

## ANEXO 1. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR ZHANG (Zhang et al., 2013)

```
%% CÓDIGO DEL ALGORITMO PROPUESTO POR ZHANG %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Autor: Diego A. Sócola E.
%% -----
%% Referencias:
%% Conteo y Etiquetado de objetos - (Diego, 2017)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear;
close all;

% Lectura de imágenes
original=imread('2.jpg');
vacio=imread('1.jpg');

% Conversión a escala de grises
A = rgb2gray(original);
B= rgb2gray(vacio);

figure;
subplot 221; imshow(original)
title('imagen original')
subplot 222;imshow(vacio)
title('Imagen vacia')
subplot 223;imshow(A)
title('Original escala de grises')
subplot 224;imshow(B)
title('Vacío escala de grises')

%Filtro de mediana
J1 = medfilt2(A);
J2 = medfilt2(B);

figure;
subplot 121; imshow(J1)
title('Imagen filtrada Original')
subplot 122;imshow(J2)
title('Imagen filtrada Vacío')

%resta de imágenes
Ip = imsubtract(J1,J2);
K = imabsdiff(J1,J2);

figure; imshow(Ip), title('Resta');

% Otsu thresholding
t1 = graythresh(Ip);
bw1 = im2bw(Ip,t1);
figure, imshow(bw1), title ('Imagen binaria');

%Supresión de ruido
h = 1/3*ones(3,1); %estructura 3x3 píxeles
H = h*h';
imfilt = filter2(H,bw1);
```

```

figure, imshow(imfilt), title ('Supresión de ruido');

%Erosión
se = strel('diamond', 10);
BWsdil = imerode(imfilt,se);

%Llenado de agujeros
BWdfill = imfill(BWsdil, 'holes');

figure;
subplot 121; imshow(BWsdil)
title('Erosion')
subplot 122;imshow(BWdfill)
title('Llenado de agujeros')

%Dilatación
se = strel('diamond', 10);
BWsdil2 = imdilate(BWdfill,se);

%Apertura binaria
sedisk = strel('disk',15);
bw = imopen(BWsdil2, sedisk);

figure;
subplot 121; imshow(BWsdil2)
title('Dilatación')
subplot 122;imshow(bw)
title('Apertura binaria')

%% Conteo y etiquetado de los objetos
[L, Ne]=bwlabel(bw);

% Calcular propiedades de los objetos de la imagen
propied= regionprops(L);
hold on

% Graficar las 'cajas' de frontera de los objetos
figure, imshow(original), title(['Total de objetos: ',num2str(Ne)])
for n=1:size(propied,1)
rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end

```

## ANEXO 2. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR BAGYAMANI (Bagyamani, 2015)

```
%% CÓDIGO DEL ALGORITMO PROPUESTO POR BAGYAMANI %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Autor: Diego A. Sócola E.
%% -----
%% Referencias:
%% Conteo y Etiquetado de objetos - (Diego, 2017)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear;
close all;

% Lectura de imágenes
original=imread('2.jpg');
vacio=imread('1.jpg');

% Conversión a escala de grises
A = rgb2gray(original);
B= rgb2gray(vacio);

figure;
subplot 221; imshow(original)
title('imagen original')
subplot 222;imshow(vacio)
title('Imagen vacía')
subplot 223;imshow(A)
title('Original escala de grises')
subplot 224;imshow(B)
title('Vacío escala de grises')

%resta de imágenes
Ip = imsubtract(A,B);
K = imabsdiff(A,B);

figure; imshow(K), title('Resta');

%Filtro de mediana
L = medfilt2(K);
figure, imshow(L), title('Imagen filtrada');

% Otsu thresholding
t1 = graythresh(L);
bw1 = im2bw(L,t1);
figure, imshow(bw1), title('Imagen binaria');

%Apertura binaria
se = strel('disk',13);
openbw = imopen(bw1, se);

%cierre binario
se1 = strel('disk',13);
BW = imclose(openbw,se1);
```

```

figure;
subplot 121; imshow(openbw)
title('Apertura binaria')
subplot 122; imshow(BW)
title('Cierre binaria')

%% Conteo y etiquetado de los objetos
[L, Ne]=bwlabel(BW);

% Calcular propiedades de los objetos de la imagen
prop=regionprops(L); hold on

% Graficar las 'cajas' de frontera de los objetos
figure, imshow(original), title(['Total de objetos: ', num2str(Ne)])
for n=1:length(prop)
    rectangle('Position', prop(n).BoundingBox, 'EdgeColor', 'g', 'LineWidth', 2)
end

% Conteo y etiquetado de los objetos
[L, Ne]=bwlabel(BW);

% Calcular propiedades de los objetos de la imagen
prop=regionprops(L);
hold on

% Graficar las 'cajas' de frontera de los objetos
figure, imshow(BW), title(['Total de objetos: ', num2str(Ne)])
for n=1:length(prop)
    rectangle('Position', prop(n).BoundingBox, 'EdgeColor', 'g', 'LineWidth', 2)
end
pause (1)

