



Implementación de un Sistema Prototipo de Luces Frontales con Segmentación para Automotores Empleando Técnicas de Visión Artificial Difusas

Implementation of a Prototype System of Front Lights with Segmentation for Automotive Using Diffuse Artificial Vision Techniques

Implementação de um sistema protótipo de luzes dianteiras com segmentação para automóveis usando técnicas de visão artificial difusa

Jhinson Edgar Coyago Tomalo ^I

jhinsoncoyago91@gmail.com

<https://orcid.org/0000-0003-3203-9391>

Jonathan Javier Coyago Tomalo ^{II}

jcoyago_94@hotmail.com

<https://orcid.org/0000-0002-9830-7005>

Cristian Jhonny Carrión Paladines ^{III}

ccjhonny@gmail.com

<https://orcid.org/0000-0003-0640-0622>

Lorena Elizabeth Cuenca Sáenz ^{IV}

cuencasaenz_lorena@hotmail.com

<https://orcid.org/0000-0002-8609-2211>

Correspondencia: jhinsoncoyago91@gmail.com

Ciencias Técnicas y Aplicadas
Artículo de Investigación

***Recibido:** 30 de Septiembre de 2021 ***Aceptado:** 30 de Octubre de 2021 *** Publicado:** 15 de Noviembre de 2021

- I. Ingeniero en Electrónica Control y Redes Industriales.
- II. Ingeniero en Electrónica Control y Redes Industriales.
- III. Ingeniero en Electrónica Control y Redes Industriales, cursando una maestría en Diseño, Gestión y Dirección de Proyectos de Cooperación Internacional en UNINI-MX, Campeche, México.
- IV. Ingeniera en Electrónica, Telecomunicaciones y Redes, cursando una maestría en Diseño, Gestión y Dirección de Proyectos de Cooperación Internacional en UNINI-MX, Campeche, México.

Resumen

Debido a la gran cantidad de accidentes de tránsito que ocurren en la noche por imprudencia e impericia de conductores que no realizan los cambios de luces de altas a bajas, se implementó un sistema de visión artificial difusa para la segmentación respectiva de luces largas, con la finalidad de lograr el no deslumbramiento hacia los demás conductores que transiten en la vía. La construcción del prototipo se realizó con la conformación de una Raspberry Pi 3, Arduino UNO, Servomotores, Barra LED, relé automotriz y elementos de protección. Para el desarrollo del sistema de detección se hizo énfasis en el algoritmo de detección "Haar Cascade", la interfaz gráfica fue desarrollada en el entorno Qt Creator. Cuando el sistema detecta la presencia de un vehículo que circule en sentido contrario o en el mismo sentido, este envía el ángulo determinado por las reglas de la lógica difusa hacia el Arduino haciendo que los servomotores se muevan, logrando la respectiva segmentación de luz. El clasificador en cascada permite crear una serie de datos admisibles al reconocimiento de vehículos determinando la posición del mismo. La interfaz gráfica será visualizada por el usuario permitiendo manipular 6 botones, cada uno cumple funciones específicas como: iniciar cámara, parar, detectar, graficar fuzzy, conectar y desconectar. Según las pruebas realizadas, se establece que la distancia máxima a la que puede detectar un vehículo es aproximadamente 200 metros; de igual manera el sistema presenta mejores resultados a bajas velocidades debido a un retardo en la adquisición de las imágenes en tiempo real.

Palabras clave: visión artificial; detección de vehiculos; deslumbramiento; segmentación de luz; algoritmo haar cascade; procesamiento de imágenes; interfaz gráfica; barra led.

Abstract

Due to the large number of traffic accidents that occur at night due to recklessness and inexperience of drivers who do not perform the lighting raises from high to low, a diffuse artificial vision system was implemented for the respective segmentation of long lights, with the purpose of achieving non-glare towards other drivers who travel on the road The prototype construction was made with the conformation of a Raspberry Pi 3, Arduino ONE, Servomotors, LED bar, automotive relay and protection elements. For the development of the detection system, emphasis was placed on the detection algorithm "Haar Cascade", the graphic interface

was developed in the Qt Creator environment. When the system detects the presence of a vehicle traveling in the opposite direction or in the same direction, it sends the angle determined by the rules of fuzzy logic to the Arduino causing the servomotors to move, achieving the respective light segmentation. The cascade classifier allows you to create a series of data admissible to the vehicle recognition determined the position of it. The graphical interface will be visualized by the user allowing to manipulate 6 buttons, each one fulfills specific functions such as: start camera, stop, detect, graph fuzzy, connect and disconnect. According to the tests carried out, it is established that the maximum distance at which a vehicle can detect is approximately 200 meters; in the same way, the system presents better results at low speeds due to a delay in the acquisition of the images in real time.

Keywords: artificial vision; vehicle detection; glare; light segmentation; haar cascade algorithm; image processing; Graphic interface; led bar.

Resumo

Devido ao grande número de acidentes de trânsito que ocorrem durante a noite por imprudência e inexperiência dos motoristas que não mudam o farol alto para baixo, foi implantado um sistema de visão artificial difusa para a respectiva segmentação dos faróis altos, com o objetivo de evitar deslumbrar outros motoristas na estrada. A construção do protótipo foi realizada com a conformação de um Raspberry Pi 3, Arduino UNO, Servomotores, Barra de LED, relé automotivo e elementos de proteção. Para o desenvolvimento do sistema de detecção, a ênfase foi colocada no algoritmo de detecção “Haar Cascade”, a interface gráfica foi desenvolvida no ambiente Qt Creator. Quando o sistema detecta a presença de um veículo viajando na direção oposta ou na mesma direção, ele envia o ângulo determinado pelas regras da lógica fuzzy para o Arduino, fazendo com que os servomotores se movam, conseguindo a respectiva segmentação leve. O classificador em cascata permite criar uma série de dados admissíveis para o reconhecimento do veículo, determinando a posição do veículo. A interface gráfica será visualizada pelo usuário permitindo a manipulação de 6 botões, cada um cumprindo funções específicas como: iniciar câmera, parar, detectar, gráfico fuzzy, conectar e desconectar. De acordo com os testes realizados, estabelece-se que a distância máxima que um veículo pode

detectar é de aproximadamente 200 metros; da mesma forma, o sistema apresenta melhores resultados em baixas velocidades devido ao atraso na aquisição das imagens em tempo real.

