Reconocimiento de Señales de Tránsito mediante Redes Neuronales Convolucionales

Andrés Orlando López Henao

Universidad de Antioquia

aorlando.lopez@udea.edu.co

1. Estructura de los notebooks

El proyecto se ha estructurado en varios notebooks para facilitar la comprensión y el manejo del código. A continuación, se describe la estructura de los notebooks:

01 - Arquitectura modelo de detección.ipynb

En este notebook se entrena y prueba el modelo YOLOv4 desde cero con el conjunto de datos de Valentyn Sichkar para la detección de solamente las señales de tránsito usando el framework de redes neuronales Darknet.

02 - Arquitectura modelo de reconocimiento.ipynb

En este notebook se define, entrena y valida la arquitectura de la red neuronal convolucional para el reconocimiento de las señales de tránsito.

03 - Aplicación reconocimiento señales de tránsito.ipynb

En este notebook se crea y evalúa la aplicación integrando el modelo de detección y el modelo de reconocimiento para detectar y reconocer las señales de tránsito presentes en imágenes de prueba.

2. Descripción de la solución

Para abordar la solución del sistema de detección y reconocimiento de señales de tránsito, el proyecto se dividió en dos componentes principales: detección de las señales de tránsito y reconocimiento de la señal siguiendo las actividades de la figura 1. La solución se enfoca en proporcionar un sistema integral que pueda identificar y clasificar señales de tránsito de manera eficiente y precisa. Este enfoque pretende alcanzar un procesamiento rápido y exacto de las señales.

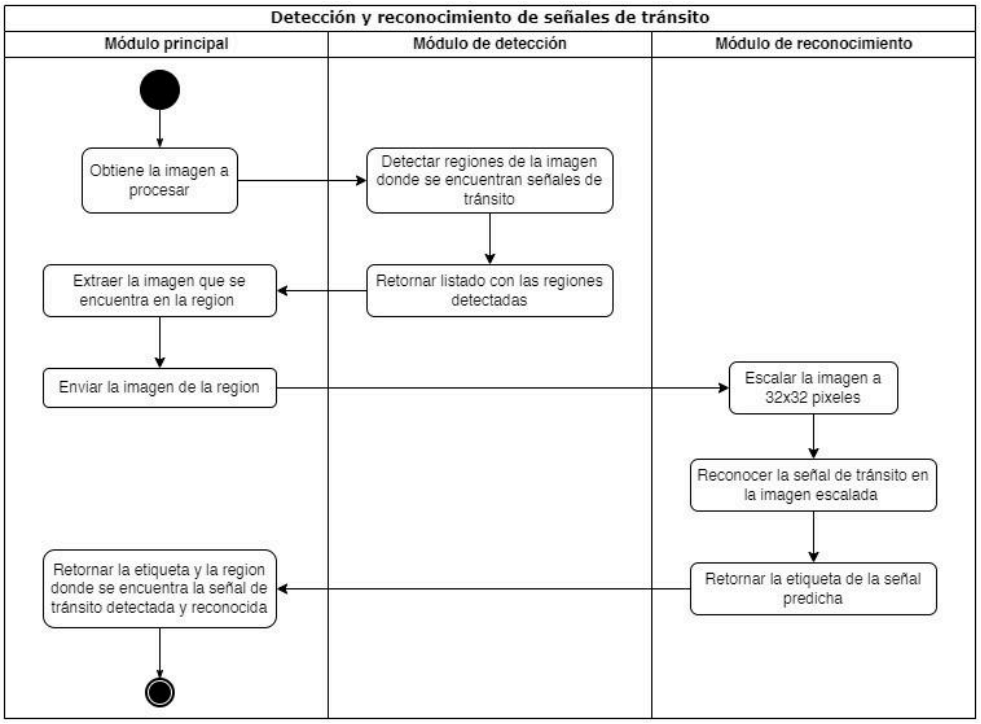

Figura 1. Diagrama de actividades para la detección y reconocimiento de señales de tránsito.

Módulo de detección

El primer componente desarrollado es el módulo de detección, basado en el modelo YOLO (You Only Look Once). Este modelo es ampliamente reconocido por su capacidad para detectar objetos en tiempo real con alta precisión. En este sistema, el modelo YOLO se encarga de identificar y localizar señales de tránsito en imágenes, proporcionando coordenadas precisas de las señales detectadas. Esta capacidad de detección rápida y confiable es esencial para el funcionamiento eficiente del sistema.

Arquitectura

Modelo YOLOv4 creado con el framework Darknet.

