

Algoritmo de programación estructurada, enfocado a la detección y conteo vehicular de manera inteligente en una intersección.



Structured programming algorithm, focused on the detection and intelligent counting of vehicles at an intersection

Edwin Fernando Mejía Peñafiel. ¹

Recibido: 13-05-2021 / Revisado: 24-05-2021 / Aceptado: 17-06-2021/ Publicado: 05-07-2021

Abstract.

DOI: <https://doi.org/10.33262/concienciadigital.v4i3.1776>

Introduction. The use of Matlab is approached through the creation of an algorithm with structured programming, implementing a software that determines the presence and classification of vehicles that pass through a road through a camera, considering it a traffic problem at peak hours. **Objective.** To propose an algorithm coded in Matlab that allows to recognize through a video the different cars on a road according to their dimensions. **Methodology.** The implemented methodology is a hybrid between software development and intelligent systems development. 8 tests were carried out to establish if the algorithm presents us with the expected results in the recognition of different cars, using tools and functions that come with Matlab. **Results.** The applied algorithm gives a margin of error of plus minus 8%, but to reach this it had to go from an error in the first test of 80% to 7.5% of it, since it is still necessary to make some adjustments in the performance of the algorithm with respect to the dimensions of the vehicles, especially when we have more of them and of different types. **Conclusion.** The importance is that based on this intelligent application, you can process videos that are captured from a camera at the intersection of the roads, with this you can obtain vehicular flow with up to 92%

¹ Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Carrera de Ingeniería Ambiental, Chimborazo – Riobamba, efmejia@epoch.edu.ec, msmejiaedwinf@yahoo.com, <https://orcid.org/0000-0001-6888-4621>

effectiveness, classification of vehicles daily and at peak hours . I consider that it is a very useful tool so that the problem of vehicular flow has a solution.

Keywords: Software development, Programming, Structured programming, artificial intelligence, pattern recognition.

Resumen.

Introducción. Se aborda el uso de Matlab a través de la creación de un algoritmo con programación estructurada, implementando un software que determine la presencia y clasificación de vehículos que pasan por una vía a través de una cámara, considerándose un problema de tráfico en las horas pico. **Objetivo.** Plantear un algoritmo codificado en Matlab que permita reconocer a través de un video los diferentes autos en una vía de acuerdo a sus dimensiones. **Metodología.** La metodología implementada es una híbrida entre desarrollo de software y desarrollo de sistemas inteligentes. Se realizaron 8 pruebas para poder establecer si el algoritmo nos presenta los resultados esperados en el reconocimiento de diferentes autos, utilizando herramientas y funciones que vienen con Matlab. **Resultados.** El algoritmo aplicado da un margen de error de más menos 8%, pero para llegar a esto se tuvo que ir de un error en la primera prueba del 80% hasta llegar al 7,5% del mismo, ya que todavía toca hacer algunos ajustes en el desempeño del algoritmo respecto a dimensiones de los vehículos, especialmente cuando tenemos mayor cantidad de ellos y de diferente tipo. **Conclusión.** La importancia radica que basado en esta aplicación inteligente, se puede processar vídeos que se captan de una cámara en la intersección de las vías, con esto se puede obtener flujo vehicular con hasta un 92% de efectividad, clasificación de vehículos diariamente y en horas pico. Considero que es una herramienta muy útil para que el problema del flujo vehicular tenga solución.

Palabras claves: Desarrollo de software, Programación, Programación estructurada, inteligencia artificial, reconocimiento de patrones.

Introducción.

En este mundo interactuamos a diario con los semáforos y vemos cada día como la cantidad de vehículos en las calles sigue en aumento, con su problema de excesivo tráfico en las ciudades de todo el mundo, con su correspondiente situación de que a veces solo en una vía está llena de autos y en la otra no, entonces nace la pregunta, ¿Por qué no implementar un software que controle ese tráfico?.

En las últimas décadas ha cobrado fuerza las hipótesis de que porque no se construyen sistemas inteligentes que ayuden de alguna manera a reorganizar el tráfico en las ciudades de todo el mundo no solo de Ecuador.

Existen innumerables formas y maneras de construir sistemas de este tipo, que estén íntimamente ligados a la inteligencia artificial. Se trata de algoritmos que permiten

programar de manera inteligente, basados en una técnica de inteligencia artificial que son las redes neuronales y dentro de ellas la visión artificial que permite el reconocimiento de imágenes o lo que es lo mismo el reconocimiento de patrones (Gualdron, A. y otros, 2013) que permitan de alguna manera mitigar este problema de congestión en las vías de la ciudad de Riobamba, provincia de Chimborazo, Ecuador.