Palavras-chave: visão artificial; detecção de veículos; brilho; segmentação leve; algoritmo de cascata haar; processamento de imagem; Interface gráfica; barra de led.

Introducción

En el presente artículo se pretende explicar la implementación de un sistema prototipo de luces frontales con segmentación para automotores empleando técnicas de visión artificial difusas. Los vehículos convencionales en el Ecuador no están dotados de un sistema con luces inteligentes que evite el deslumbramiento de los conductores que circulan por la misma vía tanto en sentido contrario como en el mismo sentido (por reflexión de luz en los retrovisores), generando directamente molestias visuales a los usuarios de automotores. Muchas de las veces el deslumbramiento es causado por la imprudencia e impericia de los conductores generando accidentes de tránsito, mismos que han sido leves, graves e incluso mortales.

La implementación del sistema prototipo pretende ser la base para el futuro desarrollo de dispositivos genéricos adaptables a cualquier tipo de vehículo convencionales, generando una solución accesible a los múltiples usuarios sin la necesidad de cambiar de vehículo, este dispositivo contará con una cámara Raspberry Pi encargada de la adquisición de imágenes en tiempo real, un algoritmo de visión artificial capaz de detectar los vehículos mediante un clasificador de cascada (haar cascade), el cual previamente ha sido entrenado con imágenes captadas por la propia cámara, esto con el fin de adquirir las variables independientes que mediante condiciones de lógica difusa procede a enviar las señales respectivas a cada actuador y así lograr el desvío del haz luminoso a partir de aquello procede a ejecutar la segmentación de luz, es decir, una o más porciones se desviarán de su proyección normal para no deslumbrar y las porciones restantes se mantendrán para aumentar la visibilidad de la carretera por la que se está circulando, de modo que el conductor no tendrá que preocuparse por este tipo de acciones y su concentración al conducir será la más óptima.

Desarrollo

Diseño e Implementación

En el sistema prototipo de luces frontales con segmentación para automotores empleando técnicas de visión artificial difusas se compone básicamente de tres elementos en general: la cámara, la caja de control y procesamiento de imágenes, barra led con sistema de persianas.



Figura 1: Sistema prototipo de luces frontales con segmentación para automotores.
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

A. Software

Python - OpenCV

Es un lenguaje de programación a diferencia con otros lenguajes como C / C++, Python presenta lentitud ante estos, pero con la facilidad de extenderlo con C / C++. Presenta dos ventajas en el código, en primer lugar, es rápido como el original de C / C++, en segundo lugar, es más fácil codificar en Python. La librería OpenCV facilita el desarrollo del sistema ya que ayuda al funcionamiento de reconocimiento de objetos mediante visión artificial.

Cascade Trainer GUI

Facilita la creación de clasificadores en cascada previamente entrenados, probados o mejorados que permite establecer diferentes parámetros como número de imágenes positivas o negativas, número de iteraciones, recurso de memoria RAM para el entrenamiento, tamaño de las muestras entre otros.

Para iniciar con el entrenamiento del clasificador en cascada, es indispensable disponer de numerosas muestras de imágenes positivas y negativas, dichas muestras deben estar contenidas

dentro de una carpeta y clasificadas en subcarpetas que contengan las imágenes positivas (carpeta “p”) y las imágenes negativas (carpeta “n”).

Arduino IDE

Se cataloga como un entorno de desarrollo en el cual se realiza la programación para cada placa de Arduino, para crear un nuevo programa de Arduino se ejecuta la aplicación, donde se escribirá las líneas de código necesarios para la comunicación con la Raspberry, y el accionamiento necesario de los servomotores cada vez que reciba el dato del ángulo que se encuentra en el puerto serial.

La programación de Arduino (Figura 2), se basa en la lectura del puerto serial para obtener el ángulo y la identificación del servo que actuará en ese momento, esta placa Arduino controla cada servomotor que están conectados a los pines PWM.