Módulo de reconocimiento

El segundo componente es el módulo de reconocimiento, implementado mediante una red neuronal convolucional (CNN) como se muestra en la figura 2. Esta red se especializa en la clasificación de imágenes, permitiendo identificar y categorizar las señales de tránsito detectadas en las regiones de la imagen por el modelo YOLO. La CNN garantiza una correcta interpretación de cada señal, asegurando que el sistema pueda distinguir entre diferentes tipos de señales de tránsito con gran exactitud. La combinación de estos dos módulos proporciona una solución robusta y eficiente para la detección y reconocimiento automático de señales de tránsito.

Arquitectura

La red neuronal convolucional diseñada para el reconocimiento de señales de tránsito consta de las siguientes capas como se muestra en la figura 2 de acuerdo al premuestreo que se verá más adelante en las interacciones:

- Capas de Convolución y Pooling: Dos bloques de capas de convolución seguidas de capas de pooling para extraer características de las imágenes.
- Capa de Dropout: Una capa para prevenir el sobreajuste durante el entrenamiento.
- Capa Densa: De 128 neuronas para realizar la clasificación a partir de las características extraídas.
- Capa de Dropout: Una capa para prevenir el sobreajuste durante el entrenamiento.
- Capa Densa de salida: Una capa de 43 neuronas para la salida de predicción de las 43 clases.
- Funciones de Activación: ReLU para las capas de convolución y densa, y softmax para la capa de salida.
- Compilación: Se utilizó el optimizador Adam, con una función de pérdida de entropía cruzada categórica y la métrica de exactitud.

