Diseño y Construcción de un Robot Móvil que Aprenda a Detectar Diferentes Señales de Tránsito Mediante Inteligencia Artificial

Estefania Ávila, Patricio Garay, Verónica Grefa, MSc., Esteban Montúfar, MSc., Mario Revelo
Escuela de Ingeniería Mecatrónica, Universidad Internacional del Ecuador (UIDE), Quito - Ecuador

***Resumen* – El paper tiene como fin presentar el diseño de un prototipo móvil que aprende a detectar señales de tránsito mediante el uso de un controlador Raspberry Pi 3. El robot móvil detecta las diferentes señales de tránsito PARE, señal de giro hacia la derecha y las luces amarillo, verde y rojo del semáforo. El sistema se implementa en el lenguaje de programación Python con la ayuda de la librería OpenCV empleando un protocolo de detección de objetos basado en el algoritmo Haar-cascade ejecutado en el dispositivo Raspberry Pi 3. Se realizan varias pruebas de funcionamiento y se obtiene un prototipo con eficiencia de detección para las señales pare y giro a la derecha de 90%, mientras que para las luces del semáforo la eficiencia alcanzada es de 73.33%. La distancia a la que cada señal es detectada varía de acuerdo a las características de la imagen e iluminación, la señal de pare comparada con la señal de giro a la derecha tiene un rango de detección diferente en 10 cm ya que la señal de giro a la derecha posee menos características a ser detectadas.**

***Índices*–Haar-cascade, OpenCV, Señales Tránsito, Python, Raspberry.**

I. Introducción

La gran mayoría de los accidentes de tránsito son provocados debido a que las personas no respetan las diferentes señales de tránsito que se encuentran en las vías siendo esta la principal causa de accidentes. Esto genera que existan problemas de tráfico y la pérdida de vida por los accidentes. La Ingeniería Mecatrónica permite hallar soluciones a diferentes problemas automatizando procesos cotidianos o industriales para mejorar su eficiencia y calidad. El presente paper muestra el desarrollo de un prototipo para la detección de señales de tránsito mediante un protocolo de aprendizaje basado en inteligencia artificial; las señales de pare, luz en rojo, luz en verde y luz en amarillo del semáforo son las evaluadas. Este prototipo busca generar una propuesta inicial para un sistema capaz de ser incorporado en los vehículos para la prevención de negligencia en los conductores.

La detección de imágenes, por medio de inteligencia artificial y algoritmos de aprendizaje, ha sido ampliamente desarrollada tanto en la parte teórica, con la generación de algoritmos nuevos o combinación de ellos, como llevada al dominio práctico con la implementación de sistemas de reconocimiento en diferentes ramas. El proyecto de detección de imágenes faciales explicado por *Maet a*.[1] muestra el desarrollo de los algoritmos Adaboost y Haar para la detección de rostros enfocados en la sección T. Los resultados de esta investigación muestran tiempos de detección de 350ms con la alimentación de imágenes positivas obtenidas a través de una base de datos. En la misma línea, *Li et al*[2] presentan un detector multivistas de rostro utilizando la combinación de algoritmo Adaboost para el entrenamiento del clasificador y las características de similitud Haar; el entrenamiento se lleva a cabo alimentando imágenes positivas y muchas más negativas, al ejecutar la detección con imágenes obtenidas de una base de datos se obtienen precisión en tiempo real, no obstante se concluye que se requieren más imágenes negativas para evitar falsos positivos.Trabajos similares se presentan en [3], [4]donde el objetivo alcanzado es el aumento del desempeño de detección de imágenes y la reducción del tiempo de ejecución. Otro estudio expone que el uso de Adaboost y Haar en conjunto han permitido alcanzar una precisión del 94% comparado con el 78% en ejecuciones de solo Adaboost [5]. Aunque los trabajos mencionados han probado su eficacia en ambientes computacionales, menos trabajos unen el desarrollo del algoritmo con la adquisición de imágenes en tiempo real y en movimiento.

El estudio de *Viola et al*[6] reúne la información necesaria acerca de las características y beneficios del uso de los algoritmos Adaboost y Haar, lo que ha servido de base para la implementación de proyectos recientes como "*Real-time Detection of Speed-Limit Traffic Signs on The Real Road using Haar-like Features and Boosted Cascade*" [7] y "*Early Detection of Sudden Pedestrian Crossing for Safe Driving during Summer Nights*" [8]. La detección de señales de tránsito en [7] ha sido aplicada a señales de velocidad de forma esférica, esto les ha permitido reducir el área de evaluación en la imagen, variar la segmentación del color y tener menos opciones de morfología. Adicionalmente, el método para la detección de colores empleado ha sido HSV y no RGB. El uso de los algoritmos Adaboost y Haar combinado con la comunicación CAN de los vehículos posibilita el ajuste de los parámetros para la evaluación en diferentes escenarios de iluminación ya que este factor es determinante durante la adquisición de la imagen y posterior evaluación. Por otro lado, [8] muestra el uso de cámaras FIR (Far Infrared) para la toma de imágenes; este recurso facilita la detección ya que al emplear cámaras infrarrojas las imágenes no se ven afectadas por la iluminación externa sino que capturan la radiación emitida por los transeúntes, debido a esto este proyecto ha sido

ejecutado en escenarios nocturnos en Corea con velocidades del vehículo de 20km/h.