La visión artificial por computadora es una disciplina que ha ido creciendo a pasos agigantados en los últimos años, creando máquinas autónomas capaces de interactuar de forma inteligente con el entorno. Para poder conseguir de una manera adecuada que las máquinas interactúen con su entorno, es necesario que se desarrolle un sistema que tenga la capacidad de percibirlo e interpretarlo (Branch & Olague, 2001).

La Visión Artificial como tal, dentro de la Inteligencia Artificial se considera como el conjunto de todas aquellas técnicas y modelos que nos permiten el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida a través de imágenes digitales (Pajares Martinsanz & Santos Peñas, 2006).

La visión artificial dentro su ámbito, utiliza sensores ópticos, como es las cámaras fotográficas y de video, con el propósito de adquirir fotografías o imágenes, para luego procesarlas mediante cierto procesador, todo esto con el fin de extraer y analizar ciertas cualidades o tipo de propiedades de la imagen capturada (Múnera Salazar & Manzano Herrera, 2012). De lo anterior, podemos concluir que la visión artificial en su afán de analizar e interpretar imágenes de una manera igual a como lo hace el ser humano, hace uso de computadoras y cámaras para adquirir dichas imágenes.

Matlab tiene algunas herramientas para el reconocimiento de objetos. El reconocimiento de objetos es una técnica de visión artificial para identificar objetos en imágenes o vídeos. El reconocimiento de objetos constituye una salida clave de los algoritmos de deep learning y machine learning. Cuando las personas miramos una fotografía o vemos un vídeo, detectamos con rapidez personas, objetos, lugares y detalles visuales. El objetivo es enseñar a un ordenador a hacer lo que resulta natural para los humanos: adquirir cierto nivel de comprensión del contenido de una imagen. (Mathworks, 2020)

La herramienta, Computer Vision System Toolbox™ proporciona algoritmos, funciones y apps para el diseño y la realización de pruebas de sistemas de procesamiento de vídeo, visión artificial y visión 3D. Es posible llevar a cabo la detección y el seguimiento de objetos, así como la detección, extracción y coincidencia de características. En el caso de la visión 3D, la toolbox soporta la calibración de cámaras simples, estéreo y ojo de pez, la visión en estéreo, la reconstrucción 3D y el procesamiento de nubes de puntos lidar y 3D. Las apps de visión artificial automatizan los flujos de trabajo de etiquetado de validación (ground-truth) y de calibración de cámaras. (Mathworks, 2020)

El objetivo principal de este artículo consiste en la detección automática de objetos, en este caso de diferentes tipos de autos con el uso de algoritmos usando programación estructurada con funciones que tiene Matlab para implementar un software que determine la presencia y clasificación de vehículos que pasan por una vía a través de una cámara,

esta investigación plantea un algoritmo muy sencillo para programar con estudiantes de Ingeniería, y así poder reconocer los diferentes autos de acuerdo a sus dimensiones. La metodología con la cual se va a implementar es una híbrida entre desarrollo de software y desarrollo de sistemas inteligentes. La importancia radica que basado en esta aplicación inteligente, el programa va a poder procesar videos captados por una cámara instalada en una intersección de las vías de la ciudad de Riobamba, los cuales son enviados a un computador a través de WiFi o cable, con esto se va a poder obtener el flujo vehicular, la clasificación de vehículos que diariamente pasan por ahí y cuáles son las horas pico.

Debemos indicar que la captura de imágenes sigue siendo un reto y una fuente muy interesante de oportunidades para el desarrollo de investigaciones y estudios dentro de la programación. El presente trabajo, busca contribuir con el crecimiento en este campo, de nuestros estudiantes y lectores, presentando de una forma muy sencilla el código que permita reconocer una gran cantidad de autos, utilizando técnicas existentes y algoritmos propios de la visión artificial que nos provee la herramienta MATLAB.

Metodología.

