

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Sistema para conteo y clasificación vehicular en tiempo real
implementado en una plataforma embebida ARM**

Artículo Académico

Liseth Coralía Pico Escobar

Ingeniería Electrónica

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniera en Electrónica

Quito, 22 de diciembre de 2017

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

**Sistema para conteo y clasificación vehicular en tiempo real implementado en
una plataforma embebida ARM**

Liseth Coralía Pico Escobar

Calificación:

Nombre del profesor, Título académico

Diego Benítez, Ph.D.

Firma del profesor

Quito, 21 de diciembre de 2017

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Lisseth Coralía Pico Escobar

Código: 00113730

Cédula de Identidad: 1804383162

Lugar y fecha: Quito, diciembre de 2017

RESUMEN

Uno de los problemas más importantes que el mundo y nuestro país debe resolver es la movilidad de personas y bienes, para ello existe la necesidad de sistemas de cuantificación vehicular confiables y versátiles como elemento crucial para los Sistemas de Transporte Inteligente. Es importante conocer datos sobre la densidad de tráfico vehicular en diferentes vías y autopistas para mejorar la gestión de la movilidad y el funcionamiento de las ciudades. El conteo y clasificación vehicular constituyen una tarea desafiante, debido a las diferentes condiciones de circulación, velocidad, clima, tiempo e iluminación. Se han propuesto numerosos trabajos sobre este tema empleando diferentes enfoques. Este documento propone la implementación de un sistema para identificar y clasificar vehículos en una plataforma ARM con sistema operativo Ubuntu compatible con una cámara de 720 P. Los algoritmos empleados se basaron en la librería de Open-CV de Intel y se usó el lenguaje de programación de Python. De las pruebas realizadas se pudo comprobar la eficiencia del algoritmo otorgando validez del proyecto y su aplicabilidad en el conteo vehicular.

Palabras clave: detección vehicular, Open CV, plataforma embebida ARM, Ubuntu, Python.

ABSTRACT

One of the most important problems that the world and our country must solve is mobility of people and goods. There is a necessity for reliable and versatile vehicle quantification systems as a crucial element for Intelligent Transportation Systems. It is a key to know the data of traffic flow in different roads and highways to improve mobility management and functioning of cities. Counting and classification of vehicles is a challenging task due to different conditions of circulation, speed, climate, time and lighting. Several works on this subject have been proposed using different approaches. This document proposes the implementation of a system to identify and classify vehicles in an ARM platform with Ubuntu operating system. The algorithms used were based on Intel's Open C-V library and the programming language used was Python. The efficiency of the algorithm was proved with different tests which granted its applicability.

Key words: vehicle detection, Open CV, ARM embedded platform, Ubuntu, Python.

TABLA DE CONTENIDO

Introducción.....	9
Investigaciones previas.....	9
Metodología.....	10
Experimentación y resultados	12
Conclusiones	14
Recomendaciones	14
Referencias	14

ÍNDICE DE TABLAS

Tabla I. Resultados evaluación de detección

Tabla II. Resultados evaluación de clasificación

ÍNDICE DE FIGURAS

Figura 1. Diagrama de flujo de la técnica empleada

Figura 2. Región de interés

Figura 3. Pre-procesamiento de imagen

Figura 4. Objetos detectados

Figura 5. Localización exacta del lugar de estudio

Figura 6. Conteo vehicular en pantalla

Figura 7. Archivo de texto de resultados

Sistema para conteo y clasificación vehicular en tiempo real implementada en una plataforma embebida ARM

