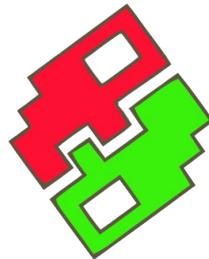


# FUNDACIÓN TARPUY

SEGUNDO NIVEL INGENIERIA

SEGUNDO SEMESTRE 2023



---

## Redes Neuronales

Aplicada a la detección de señales de tránsito

---

Tutor: Leandro Borgnino

Integrantes: Santiago Medina, Esteban Suarez y Alfonso Mouton

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación . . . . .	2
1.2. Objetivos . . . . .	2
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Detección de Objetos . . . . .	4
2.2. Clasificación de la imagen . . . . .	4
2.3. Clasificación con Localización . . . . .	5
2.4. Etapa de la detección de objetos . . . . .	6
2.4.1. Arquitecturas Integradas . . . . .	6
2.5. Ventana deslizante . . . . .	6
2.5.1. Otros datos . . . . .	8
2.6. Intersección sobre la Union (IoU) . . . . .	8
2.7. Non-Max Supression . . . . .	9
2.8. R-CNN . . . . .	10
2.9. Fast R-CNN . . . . .	10
2.10. Faster R-CNN . . . . .	11
<b>3. Implementación</b>	<b>12</b>
3.1. Extractor de características . . . . .	12
3.2. Dataset y Modelo ROI “Region Of Interest” . . . . .	12
3.3. Dataset y Modelo del Clasificador . . . . .	14
<b>4. Resultados</b>	<b>16</b>
4.1. Extractor de características . . . . .	16
4.2. Region Of Interest . . . . .	16
4.2.1. Métricas . . . . .	16
4.3. Modelo de Detección y Clasificación de Señales . . . . .	17
4.4. Arquitectura del modelo total . . . . .	18
4.5. Plataforma de implementación . . . . .	18
<b>5. Conclusión</b>	<b>19</b>

## Introducción

### 1.1. Motivación

En el vertiginoso avance de la tecnología, la aplicación de redes neuronales en el procesamiento de imágenes ha emergido como un catalizador revolucionario en diversas disciplinas. Entre las múltiples facetas que abarca esta amalgama de inteligencia artificial y visión computarizada, la detección y clasificación de señales de tránsito se destaca como un campo de estudio de gran relevancia e impacto práctico. La seguridad vial es una preocupación global de suma importancia, y el tráfico vehicular se presenta como un escenario dinámico y complejo donde la correcta interpretación de señales juega un papel crucial. La detección automatizada y la clasificación precisa de señales de tránsito no solo pueden potenciar la eficiencia de los sistemas de transporte, sino que también desempeñan un papel esencial en la prevención de accidentes y la mejora de la movilidad urbana.

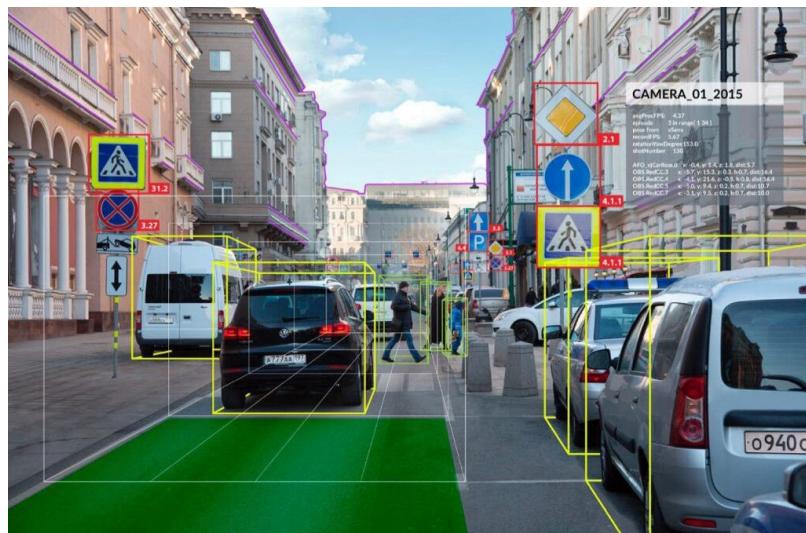


Figura 1: Reconocimiento de Objetos.

El crecimiento exponencial de datos visuales en entornos urbanos y la necesidad de respuestas rápidas ante señales cambiantes hacen imperativa la adopción de enfoques avanzados. Acá es donde las redes neuronales toman importancia, porque su capacidad para aprender patrones complejos a partir de grandes conjuntos de datos permite el desarrollo de modelos capaces de discernir con precisión las señales de tráfico en imágenes, incluso en condiciones adversas. En este contexto, la motivación subyacente es impulsar la aplicación de redes neuronales en el ámbito específico de la detección y clasificación de señales de tránsito.

### 1.2. Objetivos

Los objetivos del proyecto son los siguientes:

1. **Pre-procesamiento de los datos:** Se llevará a cabo un pre-proceso en los datasets que tendrán relevancia en el entrenamiento de los distintos modelos.
2. **Implementación de Arquitecturas Neuronales:**
  - a) **Desarrollo del modelo VGG-16:** Desarrollar la arquitectura de la red neuronal VGG-16 para la obtención de mapas de características, quitando capas para obtener resultados que necesitaremos para la siguiente etapa.
  - b) **Desarrollo del modelo ROI-Pooling:**
    - Diseñar la arquitectura de la red neuronal ROI Pooling para identificar la presencia de señales de tránsito en los mapas de características obtenidos por VGG-16.

