

Robot Móvil Inteligente (Julio 2022)

Walter Humphrey, Gerardo Peña, Javier Piña, José Manuel Neri

Estudiantes del Instituto Tecnológico y de Estudios Superiores de Monterrey

A01701730

A01701474

A01701478

A01706450

Este proyecto fue realizado para la clase de Implementación de robótica inteligente (Gpo 502) con ayuda de Manchester Robotics, quienes proporcionaron material para llevar a cabo dicho proyecto.

RESUMEN En el siguiente reporte se presentan los resultados obtenidos en el desarrollo de un robot móvil diferencial, en el cual se desarrolló un seguidor de línea para mantenerse en pista y lograr el recorrido, mientras hace continuo chequeo de los semáforos que se encuentra en el camino. Además se generó y se entrenó una red neuronal que pudiera reconocer entre distintas señales de tránsito y tomar decisiones basadas en el reconocimiento. Se explicarán los métodos usados para cada una de las partes que integran el proyecto, así como comparaciones con otras posibles soluciones para este reto. También se presentan los hallazgos obtenidos durante la realización del proyecto.

OBJETIVOS Implementar un nodo seguidor de línea que sea capaz de seguir trayectorias incluyendo curvas. Implementar un nodo detector de luces de semáforo que sea capaz de reconocer si se encuentra en verde, rojo y amarillo. Implementar una red neuronal convolucional que sea capaz de predecir las señales de tráfico con efectividad. Finalmente implementar un nodo maestro que sea capaz de unificar los nodos anteriores dando como resultado un comportamiento inteligente y autónomo al puzzlebot, un robot móvil pequeño.

PALABRAS CLAVE ROS, python, opencv, tensorflow, redes neuronales convolucionales, procesamiento de imágenes, vehículo autónomo.

I. INTRODUCTION

En este proyecto se trabajó con un robot móvil diferencial, el cual tenía como objetivo lograr cruzar una pista siguiendo una línea para completar el recorrido mientras reconocía cada señal de tránsito en su camino con ayuda de una red neuronal, así como hacer el reconocimiento de las luces de semáforos identificando la cantidad de luz en el espectro hsv.

II. MATERIALES Y MÉTODOS

Para la realización de este proyecto contamos con la ayuda de Manchester Robotics, quienes nos proporcionaron materiales necesarios y asesoramiento para el desarrollo del proyecto. El robot utilizado en el proyecto es un puzzlebot

Jetson Edition de la compañía Manchester Robotics. El puzzlebot cuenta con dos motores con encoders integrados, una Jetson Nano de 2GB, una hackerboard para el control de los motores y una cámara para la visualización de la pista.

La pista en la que el puzzlebot realiza su trayecto se puede ver en la imagen 1.1. El robot debe navegar de manera autónoma dentro de ella empezando desde la zona donde se encuentra la señal de Stop y volviendo a ella, tras hacer el trayecto obedeciendo las señales y semáforos, sin ninguna intervención humana.

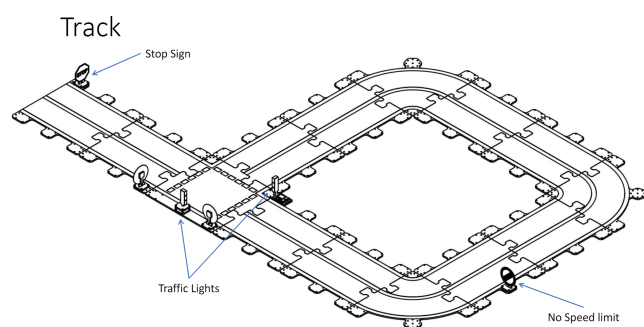


imagen 1.1. Dibujo de la pista en la que el robot

La pista cuenta con cuatro señales de tránsito: Stop, Go Ahead, Turn Right y No Limit. En la intersección se encuentran dos semáforos. El piso de la pista está marcado con tres líneas de color negro que recorren el suelo como guía para el robot, estas se interrumpen solo en la intersección.