Paralelamente, se han lanzado motores de búsqueda de imágenes exitosas como ID My Pill y Chic Engine[10]. Su desarrollador, Adrian RoseBrock, ha utilizado visión artificial en muchos de sus proyectos, varios de estos desarrollados en Raspberry Pi3, entre estos se encuentran: Detección de gatos en imágenes con OpenCV, Detección de peatones con OpenCV, Detección facial con Python y OpenCV, Detección de color OpenCV y Python, etc. Estos proyectos han servido de guía para esta investigación, además de la implementación del algoritmo Haar-Cascade con la ayuda del aprendizaje desarrollado por Naotoshi Seo [9].

Este paper, en su primera parte, describe los protocolos de aprendizaje más comunes;luego, presenta el bosquejo del prototipo desarrollado considerando el diseño mecánico, electrónico y de control empleando un Raspberry Pi 3 como controlador, servomotores para el movimiento y la Pi cámara como interfaz para la detección de señales. Finalmente, se exhiben las diferentes pruebas realizadas para la detección de distintas señales de tránsito y las conclusiones obtenidas.

II. Protocolos de Aprendizaje basados en IA

Todas las definiciones de Inteligencia Artificial están relacionadas con la siguiente idea: Desarrollo de métodos y algoritmos que permitan comportarse a las computadoras de modo inteligente[11]. Entre las aplicaciones habituales de los sistemas de aprendizaje automático se puede encontrar las tareas difíciles de programar como el reconocimiento facial, las aplicaciones auto-adaptables como los interfaces inteligentes o los filtros anti-spamy la minería de datos que supone un análisis inteligente de grandes volúmenes de datos[12]. Como disciplina no hay una única forma de aprendizaje automático, diferenciándose distintos tipos en función de las características del aprendizaje. Existen varios tipos de aprendizaje automático conocidos como protocolos de aprendizaje, los cuales se describen a continuación[13].

A. Aprendizaje supervisado

En este se produce una función que establece una correspondencia entre las entradas y las salidas deseadas. Un ejemplo de este algoritmo es la clasificación en la cual el sistema trata de etiquetar una serie de vectores utilizando una entre varias categorías. La base de conocimiento del sistema está formada por ejemplos de etiquetados anteriores[14].

B. Aprendizaje no supervisado

El proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formados tan sólo por entradas al sistema. El sistema tiene que ser capaz de reconocer patrones para etiquetar las nuevas entradas.

C. Aprendizaje semisupervisado

Esta es una combinación del aprendizaje supervisado y no supervisado para clasificar de manera adecuada. Se tienen en cuenta los datos marcados y los no marcados.

D. Aprendizaje por refuerzo

El algoritmo aprende observando el entorno que lo rodea. Su información de entrada es la realimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo y error.

E. Transducción

Similar al aprendizaje supervisado, pero no construye deforma explícita una función. Trata de predecir las categoríasde los futuros ejemplos basándose en los ejemplos de entrada,sus respectivas categorías y los ejemplos nuevos al sistema.

F. Aprendizaje multitarea

Métodos de aprendizaje que usan conocimientopreviamente aprendido por el sistema para resolver problemasparecidos a los ya observados.

G. Aprendizaje automático

De acuerdo con *Michie et al*. [15]el aprendizaje automático es un método donde se toma decisiones en nuevas situaciones en base a los casos de ejemplo proporcionados (Fig. 1).

Conociendo queun candidato es una ventana dentro de una imagen, esta ventana posee dimensiones fijas llamadas canónicas y que los clasificadores se basan en modelos que aprenden automáticamente a partir de un conjunto de muestras.

Primero se diseña un descriptor de ventanas, luego se define una frontera que separe los objetos que interesan de los que no, finalmente al clasificador le llegan una serie de candidatos para determinar si corresponde con los objetos de interés.

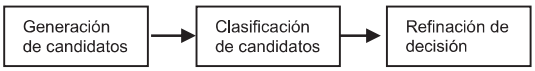

Fig. 1. Secuencia de procesamiento.

III. Bosquejo del Prototipo

El diseño del prototipo se realiza mediante el uso de una Pi cámara para el reconocimiento de las diferentes señales de tránsito, esta es colocada en la parte superior del prototipo para mejorar el rango de visión; adicionalmente se ha considerado un amplio espacio interior para la colocación de la batería de alimentación, los motores que dan el movimiento y el Raspberry Pi 3 (Fig. 2).