Lisseth Pico

Universidad San Francisco de Quito

lisseth.pico@estud.usfq.edu.ec

Abstract—Uno de los problemas más importantes que el mundo y nuestro país debe resolver es la movilidad de personas y bienes, para ello existe la necesidad de sistemas de cuantificación vehicular confiables y versátiles como elemento crucial para los Sistemas de Transporte Inteligente. Es importante conocer datos sobre la densidad de tráfico vehicular en diferentes vías y autopistas para mejorar la gestión de la movilidad y el funcionamiento de las ciudades. El conteo y clasificación vehicular constituyen una tarea desafiante, debido a las diferentes condiciones de circulación, velocidad, clima, tiempo e iluminación. Se han propuesto numerosos trabajos sobre este tema empleando diferentes enfoques. Este documento propone la implementación de un sistema para identificar y clasificar con una plataforma embebida ARM ODROID XU-4 con sistema operativo Ubuntu. Los algoritmos empleados se basaron en la librería de OpenCV de Intel y se usó el lenguaje de programación de Python. De las pruebas realizadas se pudo comprobar que la eficiencia del algoritmo es del 93,48% pudiendo ser mejorado solamente con el hecho de incrementar la muestra, por lo tanto se comprueba la validez del proyecto y su aplicabilidad en el conteo vehicular.

Keywords—detección vehicular, Open CV, Odroid XU-4, Ubuntu, Python.

I. INTRODUCCIÓN

De acuerdo a datos del INEC 2016 en Ecuador la tasa promedio anual de crecimiento del parque automotor de los últimos 10 años es del 10,6%, este rápido incremento del parque automotor a nivel nacional significa 2.056.213 vehículos circulando en el País, por tanto la adquisición de datos de tráfico ha tomado un papel importante en los últimos años. Estos datos son esenciales en estudios de tráfico vehicular ya que con ellos se puede realizar una administración de tránsito mucho más acertada.

El conteo y clasificación de vehículos siempre ha sido un tema llamativo para los Sistemas de Transporte Inteligente (ITS) [12]. Se han desarrollado diferentes métodos para dichas actividades, estos se pueden clasificar en tres grandes grupos: métodos manuales, métodos basados en hardware y métodos basados en software.

El conteo manual suele ser el más sencillo de los tres; sin embargo, al emplear el recurso humano, termina siendo el más costoso y el que consume más tiempo y sobretodo con un alto error de medición. El conteo vehicular basado en hardware es simple pero con muchas limitaciones. Entre estas limitaciones tenemos que su instalación es intrusiva es decir que requiere

la reconstrucción de las carreteras en donde se los instala, son fijos y su costo es elevado.

El uso de software para el conteo y clasificación es una técnica mucho más avanzada y versátil, que básicamente está depende de un algoritmo computacional [12]. El uso de video cámaras emparejadas con técnicas computacionales de visión son una alternativa muy atractiva frente a los sensores comunes de contacto, y que poseen de acuerdo a este estudio una certeza mayor [4] y un precio bajo en el mercado. Usando videocámaras evadimos la necesidad de colocación de sensores y el control directo del ser humano [19]; al contrario de los dos métodos anteriores. Además, permite recolectar grandes cantidades de datos lo más aproximados a la realidad existente.

Este artículo se centra en la implementación de una técnica de clasificación y conteo vehicular que detecta los vehículos de una entrada de video en tiempo real, los cuenta y a la vez los clasifica, generando un archivo de texto con los resultados.

II. INVESTIGACIONES PREVIAS

La detección y seguimiento de objetos son dos actividades muy importantes para diversas aplicaciones como sistemas de vigilancia, sistemas de transporte e incluso para aplicaciones militares [19]. La identificación de vehículos es un proceso en el cual se detecta la presencia de vehículos en una secuencia de vídeo[7].