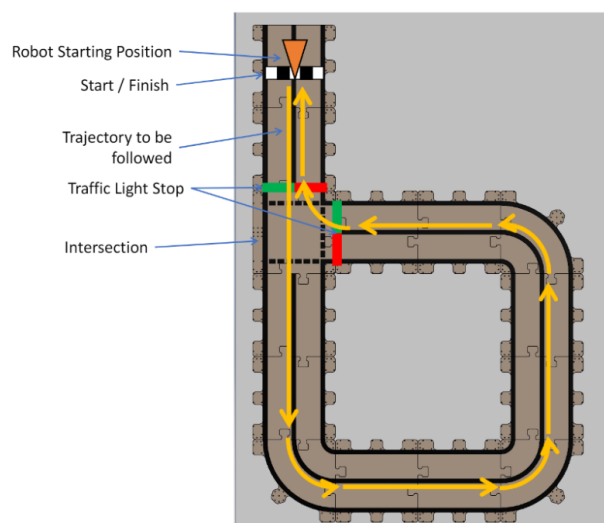


imagen 1.2. Trayecto del puzzlebot

La solución presentada consta de cuatro nodos de ROS en los que se encuentra: una red neuronal, un código de detección de semáforos, un seguidor de línea y un nodo maestro que obtiene la información de los otros nodos para controlar la respuesta del puzzlebot. Las librerías más relevantes que usamos para la detección de señales y semáforos fueron OpenCV y Tensor Flow.

Red Neuronal (CNN)

Generamos y entrenamos una red neuronal convolucional para la detección de señales de tráfico. usamos una función RELU como función de activación, usamos softmax como función de activación en la capa final para poder normalizar los valores a un porcentaje, dicho porcentaje nos dice la probabilidad de pertenencia de la imagen a cada una de las señales, una vez importado el modelo, usamos las probabilidades para decidir cuál es la señal que el robot está observando y basado en eso realiza ciertas rutinas.

La red fue entrenada con una base de datos que contenía cientos de imágenes de las cuatro señales presentes en la pista. Las fotografías fueron sacadas de señales de tránsito reales y se le agregaron fotografías de las señales en la pista para mejorar el entrenamiento de la red.

Detección de Semáforos

Para la identificación de semáforos tuvimos que procesar la imagen para limitar el espacio dónde podría aparecer un semáforo y determinar los colores que presenta este espacio de la imagen captada por la cámara. Se aplicaron filtros como el fastNIMeansDenoisingColored, el cual nos permite reducir el ruido en la imagen para después poder convertir a formato HSV y así poder diferenciar los colores basándonos en la parte del espectro de color en el que se encuentra la imagen. Se crearon 3 máscaras de color, una para la detección de verde y dos para detectar ambos espacios de rojo dentro del espectro de color. Realizamos una conversión de la imagen a formato binario para resaltar el color correspondiente para después determinar de qué color se trata comparando con cierta cantidad de píxeles encendidos en la imagen binaria. De esta forma determinamos si el puzzlebot se encuentra frente a un semáforo y el color de la luz.

Seguidor de Línea

En cuanto al seguidor de línea tuvimos que limitar el área de la imagen para evitar que objetos adyacentes causarán ruido, además de que siguiera únicamente la línea ubicada en el centro. Pasamos la imagen a escala de grises para así poder transformar la imagen a un formato binario

y resaltar la línea oscura. Aplicamos un Gaussian Blur con un kernel de (5,5) para suavizar la imagen y después hicimos 4 iteraciones de filtros erode y dilation para maximizar los negros y los blancos respectivamente. Hicimos la sumatoria de cada una de las columnas de píxeles y así determinar en qué parte de la imagen se encuentra la línea más oscura (la columna que tuviera un valor de sumatoria menor). Se utilizó un control proporcional representado por la ecuación 1.1

$$vel_w = k_w (CentroCamara - xPixelOscuro)$$

Ecuación 1.1. Control proporcional del robot

el cual nos ayuda a mantener el centro de la imagen cerca del píxel más oscuro que detectó el sistema de visión.

Nodo Maestro

Para comunicar (imagen 1.1) cada uno de los nodos generamos un script de python que funcionaba como nodo maestro, el cual se encarga de recibir los datos publicados por cada nodo y después decide qué velocidades publicar al tópico cmd_vel, el cual hace que el robot se mueva.