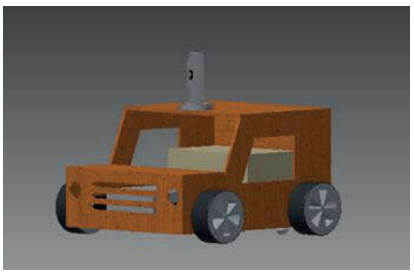

Fig. 2. Bosquejo del prototipo de robot móvil.

IV. Programación

La programación debe permitir activar la cámara para la obtención de imágenes del entorno. Cada imagen es procesada para detectar los objetos de interés, en este caso las señales de tránsito. Dependiendo de la señal que encuentre o si no encuentra ninguna señal, se determina la señal que se envía para el control de motores. Esto se ve resumido en la Fig. 3.

Fig. 3. Diagrama de bloques general de la programación.

La parte más importante es el procesamiento de imagen ya que a través de esta se hace la detección del objeto. Para la detección de los objetos se utilizan características de similitud Haar. Estas son una recopilación de características simples propias del objeto que se desea detectar, el programa busca estas características en la imagen obtenida para determinar si en esta se encuentra el objeto. Las características se definen mediante formas específicas del objeto, en la Fig. 4 se pueden apreciar los diferentes tipos de Haar.

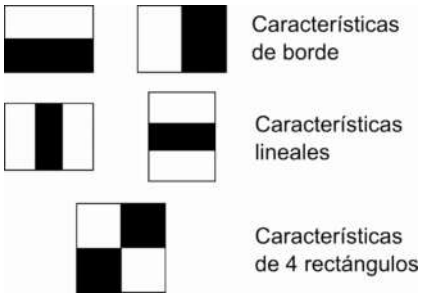

Fig. 4. Tipos de características Haar[16].

Para generar las características Haar se tiene que seguir unproceso que utiliza un protocolo basado en inteligenciaartificial como lo es el aprendizaje supervisado. Primero serecolectan varias imágenes, tanto positivas como negativas, esdecir imágenes que contengan el objeto a detectar e imágenesque no lo incluyan. Para que las características Haar seancorrectamente generadas se debe recopilar al menos 40imágenes positivas con diferentes fondos e iluminación, yaproximadamente 500 imágenes negativas. A partir de estas segeneran las muestras en las cuales se toma una imagenpositiva al azar y se coloca sobre una negativa. De este modoel sistema aprende a detectar el objeto con diferentesiluminaciones y fondos. Para explicar de mejor manera esteproceso, se detallan los pasos en la Fig. 5.

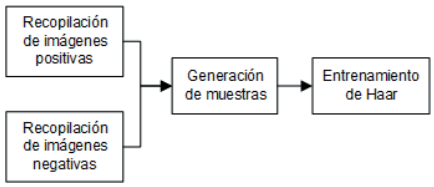

Fig. 5. Diagrama de flujo para la creación del algoritmo Haar [10].

A. *Algoritmo detectMultiScale*

Detecta objetos de diferentes tamaños en la imagen de entrada. Los objetos detectados son retornados como una lista de rectángulos. Una ventana de detección se desplaza alrededor de toda la imagen para detectar el objeto, como se muestra en Fig. 6.

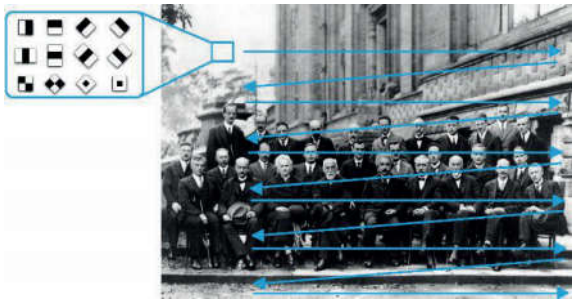

Fig. 6. Búsqueda por barrido en la imagen[17].

En la implementación proporcionada, la ventana deslizante se desplaza píxel a píxel. Cada vez que la ventana cambia, la región de la imagen dentro de la ventana pasa por el clasificador en cascada. Los parámetros que intervienen en el código de programación de Python son explicados en la Tabla I.

TABLA I
Parámetros que intervienen en el código del Algoritmo detectMultiScale[18]

Parámetros	Descripción
cascade	Haarclassifier cascade. Puede ser cargado de un archivo XML o YAML usando la función Load()
image	Matriz de tipo CV_8U que contiene una imagen donde los objetos son detectados
objects	Vector de rectángulos donde cada rectángulo contiene el objeto detectado
scaleFactor	Parámetro que especifica la escala de reducción/amplificación de cada imagen
minNeighbors	Parámetro que especifica cuántos vecinos debe tener cada rectángulo candidato para retenerlo
flags	Parámetro con el mismo significado que un cascade antiguo como en la función cvHaarDetectObjects. No es usado para un nuevo cascade
minSize	Mínimo tamaño posible del objeto. Objetos más pequeños que este son ignorados
maxSize	Máximo tamaño posible del objeto. Objetos más grandes que este son ignorados

B. *Múltiples subprocesos con Threading*

Threading en Python se utiliza para ejecutar múltiples subprocesos (tareas, llamadas de función) al mismo tiempo. Se trabaja con Threads en los casos en que la ejecución de una tarea implica una espera. Un ejemplo es la interacción con un servicio alojado en otro equipo, como un servidor web. Threading permite a Python ejecutar otro código mientras espera. Para implementar un nuevo subproceso utilizando el módulo de subprocesamiento, debe realizar lo siguiente:

- Importar la librería threading
- Definir la o las funciones que deben ejecutarse paralelamente
- Iniciar el o los nuevos subprocesos invocando el

método start().