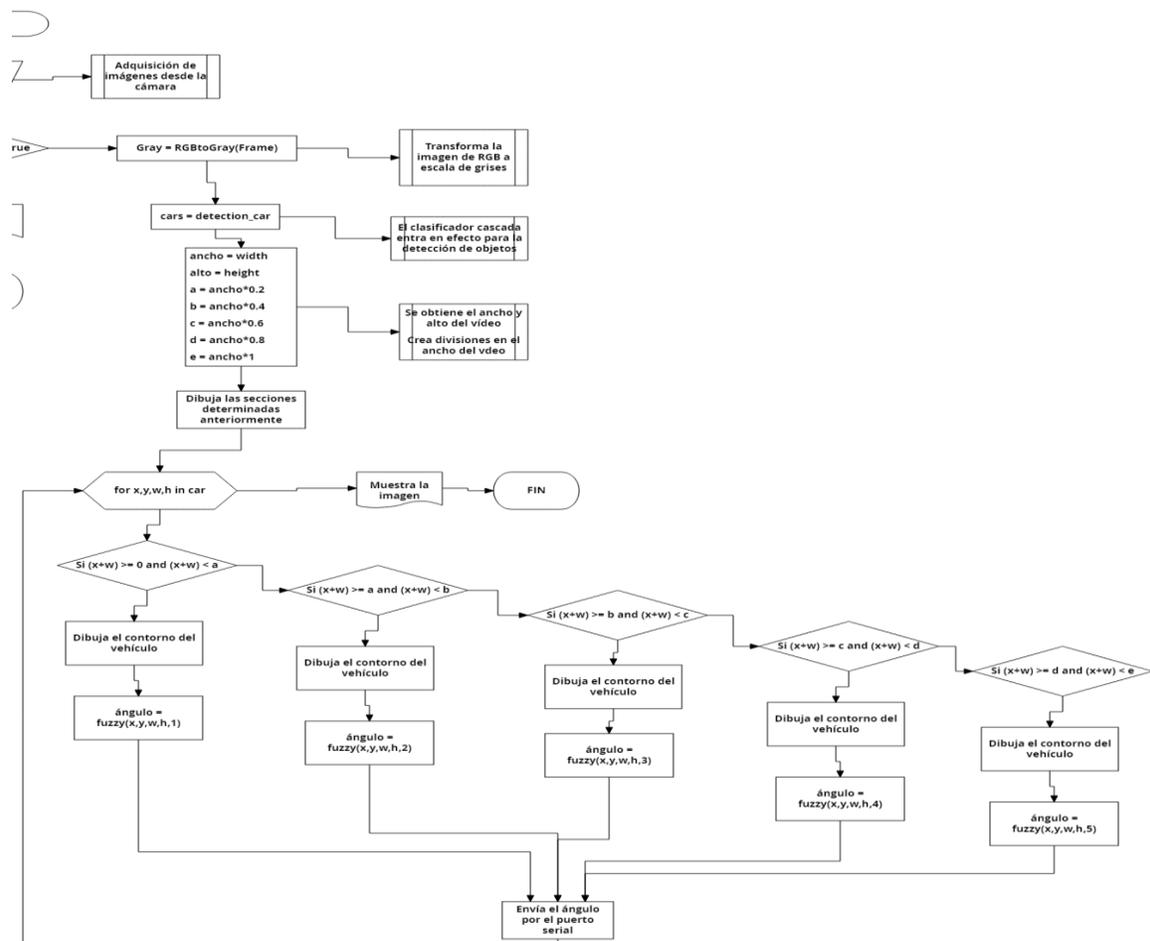


Figura 2: Diagrama de Flujo - Arduino Nano y Raspberry Pi
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Qt Designer.

Es una plataforma IDE programado en C++, JavaScript y QML, desarrollada por TrollTech, diseñado para ser compatible con los sistemas operativos más usados, como lo son: GNU/LINUX, MAC OS X y Windows. En la figura 3 se muestra la interfaz gráfica final diseñada en la plataforma de Qt Designer, con botones que cumplen una función específica cada una, que se detallan a continuación.



Figura 3: Interfaz Gráfica de Usuario
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

- 1) **Iniciar:** Inicia la captura del video.
- 2) **Stop:** Detiene la captura de video.
- 3) **Detectar:** Inicializa el Clasificador en Cascada.
- 4) **Detectar Fuzzy:** Muestra las gráficas de la lógica difusa, tanto entradas como salidas.
- 5) **Puerto COM:** Permite establecer el puerto en el que se encuentra el Arduino.
- 6) **Conectar:** Establece la comunicación con el Arduino.
- 7) **Desconectar:** Finaliza la comunicación con el Arduino.
- 8) Espacio designado para mostrar el video
- 9) **Group Box (Servos):** Muestra el ángulo de cada servo accionado.

B. Hardware.

Lámpara LED

Para la aplicación del sistema en carreteras con poca iluminación es favorable incorporar barras LED, como se muestra en la Figura 4, para obtener mayor cantidad de visibilidad. Esta barra

LED tiene una potencia de consumo de 240W a 12VDC, intensidad lumínica de 1500lm, protección IP68 y una vida útil de 50000 horas.



Figura 4: Lámpara de barra LED
Fuente: <https://goo.gl/SrQTnN>.2019

Servomotor SG90

Debido al sistema de componentes electrónicos y electromagnéticos permite controlar la posición del eje de acuerdo a una cantidad de grados acoplando varios engranajes en una transmisión similar que permite potenciar el torque del motor manteniendo fijo el eje una vez culminado el movimiento.

Arduino Nano

El Arduino Nano cumple la función de comunicar la Raspberry con los servomotores, por medio de comunicación serial, que es conectado con el cable de Arduino al puerto USB de la Raspberry. Para controlar los servomotores, la señal PWM se ha conectado a los pines del Arduino que viene configurado para usar como tal, es decir, los pines 3, 5, 6, 9, 10, tal y como se muestra en la Figura .

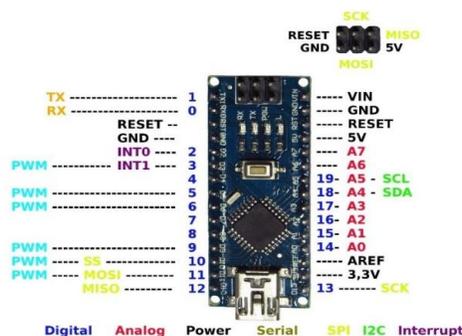


Figura 5: Esquema de los pines en el Arduino Nano
Fuente: <https://goo.gl/DJsyhp>.2019

Raspberry Pi 3 Modelo B



Figura 6: Raspberry Pi 3 Modelo B
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Este modelo de Raspberry sirve para desarrollar proyectos de visión artificial entre otros, también se usa de manera personal, reemplazando un ordenador convencional por un microordenador.

