

Sistema detección de trayectoria sobre surcos simulados en cultivo

1st Christian Camilo Torres Castillo
College Program in Electronic Engineering
Escuela Colombiana de Ingeniería Julio Garavito
Bogotá, Colombia
christian.torres-c@mail.escuelaing.edu.co

2st Javier Alberto Chaparro
Master's Program in Electronic Engineering
Escuela Colombiana de Ingeniería Julio Garavito
Bogotá, Colombia
javier.chaparro@escuelaing.edu.co

Resumen—En el este proyecto se analizan y prueban diferentes métodos de deep learning y procesamiento de imágenes para la estimación de trayectorias en entornos de cultivo simulados.

Index Terms—Visión artificial, TinyML, Deep Learning.

I. INTRODUCCIÓN

Como es sabido desde hace algún tiempo se ha ido buscando la automatización de tareas de cultivo como puede ser el riego, siembra entre otras, para los cuales se han diseñado robots autónomos tipo ROVER adaptados para cultivo los cuales deben de estimar las trayectorias a seguir y desplazarse a lo largo de estas identificando de manera autónoma qué es un camino transitable y que no entre otras tareas, en terrenos de cultivo en regiones diferentes a Latinoamérica donde el terreno y el tipo de cultivo es muy plano y lineal siendo una tarea relativamente fácil de desarrollar sin embargo en el terreno colombiano y latinoamericano principalmente los surcos de cultivo se encuentran en regiones montañosas con muchos desniveles y así mismo con curvas a lo largo de estos dejando ineficaces aquellos desarrollos para otra clase de terrenos. Con algunos trabajos que se han venido realizando previamente se han analizado técnicas de identificación de trayectoria como puede ser la transformada de Job sobre imágenes de surcos con procesamiento de imágenes previo sin embargo esto no siempre es eficaz además de que estas tareas se han salido a resolver mediante tomas con drones lo cual no tiene la misma perspectiva que un vehículo el cual está en medio del cultivo por lo tanto en este documento se intenta no tener un método adecuado para realizar entrenamiento mediante deep learning o TinyML.

II. METODOLOGÍA

Para el desarrollo de este proyecto se plantea una metodología con los siguientes pasos los cuales consisten en toma de imágenes para la creación del dataset luego un preprocesamiento y entrega de datos a una red neuronal en entrenamiento. Cabe mencionar que la red neuronal que se plantea no aspira a realizar una tarea de regresión sino de clasificación donde se plantean ángulos posibles para que el

vehículo pueda tomar de entre 45 a 135 grados con una resolución de 5 grados, adicionando un caso en el que el vehículo no pueda seguir adelante dándonos un total de 19 clases para la red neuronal.

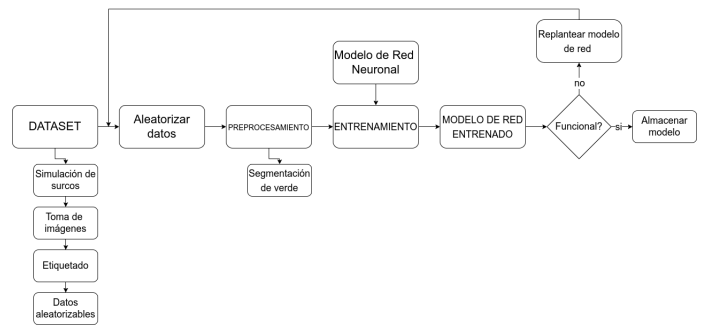


Figura 1: Diagrama de la metodología

III. DATASET

III-A. Definición de parámetros

Para la creación del data set se realizará una toma de imágenes de surco simulados para luego ser etiquetadas según la clase a la que pertenecen, pero para poder realizar la toma de imágenes es necesario asegurar de que estas han sido realizadas bajo el mismo conjunto de parámetros de la cámara lo que corresponde al ángulo de inclinación, región de visibilidad de la cámara y distancia a la frontera de visión; es decir que la región de captura de las imágenes sea la misma para todas.

En la Tabla I se listan las especificaciones de la cámara, la cual es una Logitech C270.