Se emplea una metodología de avances por fases que se presenta en la figura 1:

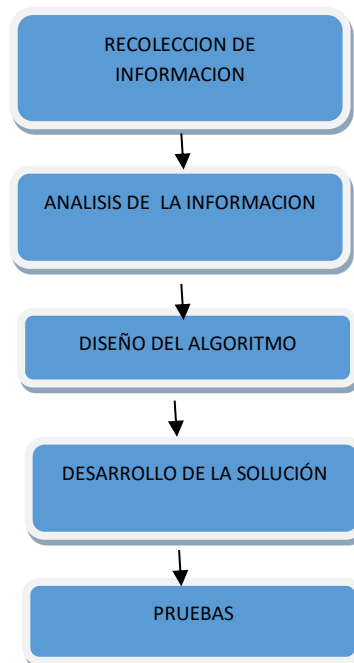


Figura 1. Metodología de avances por fases

Fuente. Elaboración propia

Proyecto propuesto

Se pretende dar una solución para contar vehículos con programación estructurada. El uso de un algoritmo de este tipo puede ser genial al momento de ejecutarlo en un ambiente

real y más aún si los proyectos tienen la adaptabilidad a procesar por ellos mismos los diferentes cambios que de acuerdo al número de autos que se encuentren puedan resolverlo por sí solos.

Infraestructura del proyecto – Para este proyecto utilizaremos el algoritmo implementado por el autor de una manera sistemática y bajo las condiciones que necesitamos para el semáforo inteligente que se irá implementando luego de terminado esto. Codificamos las líneas de código en el lenguaje M que es de diferente formato a los otros lenguajes de programación que tenemos como java y c++.

La idea principal de este proyecto es dar solución al tráfico en las vías y para ello se implementa un algoritmo y código con la herramienta Matlab, donde a través de varias funciones con el uso de visión artificial dentro de la misma, permiten hacer el desarrollo de este proyecto entendible para nuestros autores (Sisalima, F., 2018). La implementación del algoritmo se basa en 8 pruebas las cuáles me permitirán ir corrigiendo sobre la marcha cada una de las líneas de código realizadas y que además se basan en las respuestas de contar el número de autos, basados en la realidad que tiene la intersección donde se realiza el estudio. Cada uno de los videos tomados me permitirán obtener diferentes respuestas y que cada una de las mismas me llevará al óptimo funcionamiento del algoritmo para ponerlo en funcionamiento.

Nuestro estudio se realiza en la ciudad de Riobamba en las calles horizontal Avenida La Prensa y vertical Avenida Lizarzaburu y calle Veloz, con los 4 semáforos que se muestran en la figura 2 en las diferentes intersecciones.

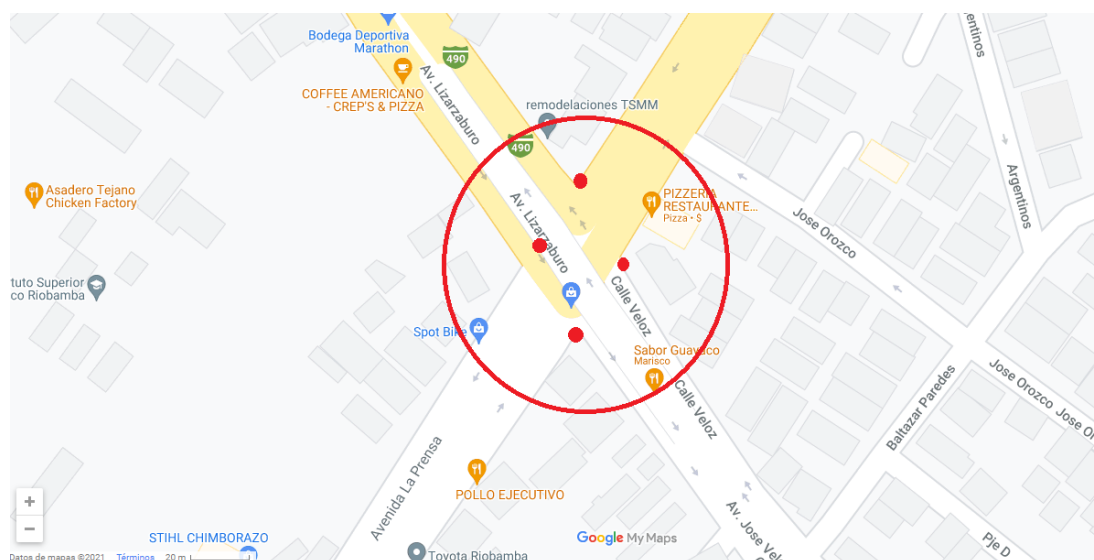


Figura 2. Intersecciones de calles de la ciudad de Riobamba y ubicación de semáforos.
Fuente. Elaboración propia

Resultados.