Los principales métodos de localización vehicular incluyen: detección de puntos, detección de bordes, diferenciación de fotogramas, umbralización y segmentación, pareo y finalmente métodos de flujo óptico [4]. Los métodos de detección de puntos son rápidos y ofrecen mejores resultados para cambios inesperados en la iluminación [10]. La detección de bordes también constituye un método rápido y mucho mejor frente a cambios de iluminación; sin embargo es sensible al ruido [17]. La diferenciación de fotogramas; por otro lado, requiere un fondo estático o una imagen de referencia, lo que no lo hace apto para emplearlo con vehículos en movimiento [14]. Los métodos de emparejamiento son comprensivos y proporcionan exactitud [2]. En los métodos de segmentación y extracción de características y pareo se toma un vídeo y se lo convierte en una secuencia de imágenes, se realiza la diferenciación y los píxeles que tienen el mismo valor son eliminados. Se registra el fondo y luego se realiza la detección de las imágenes del primer plano. Posteriormente se realiza la

segmentación de la imagen y finalmente se los compara con una imagen de "plantilla".

Se han realizado varios trabajos de investigación en cuanto al conteo vehicular a través de procesamiento de imágenes. Los cuales han tenido ventajas y limitaciones. Se puede mencionar a [11] que se centra en el mejoramiento de la imagen obtenida mediante un enfoque de Ecuación Adaptativa del Histograma (CLAHE). Esta técnica consta de los siguientes pasos básicos:

- Adquisición de imagen
- Pre procesamiento
- Modelado y segmentación del segundo plano
- Filtro y detección

Este método ha dado mejores resultados en imágenes reales de tráfico, incluyendo alta congestión, ruido, clúster, nieve y lluvia que contienen sombras proyectadas y oclusiones, tomadas tanto de las carreteras al aire libre como de las carreteras de la ciudad [17]. Las limitaciones de se centran en algoritmos débiles y costo elevado.

En otro estudio [12] se propone un algoritmo de transformación de características de escala invariantes (SIFT) para la detección y clasificación de vehículos. Esta técnica consta de las siguientes etapas:

- Detección binaria de objetos en movimiento
- Segmentación de la imagen
- Uso de SIFT para extracción de características
- Emparejamiento de características
- Conteo y detección vehicular

El empleo de la SIFT se han podido realizar la extracción de características invariantes de la imagen que son estables en cuanto a rotación, traslación, escalamiento y cambios en iluminación

Se empleó una transformación morfológica para la identificación y conteo de vehículos en [5] y [22], en donde se toma un vídeo y se lo convierte en una secuencia de imágenes. Para extraer las diferencias entre las imágenes, los píxeles que contienen el mismo valor son eliminados. Después de una actualización del segundo plano, se extraen las características importantes y objetos del primer plano. Se han usado operaciones morfológicas como dilatación y erosión para la segmentación de las imágenes. Finalmente se remueve el ruido de los dos planos de la imagen y se realiza el conteo. Este método ha sido efectivo en situaciones de oclusiones parciales y ruido en el segundo plano.

La investigación realizada propone un contador y clasificador basado en una mezcla de diferentes métodos de procesamiento de imágenes que incluyen detección de objetos, detección de bordes, diferenciación de fotogramas y filtros de Kalman de la siguiente manera:

- Pre procesamiento de imágenes
- Conversión de RGB a escala de grises

- Transformación de la ley potencial
- Filtro de Kalman
- Detección de vehículos

En muchas investigaciones como en [9] se ha empleado el Modelo Gaussiano Mixto para el modelado del segundo plano en la detección de vehículos. En este caso específico se ha seleccionado una región de interés (ROI), una transformada proyectiva y una matriz de homografía para la conversión de coordenadas. Para el análisis del primer plano se ha empleado un análisis con una estructura piramidal para la detección de objetos en movimiento.

Para la detección de vehículos existen algunos desafíos que enfrentar entre ellos:

- Rápida detección
- Posición adaptable del aparato
- Calibración sencilla
- Eficacia en el conteo y clasificación

El objetivo de este trabajo es implementar un sistema de detección vehicular de bajo costo empleado una cámara de baja resolución que permita obtener datos reales de conteo y clasificación de vehículos para análisis de sistemas de tránsito inteligentes.