- Entrenar la red ROI Pooling con diversos conjuntos de mapas de características generados a partir de imágenes de tráfico.

c) **Implementación del modelo de clasificación:**

- Investigación y evaluación de diferentes arquitecturas de redes neuronales. Se seleccionará la arquitectura más adecuada para el clasificador de señales de tránsito, teniendo en cuenta la precisión, la eficacia computacional y el tamaño del modelo.
- Entrenar la red de clasificación utilizando un conjunto de datos exclusivo que incluye imágenes de señales de tránsito etiquetadas.

3. **Integración de las redes neuronales:** Integrar las arquitecturas VGG-16, ROI Pooling y la red de clasificación para formar un modelo coherente y funcional de clasificación de señales de tránsito.

4. **Evaluación del rendimiento del modelo:** Evaluar la precisión y eficacia del modelo completo mediante métricas relevantes de clasificación.

5. **Comunicación:** Analizar críticamente los resultados, identificar posibles mejoras y proporcionar recomendaciones para futuras investigaciones.

## Marco Teórico

### 2.1. Detección de Objetos

La detección de objetos constituye una rama esencial en el campo del procesamiento de imágenes y la visión por computadora. Su objetivo principal es discernir y localizar la presencia de uno o más objetos dentro de una imagen completa, asignándoles una identidad específica. Las técnicas de detección de objetos se han desarrollado de manera significativa en respuesta a la creciente necesidad de sistemas capaces de comprender y responder a entornos visuales complejos. Uno de los enfoques más destacados y exitosos en este ámbito es el uso de redes neuronales convolucionales (CNN), que han demostrado una eficacia excepcional en la extracción de características relevantes de las imágenes. La arquitectura típica para la detección de objetos a menudo involucra dos etapas cruciales: la generación de propuestas y la clasificación de esas propuestas. En la primera etapa, se utilizan técnicas como Region Proposal Networks (RPN) para proponer regiones de interés que podrían contener objetos. Posteriormente, estas regiones son clasificadas y refinadas utilizando capas de clasificación y regresión.



Figura 2: Reconocimiento facial

Un enfoque más reciente y potente en la detección de objetos es la utilización de modelos de detección de objetos de una sola etapa, como YOLO (You Only Look Once) y SSD (Single Shot Multibox Detector). Estos modelos permiten la detección de objetos en tiempo real al abordar la tarea de manera conjunta, prediciendo las clases y las ubicaciones de los objetos de una sola vez. Además, existen distintos tipos de detecciones y clasificaciones:

- Clasificación de la imagen.
- Clasificación con Localización.
- Detección.

### 2.2. Clasificación de la imagen

En líneas generales, la clasificación de imágenes implica el uso de CNN seguidas de capas totalmente conectadas. Estas redes convolucionales son particularmente eficientes en la extracción de características relevantes, como bordes, texturas y patrones, mientras que las capas totalmente conectadas permiten la interpretación global de estas características para realizar la clasificación final. Por ejemplo, en el contexto específico de la detección de vehículos, el objetivo de la clasificación se centraría en determinar la probabilidad de que la imagen contenga un automóvil. Esta probabilidad se obtiene al alimentar la imagen a través de la red neuronal, que ha sido previamente entrenada para reconocer patrones asociados con la presencia de vehículos.

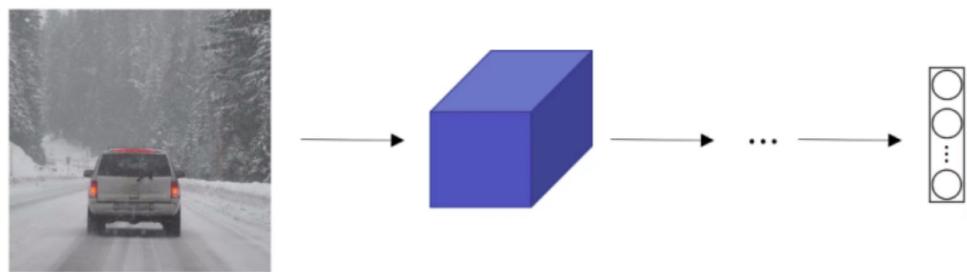


Figura 3: Representación del proceso de clasificación

### 2.3. Clasificación con Localización

Para refinar aún más esta tarea, se pueden agregar neuronas de salida adicionales que proporcionen información sobre la localización del vehículo. Este enfoque, a menudo utilizado en sistemas de detección y clasificación de objetos, utiliza una bounding box (caja delimitadora) para representar la región en la que se encuentra el objeto de interés. En el caso de la detección de vehículos, esta **bounding box** se define mediante cuatro valores: **bx** (coordenada x del centro), **by** (coordenada y del centro), **bh** (altura) y **bw** (ancho)

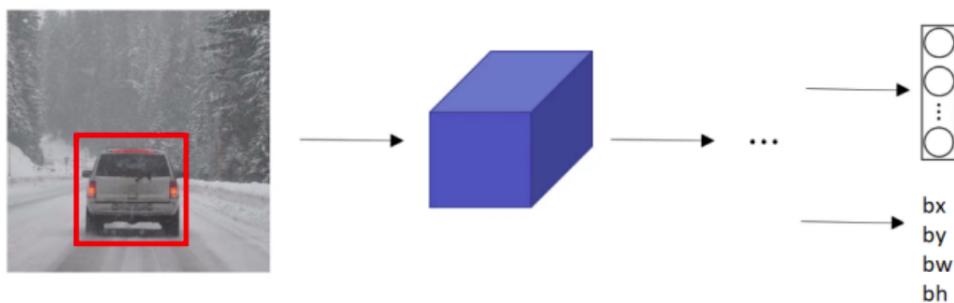


Figura 4: Representación del proceso de clasificación con localización

La inclusión de esta información espacial permite no solo identificar la presencia de un vehículo en la imagen, sino también delimitar su ubicación precisa. Este enfoque, combinado con técnicas avanzadas de clasificación, potencia la capacidad del sistema para comprender la escena visual en su totalidad y, en el caso específico de la detección de vehículos, proporcionar información detallada sobre su posición en la imagen.