Preprocesado

- Normalización: Las imágenes se normalizaron dividiendo los valores de los píxeles por 255.
- Escalado: Las imágenes se redimensionaron a un tamaño estándar de 32x32 píxeles.
- Aumento de Datos: Se aplicaron las siguientes técnicas de aumento de datos como rotaciones, traslaciones y cambios de brillo para aumentar la robustez del modelo.
    - Rango de zoom = entre 50% y 150%
    - Movimiento vertical = 15%
    - Movimiento horizontal = 15%
    - Modo de llenado = El píxel más cercano
    - Rango de rotación = 5%

```
Model: "C2_F28_D128"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 32, 32, 3)         0

 conv2d (Conv2D)             (None, 32, 32, 28)        784

 max_pooling2d (MaxPooling2  (None, 16, 16, 28)        0
 D)

 conv2d_1 (Conv2D)           (None, 16, 16, 56)        14168

 max_pooling2d_1 (MaxPoolin  (None, 8, 8, 56)          0
 g2D)

 dropout (Dropout)           (None, 8, 8, 56)          0

 flatten (Flatten)           (None, 3584)              0

 dense (Dense)               (None, 128)               458880

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 43)                5547

=================================================================
Total params: 479379 (1.83 MB)
Trainable params: 479379 (1.83 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

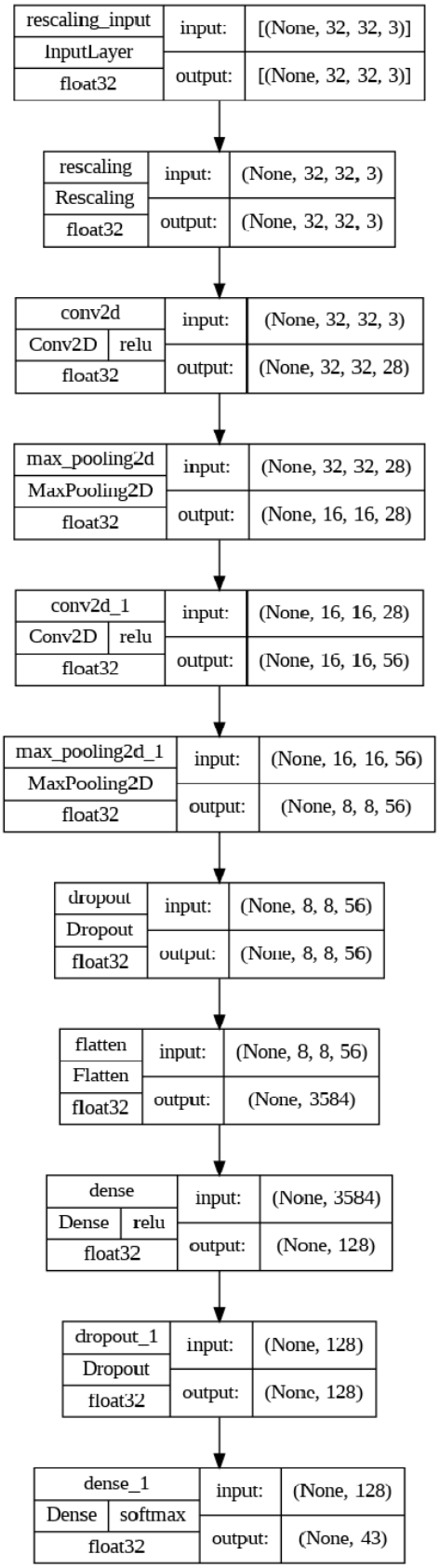

Figura 2. Arquitectura del modelo CNN para reconocimiento de señales de tránsito

3. Descripción de las iteraciones

Módulo de detección

La primera aproximación consistió en probar el modelo YOLOv7 con imágenes que contenían señales de tránsito para validar su capacidad de detección. Si bien el modelo detectaba con gran precisión vehículos, personas y otros objetos, no lograba identificar todas las señales de tránsito de manera consistente.

A continuación, se investigaron modelos YOLOv7 preentrenados específicamente para reconocer señales de tránsito. Sin embargo, el propósito no era utilizar estos modelos preentrenados, sino encontrar un dataset adecuado para entrenar un modelo que únicamente detectara la presencia de una señal sin necesidad de identificarla.