En el robot móvil se trabaja con Threads para continuar con la obtención de imagen paralelamente con la generación de PWM.

C. Hough Circle

La función Hough Circle es una función de Open CV [19]para encontrar círculos en una imagen en escala de grises usando la transformada de Hough. La transformada de Hough es una técnica para la detección de figuras en imágenes digitales. Esta técnica es mayormente usada en el campo de Visión por Computadora. Con la transformada de Hough es posible encontrar todo tipo de figuras que puedan ser expresadas matemáticamente, como rectas, circunferencias o elipses. Los parámetros que intervienen en el código de programación de Python son explicados en la TABLAII.

TABLA II
PARÁMETROS QUE INTERVIENEN EN EL CÓDIGO DEL ALGORITMO HOUGH CIRCLE[19]

PARÁMETROS	DESCRIPCIÓN
image	Imagen de entrada en escala de grises, 8-bit.
circles	Vector de salida de los círculos encontrados. Cada vector es como un 3-element floating-point vector
circle_storage	Memoria de almacenamiento en un función C que contiene la secuencia de salida de los círculos encontrados
method	Método de detección a usar. Actualmente, el único método implementado es CV_HOUGH_GRADIENT
dp	Relación inversa de la resolución del acumulador a la resolución de la imagen
MinDist	Distancia mínima entre los centros de los círculos detectados
param1	Primer parámetro específico del método. En el caso de CV_HOUGH_GRADIENT, es el umbral más alto de los dos pasados al detector de borde de Canny ()
param2	Segundo parámetro específico del método. En el caso de CV_HOUGH_GRADIENT, es el umbral de acumulación para los centros de círculo en la etapa de detección
minRadius	Mínimo radio del círculo
maxRadius	Máximo radio del círculo

V. DIMENSIONAMIENTO MECÁNICO

Para el movimiento y transporte del Raspberry Pi 3, el módulo Pi cámara y la batería de alimentación se implementa un prototipo automóvil; el cual posee dos servomotores truncados en las llantas delanteras y dos garruchas en la parte trasera como se muestra en Fig. 7.

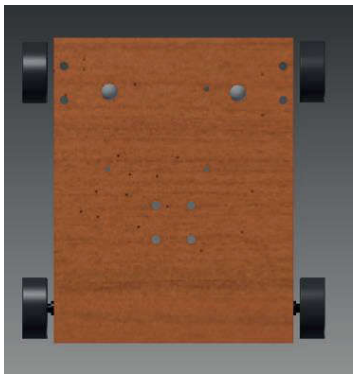

Fig. 7. Posición de las garruchas en la base del prototipo.

A. Dimensionamiento del Motor

Se calcula el torque que debe poseer cada motor para que el prototipo pueda moverse sin restricciones. Las variables que intervienen en este cálculo se muestran en la TABLA III.

TABLA III
VARIABLES CONSIDERADAS EN EL CÁLCULO DEL TORQUE DEL SERVOMOTOR

VARIABLE	DESCRIPCIÓN
m	Masa total del automóvil
R	Radio de la llanta
us1	Coeficiente estático de rozamiento entre el material de la llanta y el material donde será empleado
us2	Coeficiente estático de rozamiento entre el material de la rueda pivote y el material donde será empleado
g	Gravedad de la tierra (9.81 m/s^2)

En la Fig. 8 se muestra el diagrama de cuerpo libre del automóvil en donde se grafican las fuerzas a las que está sometido.

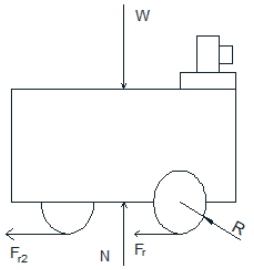

Fig. 8. Diagrama del cuerpo libre del robot mòvil.

El material utilizado para las llantas y las garruchas es plástico, el coeficiente de rozamiento estático entre el plástico y el piso de cerámica de 0.6 (medio de trabajo). A continuación, se muestran las fórmulas y el desarrollo de las mismas para la obtención del torque de los servomotores en donde intervienen las variables explicadas y analizadas anteriormente.

$$W = m \cdot g = N \quad (1)$$

$$Fr = us \cdot N \quad (2)$$

$$M = \left(\frac{F_{r2}+F_{r1}}{2}\right) \cdot R \quad (3)$$

$$T_{motriz} = 2 \cdot M \quad (4)$$

$$T_{servo} = \frac{T_{motriz}}{0.4} \quad (5)$$

Donde m es la masa, en kg; g gravedad, en m/s^2; us coeficiente de rozamiento estático; F_{r1} fuerza de rozamiento 1, en N; F_{r2} fuerza de rozamiento 2, en N; M momento motriz, en N·m; T_{motriz} torque motriz, en N; T_{servo} torque servomotor, en N·m. El valor obtenido para el torque del servomotor es 0.1765N·m (1.8kg·cm).