III. METODOLOGÍA

Este trabajo consiste en la detección de presencia de vehículos en un ambiente en tiempo real empleando Open CV, el cual es una fuente abierta de software para la creación de aplicaciones relacionadas con visión computacional. Open CV está disponible para lenguajes de programación como C, C++, Java y Python. Para esta investigación se escogió el último de ellos. Al analizar imágenes en tiempo real se requirió un procesador muy eficiente. Se seleccionó ODROID XU4 el cual es un ordenador de monoplaca con tecnología ARM® big.LITTLE™ con una velocidad de procesamiento de 1.6GHz y una tarjeta de memoria eMMC 5.0 de 16Gb para mejor desempeño. El algoritmo se resume en las siguientes etapas de la figura (1).

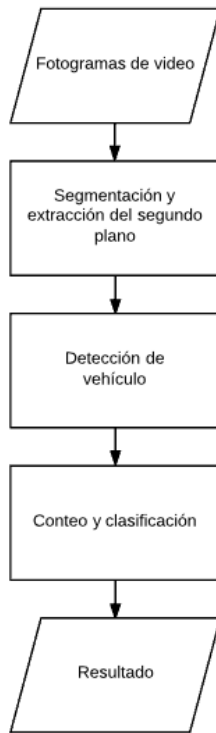


Fig. 1: Diagrama de flujo de la técnica empleada

A. Fotogramas de video

Para esta etapa, los datos de entrada son adquiridos mediante la cámara Odroid USB-CAM 720P de bajo costo. Posee algunas limitaciones que se pueden evitar empleando una cámara con mejor resolución, lo que aumentaría la calidad y el resultado del algoritmo.

B. Selección de la región de interés

En el algoritmo propuesto se debe especificar un área delimitada en la cual se contarán los vehículos que pasen por la misma. El ancho de esta sección debe ser menor que el ancho de los vehículos ya que así se evitan conteos falsos. En otras aplicaciones como en el registro de velocidad de vehículos, el ancho de esta sección debe ser más grande que la de los vehículos.



Fig. 2: Región de Interés

C. Preprocesamiento

En esta etapa se realizó la conversión a escala de grises para que sea mucho más manejable por el algoritmo y el sistema operativo.

1) *Modelo Gaussiano Mixto de extracción de fondo*: Con este modelo, cada uno de los píxeles es modelado de acuerdo a una distribución Gaussiana. Los píxeles son divididos por su intensidad en RGB, luego se calcula de acuerdo a su probabilidad si deben ser incluidos en el fondo FG o en el primer plano BG con la siguiente ecuación:

$$P(X_t) = \sum_{i=1}^N \omega_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

en donde

- X_t : Píxel actual en el marco t
- N : es el número de distribuciones en el modelo
- $\omega_{i,t}$: es el peso de la distribución N en el marco t
- $\Sigma_{i,t}$ es la desviación estándar de la distribución N
- $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ es la función de densidad de probabilidad y está dada por la siguiente ecuación:

$$\eta(X_t, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu) \Sigma^{-1} (X_t - \mu)} \quad (2)$$

En el método de [13] cada RGB no está relacionado con el otro. Por ello, se asume que la diferencia en intensidad posee una desviación estándar uniforme. La matriz de covarianza entonces está representada por:

$$\Sigma_{i,t} = \sigma_{i,t}^2 I \quad (3)$$

El fondo de la imagen está dado por cada uno de los píxeles que es mayor que el umbral designado; mientras que los píxeles menores constituyen el primer plano de la siguiente forma:

$$B = \operatorname{argmin}_b \left(\sum_{i=1}^b \omega_{i,t} > T \right) \quad (4)$$

Cuando el valor de los píxeles concuerda con alguno de los N valores Gaussianos, ω , μ y σ son actualizados

$$\omega_{i,t} + 1 = (1 - \alpha)\omega_{i,t} + \alpha \quad (5)$$

Mientras que cuando ninguno de los píxeles coincide con ninguno de los N Gaussianos, solo ω es actualizado.