El nodo maestro revisa el color de los semáforos para saber si el robot puede avanzar, si está enfrente de una señal el maestro ejecuta ciertas rutinas para realizar el movimiento requerido por cada señal de tránsito.

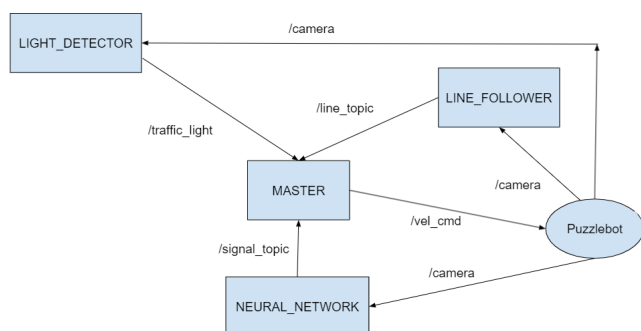


imagen 1.3. Esquema de comunicación entre nodos

III. RESULTADOS

En las pruebas realizadas con la última versión del robot se logró completar el circuito completo en 1:15 minutos, se identificaron las señales de tránsito en su totalidad, además de que el detector de semáforos funcionó en todo momento.

En cuanto a la detección de semáforos, se logró aislar el color de la luz para poder diferenciar el estado del semáforo, como podemos observar en la imagen 2.1 las máscaras empleadas funcionan muy bien al resaltar el color deseado.

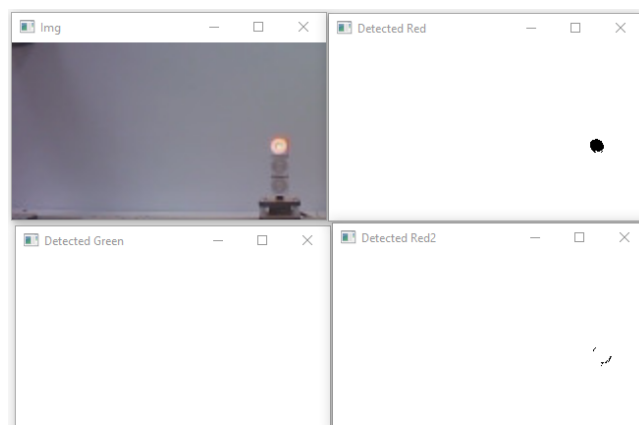


Imagen 2.1. Identificación de semáforos

Como ya se mencionó anteriormente, para el seguidor de línea la imagen fue recortada para evitar que el robot intentará seguir objetos oscuros fuera de la pista, y como se muestra en la imagen 2.2, se logró enfocar la visión justo en el área en la que se encuentra la línea a seguir, de este modo el robot siempre se mantiene en el camino.



Imagen 2.2. Imagen de la línea vista por el robot

La red neuronal generada fue capaz de identificar con éxito cada una de las señales dispuestas en la pista de pruebas.

Aquí se encuentra un video de demostración del funcionamiento del robot:

<https://youtu.be/58Kq-zgGJ4>

IV. DISCUSIÓN DE RESULTADOS

Gerardo: Obtuvimos buenos resultados en las pruebas finales, ya que se demostró que el robot puede obtener información de su propio entorno, detectando de manera correcta las señales de tráfico, así como una adecuada identificación de los colores de los semáforos. Algunas fallas que nos encontramos en el proceso fueron con respecto a las rutinas que se deben realizar al momento de enfrentarse a ciertas señales, como lo es el caso del “turn right”, que debía avanzar durante cierto tiempo y después girar, al hacer este proceso el robot choca en ocasiones contra el semáforo, esto es gracias a que no siempre se detecta la señal en el mismo momento, por lo tanto a veces el robot comienza la rutina antes de tiempo ocasionando que choque. Este problema se podría arreglar mejorando el modelo de la red neuronal, permitiendo que la señal se identifique con incluso menos error y limitando que solo se haga caso a una señal si está suficientemente cerca. Otra área de oportunidad es que podríamos lograr que el robot se mueva más rápido, para esto necesitamos calcular nuevos valores para las ganancias en el sistema de control, y posiblemente mejorar todo el control, tal vez optar por un controlador PID que sea capaz de mantener sobre la línea al robot incluso a velocidades más altas.