VI. DISEÑO ELECTRÓNICO

En la Fig. 9. se presenta el diagrama de bloques del proyecto, considerando las conexiones entre los diferentes módulos y el controlador Raspberry Pi 3.

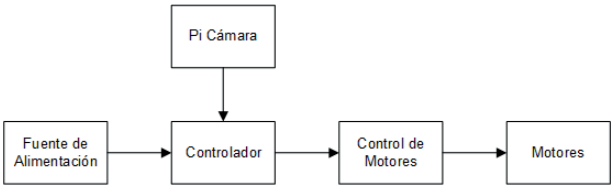

Fig. 9. Diagrama de bloques del circuito a diseñar.

A. Etapa 1: Controlador

El controlador Raspberry Pi 3 realiza la detección de objetos. Este debe ser alimentado con 5V ±5% y necesita una corriente de máximo 2.5A. También, tiene pines de entrada y salida para propósitos generales, conocidos también como GPIO por sus iniciales en inglés (general purpose input/output). Estos tienen dos posibles estados de salida que son 3.3V en ALTO y 0V en BAJO, con los cuales se controlan los motores.

B. Etapa 2: Pi Camera

La Pi cámara es un producto original de la compañía de Raspberry Pi 3. Puede ser utilizada para grabar videos de alta definición, así como fotografías. Está diseñada para ser fácil de usar, pero a su vez tiene bastante que ofrecer en aplicaciones avanzadas. Sus ventajas son: dimensiones pequeñas, bajo peso y resolución de 5 u 8 megapíxeles de acuerdo al modelo. La conexión con el Raspberry Pi 3 se establece a través de un bus de datos de 15 pines, dedicados por completo a una interfaz serie para cámaras (MIPI CSI-2).

C. Etapa 3: Motores

Para dar movimiento al robot móvil se necesitan 2 servomotores, los cuales necesitan entre 4.8 y 6V con una corriente de 180 mA sin carga y 800 mA con carga. Estos valores son importantes de considerar al momento de realizar el diseño de la siguiente etapa.

D. Etapa 4: Control de Motores

Se realiza a través del control PWM en los pines del Raspberry Pi 3. Debido a la corriente que necesitan para su accionamiento se ha empleado transistores Darlington de potencia (TIP142) [20]y un relé de 5V para la inversión del sentido de giro.

E. Circuito a Implementar

En la Fig. 10 se encuentra el esquema del circuito con las conexiones de todos los componentes que están presentes en la tarjeta. En la bornera B1 se conecta el ventilador para disipar el calor del Raspberry Pi y el resto de componentes, en B2 se conecta el motor para la rueda derecha del robot móvil y en la bornera B3 se conecta el motor para la rueda izquierda.

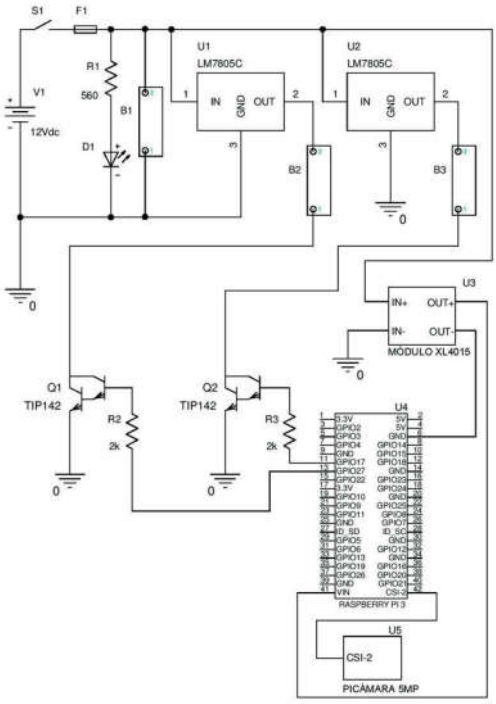

Fig. 10. Circuito a implementar para el prototipo.

Finalmente se muestra el prototipo del robot móvil implementado en la Fig. 11.

Fig. 11. Prototipo de robot móvil que aprende a detectar diferentes señales de tránsito mediante inteligencia artificial.

VII. RESULTADOS OBTENIDOS

A. Prueba N° 1. Detección de Rango de Colores para Semáforo

Mediante esta prueba se determina el rango de colores RGB para cada luz en el semáforo, verde, amarilla y roja. La prueba consiste en tomar varias fotografías a través de la Pi cámara de las diferentes luces del semáforo y obtener varias muestras del color. La TABLA IV presenta los resultados.