Se encontró un dataset apropiado para esta tarea del autor Valentyn Sichkar disponible en https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-dataset-in-yolo-format, que contiene imágenes de señales de tránsito en calles etiquetadas en las siguientes clases: prohibitory, danger, mandatory, y other. Este dataset, diseñado originalmente para entrenar el modelo YOLOv3, y en este proyecto fue utilizado para entrenar un modelo YOLOv4 desde cero. El proceso de entrenamiento aseguró que el modelo fuera capaz de detectar la presencia de señales de tránsito de manera efectiva sin necesidad de realizar una clasificación detallada.

Módulo de reconocimiento

Para el módulo de reconocimiento se realizó un experimento factorial para determinar qué configuración básica de red neuronal convolucional alcanzaba la mejor exactitud en el menor tiempo debido a las restricciones computacionales del entorno de Google colab.

Para el experimento se tomaron los factores que se muestran en la tabla 1.

Tabla 1. Factores

Factor	Configuración propuesta	Efecto predicho
Número de capas convolucionales	[1, 2]	La exactitud de predicción aumentará a mayor número de convoluciones.
Número de filtros	[4, 16]	La exactitud de predicción aumentará a mayor número de filtros.
Neuronas de la capa densa	[65, 128]	La exactitud de predicción aumentará a mayor número de neuronas en la capa densa.

La combinación de estos factores dió como resultado los tratamientos que se muestra en la tabla 2.

Tabla 2. Tratamientos

Identificador	Convoluciones	Filtros	Neuronas capa densa
C1_F4_D64	1	4	64
C1_F4_D128	1	4	128
C1_F16_D64	1	16	64
C1_F16_D128	1	16	128
C2_F4_D64	2	4	64
C2_F4_D128	2	4	128
C2_F16_D64	2	16	64
C2_F16_D128	2	16	128

Al realizar el entrenamiento se obtuvieron las medianas que se muestran en el siguiente gráfico.

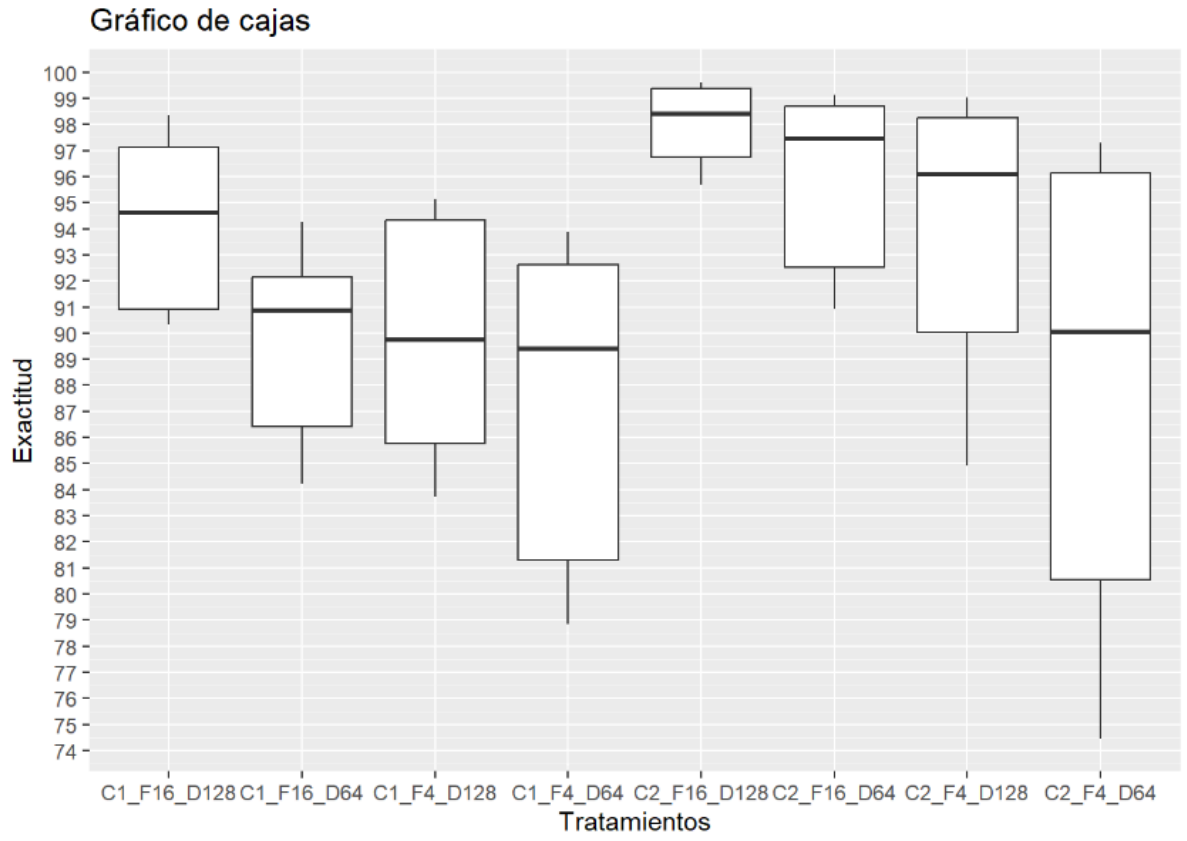

Figura 3. Gráfico de cajas con las medianas de la exactitud correspondiente a los tratamientos

Lo que permitió concluir que el mejor tratamiento era el de 2 capas convolucionales, 16 filtros y 128 neuronas en la capa densa.