Por ejemplo, la salida de la red de clasificación con localización, considerando tres clases de objetos diferentes, sería la siguiente:

$$\mathbf{y} = [p_c \ b_x \ b_y \ b_w \ c_1 \ c_2 \ c_3]^T$$

Si ahora planteamos la función de pérdida de la salida (con MSE) tenemos:

- Si  $p_c = 1$

$$L(\mathbf{y}, \hat{\mathbf{y}}) = (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots + (\hat{y}_8 - y_8)^2$$

- Si  $p_c = 0$

$$L(\mathbf{y}, \hat{\mathbf{y}}) = (\hat{y}_1 - y_1)^2$$

Por lo general se utiliza la función de pérdida log likelihood para las clases, mse para las coordenadas de la región limitante y logistic regression para  $p_c$ .

## 2.4. Etapa de la detección de objetos

En el campo de la detección de objetos, las arquitecturas aplicadas a menudo se estructuran en dos etapas distintas, cada una desempeñando un papel crucial en el proceso global de reconocimiento y localización de objetos en imágenes. Estas etapas son la detección de la región y la detección y clasificación del objeto.

1. Detección de la Región: En la primera etapa, la detección de la región se centra en identificar áreas candidatas que podrían contener objetos de interés. Diversas técnicas han sido desarrolladas para esta tarea, entre las cuales se destacan la ventana deslizante y la búsqueda selectiva, entre otras similares. La ventana deslizante implica el escaneo sistemático de la imagen mediante una ventana móvil, evaluando cada región para determinar la probabilidad de contener un objeto. Por otro lado, la búsqueda selectiva utiliza propuestas generadas previamente para enfocarse en áreas más prometedoras de la imagen, reduciendo así la carga computacional.
2. Detección y Clasificación del Objeto: Una vez identificadas las regiones de interés, la siguiente etapa involucra la detección y clasificación del objeto dentro de estas regiones. Aquí, se pueden utilizar clasificadores clásicos o redes neuronales convolucionales entrenadas específicamente para reconocer patrones visuales asociados con categorías de objetos. Este paso es esencial para asignar una etiqueta a cada objeto detectado, proporcionando información sobre su naturaleza y características

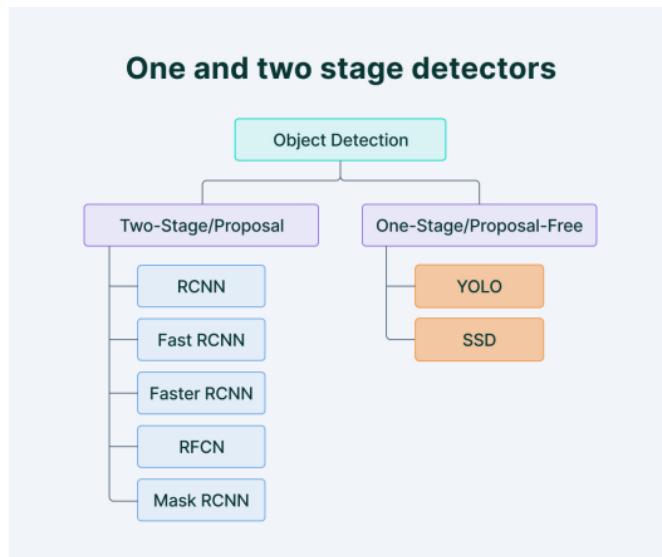


Figura 5: Esquema de detectores por etapas

### 2.4.1. Arquitecturas Integradas

Aunque las dos etapas mencionadas anteriormente son comunes, han surgido arquitecturas más avanzadas que realizan ambas tareas simultáneamente. Estas arquitecturas integran la detección de la región y la clasificación del objeto en una única red neuronal, permitiendo una inferencia más eficiente y rápida. En el panorama de la detección de objetos, la elección entre arquitecturas de una o dos etapas depende de los requisitos específicos de la aplicación y las limitaciones computacionales.

## 2.5. Ventana deslizante

La técnica de ventana deslizante, que implica el desplazamiento de rectángulos a través de una imagen en busca de objetos, se convierte en una estrategia aún más potente cuando se incorporan optimizaciones

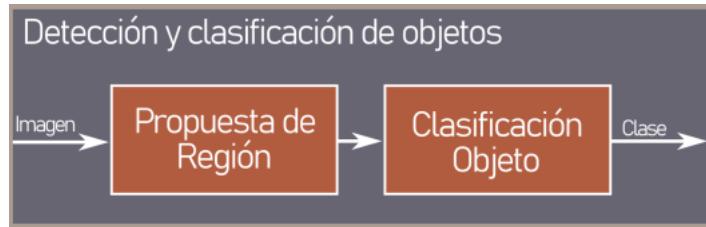


Figura 6: Arquitectura con dos etapas

clave. Este enfoque exhaustivo se beneficia de la flexibilidad para cambiar el tamaño de la imagen o de la propia ventana deslizante, permitiendo la obtención de cajas de delimitaciones más precisas.