Pantalla Táctil Raspberry

Para el desarrollo de este sistema prototipo se utiliza una pantalla de 7", con la finalidad de visualizar el sistema operativo, desarrollar la interfaz y el funcionamiento final del sistema.



Figura 7: Pantalla LCD 7"
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Pi Camera

La placa de esta cámara se conecta al puerto CSI de la Raspberry Pi, ofrece una resolución de 5MP y captura de video HD de 1080p a 30fps. La cámara Raspberry cuenta con un sensor omnivision 5647 de 5MP (2592 x 1944 píxeles) dentro de un módulo de enfoque fijo.



Figura 8: Cámara Raspberry Pi
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Diseño de mecanismos para el sistema prototipo

Mecanismo para las luces frontales

Con el uso de la plataforma SolidWorks, se crea el mecanismo que permita la segmentación del haz de luz, dividida en cinco segmentos, cada segmento tiene una longitud máxima de 204mm y una altura total de 88.50mm.

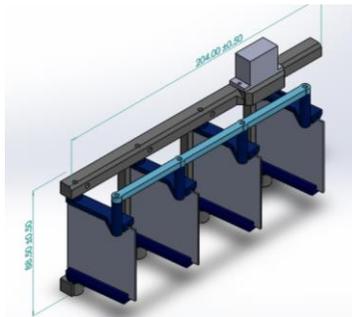


Figura 9: Ensamble del mecanismo tipo cortina - Solidworks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Estos soportes se usan para colocar una corte en acrílico transparente que sirva de cubierta y rompe vientos, permitiendo facilidad al servomotor de cumplir su respectiva función.

Diseño de la caja del sistema prototipo

El diseño de la caja posee las siguientes medidas: 236mm de ancho, 235.80mm de profundidad y 153mm de altura, con estas dimensiones se tiene la capacidad de albergar todos los componentes y mantener todo en un solo lugar para poder ser transportado, de igual manera depende de las conexiones de electricidad tanto de la batería de 12V para la barra LED, y de un

toma de 110VAC para la alimentación interna de la Raspberry, pantalla LCD, Arduino y demás elementos, esta fuente AC es proporcionada por un inversor.

Se incluyó al sistema un ventilador de PC con medidas de 80x80mm haciendo la función de propiciar un ambiente fresco de todos los elementos conectados, aportando en el rendimiento del sistema a pleno funcionamiento. El diseño de la caja posee un corte rectangular en un lateral permitiendo que los puertos USB y LAN queden con espacio suficiente para conectar dispositivos adicionales.

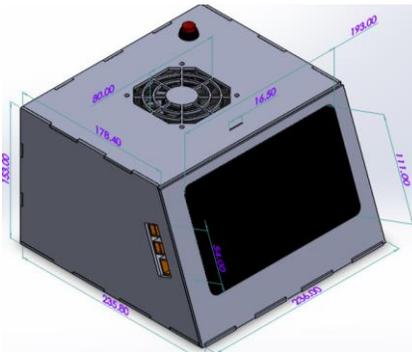


Figura 10: Ensamble de la caja del sistema prototipo - SolidWorks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Resultados

Clasificadores en cascada en espacios controlados

El primer clasificador en cascada que se utilizó fue entrenado con un número de muestras específicas captadas por la cámara oficial de Raspberry, este entrenamiento posee una base de 330 muestras positivas y 413 negativas, con el archivo de extensión “.xml”, este clasificador presenta un gran margen de error ya que no es capaz de detectar la presencia de los vehículos.

En el segundo entrenamiento se añade más muestras tanto positivas como negativas, siendo un total de 435 muestras positivas y 679 muestras negativas. Presenta falsas detecciones debido a que en la noche existen mucha iluminaria que puede ser detectada como si fuese un vehículo ya sean estas de casas, alumbrado público o reflejo de la señaléticas viales.

Para mejorar la efectividad en la detección de los vehículos, se optó por renovar el clasificador en cascada, esto se logra aumentando el número de muestras, generando una base de datos de un total de 1000 muestras positivas y 4019 muestras negativas, al momento de probar dicho

clasificador se obtiene un aproximado del 95% en detecciones positivas. En comparación con el primer y segundo clasificador este último llega a detectar más vehículos.

Para ser un sistema prototipo el porcentaje de efectividad del clasificador cumple con el requerimiento inicialmente propuesto, que es la capacidad de detectar los vehículos que transiten en la vía y con carencia de visibilidad nocturna. Se utilizó este último entrenamiento para realizar las pruebas necesarias en la carretera y en tiempo real.

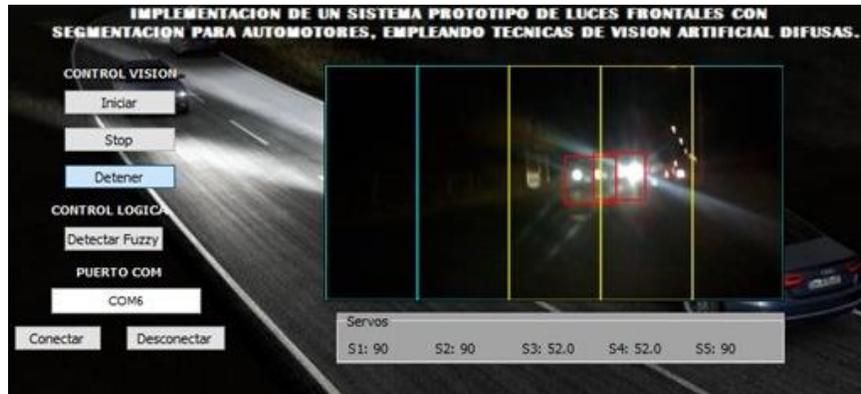


Figura 11: Prueba tercer clasificador
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Tabla 1, se muestra los resultados recopilados de los clasificadores en cascada puestos a prueba.