Rankin de los lenguajes de programación más usados

En la tabla 1 se muestra el rankin de los lenguajes de programación más usados en el año 2018 – 2019: <https://www.tiobe.com/tiobe-index/> (tiobe, 2019) [12]

Tabla 1. Rankin de los lenguajes de programación año 2018 – 2019

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	14.243%	+0.24%
3	3		C++	8.095%	+0.43%
4	4		Python	7.830%	+2.64%
5	6	▲	Visual Basic .NET	5.193%	+1.07%
6	5	▼	C#	3.984%	-0.42%
7	8	▲	JavaScript	2.690%	-0.23%
8	9	▲	SQL	2.555%	+0.57%
9	7	▼	PHP	2.489%	-0.83%
10	13	▲	Assembly language	1.816%	+0.82%
11	15	▲	Objective-C	1.626%	+0.69%
12	12		Delphi/Object Pascal	1.406%	+0.39%
13	18	▲	Perl	1.394%	+0.48%
14	16	▲	MATLAB	1.366%	+0.44%
15	10	▼	Ruby	1.343%	+0.16%
16	17	▲	Visual Basic	1.317%	+0.40%
17	91	▲	Groovy	1.173%	+1.06%
18	19	▲	Swift	1.150%	+0.24%

Fuente: índice TIOBE 2019

Como vemos en la tabla 1, se listan los 15 primeros lenguajes de programación más usados en el 2018 (codinghub, 2018) <https://codinghub.es/lenguajes-de-programacion-mas-usados/> [12], ahora en la figura 3 tenemos a Matlab como uno de los lenguajes de programación para visión artificial desde hace años:

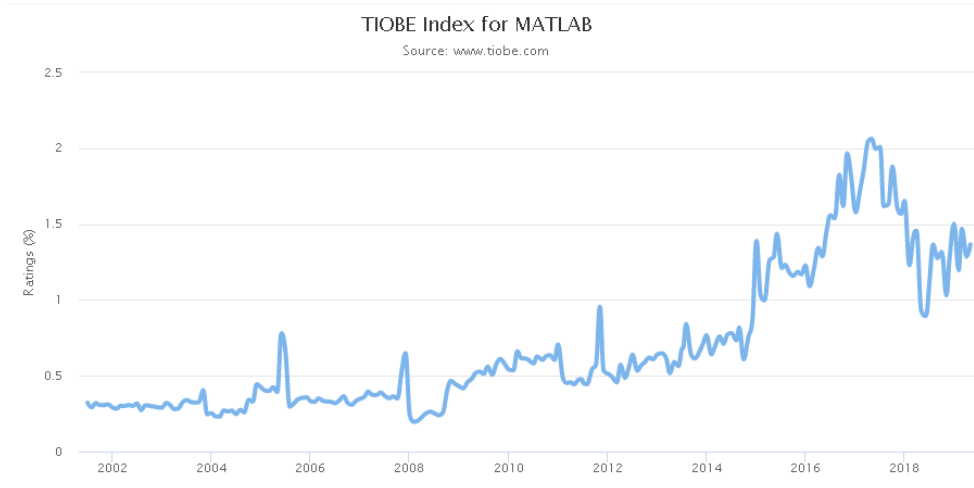


Figura 3. Estadística del lenguaje Matlab desde 2002 hasta 2018

Fuente: índice TIOBE 2019

En la figura 3 se muestra como el lenguaje Matlab escogido para este proyecto ha venido siendo uno de los mejores durante varios años hasta la actualidad. The Importance Of Being Earnest (TIOBE) comprueba más de 502 millones de líneas de código de software para sus clientes en todo el mundo, en tiempo real, cada día.

3.2. Características del proyecto

Nuestro sistema tiene las siguientes características mostradas en la Tabla 2:

Tabla 2. Características del proyecto

SISTEMA PARA CONTAR AUTOS	
CARACTERISTICAS	DESEMPEÑO
Tipo de inteligencia	El reconocimiento de patrones lo hace como las mismas personas
Capacidad sensorial	Recibe comunicaciones del entorno mediante del empleo de una cámara
Objetivo	Contar autos en una vía principal
Reglas de actuación	Interpreta las imágenes adquiridas dentro del sistema y las compara con los tipos de autos
Sistematización	Correlación entre los subsistemas que tiene el proyecto para permitir el mejor desempeño
Robustez	Reglas de inferencias bien definidas

Fuente: Elaboración propia

Reconocimiento de patrones

En el caso del prototipo resultó bastante eficiente el algoritmo planteado, ya que como contamos con una cámara de alta definición, entonces funcionó correctamente.