```

### ANEXO 3. CÓDIGO BASADO EN EL ALGORITMO PROPUESTO POR ABINAYA (Abinaya et al., 2013)

```
%% CÓDIGO DEL ALGORITMO PROPUESTO POR ABINAYA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Autor: Diego A. Sócola E.
%% -----
%% Referencias:
%% Segmentación "watersheed" - ("Marker-Controlled Watershed Segmentation -
MATLAB & Simulink Example," 2017)
%% Conteo y Etiquetado de objetos - (Diego, 2017)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear;
close all;

% Lectura de imágenes
original=imread('2.jpg');
vacio=imread('1.jpg');

% Conversión a escala de grises
A = rgb2gray(original);
B= rgb2gray(vacio);

figure;
subplot 221; imshow(original)
title('imagen original')
subplot 222;imshow(vacio)
title('Imagen vacia')
subplot 223;imshow(A)
title('Original escala de grises')
subplot 224;imshow(B)
title('Vacio escala de grises')

% resta de imágenes
Ip = imsubtract(A,B);
figure, imshow(Ip), title ('Resta');

%% SEGMENTACIÓN "WATERSHEED"

%Utilizar la magnitud del gradiente como la función de segmentación
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ip), hy, 'replicate');
Ix = imfilter(double(Ip), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);
figure, imshow(gradmag,[]), title('Magnitud del gradiente (gradmag)')

% Marcar los objetos en primer plano
se = strel('disk', 20);
Io = imopen(Ip, se);

Ie = imerode(Ip, se);
Iobr = imreconstruct(Ie, Ip);

Ioc = imclose(Io, se);
```

```

figure;
subplot 121; imshow(Io)
title('Apertura (Io)')
subplot 122; imshow(Ioc)
title('Apertura-cierre (Ioc)')

Iobrd = imdilate(Iobr, se);
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));
Iobrcbr = imcomplement(Iobrcbr);
figure, imshow(Iobrcbr), title('Apertura-cierre por reconstrucción
(Iobrcbr)')

fgm = imregionalmax(Iobrcbr);
figure, imshow(fgm), title('Regional máxima de apertura-cierre por
reconstrucción (fgm)')

I2 = Ip;
I2(fgm) = 255;
figure, imshow(I2), title('Regional máxima superpuesta sobre la imagen
original (I2)')

se2 = strel(ones(5,5));
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);

fgm4 = bwareaopen(fgm3, 20);
I3 = Ip;
I3(fgm4) = 255;
figure, imshow(I3)
title('Regional máxima modificada superpuesta sobre la imagen original
(fgm4)')

% Calcular marcadores en el primer plano

bw = im2bw(Iobrcbr, graythresh(Iobrcbr));
figure, imshow(bw), title('Thresholded apertura-cierre por reconstrucción
(bw)')

D = bwdist(bw);
DL = watershed(D);
bgm = DL == 0;
figure, imshow(bgm), title('Watershed ridge lines (bgm)')

% Calcular la transformada Watershed de la función de segmentación
gradmag2 = imimposemin(gradmag, bgm | fgm4);
L = watershed(gradmag2);

%Visualizar el resultado
I4 = Ip;
I4(imdilate(L == 0, ones(3, 3)) | bgm | fgm4) = 255;
figure, imshow(I4)
title('Marcadores y bordes del objeto superpuesto sobre la imagen original
(I4)')

Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
figure, imshow(Lrgb)
title('Matriz de etiquetas Watershed coloreadas (Lrgb)')