Tabla 1: Datos comparativos entre los clasificadores en cascada

COMPARACIONES DE CLASIFICADOR EN CASCADA							
Entrenamiento del Clasificador			Resultados del Clasificador				
Número de entrenamiento	Muestras Positivas	Muestras Negativas	Verdaderos Positivos	Falsos Positivos	No detectados	Total de vehículos	Efectividad Verdaderos Positivos (%)
1	330	413	2	5	77	79	2.53
2	435	679	52	8	24		65.82
3	1000	4019	75	3	4		94.94

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para obtener el porcentaje de efectividad por cada clasificador se usa la siguiente relación (1), sin tomar en cuenta los falsos positivos porque el nivel de efectividad que se requiere está basado en la detección de los vehículos.

$$Efectividad(\%) = \frac{Verdaderos\ Positivos}{Número\ total\ de\ vehículos} \quad (1)$$

Las falsas detecciones se obtiene debido al alumbrado público, casas o el reflejo de la señalética vial presente en las carreteras, ya que en ocasiones particulares se asemeja a los faros de un vehículo, este tipo de errores se puede reducir incrementando el número de muestras.

Tiempos de respuesta del sistema en espacios controlados

Para determinar el tiempo de respuesta del algoritmo de detección y lógica difusa, se utiliza un cronómetro de celular y se procede a poner en marcha el sistema en ciertos casos, de igual manera se realizó las pruebas en dos computadoras que cuentan con mejor procesador y memoria RAM en comparación a la Raspberry.

Al inicializar el sistema se carga un vídeo almacenado en la carpeta que contiene los archivos del mismo que tiene una resolución de 1920x1080 pixeles y un tiempo de duración de 2 min. y 22 s, el cual se utiliza en la Raspberry y en las computadoras.

Las características que presenta las computadoras se detallan en la Tabla 2.

Tabla 2: Características de las CPU

CPU	PROCESADOR	RAM	GPU
Raspberry Pi 3	Quad Cortex A53 a 1.2GHZ	1 GB SDRAM @ 400MHz	VideoCore IV de doble núcleo
Sony Vaio – VPCEB15FM	Intel Core i3 – 2.13 GHz	4 GB, DDR2	Intel HD Graphics
Lenovo – G40-80	Intel Core i5 – 2.2 GHz	8 GB, DDR3	AMD - 2GB

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Los casos en los que se analiza el sistema son cuando: se muestra el vídeo, empieza a detectar a los vehículos, comunicación con el Arduino y finalmente evitando mostrar el vídeo.

Tabla 3: Tiempo de ejecución entre las CPU

Tiempo de ejecución			
Computadora /Acción realizada	Raspberry Pi 3	Laptop Core i3	Laptop Core i5
Mostrar Video	00:02:35.65	00:00:41.11	00:00:32.51
Mostrar Video + Detección	00:05:36.63	00:02:36.10	00:01:34.65
Mostrar Video + Detección + Comunicación Arduino	00:07:01.78	00:03:13.57	00:02:41.92
Detección sin mostrar Video	00:03:30.18	00:01:25.14	00:01:05.46
Detección + Comunicación Arduino sin mostrar Video	00:04:58.68	00:02:43.72	00:02:17.11

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Como se puede observar en la Tabla 3, existe una gran diferencia en la ejecución del sistema entre la Raspberry y una Laptop; existe un mayor consumo de recursos computacionales al momento de estar en completo funcionamiento.

Consumo de corriente

Con la ayuda de un multímetro “ingco - dm200”, se realiza mediciones del consumo de corriente del sistema prototipo alimentado con una batería 12V – 18Ah.

Tabla 4: Mediciones de corriente del sistema

Fuente	Encendido Raspberry	En reposo	Inicialización de la cámara	Detección de vehículos	Comunicación Arduino
12VDC	0.03 Amp	0.03 Amp	0.03 Amp	0.03 Amp	[1.71; 2.03] Amp
110VAC	0.051 Amp	0.012 Amp	0.071 Amp	0.071 Amp	0.071 Amp

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Iluminación del entorno en espacios controlados

El vehículo para las pruebas pertinentes fue una camioneta Mazda BT-50 doble cabina año 2013, la barra es colocada en la parte frontal del vehículo adecuado a unas bases metálicas que se sujetan al chasis, en la Figura 12 se observa el montaje antes mencionado.

Se aprecia la vista frontal que es percibida por los conductores que circulan en la vía y en sentido opuesto como se muestra en la Figura 12, enumerando los sectores de derecha a izquierda como S1, S2, S3, S4 y S5.



Figura 12: Identificación de segmentos desde la vista frontal
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Con la barra situada en su lugar, se procede a verificar la visibilidad del entorno por parte del vehículo, comparando con las luces bajas, altas y con la ayuda de la barra LED.

En la Figura 13, la visibilidad del entorno mejora notablemente con la ayuda de la barra LED, en comparación con las luces bajas y altas originales del vehículo, se obtiene un campo de visualización con una iluminación óptima en la vía, evitando eventos de riesgo o colisiones.