A pesar de estas áreas de mejora en nuestro sistema, considero que se logró dotar al robot con cierto grado de autonomía, ya que con ayuda de todos los nodos mencionados anteriormente se logra completar el recorrido.

Entre los principales retos que nos encontramos en el desarrollo de este proyecto resalta tener que acoplar los algoritmos al entorno del robot, como la luz natural que hay en el ambiente, específicamente en la detección de semáforos, ya que las variaciones en la luz cambian los rangos de color que se deben considerar para diferenciar el estado del semáforo.

Manuel: La demostración de lo que fue nuestro proyecto salió como esperábamos, funcionó de la manera correcta, esto debido a que el robot logró seguir la línea correctamente, identificar las señales de tránsito y actuar respectivamente con los semáforos, pero la unión de lo que fue todo esto no resultó nada fácil en lo que fue el caso de red neuronal la cual necesitaba simplemente la imagen de la señal, si el círculo le faltaba un pedazo u hasta la imagen se excede de lo que es la circunferencia generaba ruido y no lograba identificar al 100% la señal esto debido a la base de datos que utilizamos la cual, exige usar una imagen parecida a las de entrenamiento, otro detallito que llegamos a notar es la latencia que existe entre la Jetson y la computadora esto por enviar una imagen grande, pero este problema se soluciono disminuyendo la resolución de la imagen, mientras que la detección de colores también causó un grande problema debido a la iluminación de lo que fueron los semáforos, como cambia la luz del ambiente o el ruido que llegaba a generar el exterior de la pista, todo esto se llevo a solucionar modificando los valores y probando lo que fueron nuestros nuevos nodos.

Walter: El funcionamiento del robot fue adecuado según los objetivos propuestos. El detector de luces de semáforo logro distinguir si se encontraba ante un semáforo verde, rojo, amarillo o no habia ningún semaforo dando asi la orden al nodo maestro de continuar con el movimiento. Por parte del seguidor de línea, el robot se mantuvo dentro de la pista en todo momento, esto gracias a que se obtuvo correctamente el punto más oscuro y se aseguro que fuera el de la línea a seguir. Finalmente la red neuronal convolucional pudo predecir correctamente las 4 señales a lo largo de su recorrido, asegurando así que el nodo maestro indicará de manera correcta el movimiento a seguir. Gracias a estos resultados se obtuvo el grado de inteligencia y autonomía deseado. Los principales retos enfrentados durante este proyecto fueron hacer los ajustes necesarios en cuanto a la visión computacional con el uso de opencv para poder asegurar un

correcto procesamiento de nuestro entorno. Nos percatamos que incluso los cambios de luz o presencia de personas cerca de la pista eran perturbaciones suficientes para impedir el correcto funcionamiento de nuestro sistema. Otro gran reto al cual nos enfrentamos fue la velocidad de transferencia entre nuestros nodos, esto por el poder computacional de la jetson y la computadora donde corríamos la red neuronal. Muchas veces aunque el código era el correcto, esta velocidad de transferencia de información impedía que el robot actuara a tiempo y generaba comportamientos no deseados. correctamente. Para solucionar esto optamos por reducir la calidad y tamaño de las imágenes que estábamos procesando para así poder aligerar la carga y transferencia de datos.

Javier: La solución implementada permitió el cumplimiento del reto propuesto por Manchester Robotics. El puzzlebot pudo realizar la ruta en la pista mientras detectaba las señales y los semáforos en su camino. La red neuronal demostró estar lo suficientemente bien entrenada para detectar todas las señales en cada recorrido. Esto también demuestra que el procesamiento de imagen que se realizó para cortar la zona de la imagen en la que se detectaban círculos, las zonas donde aparecían las señales, fue correcto. En la implementación final la velocidad del robot fue algo lenta pero en ningún momento se desvió de la línea de referencia, por lo que podemos argumentar que el control del robot fue el adecuado a pesar de ser una simple ganancia proporcional. En cuanto a la detección de semáforo sí se presentan algunos errores debido al cambio constante de iluminación dentro de la habitación. Para resolver este problema antes de cada prueba se deben configurar los rango de píxeles de color detectados dentro de la detección de semáforos. Una vez que se configuran los parámetros de manera adecuada para la iluminación presente el robot no presenta errores, hasta que vuelve a cambiar la iluminación. Es adecuado mencionar que para que los códigos corran bien el nodo de la red neuronal debe ser ejecutado dentro de una computadora que soporte

de manera fluida ROS. De lo contrario los programas empiezan a fallar debido al delay en la cámara del robot.