TABLA IV
PRUEBA DE DETECCIÓN DE RANGO DE COLORES PARA SEMÁFORO

PRUEBA	LUZ VERDE			LUZ AMARILLA			LUZ ROJA		
	R	G	B	R	G	B	R	G	B
1	0	255	1	255	255	138	255	141	154
2	49	153	68	251	184	31	255	8	110
3	49	176	73	253	254	0	251	27	89
4	3	253	6	203	141	30	231	49	87
5	9	115	25	211	155	20	255	6	110
6	0	220	36	182	137	0	254	43	96
7	0	255	68	200	124	31	203	41	79
8	42	131	51	233	172	19	255	1	113
9	0	185	40	244	222	1	249	21	98
10	0	254	52	201	213	1	247	40	96
Rango menor	**0**	**115**	**1**	**182**	**124**	**0**	**203**	**1**	**79**
Rango mayor	**49**	**255**	**73**	**255**	**255**	**138**	**255**	**141**	**154**

B. Prueba N° 2. Distancia de Reconocimiento de Señal de Tránsito

Con esta prueba se obtiene el rango de distancia, en centímetros, para el reconocimiento de las luces del semáforo. Se realizan diferentes pruebas variando la distancia entre el semáforo y el prototipo. Los resultados obtenidos se muestran en la TABLAV. La distancia máxima a la que son reconocidaslas luces del semáforo es a 30 cm. A distancias mayores no sereconoce.

TABLA V
PRUEBA DE DISTANCIA DE RECONOCIMIENTO DE LOS COLORES DEL SEMÁFORO

DISTANCIA	LUZ ROJA	LUZ AMARILLA	LUZ VERDE
5 cm	Si	Si	Si
10 cm	Si	Si	Si
20 cm	Si	Si	Si
30 cm	Si	Si	Si
40 cm	No	No	No
50 cm	No	No	No

C. Prueba N° 3. Velocidad de Funcionamiento

Con esta prueba se obtiene la velocidad en m/s a la que debe funcionar los motores para obtener los mejores resultados de reconocimiento de todas las señales de tránsito. Para variar la velocidad de giro de los motores se varía el ciclo de trabajo del PWM, la velocidad en m/s se obtiene midiendo el tiempo en una distancia recta de 1 m (TABLA VI).

TABLA VI
PRUEBA DE VELOCIDAD DE FUNCIONAMIENTO PARA DIFERENTES VALORES DE CICLO DE TRABAJO – PWM

CICLO DE TRABAJO	TIEMPO (S)	VELOCIDAD (M/S)	DETECCIÓN
65%	11,42	0,088	Si
75%	11,01	0,091	Si
85%	10,07	0,099	Si
95%	9,48	0.105	Si

Con un ciclo de trabajo del 85% no disminuye la eficiencia de detección; es decir, detecta la señal de tránsito en el menor tiempo posible.

D. Prueba N° 4. Eficiencia de Detección

Mediante esta prueba se obtiene el porcentaje de eficienciaen la detección de las señales de tránsito (Fig. 12) empleando lavelocidad de funcionamiento presentada en la TABLA VI, serealizan 10 intentos para cada señal de tránsito.

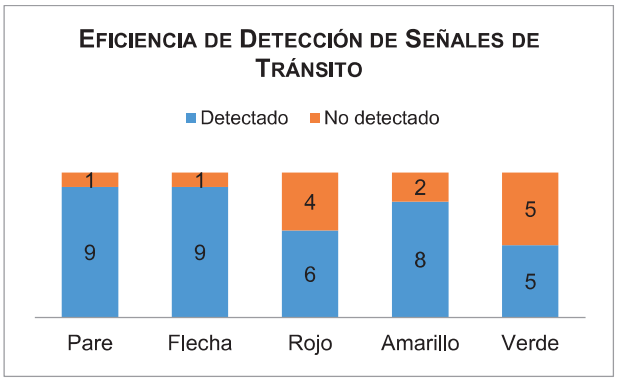

Fig. 12. Prueba de Eficiencia de Detección

E. Prueba N° 5. Tiempo de Giro

Mediante esta prueba se obtiene el tiempo necesario que debe estar detenida la rueda derecha para un correcto giro del prototipo (90°) y su posterior desplazamiento en línea recta. Para realizar esta prueba se varía el tiempo de apagado del motor derecho y mediante observación se determina si el giro es correcto basándose en el ángulo de giro (TABLA VII). El tiempo adecuado para que el auto gire 90° es de 3,2s.

TABLA VII
PRUEBA DE TIEMPO DE GIRO

TIEMPO (S)	ÁNGULO
2	50°
2.5	65°
3	80°
3.2	**90°**
3.5	105°

F. Prueba N° 6. Tiempo entre la Detección de la señal PARE y frenado

Mediante esta prueba se obtiene el tiempo que debe esperar entre la detección de la señal PARE y el frenado del prototipo, empleando la velocidad de funcionamiento encontrada en la prueba N°3 (TABLA VI). Para realizar esta prueba se emplea el método de observación determinando si el prototipo se detiene o no en donde debe frenar, modificando el tiempo de espera entre la detección de la señal y el control de los motores. Se emplea un flexómetro para medir la distancia. En la TABLA VIII se puede observar que el tiempo de espera que mejor resultados genera es de 4s.