Figura 7: Representación de la técnica

Cuando se trabaja con imágenes de tamaños variados, ajustar la escala de la ventana deslizante es esencial para garantizar la detección efectiva de objetos en diferentes contextos. Posteriormente, basándose en las ventanas donde se detecta el objeto en imágenes más pequeñas, se puede escalar nuevamente y unir las detecciones. Este proceso contribuye significativamente a obtener resultados más precisos y detallados, especialmente en escenarios donde la escala de los objetos varía considerablemente.

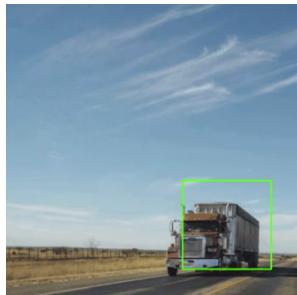


Figura 8: Representación de la técnica

La versatilidad de la ventana deslizante también puede conducir a situaciones en las que varias cajas de delimitaciones detectan el mismo objeto. El desafío radica en seleccionar la mejor candidata entre estas detecciones redundantes. Para abordar este problema, se emplea la técnica de **Non-Maximum Suppression** (supresión de no máximos) entre las candidatas. Esta estrategia se centra en retener únicamente la detección más confiable, descartando las demás para evitar duplicaciones innecesarias.

La evaluación de la calidad de las detecciones se realiza mediante la métrica Intersection Over Union (Intersección sobre Unión, IoU). Esta métrica calcula la proporción entre la intersección y la unión de dos regiones delimitadas por cajas. Al establecer un umbral específico de IoU, se puede determinar la superposición aceptable entre dos cajas para considerarlas como duplicadas o no. Esto asegura que la mejor candidata seleccionada sea la que mejor se ajusta a la verdadera posición y forma del objeto.



Figura 9: Representación de la técnica NMS

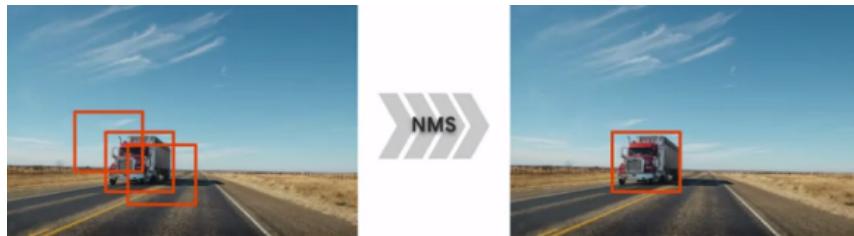


Figura 10: Representación de la técnica NMS

### 2.5.1. Otros datos

La técnica de ventana deslizante, a pesar de ser efectiva en la detección de objetos, enfrenta desafíos computacionales significativos, especialmente en términos de eficiencia. Su alto costo computacional se debe a que cada recorte generado por el desplazamiento de la ventana debe procesarse individualmente por la red convolucional, siendo más intensivo en recursos con imágenes de alta resolución o detecciones más precisas.

La situación empeora al buscar un desplazamiento más preciso, ya que explorar un espacio de búsqueda más fino implica procesar más regiones, aumentando exponencialmente la carga computacional. La ventana deslizante no es óptima para Redes Neuronales Convolucionales (CNN), ya que la complejidad de las CNN hace que el procesamiento independiente de cada recorte sea inefficiente y costoso.

Para superar estas limitaciones, se propone la ventana deslizante convolucional, que optimiza el proceso al introducir la convolución directamente en la ventana. Esto permite a la red compartir cálculos entre regiones superpuestas, reduciendo redundancias y mejorando la eficiencia global del modelo.

## 2.6. Intersección sobre la Unión (IoU)

La métrica de Intersección sobre la Unión (IoU) es un elemento crucial en la evaluación de la precisión y calidad de las predicciones en la detección de objetos. Esta métrica proporciona una medida del grado de superposición entre dos regiones delimitantes, ofreciendo una indicación clara de la similitud y, por ende, de la efectividad del modelo predictivo.

El cálculo de IoU está dado por la siguiente expresión:

$$IoU = \frac{\text{área de la intersección}}{\text{área de la unión}}$$

Donde el numerador representa el área compartida entre las dos regiones delimitantes, y el denominador representa el área total cubierta por ambas. Este cálculo proporciona un valor normalizado que varía entre 0 y 1, donde 0 indica ninguna superposición y 1 indica una coincidencia perfecta entre las regiones.

**La interpretación del resultado de IoU es directa:**

- A medida que el valor de IoU se acerca a 1, se indica una mayor similitud y precisión en la predicción
- Un IoU de 0.5 se considera un umbral comúnmente aceptado para determinar si una predicción es correcta

- Si el IoU es mayor a 0.5, se considera que la predicción es precisa, lo que implica que la región predicha se superpone significativamente con la región real del objeto.

Esta métrica es especialmente valiosa en situaciones donde es esencial evaluar no solo la detección de un objeto, sino también la precisión de su ubicación y forma predicha. Al establecer un umbral significativo, se puede establecer un estándar para la aceptabilidad de las predicciones, contribuyendo así a la toma de decisiones en la optimización de modelos de detección de objetos.