$$\omega_{j,t+1} = (1 - \alpha)\omega_{j,t} \quad (6)$$

2) *Cascade Classifier*: El fondo de la imagen debe actualizarse en cada fotograma para así poder determinar la presencia de los vehículos en la región de interés. Se empleó el clasificador en cascada que contiene distintas etapas formadas por clasificadores más básicos. La estimación y actualización del fondo de la imagen se obtuvo por medio de:

$$B_{k+1} = \alpha I_{k+1} + (1 - \alpha)B_k \quad (7)$$

en dónde

- B_{k+1} e I_{k+1} indican el valor estimado del fondo de la imagen en el $(k + 1)$ ésimo fotograma.
- α controla la proporción de actualización

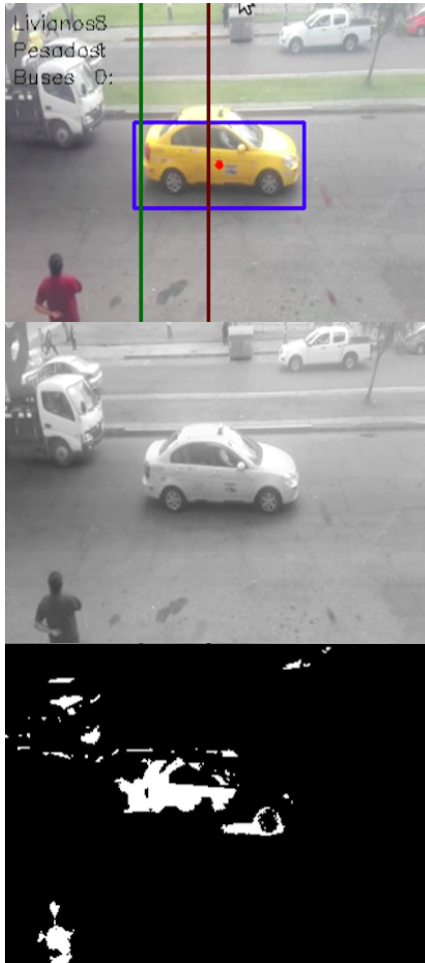


Fig. 3: Preprocesamiento de imagen

D. Detección de vehículos

Se empleó un algoritmo de detección que está basado en un cascade classifier para identificación de los patrones de vehículos. Estos clasificadores se entrenan con diferentes imágenes de un objeto en particular (en este caso diferentes imágenes laterales de vehículos) que constituyen ejemplos positivos y con imágenes arbitrarias del mismo tamaño las cuales son los ejemplos negativos. Con la finalidad de obtener un algoritmo robusto, se empleó la base de datos de la Universidad de Illinois Urbana-Champaign (UIUC) para detección de vehículos la cual posee:

- 1050 imágenes de entrenamiento
- 170 imágenes de prueba de una sola escala
- 108 imágenes de prueba a múltiples escalas

La detección se realiza por medio de una "ventana" que se mueve en el área de la imagen. Este clasificador está diseñado para que pueda cambiar de tamaño y así encontrar los objetivos

en varios tamaños. Los clasificadores se basan en el árbol de decisión con al menos 2 hojas.

Cuando se obtiene un resultado negativo, la ventana pasa al siguiente región de la imagen hasta obtener un resultado positivo. Si se obtiene un resultado positivo, la región de la imagen pasa a la siguiente etapa de clasificación. Al final, el clasificador indica la presencia de un vehículo en la imagen cuando en todas las etapas anteriores se obtuvieron resultados positivos.

Después de detectar el objeto, el algoritmo dibuja un rectángulo alrededor del vehículo para su mejor identificación visual como en la figura (4).