TABLA VIII
PRUEBA DE TIEMPO ENTRE LA DETECCIÓN DE PARE Y FRENADO

TIEMPO (S)	DISTANCIA DEL PARE (CM)
1	25
2	17
3	8
4	**0**
5	-7

G. Prueba N° 7. Eficiencia en el Giro del Prototipo con los Parámetros Finales

Mediante esta prueba se obtiene el porcentaje de eficiencia en el giro correcto del prototipo luego de implementar los resultados obtenidos en pruebas anteriores; se utiliza el método de observación para determinar si el giro es correcto, es decir, si el robot gira 90° y continua su trayectoria de manera recta. De 10 pruebas, en 8 ocasiones el auto gira correctamente las cuales se presentan en la Fig. 13.

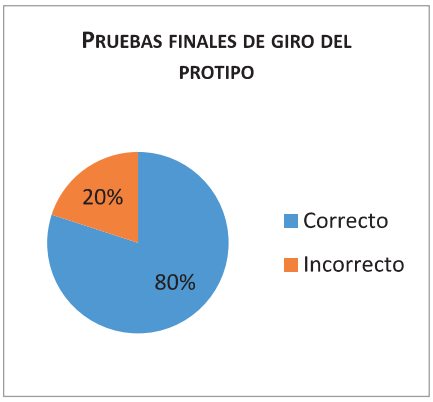

Fig. 13. Pruebas finales de eficiencia de giro del prototipo

VIII. Conclusiones

La obtención de los rangos para la detección de color de cada luz del semáforo varía de acuerdo a la intensidad de luz que presente el medio ambiente en el momento del procesamiento de la misma ya que el enfoque de la cámara y el tono de los colores varían. La distancia a la que cada señal es detectada varía por el tipo de procesamiento y las características que esta posee, en el caso de la señal de pare en comparación con la señal giro a la derecha poseen una rango de detección diferente en 10 cm ya que la señal giro a la derecha posee menos características a ser detectadas en la imagen. La eficiencia de detección de las señales pare y giro a la derecha es del 90% mientras que la de las luces del semáforo tiene un promedio de 73.33% por lo cual se puede concluir que la detección de objetos por medio de las características Haar mejora la eficiencia del proceso. Ha sido necesario controlar el entorno y su iluminación para evaluar los resultados, finalmente, el entrenamiento para diferentes imágenes requiere un aumento de las imágenes positivas y negativas con el objetivo de mejorar el porcentaje de detección.

IX. Referencias

[1] S. Ma and L. Bai, "A Face Detection Algorithm Based on Adaboost and New Haar-Like Feature," pp. 651–654, 2016.

[2] Q. Li, Z. Chen, P. Liang, L. Deng, J. Zhong, X. Liu, S. Qi, H. Zhang, and T. Wang, "Multi-view face detector using a single cascade classifier," in *Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on*, 2016, pp. 464–468.

[3] S. Chen, M. Dong, J. Le, and M. Rao, "A Method for Vehicle Occupant Height Estimation," 2017.

[4] S. Guennouni and A. Ahaitouf, "Face Detection : Comparing Haar-like combined with Cascade Classifiers and Edge Orientation Matching," 2017.

[5] W. Yu, J. Xiu, C. Liu, and Z. Yang, "A depth cascade face detection algorithm based on adaboost," in *Network Infrastructure and Digital Content (IC-NIDC), 2016 IEEE International Conference on*, 2016, pp. 103–107.

[6] P. Viola, O. M. Way, and M. J. Jones, "Robust Real-Time Face Detection," vol. 57, no. 2, pp. 137–154, 2004.

[7] W. J. Jeon, T. Lee, G. Adrian, and R. Sanchez, "Real-time Detection of Speed-Limit Traffic Signs on The Real Road using Haar-like Features and Boosted Cascade," pp. 2–6, 2014.

[8] M. J. Non-member, B. C. Ko, and J. Nam, "Early Detection of Sudden Pedestrian Crossing for Safe Driving during Summer Nights," vol. 1, no. c, pp. 1–13, 2016.

[9] A. Rosebrock, "Start here: Learn computer vision &OpenCV," 2017. [Online]. Available: http://www.pyimagesearch.com/start-here-learn-computer-vision-OpenCV/. [Accessed: 02-Feb-2017].

[10] N. Seo, "OpenCV," 2016. [Online]. Available: http://note.sonots.com/OpenCV.html. [Accessed: 15-Jan-2017].

[11] A. Sánchez Calle, "*Aplicaciones de Visión Artificial y la Biometría Informática*", Dykinson, España, Madrid, 2005. Available: http://www.ebrary.com.