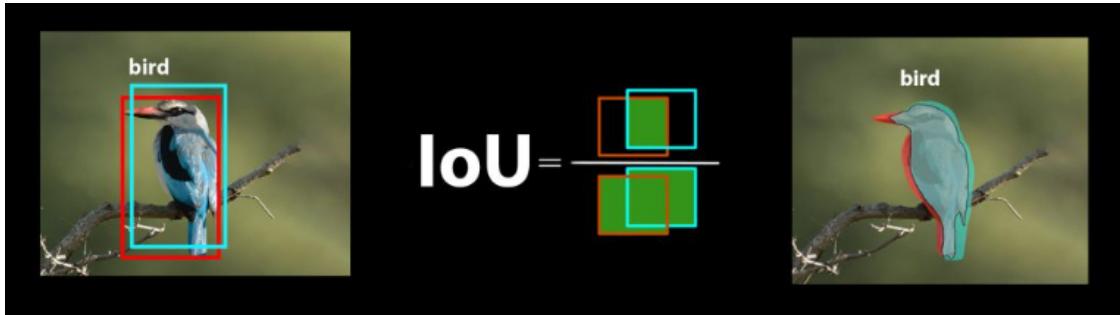


Figura 11: Representación de la métrica IoU

## 2.7. Non-Max Supression

La técnica de Non-Maximum Suppression (NMS) es una estrategia esencial en el postprocesamiento de detecciones, particularmente cuando se enfrenta a la presencia de múltiples celdas o ventanas que comprenden un solo objeto. El objetivo principal de NMS es seleccionar la ventana que mejor encuadre al objeto en cuestión, eliminando redundancias y asegurando una salida precisa. El proceso de Non-Maximum Suppression inicia evaluando las probabilidades asociadas con cada detección (pc) y seleccionando la ventana con la probabilidad más alta. Este paso inicial garantiza que la predicción más confiable se mantenga, sirviendo como referencia para la supresión de detecciones redundantes.

Se calcula el valor de solapamiento (IoU) entre estos rectángulos y el rectángulo de mayor probabilidad.

Aquellos rectángulos que tienen un valor de IoU significativo con el rectángulo de referencia son suprimidos, ya que representan detecciones superpuestas o redundantes.

Este proceso, además de supresión, garantiza que solo se retenga la detección más confiable y precisa, eliminando aquellas que no aportan información adicional significativa. La técnica de Non-Maximum Suppression, por lo tanto, contribuye a la generación de resultados más limpios y coherentes en el contexto de la detección de objetos.

Es importante destacar que, en escenarios donde hay varios objetos a detectar, NMS puede ejecutarse de manera independiente para cada salida, permitiendo así un manejo eficiente de múltiples detecciones en una única imagen, y así asegurando que la selección de ventanas se realice de manera óptima para cada objeto individual.

A pesar de la optimización que proporciona la implementación de la técnica de ventana deslizante en conjunto con redes neuronales convolucionales (CNN), surge una limitación.

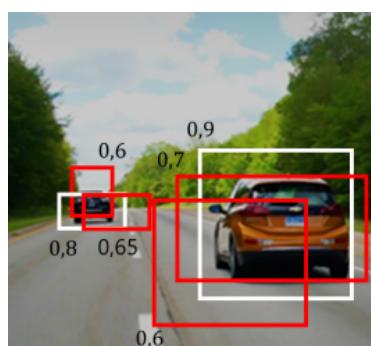


Figura 13: Representación NMS-IOU

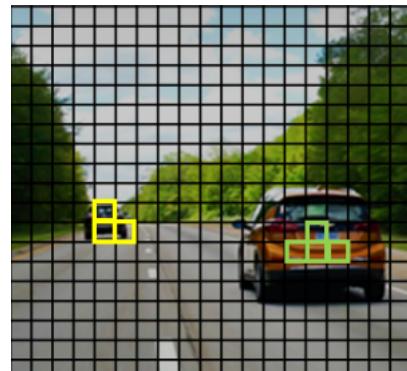


Figura 12: Referencia a la técnica NMS

Esta limitación se manifiesta en situaciones donde ninguna ventana deslizante coincide de manera precisa con la posición real del objeto, como un vehículo. Además, la forma y tamaño predeterminados de la ventana deslizante pueden no ser los más apropiados para delimitar con precisión la región de interés.

La falta de precisión en los límites del cuadro delimitador puede comprometer la exactitud global del sistema de detección de objetos. Esto se convierte en un desafío importante, ya que la correcta delimitación de la región ocupada por el objeto es esencial para comprender su ubicación y forma con precisión. Para abordar esta limitación, se exploran enfoques avanzados que buscan mejorar la predicción de los cuadros delimitadores, permitiendo una detección más precisa y detallada de los objetos en la imagen. Estos enfoques a menudo involucran técnicas de regresión que buscan ajustar dinámicamente los límites del cuadro en función de las características específicas del objeto detectado.

## 2.8. R-CNN

El enfoque R-CNN es una metodología que incorpora redes neuronales convolucionales (CNN) basadas en regiones para implementar la búsqueda selectiva de objetos en una imagen. El proceso R-CNN inicia con la fase de búsqueda selectiva, donde se exploran alrededor de 2000 posibles regiones de interés en la imagen. Esta etapa utiliza la técnica de búsqueda selectiva para identificar áreas prometedoras que podrían contener objetos de interés. Esta primera fase de selección de regiones es esencial para reducir la complejidad computacional y centrar el análisis en áreas relevantes.