El sistema de reconocimiento tiene la capacidad de determinar la presencia de los diferentes tipos de autos en las dos áreas que se definieron con el propósito de extraer la información que permita contar cuantos autos pasan por esa vía.

Fases del reconocimiento de patrones

Con el propósito de emplear reconocimiento de patrones se tiene a bien hacer uso de una cámara web para obtener imágenes dentro de un video de la realidad en la intersección de la vía. Con esta información se envía para poder realizar una comparación de las características de cada imagen de autos, en donde se deben extraer del video las mismas, pero eliminando datos que nada tienen que hacer con el estudio o redundantes. En la figura 4 se muestra las fases por las cuales deben pasar un reconocimiento de este tipo:

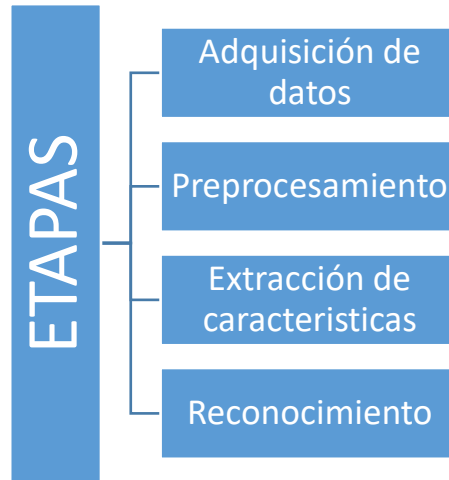


Figura 4. Etapas del reconocimiento a patrones
Fuente: Elaboración propia

Elementos del proyecto propuesto

Para el proyecto se utilizó todos los elementos que se observan en la figura 5, donde se muestra el proceso mismo del sistema en mención