Cuando las persianas de la barra se encuentran cerradas, es decir, a su ángulo mínimo de 11° en todos los sectores, se puede apreciar que no encandila, ya que la fuente lumínica está siendo obstruida casi por completo

Vista posterior de la iluminación del entorno

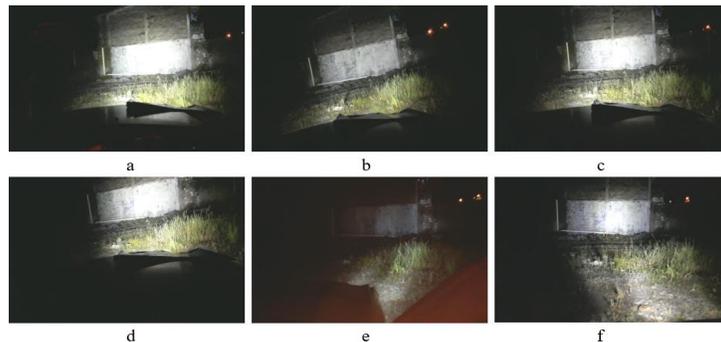


Figura 13: Vista posterior de las pruebas de iluminación
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Se realizaron las pruebas de iluminación que provee la barra LED en distintos casos como se muestra en la Figura 13, con los valores obtenidos se puede observar el cambio de visibilidad del entorno debido a variaciones en los servomotores según el sector correspondido, mismos que varían en un rango de 79° grados como máximo y un mínimo de 11° siempre y cuando se detecte un vehículo para obstruir o dejar pasar el haz de luz, sin tener en cuenta las luces que posee el carro.

Por otro lado, cuando el sistema no detecta la presencia de automotores circulantes durante una iteración de 15 fps o en su defecto aproximadamente 500ms debido a que la cámara trasmite vídeo a una velocidad de 30fps, las persianas se abrirán completamente a 90° todos los sectores dejando de lado la lógica difusa.

En la Tabla 5 se presenta una serie de posibles sucesos con mayor relevancia observado durante las pruebas mostradas en la Figura 13, se marca como “X” a la ausencia de vehículos en el sector y con “√” los sectores en los que detecta la presencia de vehículos y que el accionar se rigen al valor del ángulo emitido por el algoritmo.

Tabla 5: Presencia de vehículos en diferentes sectores

Sector	S1	S2	S3	S4	S5	Lógica difusa	Visibilidad del entorno	Encandila
a	X	X	X	X	X	No	Buena	Si
b	X	√	√	√	X	Si	Buena	Si
c	X	X	√	√	√	Si	Buena	Si
d	√	√	√	X	X	Si	Buena	No
e	√	√	√	√	√	Si	Mala	No
f	√	√	X	√	√	Si	Buena	Si

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Vista frontal de la iluminación del entorno

En la tabla 5, se puede observar el encandilamiento que será recibido por una persona que se encuentre en la misma vía con la vista de frente hacia la barra LED montada en el vehículo.

En la Figura 14 se observa el encandilamiento que presenta la barra hacia los demás conductores, en el primer caso es cuando no se detecta ningún vehículo por lo que los sectores

se establecen en 90° y se aprecia un encandilamiento alto, para el segundo caso existe presencia de vehículos en los sectores centrales propiciando un encandilamiento medio hacia el conductor, de igual manera sucede en el tercer caso debido a que los sectores S1 y S2 son lo que más afectan al conductor que transita en sentido contrario y por ello el encandilamiento que reciben es medio.

En el cuarto y quinto caso el encandilamiento es bajo debido a que los sectores más propensos a encandilar se encuentran con presencia de vehículos, para el último caso presenta un encandilamiento medio para los conductores que se encuentren lejos, porque el sector medio no detecta vehículos.

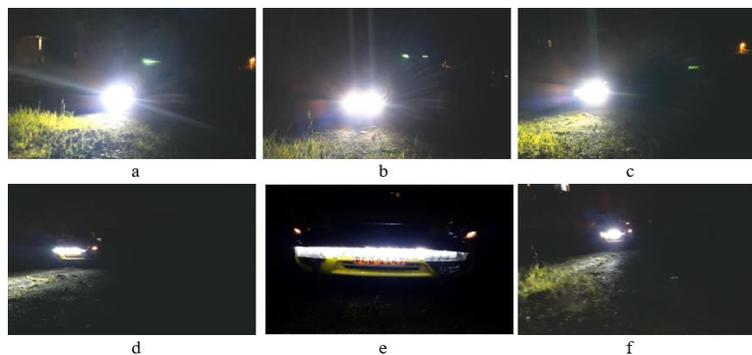


Figura 14: Vista Frontal de las pruebas de iluminación
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Los resultados presentados en esta sección no son los únicos que el sistema puede generar debido a que cada sector tiene 68 posibles valores de ángulos, depende de la posición que se encuentre el vehículo tanto en el eje X como en el eje Y, los ángulos varían entre los 11° y 79° .

Prueba de distancias en espacios controlados

Se realizaron pruebas de acuerdo a la distancia en la que puede detectar el clasificador en cascada las luces de los vehículos que llegue a ser captadas por la cámara, para ello se posicionó al sistema en un punto específico y partiendo de ahí se identificó referencias que se encuentren a 50, 100, 150 y 200 metros, como se muestra en la Figura 15, se toma como máximo los 200 metros, debido a que ese es el valor límite en el que un conductor debe hacer el cambio de luces respectivos, según lo expuesto en el *REGLAMENTO GENERAL PARA LA APLICACIÓN DE*

LA LEY ORGÁNICA DE TRANSPORTE TERRESTRE, TRÁNSITO Y SEGURIDAD VIAL.
 Artículo 189.