Posteriormente, cada región seleccionada se somete a un proceso de extracción de características utilizando una red neuronal pre-entrenada. Esta red, habitualmente diseñada para tareas de clasificación de imágenes a gran escala, se adapta para capturar las características específicas de las regiones de interés. Esta adaptación permite que la red aprenda representaciones significativas de los objetos contenidos en las regiones, contribuyendo a una detección más precisa.

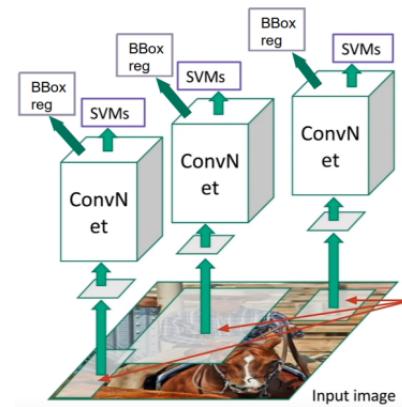


Figura 14: Representación de R-CNN

## Material Consultado [2].

## 2.9. Fast R-CNN

Para superar las limitaciones de velocidad inherentes a la arquitectura R-CNN, surge Fast R-CNN. R-CNN, aunque efectiva, enfrenta desafíos computacionales significativos debido a su técnica de búsqueda selectiva en toda la imagen y al procesamiento lento de 2000 áreas de interés a través de las CNN.

Fast R-CNN aborda estas limitaciones implementando una estrategia más eficiente. En lugar de enviar cada región de interés por separado a la red neuronal, Fast R-CNN integra una CNN que opera en toda la imagen. Esta CNN de toda la imagen extrae características generales y crea feature maps que luego se utilizan para cada región de interés.

Esta arquitectura optimizada ahorra tiempo y recursos computacionales al evitar el procesamiento repetitivo de la imagen completa para cada región. Al utilizar feature maps compartidos, Fast R-CNN logra una mayor eficiencia en la extracción de características, permitiendo una detección más rápida y precisa de objetos en la imagen.

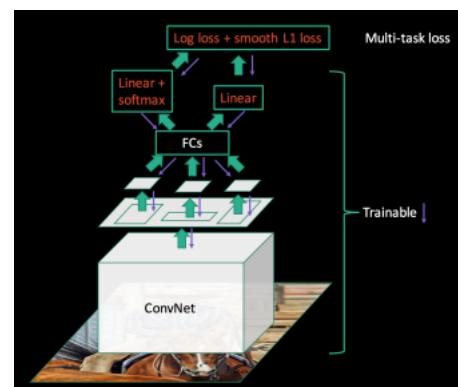


Figura 15: Representación de Fast R-CNN

## 2.10. Faster R-CNN

Faster R-CNN integra de manera eficiente el algoritmo de proposición de regiones de interés directamente en la red neuronal convolucional (CNN).

La principal innovación de Faster R-CNN radica en su capacidad para unificar el enfoque eficiente de extracción de características de Fast R-CNN con un algoritmo de proposición de regiones, creando así una arquitectura integral y ágil. Esta integración permite que la red proponga automáticamente las regiones de interés, eliminando la necesidad de procesos separados y mejorando drásticamente la eficiencia del modelo.

Comparado con su predecesor, R-CNN, Faster R-CNN se destaca por su velocidad. Es 250 veces más rápido que R-CNN, y supera significativamente a Fast R-CNN siendo 25 veces más rápido. Estas mejoras en velocidad no solo aceleran el proceso de detección, sino que también abren posibilidades para la implementación de sistemas en tiempo real y aplicaciones de visión por computadora más exigentes.

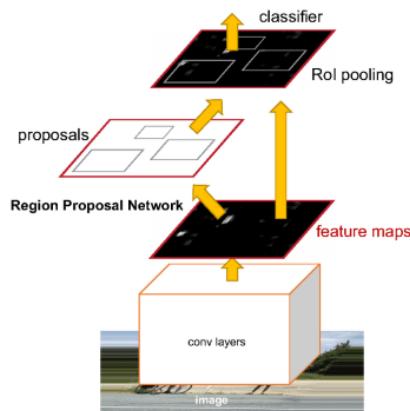


Figura 16: Representación de Faster R-CNN

**Material Consultado [3].**

## Implementación

### 3.1. Extractor de características

Vamos a emplear una red pre-entrenada, específicamente la 'VGG16', como extractor de características para una imagen de entrada con dimensiones de 400x240 píxeles y 3 canales, previamente normalizada. En este caso, hemos truncado la red hasta la capa 'block4-conv3', generando así un mapa de características de 30x50 con 512 características. Con esta configuración, al aplicar la ventana deslizante, obtenemos **30 filas y 50 columnas**. Donde cada ventana corresponde a una dimensión en la imagen original de **8x8 píxeles**.

A continuación se representará gráficamente el modelo truncado para nuestra aplicación:

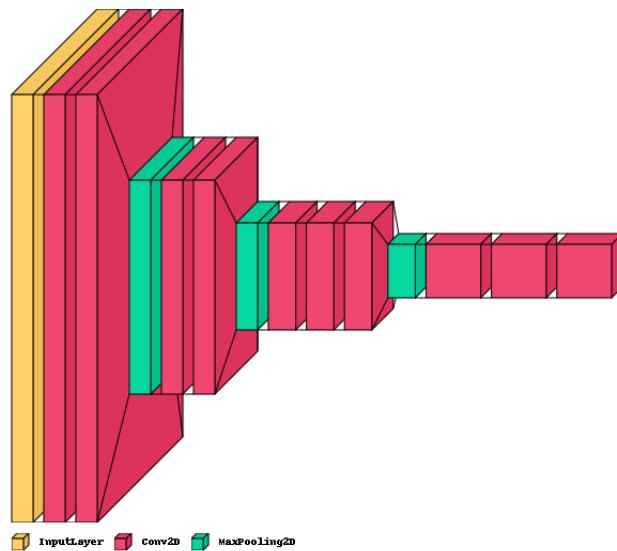


Figura 17: Arquitectura del extractor de características

### 3.2. Dataset y Modelo ROI “Region Of Interest”

Este modelo se compone exclusivamente de una red totalmente conectada que emplea la salida del extractor de características en conjunto con una ventana deslizante. Por lo tanto recibe mapas de características de dimensiones 1x1x512 y genera predicciones binarias para determinar la presencia o ausencia de una señal de tránsito en cada ventana.