V. CONCLUSIONES

Manuel: Los robots autónomos cada vez se vuelven más una realidad donde poco a poco se van adentrando en nuestras vidas, esto debido a que estos pueden llegar a facilitar una tarea tan compleja como lo es conducir, no ayudarán a reducir lo que es los accidentes por simples cosas como lo son puntos ciegos, distracciones, cansancio, entre diversos factores. Esto debido a los algoritmos que estos traen, hasta los sensores los cuales se encargan de recopilar información en tiempo real la cual se vuelve altamente valiosa para preservar la vida del conductor.

Gerardo: En general se desarrollaron algoritmos sencillos que lograran cumplir con cada uno de los objetivos planteados para el robot. Se dio mucha importancia al procesamiento de imágenes, como el uso de filtros en tiempo real, uso de máscaras para aislar colores, entre otros métodos muy usados actualmente en la visión computacional, esto para facilitar el trabajo de la detección de señales, detección de semáforos y el seguimiento de la línea.

Una posible área de mejora es lograr que el robot alcance mayor velocidad, para esto se deberán cambiar los valores de las ganancias y posiblemente cambiar el sistema de control considerando un control más robusto que permitan mantener al robot en pista.

Javier: La funcionalidad del robot fue exitosa al poder completar el recorrido de la pista identificando las señales y semáforos. Se logró la autonomía del robot usando redes neuronales, visión computacional, control y ROS. A pesar de ser una implementación de un vehículo autónomo al ser de pequeña escala se ve limitado a las condiciones ideales del espacio de prueba. Para lograr una mejora sustancial se tendría que afinar la detección de luz para asegurar que la detección del color provenga de un semáforo. También se

puede mejorar el control por medio de un PID o incluso un control difuso.

Walter: El robot logró el recorrido reconociendo siempre las luces de los semáforos y las señales a lo largo de él, así como siguiendo la trayectoria establecida. Se logró este grado de autonomía gracias a que todos los nodos cumplían con su función en tiempo y forma, asegurando así que el robot completará el circuito con todas las variables que este involucraba. A futuro considero que hay mucho campo de mejora en nuestro sistema, las luces de semáforo pueden ser detectadas con mayor precisión, el seguidor de línea puede detectar las líneas con mayor procesamiento de imagen para así asegurar la trayectoria y el controlador de nuestro robot podría ser más robusto para incrementar el control sobre el movimiento. Dando una explicación más concreta de las mejoras que se podrían realizar sería incluir la detección de círculos junto con la detección de colores para así evitar ruido e información falsa al momento de observar las luces de un semáforo y el entorno general donde el robot se encuentre.

REFERENCIAS

API documentation : *Tensorflow Core v2.9.1*. TensorFlow. (n.d.). Retrieved June 18, 2022, from https://www.tensorflow.org/api_docs

Autopilot. Autopilot | Tesla México. (2022). Retrieved June 18, 2022, from https://www.tesla.com/es_MX/autopilot

Nvidia. (2022, April 13). Nvidia Drive Solutions for autonomous vehicles. NVIDIA Developer. Retrieved June 18, 2022, from <https://developer.nvidia.com/drive>

OpenCV modules. OpenCV. (n.d.). Retrieved June 18, 2022, from <https://docs.opencv.org/4.x/>

Wiki. ros.org. (n.d.). Retrieved June 18, 2022, from <http://wiki.ros.org/ROS/Tutorials>

1. Diagrama de comunicación en ROS:
<https://docs.google.com/drawings/d/1BFyG15PDkxO3gXqaw5EMeh1MlcEPTi3nCbyzRMxQfOA/edit?usp=sharing>
2. Códigos del proyecto:
<https://github.com/Javier-P-C/Autonomous-Navigation-Robot>

ANEXOS