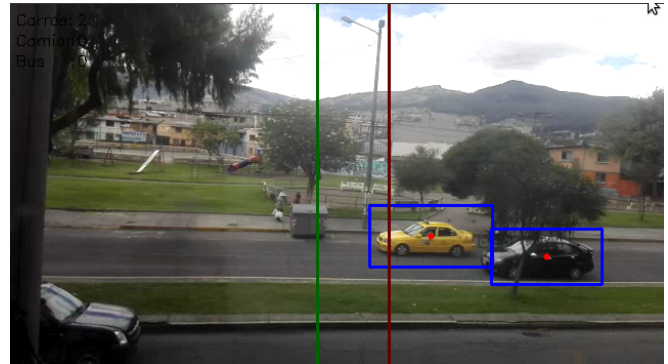


Fig. 4: Objetos detectados

IV. EXPERIMENTACIÓN Y RESULTADOS

Se realizaron varias pruebas en tiempo real en diferentes calles de la ciudad. La prueba más significativa se realizó en la intersección de salida de la Parroquia Benítez hacia el Cantón Pelileo, Provincia de Tungurahua figura (5), que por solicitud del GAD por el alto índice de accidentabilidad solicita la colocación de semáforos en dicha intersección, siendo necesario el conteo vehicular para identificar la mejor opción de seguridad vial

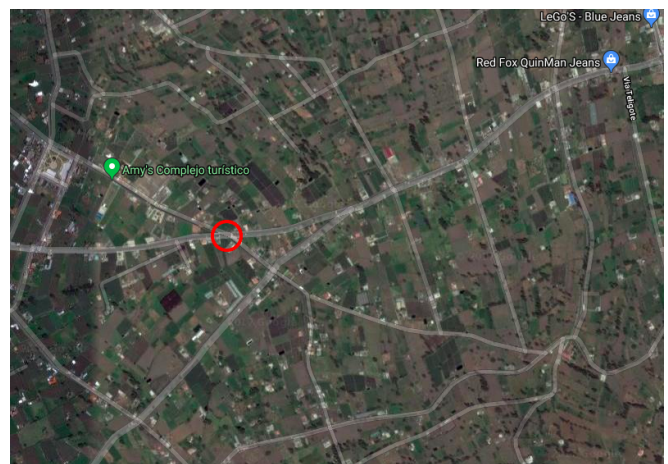


Fig. 5: Localización exacta del lugar de estudio

Se instaló el equipo a una distancia aproximada de 6 metros del borde de la vía a una altura aproximada de 1.70

metros desde el suelo. Se realizó una simple calibración de la cámara debido a las condiciones de iluminación del día y una calibración del algoritmo de acuerdo a las áreas detectadas para cada tipo de vehículo. Para estudios de semaforización y señalización se deben tomar datos de flujo vehicular durante una tasa horaria equivalente (vehículos/hora) que sea menor a una hora. Para este estudio se recogió datos en los puntos ya mencionados durante intervalos de 15 minutos cada media hora desde las 8:00am hasta las 12:15pm dando como resultado 9 mediciones. Se puede observar el funcionamiento del algoritmo en la figura(6)



Fig. 6: Conteo Vehicular en pantalla

El análisis del proceso experimental se dividió en dos etapas: conteo de vehículos en general (prueba de detección) y exactitud en la clasificación.

1) *Evaluación de detección*: Para este caso se compararon los datos del conteo manual del flujo vehicular de las 9 mediciones con los datos obtenidos del algoritmo y se obtuvo la siguiente tabla de resultados:

TABLE I: Resultados evaluación de detección

Prueba	Conteo manual	Conteo automático	Exactitud
1	270	266	98.51%
2	225	219	97.33%
3	192	196	97.91%
4	140	143	97.85%
5	190	186	97.89%
6	143	139	97.20%
7	135	128	94.81%
8	153	150	98.03%
9	155	148	95.48%