A continuación se representará gráficamente el modelo:

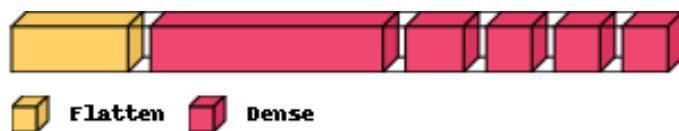


Figura 18: Arquitectura del modelo ROI

Para el entrenamiento de esta red, se procedió a generar un dataset que contiene mapas de características de las señales de tránsito y de las que no son señales, cada uno con su respectiva etiqueta. Se sabe que la salida del extractor de características genera un mapa con una dimensión determinada, donde cada “píxel” o 1x1x512 representa en la imagen original una dimensión de 16x16 píxeles y 3 canales.

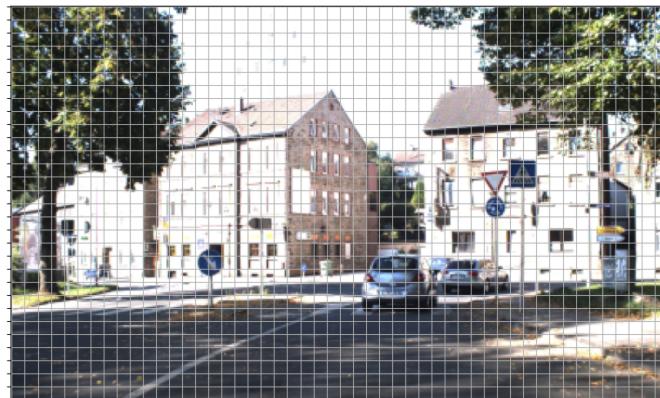


Figura 19: Grilla Generada por el extractor de características

También se conoce la ubicación de cada señal de tránsito, la cual es obtenida en la documentación del dataset.

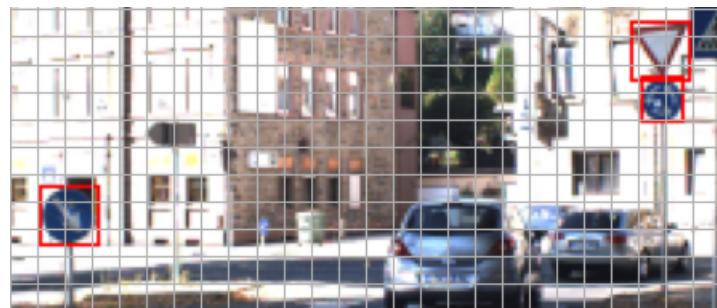


Figura 20: Grilla y Ubicación de las Señales de Tránsito

Por lo que habrá que definir una métrica para determinar si en cada cuadrante de la grilla, este pertenece o no a una señal de tránsito, para eso se consideró la métrica de la Intersección sobre la Unión y la de solamente el porcentaje de intersección:

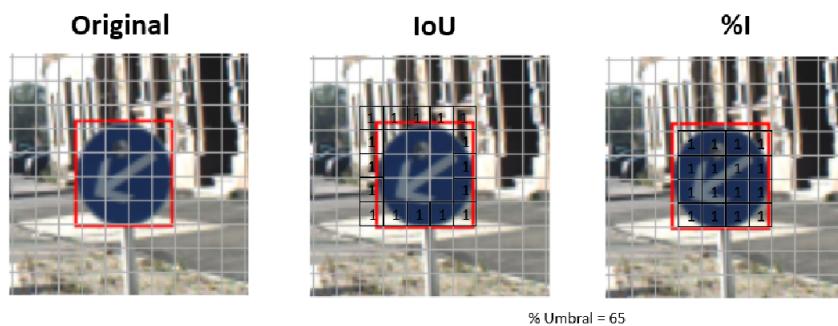


Figura 21: IoU vs. I

Como se aprecia, la métrica del porcentaje de intersección en este caso es mucho mejor, ya que en el caso de la intersección sobre la unión, se eliminarán datos de suma importancia como es el interior de la señal de tránsito.

Para la generación de los mapas de características de las señales de tránsito, simplemente debemos extraer el mapa de característica en cada columna y fila donde el binarizado de la métrica sea 1. En cambio para los mapas de características de aquellas partes donde no se encuentra una señal de tránsito, se utilizó un generador aleatorio de números enteros para colocarlos en la grilla con el cuidado de que este generador no

nos de un par fila-columna tal que sea una señal de tránsito, este generador aleatorio fue puesto con una distribución normal con media la mitad de la grilla y varianza 4; Esto nos permite obtener más información del centro de la imagen y en zonas donde es más probable encontrar una señal de tránsito.