Posteriormente se aplicó la metodología de superficie respuesta para encontrar la mejor arquitectura de red neuronal que maximice el rendimiento, tomando como punto medio el hallado en el experimento factorial mencionado anteriormente. Para ello solo se tomaron en cuenta los factores Número de convoluciones y Número de filtros aplicados en cada convolución con los valores de la tabla 3 y se establecieron las neuronas de la capa densa con un valor fijo de 128.

Tabla 3. Factores

Factor	Configuración propuesta	Efecto predicho
Número de capas convoluciones	[1, 2, 3]	La exactitud de predicción aumentará a mayor número de convoluciones.
Número de filtros	[4, 16, 28]	La exactitud de predicción aumentará a mayor número de filtros.

De la combinación de los factores se generaron los siguientes tratamientos:

Tabla 6. Tratamientos

Identificador	Convoluciones	Filtros
C1_F4	1	4
C1_F16	1	16
C1_F28	1	28
C2_F4	2	4
C2_F16	2	16
C2_F28	2	28
C3_F4	3	4
C3_F16	3	16
C3_F28	3	28

Al realizar el entrenamiento se obtuvieron las medianas que se muestran en el siguiente gráfico.

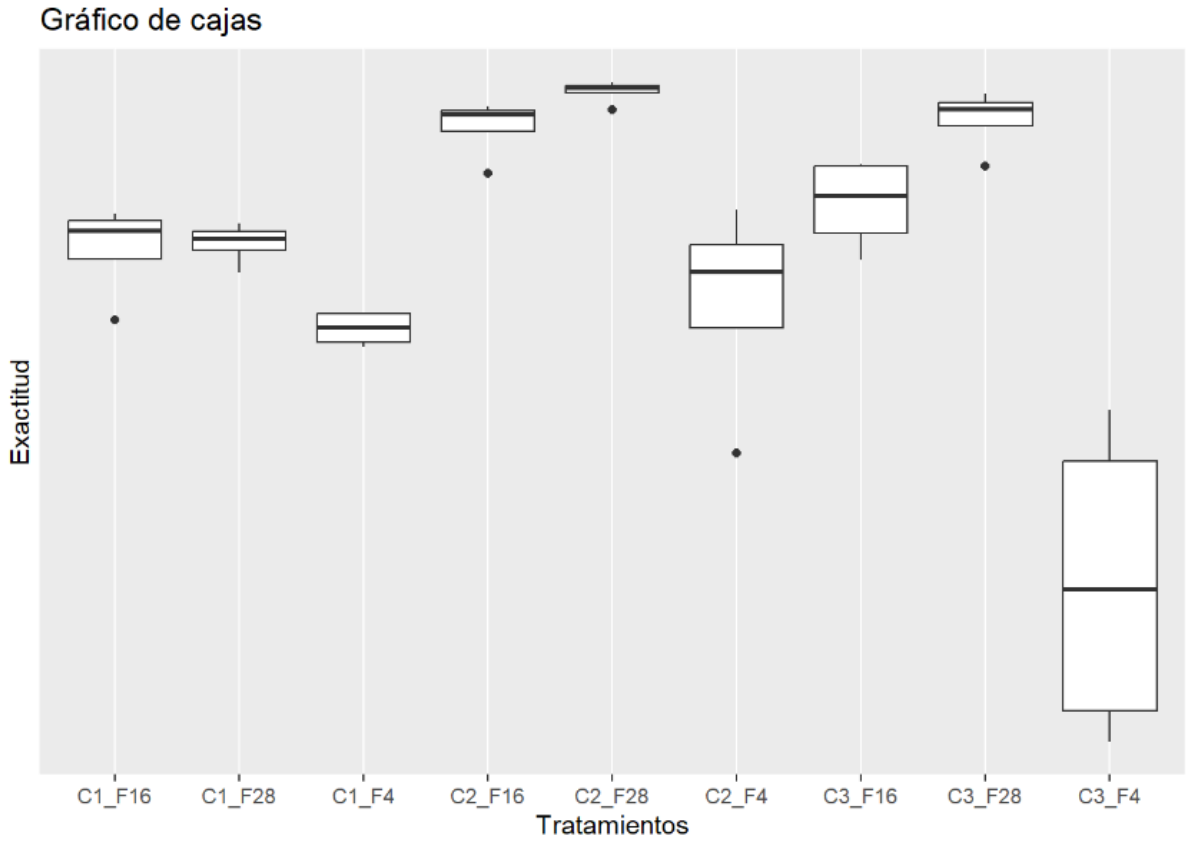

Figura 4. Gráfico de cajas con las medianas de la exactitud correspondiente a los tratamientos

De lo que se podía determinar que el mejor tratamiento era el de 2 capas convolucionales y 28 filtros.

Aplicando la metodología de superficie respuesta se halló que la configuración que maximiza la exactitud era la de 2 capas convolucionales y entre 27 y 20 filtros como se muestra en el siguiente gráfico.

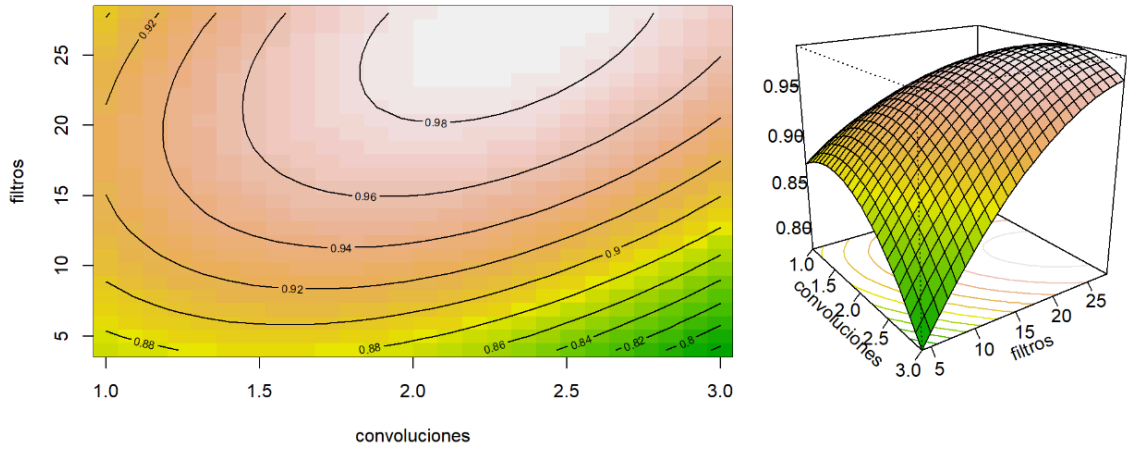

Figura 5. Gráfico de superficie respuesta

4. Resultados

Módulo de detección

Para el módulo de detección se obtuvo una exactitud de entrenamiento del 60% y una exactitud de validación del 58%.

Aplicación

En el uso de la aplicación el 50% de las pruebas realizadas detectaba las señales de tránsito y el 50% de cuando las detectaba las reconocía correctamente.

5. Acceso a datos

Los cuadernos de google colab y los conjuntos de dato de entrenamiento se pueden encontrar en https://github.com/aorlandolopezudea/trafficsignrecognitionML.