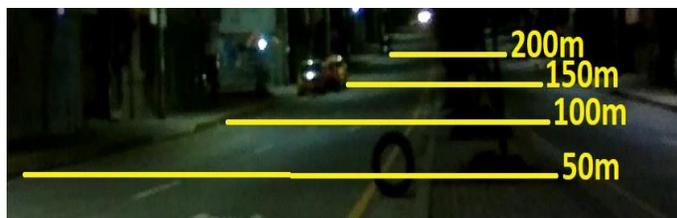


Figura 15: Distancias establecidas para las pruebas
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Los datos obtenidos se muestran en la Tabla 6.

Tabla 6: Pruebas de detecciones a distancias

Distancia (metros)	Prueba 1		Prueba 2		Prueba 3	
	Vehículos detectados	Vehículos no detectados	Vehículos detectados	Vehículos no detectados	Vehículos detectados	Vehículos no detectados
50	56	1	35	0	46	2
100	54	3	33	1	40	8
150	50	7	26	9	38	10
200	24	33	20	15	30	18
Total de vehículos	57		35		48	

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De acuerdo a la distancia del vehículo con respecto a la cámara, en una distancia de 200 metros el sistema no llega a reconocer los vehículos en un 46.1% de la totalidad, es un porcentaje por debajo de la mitad, por tanto, se determina que el sistema si llega a identificar los vehículos, aunque mayor efectividad presenta cuando el vehículo entra en el rango de los 150 metros.

Pruebas de funcionamiento final del sistema prototipo en espacios no controlados

Las pruebas finales del sistema prototipo se las realizó en el trayecto de Riobamba-Ambato, en un ambiente despejado y sin quebrantar las leyes de tránsito circulando a la velocidad permitida y realizando maniobras seguras y permitidas, el tiempo que se demoró en llegar a la ciudad de

Ambato fue de aproximadamente de 50 minutos, para el trayecto de ida se utilizó la cámara oficial de Raspberry de 1080p, y para el trayecto de regreso se cambió la cámara por una webcam con resolución de 640x480 pixeles de igual manera el tiempo de retorno es de 55 minutos, sumando un total de 1 hora con 45 minutos de funcionamiento del sistema prototipo en ambiente no controlado.

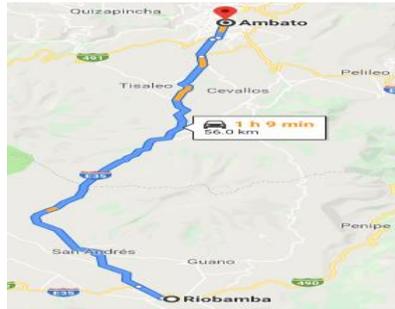


Figura 16: Trayecto recorrido para las pruebas finales
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Tabla 7 se muestra los datos recopilados durante la trayectoria hacia la ciudad de Ambato, los datos tabulados son de acuerdo a la velocidad en la que se circulaba arbitrariamente en diferentes tramos de la vía.

Tabla 7: Datos de vehículos detectados en espacios no controlados

Velocidad (km/h)	Vehículos en sentido contrario		Vehículos en el mismo sentido	
	Detectados	No detectados	Detectado	No detectados
50	18	3	7	2
55	17	6	5	3
65	20	5	9	8
70	12	11	12	6
75	12	5	8	13
80	10	16	6	8
90	6	14	5	4

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De los vehículos que circulaban en sentido contrario se lograron observar en la carretera un total de aproximadamente 155 vehículos, de los cuáles el 61% de su totalidad fueron detectados, la mayor parte de esos se detectaron cuando se circulaba a una velocidad moderada, mientras que

a mayor velocidad se reducía el número de vehículos detectados debido a que el tiempo de respuesta del sistema no es lo suficientemente rápido para actuar ante este tipo de situaciones, los conductores que transiten en sentido contrario generalmente circulan a velocidades diferentes a la nuestra (por lo general mayores) y es un factor que tiene un impacto negativo en el sistema, aumentando el número de vehículos no detectados.

Para los vehículos que circulan en el mismo sentido es posible notar que las velocidades de circulación tras otro automotor son similares entre sí, se puede decir que la diferencia de velocidades son muy pequeñas, se llegaron a observar un total de 96 vehículos, los cuáles fueron detectados 52 vehículos pertenecientes al 54% de su totalidad, en este caso, el tiempo de respuesta del sistema se considera optimo, por estas razones la cantidad de vehículos detectados depende más del entrenamiento previo para obtener mayor efectividad en los resultados, considerando que se evitó en todo momento maniobras de adelantamientos.

En el retorno hacia la ciudad de Riobamba, se pone en marcha el sistema, pero ahora con un computador “Lenovo – G40-80” y una cámara web “LOGITECH C910” de 1920x1080 pixeles en comparación a la cámara de Raspberry usada en el presente trabajo. A continuación, se muestra en la Tabla 8 los datos recopilados durante la nueva trayectoria, del mismo modo estos valores son obtenidos considerando los mismos parámetros de la Tabla 7.

Tabla 8: Datos de vehículos detectados en el trayecto de retorno

Velocidad (km/h)	Vehículos en sentido contrario		Vehículos en el mismo sentido	
	Detectados	No detectados	Detectado	No detectados
50	22	4	6	3
55	19	4	8	1
65	7	0	3	1
70	15	7	7	2
75	17	11	8	5
80	8	10	9	4
90	10	12	3	0

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De los vehículos que circulaban en sentido contrario se lograron observar en la carretera un total de aproximadamente 146 vehículos, el 67% fueron detectados, en comparación al 61% de

efectividad de la Raspberry, el computador “Lenovo – G40-80” presenta mejores resultados. Los vehículos que circulan en el mismo sentido se registra un total de 60 vehículos, los cuáles se captaron positivamente 44 unidades que conforma el 73% de su totalidad, nuevamente el computador nos refleja mejores resultados en comparación a la Raspberry.