De esta prueba se obtuvo un error de aproximadamente 2,78%. Los resultados de esta evaluación muestran que la mayoría de errores ocurre cuando dos autos con colores y tamaños similares se mueven uno cerca del otro. Para superar estos obstáculos se pueden emplear diferentes métodos como el de [21]. Otro punto importante a considerar es la detección de objetos que no necesariamente eran vehículos; es decir que hay que tomar en cuenta que existieron falsos positivos.

```

archivo.txt (~/Desktop/Tesis) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
archivo.txt
260 Buses: 11
261
262 -----Informe de visualizacion-----
263 Inicio:
264 15/12/17---10:40:30
265 Fin:
266 15/12/17---10:55:40
267
268 Carros: 121
269 Camiones: 22
270 Buses: 1
271
272 -----Informe de visualizacion-----
273 Inicio:
274 11/02/16---11:01:05
275 Fin:
276 15/12/17---11:15:32
277
278 Carros: 109
279 Camiones: 16
280 Buses: 3
281
282 -----Informe de visualizacion-----
  
```

Fig. 7: Archivo de texto de resultados

2) *Evaluación de clasificación* : Como segunda parte, se realizó una prueba de exactitud del algoritmo en la clasificación de los automotores. Se empleó la siguiente matriz para visualizar el rendimiento de la técnica de clasificación.

TABLE II: Resultados evaluación de clasificación

Tipo de vehículo	Número real	Número detectado	Exactitud
Livianos	1241	1207	97.26%
Pesados	178	180	99.87%
Buses	23	25	91.30%

De estos análisis se obtienen que el error promedio para la clasificación es de aproximadamente 6.52%. Para estudios de tránsito se admite un error de 5% en el conteo manual. Si bien el error es mayor al 5% está dentro de un rango cercano.

Para disminuir el error y aumentar la exactitud del prototipo se debe aumentar la muestra. Cabe destacar que el mayor

porcentaje de error proviene de las categorías de Pesados y Buses ya que estos vehículos son más escasos en las vías; por ello, una pequeña desviación en la clasificación causa un error muy grande.

V. CONCLUSIONES

En esta investigación se ha implementado un pequeño prototipo para el conteo y clasificación vehicular basado en un algoritmos de procesamiento de imágenes. Este método propone una alternativa viable y de bajo costo, ya que se basa en el uso de software libre y además ofrece calidad y exactitud en los resultados.

La eficiencia general del contador vehicular es del 95.35% que puede disminuir por el incremento de la muestra.

La versatilidad del aparato es importante, ya que tiene un tamaño muy pequeño y además la calibración del algoritmo de acuerdo al lugar de colocación del mismo no es complicado.

Los resultados de las pruebas realizadas arrojaron un porcentaje de error dentro de los valores aceptables para estudios de Sistemas de Transporte Inteligente.

Es indispensable en la implementación de infraestructura y equipamiento vial tener datos reales de flujos vehiculares para optar por la mejor opción de seguridad vial.

Para las instituciones públicas que manejan el tránsito a nivel nacional se requiere de manera urgente contadores vehiculares de este tipo para tomar buenas decisiones de infraestructura equipamiento vial.

VI. RECOMENDACIONES

Si bien los valores de exactitud del prototipo son buenos, se podría mejorar mediante el uso de una cámara más avanzada, para lo cual se requeriría la adquisición de drivers, aumentando el costo total del proyecto.

Para disminuir el error del experimento sería necesario incrementar la muestra, lo que arrojaría resultados inferiores al error aceptable.

Se puede mejorar el algoritmo para que la clasificación sea mucho más precisa.

De las pruebas iniciales existe interés por la adquisición del contador vehicular, pudiendo empezar la producción y comercialización del mismo a escala nacional.