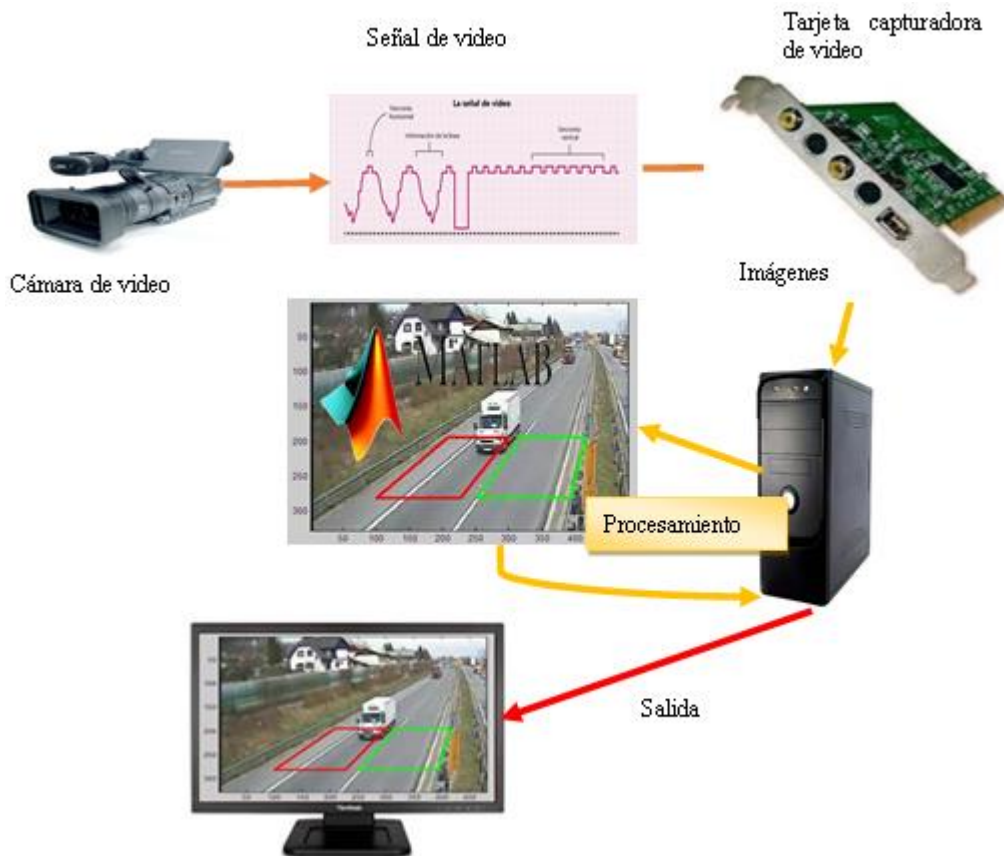


Figura 5. Elementos del proyecto propuesto
Fuente: Elaboración propia

Diseño y representación del proyecto

En la figura 5 se observa como es el funcionamiento del proyecto propuesto y en donde también se enmarca los procesos por donde seguirá la transformación de las imágenes del video capturado y bajo las reglas de inferencia permitidas dentro del mismo.

Siendo una etapa esencial el reconocimiento de patrones dentro de la inteligencia artificial y visión artificial específicamente, esto implica que se debe contar con imágenes con el menor ruido posible para poder detectar a los autos que pasan en la vía y así poder evitar errores en la detección. (Lopez, G., 2016)

El uso de filtros ayuda mucho en este aspecto, pero si las imágenes que nos llegan de la cámara son de mala calidad nuestro sistema podrá tener errores que en un próximo artículo estaremos publicando. (Toledano, L., 2019)

Como los cuadros de entrada son monoculares (tomados de una sola cámara), el método de calibración más simple consiste en seleccionar cuatro puntos en la vista en perspectiva y asignarlos a las esquinas de un rectángulo a vista de pájaro. Esto, para el sistema, significará que cada auto está parado en el mismo plano. (Meza, D., 2020)

Algoritmo Usado

El algoritmo con el que se ejecutó el prototipo tiene la forma siguiente:

- **Inicio**
 - Captación del Video
 - Visualización de imagen
 - Configurando la posición a través de líneas de conteo
 - Configura la fuente del video
 - Se puede detectar algunos vehículos?
 - Sisi: Rastrear vehículos dentro de la imagen
 - Contar vehículos de acuerdo a la línea de conteo
 - Se genera el reporte de acuerdo con el conteo vehicular
 - Se debe exportar el reporte de este conteo
 - Actualizar y presentar resultados en la vista preliminar del conteo de los vehículos captados
- **Fin**

Este algoritmo se traslada con la siguientes líneas de código:

```
clear all, close all,  
  
trafficVid = VideoReader('video4_3.mp4'); %Acceder al video  
%imshow('video1.mp4'); %Visualizar el video  
  
nframes = trafficVid.NumberOfFrames;  
figure, pause(0.3);  
mask=rgb2gray(read(trafficVid,4));
```

```

for i=20:nframes
    vid = rgb2gray(read(trafficVid,i));
    vide=vid-mask;
    video = vide > 30;
    video2 = imfilter(video, ones(16)/256); % Con kernel de 7x7 es apropiado
    subplot(2,2,2), imshow(read(trafficVid,i)),title('Imagen Original');

    % Cálculo de área de cada blob
    area_objetos = regionprops(L, 'Area');
    areas = []; % Vector para tener las áreas desde la estructura entregada por 'reg
    for index = 1:NUM [
        areas = [areas area_objetos(index).Area];
    end
    autol = L == 2; % Selecciona auto con etiqueta "2"
    %subplot(2,2,3), stem(areas),title('Areas'); grid %GRAFICA DE AREAS
    %subplot(2,2,3), imshow(autol),title('Imagen Filtro Area');
    % Extrae la máscara para un auto

    subplot(2,2,4), imshow(read(trafficVid,i)),title('Imagen Final');
    stats = regionprops('table',autol,'Centroid','MajorAxisLength','MinorAxisLength')
    centers = stats.Centroid;
    diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);
    radii = 20+diameters/2;
    subplot(2,2,4), viscircles(centers,radii,'EdgeColor','b','LineWidth',1);
end

```

Las líneas de código que se encuentran aquí, en especial algunas son tomadas del libro de Matlab básico para ingenierías (Mejía et al., 2019)

Pruebas

Las pruebas que se realizaron al prototipo fueron las siguientes:

Para la prueba No. 1 se contó con las siguientes situaciones, como se muestra en la tabla3:

Tabla 3. Prueba con video1

Observación	Información
Video	Video1.mp4
Duración	4.50 minutos
Conteo automático	84 vehículos
Conteo manual	46 vehículos
Error	82,60%

Fuente: Elaboración propia

Para la prueba No. 2 se contó con las siguientes situaciones, como se muestra en la tabla4:

Tabla 4. Prueba con video2

Observación	Información
Video	Video2.mp4
Duración	3.50 minutos
Conteo automático	73 vehículos
Conteo manual	40 vehículos
Error	82,50%

Fuente: Elaboración propia

Para la prueba No. 3 se contó con las siguientes situaciones, como se muestra en la tabla5:

Tabla 5. Prueba con video3

Observación	Información
Video	Video3.mp4
Duración	4.50 minutos
Conteo automático	63 vehículos
Conteo manual	46 vehículos
Error	36,95%

Fuente: Elaboración propia

Para la prueba No. 4 se contó con las siguientes situaciones, como se muestra en la tabla6:

Tabla 6. Prueba con video4

Observación	Información
Video	Video4.mp4
Duración	3.50 minutos
Conteo automático	56 vehículos
Conteo manual	40 vehículos
Error	40%

Fuente: Elaboración propia

Para la prueba No. 5 se contó con las siguientes situaciones, como se muestra en la tabla7:

Tabla 7. Prueba con video5

Observación	Información
Video	Video5.mp4
Duración	4.50 minutos
Conteo automático	51 vehículos
Conteo manual	46 vehículos
Error	10,87%

Fuente: Elaboración propia

Para la prueba No. 6 se contó con las siguientes situaciones, como se muestra en la tabla8:

Tabla 8. Prueba con video6

Observación	Información
Video	Video6.mp4
Duración	3.50 minutos
Conteo automático	46 vehículos
Conteo manual	40 vehículos
Error	15%

Fuente: Elaboración propia

Para la prueba No. 7 se contó con las siguientes situaciones, como se muestra en la tabla9:

Tabla 9. Prueba con video7

Observación	Información
Video	Video7.mp4
Duración	4.50 minutos
Conteo automático	50 vehículos
Conteo manual	46 vehículos
Error	8,70%

Fuente: Elaboración propia

Para la prueba No. 8 se contó con las siguientes situaciones, como se muestra en la tabla10:

Tabla 10. Prueba con video8

Observación	Información
Video	Video8.mp4
Duración	3.50 minutos
Conteo automático	43 vehículos
Conteo manual	40 vehículos
Error	7,50%

Fuente: Elaboración propia

Como podemos observar se tuvo que replantear las líneas de división para que el algoritmo pueda realizar de manera correcta la detección de los autos que pasan por las dos vías tomadas en los videos, tal cual como se muestran los datos desde las tablas6 hasta la 10.

Se realizaron 8 pruebas dando como resultado las tablas que se pudieron observar anteriormente. Entonces nos toca medir bien los autos en la realidad para ingresar con la escala correspondiente al sistema.

Discusión

La toma de decisiones la debe generar el algoritmo planteado para presencia o ausencia de vehículos en la vía, en el caso planteado que es la de contar los vehículos que llegan al semáforo en la vía principal, resulta muy beneficioso utilizar esto.

La dimensión del tiempo en cuanto a respuesta del sistema esta medido en segundos.

El algoritmo aplicado da un margen de error de más menos 8%, pero para llegar a esto se tuvo que ir de un error en la primera prueba del 80% hasta llegar al 7,5% del mismo, ya que todavía toca hacer algunos ajustes en el desempeño del algoritmo respecto a dimensiones de los vehículos, especialmente cuando tenemos mayor cantidad de ellos y de diferente tipo. A pesar de esto el algoritmo planteado está dando resultado en la medición del tráfico y el conteo de los vehículos, con la altísima responsabilidad de que

cuando el sistema esté completo pueda dar prioridades a los semáforos para cambiar de verde a rojo y viceversa.

Un resultado final inesperado fue que cuando se quiso aplicar otra cámara, el algoritmo no podía asumir muy bien qué tipo de autos eran ya que las entradas en el sistema tienen un tipo de dimensión y con este otro tipo de cámara pues no había como llegar a detectar que auto mismo era y entonces dicho objetivo no se estaba cumpliendo.

Como dijimos anteriormente, ya, en el momento de ejecutar el prototipo, se tiene que ver bien las dimensiones de los autos. Hay que prever que los automóviles tienen una dimensión, las camionetas, las busetas, los buses, camiones y tráileres tienen otras; por eso se tomó la situación sólo para automóviles.