Conclusiones

- De acuerdo a la investigación realizada se experimentó con varios tipos de algoritmos que permiten el reconocimiento de objetos, se eligió Haar Cascade porque permite desarrollar un entrenamiento clasificador de aprendizaje supervisado, destinado a detectar los patrones de faros vehiculares circulantes durante la noche, con el entrenamiento que se obtuvo el 95% de efectividad en las detecciones se procedió a realizar pruebas en espacios no controlados.
- Un método sencillo que permitió la segmentación de luz, fue obstruir el paso de iluminación que provee un faro como fuente lumínica, al realizar esta acción se aprecia de inmediato el cambio de visibilidad del entorno, se optó por una barra LED debido a su facilidad para añadir el mecanismo tipo persiana.
- En el desarrollo del algoritmo en Python usando OpenCV como librería de visión artificial, se debe determinar las variables de entradas y salidas que nos ayudó a establecer las reglas de lógica difusa para obtener el ángulo de salida en cada actuador y así el haz de luz emitido por la barra LED sea obstruido disminuyendo el encandilamiento hacia los demás conductores.
- La implementación del sistema prototipo al estar conectados todos los elementos muestra un retardo de aproximadamente 670 milisegundos desde la adquisición de imagen hasta el accionamiento del servomotor correspondiente, el tiempo determinado varía de acuerdo a las tareas impuestas especialmente en el algoritmo, es decir, tarda menos si se evita mostrar imágenes en tiempo real.
- Las condiciones climatológicas y velocidades de circulación afectaron directamente el comportamiento del prototipo durante las pruebas realizadas, debido al cambio drástico en el campo visual de la cámara por ende las detecciones vehiculares disminuyeron y los actuadores no respondían correctamente en el rango de los 200 metros.

Recomendaciones

- Para obtener un archivo Haar Cascade que permita la detección de objetos específicos con mayor calidad y aceptabilidad, en este trabajo los faros de vehículos circulantes en las noches, se debe aumentar el número de muestras positivas y negativas, además no se debe repetir o extraer sub-muestras de una misma imagen.
- Verificar el mecanismo de las persianas que se encuentren en buenas condiciones, es decir, que la movilidad de las persianas sea suave y ágil antes de poner en marcha el sistema, es necesario mantener el acrílico frontal que sirve de romper vientos para que el trabajo del mecanismo no sea forzado provocando daños en los servomotores.
- Para trabajar adecuadamente con la librería OpenCV es necesario conocer los requerimientos de cada algoritmo, en este trabajo se usa el Haar Cascade para el reconocimiento de objetos por lo cual se debe instalar la versión 3.2 o superior de dicha librería.
- Para mejorar el tiempo de respuesta del algoritmo que gobierna el prototipo se debe optar por un ordenador más eficiente con el fin de cubrir los recursos computacionales necesarios para procesar el algoritmo destinado a la detección de vehículos y el accionamiento de los actuadores para segmentar el haz de luz. Por otro lado, se sugiere realizar pruebas reemplazando las Raspberry Pi 3 por una Raspberry Pi 4 b+ lanzada este año.
- Se puede reducir el uso de consumo computacional suspendiendo la visualización de la adquisición de imágenes en tiempo real para dejar de usar la pantalla, de tal manera que el módulo central se lo pueda rediseñar para que sea de menor tamaño y maniobrable.

Referencias

1. ALONSO, R; et al. Análisis: Raspberry Pi 3 Modelo B+. [en línea] Raspberry Pi 3 Model B+. 2018 [Consulta: 15 marzo 2019]. Disponible en: <https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>
2. FIDALGO, R. Tipos de faros en el automóvil [en línea] Autocasión. 2013 [Consulta: 15 marzo 2019]. Disponible en: <https://www.autocasion.com/actualidad/reportajes/tipos-de-faros-en-el-coche>.

3. FLORES, P. & BRAUN, J. Algoritmo SIFT: fundamento teórico [en línea] Detección de extremos en el espacio-escala. 2011 [Consulta: 26 febrero 2019]. Disponible en: <http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoSIFT.pdf>
4. HERNÁNDEZ, J. Aplicación de la lógica difusa para detección de defectos en rodamientos [en línea] (Trabajo de titulación) (Pregrado) Universidad Carlos III De Madrid, Departamento de Ingeniería Mecánica. España, Madrid. 2010 [Consulta: 16 marzo 2019]. Disponible en: <https://core.ac.uk/download/pdf/30043369.pdf>
5. IBAÑEZ, P. Sistemas de iluminación avanzados en coches [en línea] Tipos de lámparas o emisores de luz. 2011 [Consulta: 14 marzo 2019]. Disponible en: <https://www.xataka.com/automovil/sistemas-de-iluminacion-avanzados-en-coches>
6. INEN. Vehículos automotores. Dispositivos para mantener o mejorar la visibilidad [en línea] 2009. [Consulta: 7 mayo 2019] Disponible en: <http://canfacecuador.com/normas/1155.pdf>
7. JIMÉNEZ, J. El sistema de alumbrado del vehículo [en línea] El sistema de alumbrado permite ejercer la conducción con seguridad al aportar la iluminación necesaria para ver y ser vistos con claridad. 2018 [Consulta: 14 marzo 2019]. Disponible en: <https://www.ro-des.com/mecanica/sistema-alumbrado-del-coche-que-es/>
8. JULIO. La imagen digital [en línea] 2012. [Consulta: 8 mayo 2019]. Disponible en: http://cuartoinformatica.tecn julio.com/wp-content/uploads/2012/02/Tema3.-Imagen_Digital.pdf

© 2021 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)
(<https://creativecommons.org/licenses/by-nc-sa/4.0/>).