Cuadro I: Características técnicas

Resolución	720p
FPS	30
dFoV	55°
Tamaño de imagen	640x480

Para las tomas de imágenes se dispone del siguiente montaje para desplazar la cámara, y el cual cumple con las siguientes características

Los valores de ángulo de inclinación y altura de la cámara se escogieron considerando que el vehículo debería ser capaz de capturar píxeles hasta un rango de aproximadamente 1m de distancia a partir de la frontera de visión, reconociendo que para un ángulo en el cual el alcance de visión era de hasta 1.5m, la nueva región de visibilidad en la imagen representaba una altura en la imagen inferior a 100 píxeles, información aunque útil, significaba el incremento de la distancia hasta la frontera de visión, lo que descarta esta opción del rango de visión. Se presenta la región de visibilidad de la cámara con las características escogidas

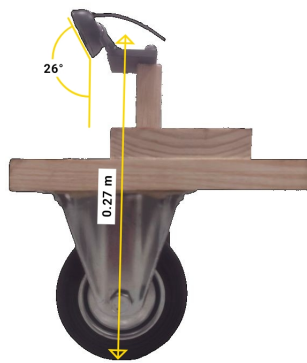


Figura 2: Geometría del montaje

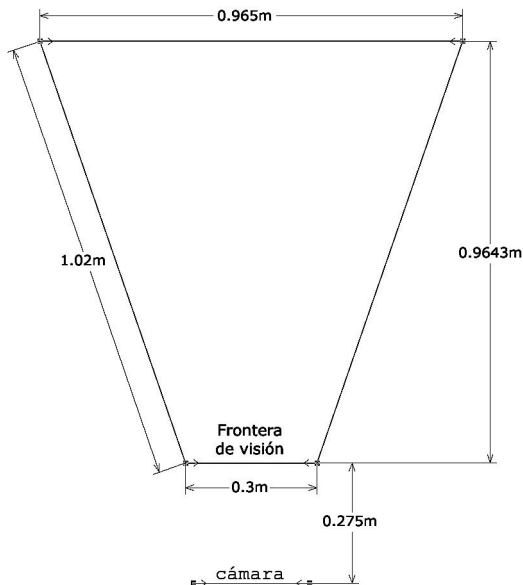


Figura 3: región de visibilidad

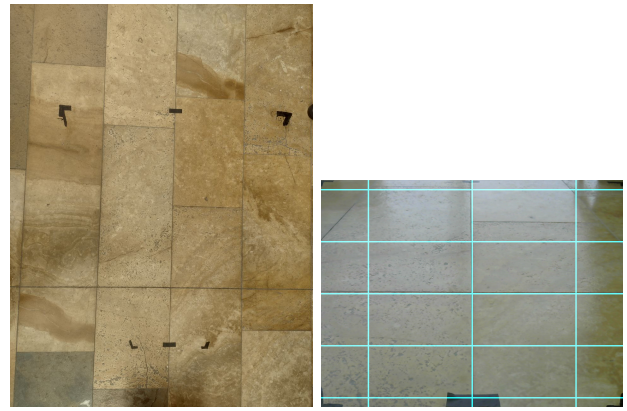


Figura 4: Captura de la región de visibilidad

III-B. Toma de imágenes y Etiquetado

Para la toma de imágenes es preferible que estas provengan de imágenes de surcos reales, pero con la dificultad de conseguir surcos reales y con las características deseadas (que sean curvos) se opta por surcos simulados con plantas enredaderas artificiales. Con el entorno para la toma de imágenes listo es necesario decidir como se etiquetarán las imágenes capturadas, para lo cual se plantean dos códigos automatizados, uno que toma un vídeo pre-grabado y se apunta con el cursor del mouse sobre la imagen la trayectoria en cada instante, almacenando el frame instantáneo del vídeo en una carpeta con el nombre del ángulo al que este se orienta, estimando el ángulo con una resolución de 5° , el segundo código se ejecuta en tiempo real mostrando sobre el streaming del vídeo un par de líneas (número ajustable) a determinado ángulo, para hacer coincidir estas líneas con las trayectorias del cultivo simulado observable, dando malos resultados por la dificultad de alinear estas líneas sin perspectiva respecto a las observadas del cultivo que si presentan perspectiva.