```

```

figure, imshow(Ip), hold on
himage = imshow(Lrgb);
set(himage, 'AlphaData', 0.3);
title('LRGB superpuesta transparente en la imagen original')

%% Conteo y etiquetado de los objetos
[L, Ne]=bwlabel(bw);

% Calcular propiedades de los objetos de la imagen
prop=regionprops(L);
hold on
% Graficar las 'cajas' de frontera de los objetos
figure, imshow(original), title(['Total de objetos: ', num2str(Ne)])
for n=1:length(prop)
    rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
    %x=prop(n).Centroid(1);
    %y=prop(n).Centroid(2);
    %plot(x,y,'*')
    % text(x,y,'O')
end

```

## ANEXO 4. CÓDIGO DEL ALGORITMO MAT

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CÓDIGO DEL ALGORITMO MAT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Autor: Diego A. Sócola E.
%% -----
%% Referencias:
%% Segmentación "watersheed" - ("Marker-Controlled Watershed Segmentation -
MATLAB & Simulink Example," 2017)
%% Conteo y Etiquetado de objetos - (Diego, 2017)
%% Búsqueda y eliminación de áreas menores - (Diego, 2017)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear;
close all;

% Lectura de imágenes
original=imread('2.jpg');
vacio=imread('1.jpg');

% Conversión a escala de grises
A = rgb2gray(original);
B= rgb2gray(vacio);

figure;
subplot 221; imshow(original)
title('imagen original')
subplot 222;imshow(vacio)
title('Imagen vacia')
subplot 223;imshow(A)
title('Original escala de grises')
subplot 224;imshow(B)
title('Vacío escala de grises')

%Ajuste de intensidad
M1 = imadjust(A);
M2 = imadjust(B);

figure;
subplot 121; imshow(M1)
title('Ajuste de intensidad en original')
subplot 122;imshow(M2)
title('Ajuste de intensidad en vacío')

% resta de imágenes
Ip = imsubtract(M1,M2);
figure, imshow(Ip), title ('Diferencia entre Imágenes');

%% SEGMENTACIÓN "WATERSHEED"

%Utilizar la magnitud del gradiente como la función de segmentación
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ip), hy, 'replicate');
Ix = imfilter(double(Ip), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);
figure, imshow(gradmag,[]), title('Magnitud del gradiente (gradmag)')
```

```

% Marcar los objetos en primer plano
se = strel('disk',15);
Io = imopen(Ip, se);

Ie = imerode(Ip, se);
Iobr = imreconstruct(Ie, Ip);

Ioc = imclose(Io, se);

subplot 121; imshow(Io)
title('Apertura')
subplot 122; imshow(Ioc)
title('Apertura-cierre')

Iobrd = imdilate(Ioc, se);
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Ioc));
Iobrcbr = imcomplement(Iobrcbr);
figure, imshow(Iobrcbr), title('Apertura-cierre por reconstruccion')

% Calcular marcadores en el primer plano
bw = im2bw(Iobrcbr, graythresh(Iobrcbr));
figure, imshow(bw), title('Thresholded apertura-cierre por reconstruccion')

%Cierre binario
sel = strel('disk', 30);
BW = imclose(bw, sel);
figure, imshow(BW), title('Cierre binario');

%% Conteo y etiquetado de los objetos
[L, Ne]=bwlabel(BW);

% Calcular propiedades de los objetos de la imagen
prop=regionprops(L);
hold on
% Graficar las 'cajas' de frontera de los objetos
figure, imshow(BW), title(['Total de objetos: ', num2str(Ne)])
for n=1:length(prop)
    rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
pause (1)

areas=[prop.Area]

%% Búsqueda y eliminación de áreas menores
s=find([prop.Area]<3200);

% Marcar áreas menores
for n=1:size(s,2)
    rectangle('Position',prop(s(n)).BoundingBox,'EdgeColor','r','LineWidth',2)
end
pause (2)

% Eliminar áreas menores
for n=1:size(s,2)
    d=round(prop(s(n)).BoundingBox);
    BW(d(2):d(2)+d(4),d(1):d(1)+d(3))=0;
end

```