REFERENCES

- [1] S. Agarwal and D. Roth "Learning a Sparse Representation for Object Detection," Proceedings of the Seventh European Conference on Computer Vision, 2002.
- [2] A. Bensrhair et al "A Cooperative Approach to Vision-BAsed Vehicle Detection," IEEE Intelligent Transportation Systems, pp. 207-212, 2001.
- [3] K. Chang and K. Wang "Design and Implementation of Traffic Safety Guardian System for Android Based on OpenCV," 2012 International Conference on Connected Vehicles and Expo (ICCVE), pp. 288-289, 2012.
- [4] N. Chintalacheruvu and V. Muthukumar, "Video Based Vehicle Detection and its Application in Intelligent Transportation Systems," Journal of Transportation Technologies, vol. 2, pp. 305-314, 2012.
- [5] R. Cucchiara, M. Piccardi and P. Mello "Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Transactions on Intelligent Transportation Systems, vol.1, pp. 758-763, 2000.
- [6] P. Daigavane and P. Bajaj, "Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways," IJCSNS International Journal of Computer Science and Network Security, vol. 10 , pp. 112-117, 2010.
- [7] M. Jalalat, M. Nejati and A. Majadi "Vehicle detection and speed estimation using cascade classifier and sub-pixel stereo matching," 2016 International Conference of Signal Processing and Intelligent Systems (ICSPIS), pp. 1-5, 2016.
- [8] H. Jang, I. Won and D. Jeong "Automatic Vehicle Detection and Counting Algorithm," IJCSNS International Journal of Computer Science and Network Security, vol.14 no.9, pp. 99-102, 2014.
- [9] P. Kaewtrakulpong and R. Bowden "An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection," 2001 2nd European Workshop on Advanced Video Based Surveillance Systems, pp. 1-5, 2001.
- [10] D. Lowe "Distinctive Image Features from Scale Invariant Keypoints," International Journal of Computer Vision, vol. 60, pp. 91-110, 2004.
- [11] E. Mosabbeh, M. Sadegi and M. Fathe "A low-cost strong shadow-based segmentation approach for vehicle tracking in congested traffic scenes," International Conference on Machine Vision ICMV, 2007.
- [12] M. Narhe and M. Nagmode, "Vehicle Counting using Video Image Processing," International Journal of Computing and Technology (IJCAT), vol. 1, pp. 358-362, 2014.
- [13] A. Nurhadiyatna et al. "Background Subtraction Using Gaussian Mixture Model Enhanced by Hole Filling Algorithm (GMMHF)," 2013 IEEE International Conference on Systems, Man and Cybernetics (SMC) , pp 4006-4011, 2013.
- [14] N. Paragios and R. Deriche "Geodesic Active Contours and Level Sets for Detection and Tracking of Moving Objects," 2013 IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp 266-280, 2000
- [15] D. Sarkar , A. Chowdhury., "A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection," International Journal of Engineering Trends and Technology , vol 10, pp 461-465, 2014.
- [16] J. Seth et al, "Advanced license plate recognition system for car parking," 2014 International Conference on Embedded Systems (ICES), pp. 162-165, 2014.
- [17] T. Sridevi, K. Harinath and P. Swapna, "Automatic Generation of Traffic Signal Based on Traffic Volume," 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 423-428, 2017.
- [18] N. Srinivasa, "A Vision-Based Vehicle Detection and Tracking Method for Forward Collision Warning," IEEE Intelligent Vehicle Symposium, pp. 626-631, 2002.
- [19] A. Tourani and A. Shahbahrami "Vehicle counting method based on digital image processing algorithms," 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), 2015.
- [20] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001., vol. 1, pp I-511,I518, 2001
- [21] D. Zhao, Y. Chen and L. Lv "Deep Reinforcement Learning With Visual Attention for Vehicle Classification," IEEE Transactions on Cognitive and Developmental Systems, vol. 9, pp. 356-366, 2017.
- [22] J. Zhou, D Gao and D. Zhang, "Moving Vehicle Detection for Automatic Traffic Monitoring," IEEE Transactions on Vehicular Technology, vol. 56, pp. 51-59, 2007.