[12] P. Garzón, J. Sola, "*Diseño e implementación de un robot móvil provisto de visión artificial para reconocimiento de objetos y posicionamiento del móvil a los objetos*", Proyecto Pregrado, Fac. Ing. Eléctrica y Electrónica, Quito, Ecuador, 2016

[13] Parker J., "*Algorithms for image processing and computer vision,*" Second Edition. John Wiley & Sons. 2010.

[14] P. Palma and J. Marín, "*Inteligencia artificial: métodos, técnicas y aplicaciones*," ES: McGraw-Hill España, Madrid, 2008. Available: http://www.ebrary.com.

[15] D. Mitchie, D. Spiegelhalter, C. Taylor, "Machine Learning, Neural and Statistical Classification", USA, NJ, 1994. Available: https://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf

[16] C. Osimani, "Análisis y procesamiento de imágenes para la detección del contorno labial en pacientes de odontología," *Conference: 2do Congreso Nacional de Ingeniería Informática / Sistemas de Información*, Universidad Blas Pascal, At San Luis, Argentina, 2016.

[17] Z. Ye, 5KK73 GPU, "Viola-Jones Face Detection," 2012. [Online]. Available: https://sites.google.com/site/5kk73gpu2012/assignment/viola-jones-face-detection

[18] OpenCV 2.4.13.3 documentation, OpenCV API reference. "Object Detection," 2017 [Online] Available: http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html?highlight=detectmultiscale

[19] OpenCV 2.4.13.2 documentation, OpenCV API reference. Available: http://docs.OpenCV.org/2.4/modules/imgproc/doc/feature_detection.html? highlight=houghcircles#houghcircles

[20] ON Semiconductor, "TIP 140, TIP141, TIP142, (NPN); TIP145, TIP146, TIP147, (PNP) Darlington Complementary Silicon Power Transistor", USA.

X. Biografías

Estefanía Ávila Proaño, nació en Quito-Ecuador el 25 de febrero de 1995. Realizó sus estudios primarios y secundarios en el Colegio Los Pinos, donde también obtuvo el Diploma del Bachillerato Internacional. Alcanzó mención de honor en la X Edición de las Olimpiadas de la Sociedad Ecuatoriana de Matemáticas, categoría Juvenil III en mayo de 2013. Actualmente es estudiante de Ingeniería Mecatrónica en la Universidad Internacional del Ecuador, becada por excelencia académica.
(t.fa111@hotmail.com)

Patricio Garay, nació en Quito-Ecuador el 15 de Octubre de 1995. Realizó sus estudios primarios en la Unidad Educativa Municipal Experimental "Sucre" donde consiguió el primer lugar en el concurso de Matemáticas organizado por la Universidad Central del Ecuador, además del campeonato interescolar de fútbol en el año 2006. Realizó sus estudios secundarios en el Colegio La Salle de Conocoto con la especialidad Físico-Matemático. Participó en cursos realizados por la empresa de capacitación Amatic de electrónica básica y microcontroladores PIC obteniendo los respectivos certificados ADEF. Actualmente cursa el sexto semestre en la Universidad Internacional del Ecuador, carrera de Ingeniería Mecatrónica, para la obtención del título de tercer nivel.
Áreas de interés: robótica, programación, campo automotriz, automatización y control industrial.
(serta.2h@gmail.com)

Verónica Grefa, nació en Quito-Ecuador el 19 de noviembre de 1984. Realizó sus estudios universitarios en la Escuela Politécnica Nacional, obteniendo el grado de Ingeniera en Electrónica y Control en 2008. Obtuvo su título de Master en Nanoscience and Nanotechnology en University of Southampton, Reino Unido, en 2014. Actualmente se desempeña como docente en las áreas de Electrónica y Física Aplicada en la Escuela de Ingeniería Mecatrónica en la Universidad Internacional del Ecuador.
Áreas de interés: nanoelectrónica, fotónica, microcontroladores, física aplicada.
(vgrefa@uide.edu.ec)

Esteban Montúfar, nació en Sangolquí-Ecuador el 18 de Septiembre de 1980. Realizó sus estudios secundarios en la Unidad Educativa La Salle. Se graduó en la Escuela Superior Politécnica del Ejercito como Ingeniero en Sistemas e Informática en el 2004. Obtuvo su título de Magíster en Docencia Universitaria y Administración Educativa en el 2011 en la Universidad Tecnológica Indoamérica. Actualmente desempeña el cargo de Docente T/C en laFacultad de Ciencias Exactas y Tenologías Aplicadas de la Universidad Internacional del Ecuador.(emontufar@uide.edu.ec)

Mario Revelo,nació en Quito-Ecuador el 19 de Abril de 1994. Realizó sus estudios secundarios en la Unidad Educativa Naval "CMDTE. César Endara Peñaherrera". Actualmente se desempeña como estudiante en la Universidad Internacional del Ecuador en la carrera de Ingeniería Mecatrónica cursando 6 ^{to} semestre.
Áreas de interés: domótica, automatización industrial y automatización.

(mariorevelo_14@hotmail.com;marevelolo@internacional.edu.ec).