Como trabajos futuros se propone realizar para todos los vehículos pero usando ya reconocimiento de patrones pero para diferentes tipos de vehículos, es decir usando 2 cámaras inalámbricas para el conteo de autos y usando algoritmos para el día y otro para la noche. Se estará mostrando las simulaciones realizadas con ejemplos prácticos y ya con semáforos que cambien de color verde a rojo y viceversa.

Conclusiones.

- En este artículo se ha realizado una presentación del prototipo, basado en un algoritmo usando programación estructurada con herramientas de visión artificial que tiene Matlab para solucionar el conteo de vehículos en las vías de la ciudad de Riobamba, provincia de Chimborazo, país el Ecuador.
- El uso de cámara web es de gran ayuda para este tipo de proyectos, pero hay que saber escogerla, al momento de realizarlo en la realidad.
- Los problemas por donde hemos ido en este campo de la creación del proyecto de investigación son muy grandes pero no complejos ya que en el camino vamos encontrando las soluciones deseadas.
- Del 80% de errores al principio, ahora tenemos un más menos 8%, que es superable y entendible, hasta que el algoritmo y las dimensiones correctas de los autos cumplan con el objetivo para la cual fue creado.
- La importancia radica que basado en esta aplicación inteligente, el programa procesa videos captados por una cámara instalada en una intersección los cuales son enviados a un computador a través de WiFi o cable, con esto se puede obtener flujo vehicular con hasta un 92% de efectividad, clasificación de vehículos diariamente y en horas pico. Considero que es una herramienta muy útil para que el problema del flujo vehicular tenga solución

Referencias bibliográficas.

- Branch, J. & Olague, G. (2001). La visión por computador: Una aproximación al estado del arte. *Revista Dyna*, 133.
- Gualdron, O., Duque, M. and Chacón, A. (2013). Diseño de un sistema de reconocimiento de rostros mediante la hibridación de técnicas de reconocimiento de patrones, visión artificial e IA, enfocado a la seguridad de interacción robótica social. *Revista Dialnet edición 6*, pág. 16-28.
- López, G. (2016). Sistema inteligente de reconocimiento de patrones con visión artificial para la alerta automática de intrusos en las áreas de almacenamiento de las pymes. Ambato: Universidad Técnica de Ambato.
- Mejia, F., Jaramillo, K. and Gutiérrez, J. (2019). *Matlab básico para Ingenierías*. CIDEPRO Editorial. Ecuador.
- Meza, D. (2020). Esta inteligencia artificial puede detectar si las personas están guardando “la sana distancia”. Consultado el 20 de enero de 2021 en: <https://nmas1.org/news/2020/04/22/vision-ia-distancia>
- Pajares Martinsanz, G. and Santos Peñas, M. (2006). *Inteligencia artificial e ingeniería del conocimiento*. México: Alfaomega Grupo Editor
- Salazar, M. and Manzano Herrera, D. A. (2012). Sistema de visión artificial para conteo de objetos en movimiento. *El Hombre y la Máquina - Facultad de Ingeniería, Universidad Autónoma de Occidente*, 87-101.
- Sisalima, F. (2018). Sistema para la detección y conteo vehicular aplicando técnicas de visión artificial. Tesis previa la obtención del título de ingeniero en sistemas. Loja: Universidad Nacional de Loja. Consultado el 15 de febrero del 2021 de: <http://192.188.49.17/jspui/bitstream/123456789/20892/1/Sisalima%20Ortega%20C%20Fabricio%20Roberto.pdf>
- TIOBE. (2017) Historia a largo plazo de los mejores lenguajes de programación. <http://www.tiobe.com/tiobe-index/?6671423=1>
- Toledano, L. (2019). Desarrollo de App en Matlab para rehabilitación de espasticidad con ayuda del robot colaborativo Kuka LBR IIWA. Universidad Carlos III. Madrid – España.

PARA CITAR EL ARTÍCULO INDEXADO.

Mejía Peñafiel, E. F. (2021). Algoritmo de programación estructurada, enfocado a la detección y conteo vehicular de manera inteligente en una intersección. *ConcienciaDigital*, 4(3), 141-155. <https://doi.org/10.33262/concienciadigital.v4i3.1776>



El artículo que se publica es de exclusiva responsabilidad de los autores y no necesariamente reflejan el pensamiento de la **Revista Conciencia Digital**.

El artículo queda en propiedad de la revista y, por tanto, su publicación parcial y/o total en otro medio tiene que ser autorizado por el director de la **Revista Conciencia Digital**.