Se muestran ejemplos de ambos métodos

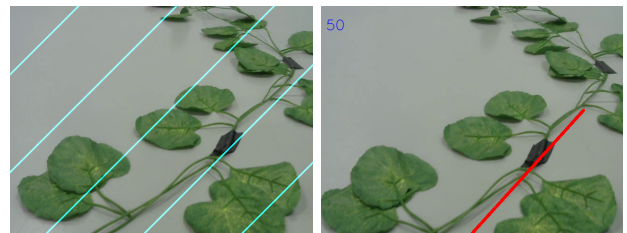


Figura 5: Métodos de etiquetado

III-C. Aleatoriedad de los datos

Para crear el Dataset se busca generar los datos de entrenamiento y validación, utilizando la función de Python `tf.keras.utils.image_dataset_from_directory`, la cual requiere que las imágenes estén almacenadas en carpetas con el nombre de la clase a la que pertenecen, entregando como resultado los datos de entrenamiento y validación, no como vectores de NumPy sino como objetos de la clase `tf.data.Dataset`, los cuales pueden ser entregados

directamente a la función de entrenamiento del modelo, pero implica cierta dificultad para manipular los datos, por ejemplo, que si se desea volver a alegorizar los datos, será necesario volver a crear el Dataset desde 0, osea ejecutar otra vez la función `tf.keras.utils.image_dataset_from_directory`, tomando un poco mas de tiempo.

IV. PREPROCESAMIENTO

El preprocesamiento en esta tarea es crucial porque es lo que va a permitir eliminar información no necesaria al momento de entrenar la red neuronal, esto es fácilmente interpretable cuando tenemos diferentes condiciones de iluminación o un entorno diferente pero un mismo ángulo a clasificar, entonces lo que se realiza es un conjunto de filtros o cambios de mapa de color de la imagen para poder extraer solamente información que se considere útil, este caso el color verde de los surcos de cultivo los cuales nos entregan información de la trayectoria.

Se toman en cuenta tres métodos de preprocesamiento todos para la extracción del color verde, el primero aplicando conversión a mapa de color a HSV y dos máscaras HSV para un tono alto y bajo de verde, un segundo método utilizando las ecuaciones de la función `GreenIdfByHSVDT` del artículo [2] adaptadas a Python, y el tercer método utilizando la ecuaciones expuestas en [1], en el cual se modifican las ecuaciones del método COM en [2]; A los tres métodos se les nombra `greenbyHSV`, `greenbyCOM` y `greenidfbyHSVDT` correspondientemente.

En las siguientes imágenes se muestra el resultado de cada método al aplicarlo sobre un surco simulado

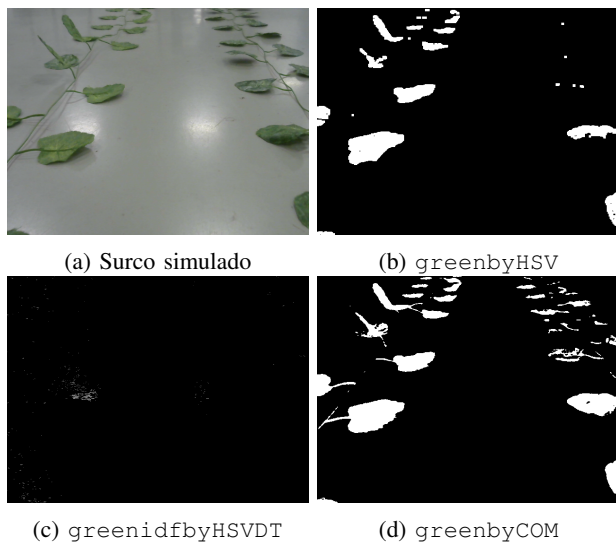


Figura 6: Comportamiento de los métodos de preprocesamiento

V. RED NEURONAL

REFERENCIAS

- [1] Manuel Eduardo Gantiva Osorio y Javier Alberto Chaparro. «Potatoes furrow lines detection using aerial RGB

image processing». En: *2022 IEEE ANDESCON*. 2022, págs. 1-6. DOI: [10.1109/ANDESCON56260.2022.9989802](https://doi.org/10.1109/ANDESCON56260.2022.9989802).

- [2] Wenzhu Yang et al. «Greenness identification based on HSV decision tree». En: *Information Processing in Agriculture* 2.3 (2015), págs. 149-160. ISSN: 2214-3173. DOI: <https://doi.org/10.1016/j.inpa.2015.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S2214317315000347>.