La estructura del dataset es la siguiente:

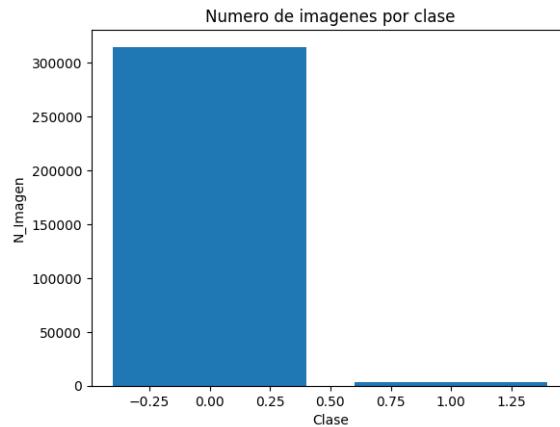


Figura 22: Estructura del Dataset ROI

### 3.3. Dataset y Modelo del Clasificador

Se empleó el conjunto de datos GTSRB, el cual proporciona información tanto para el entrenamiento como para la prueba, incluyendo etiquetas asociadas.

El conjunto de datos de entrenamiento consta de 39,209 imágenes de tamaño variable, representando señales de tránsito capturadas desde diversas cámaras, bajo condiciones de iluminación y ángulos diversos. Este conjunto abarca 43 clases distintas para su clasificación, pero se agregó una clase adicional para eliminar falsos positivos, esta clase es la denominada “Nada” y la obtuvimos mediante el dataset “JCNN” generando recortes de imágenes que no fueran señales de tránsito conociendo las regiones de la imagen que sí contienen señales de tránsito, tal información se recopiló en la documentación de los datasets, por lo tanto, todo este conjunto de datos que se utilizó para entrenar el clasificador posee la siguiente estructura:

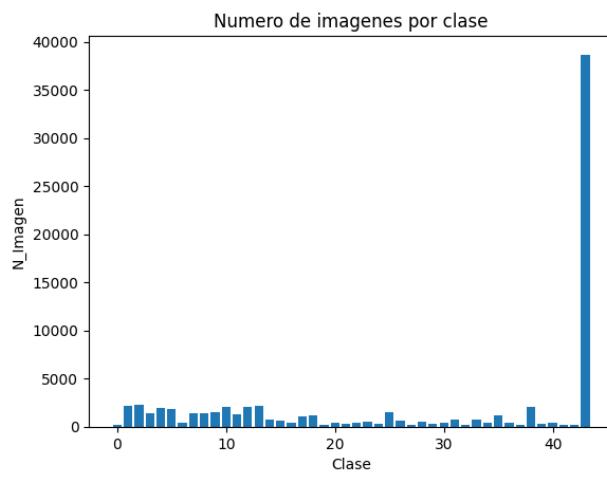


Figura 23: Estructura del Dataset

Es evidente que este conjunto de datos presenta un desequilibrio, lo que significa que no hay una distribución equitativa de datos para cada clase. Esta asimetría se ha diseñado con un propósito específico relacionado con la probabilidad de ocurrencia de cada clase. En consecuencia, la clase con menos datos

tendrá una probabilidad menor de aparecer. Se puede equilibrar este conjunto de datos de forma artificial en el entrenamiento, ajustando las ponderaciones por cada clase.

Previo al entrenamiento del modelo clasificador, cada imagen requiere de un pre-procesamiento; Se debe ajustar su dimensión a 32x32 píxeles y normalizarla. La normalización implica modificar el rango de cada canal de color de 0 a 255 a un nuevo rango de 0 a 1, garantizando así una representación estandarizada de los datos.

En cuanto al modelo del clasificador, definimos la siguiente arquitectura de una red convolucional:

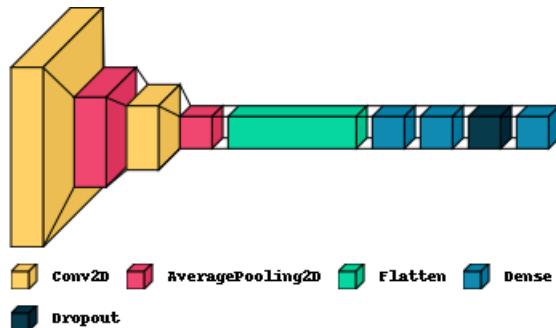


Figura 24: Arquitectura del clasificador

Al entrenar el modelo y establecer una tasa de aprendizaje inicial lo suficientemente baja, se logra:

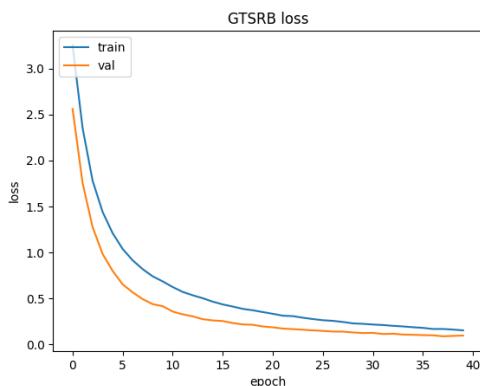


Figura 25: Pérdida del modelo

A continuación, se incluirán algunas predicciones generadas por el modelo propuesto utilizando el conjunto de datos de prueba, con el objetivo de verificar su capacidad de generalización adecuada:



Figura 26: Prueba del modelo

En la imagen previa, se presenta la clase asignada a cada imagen, acompañada del correspondiente porcentaje de confianza proporcionado por el modelo en su respectiva predicción.

## Resultados

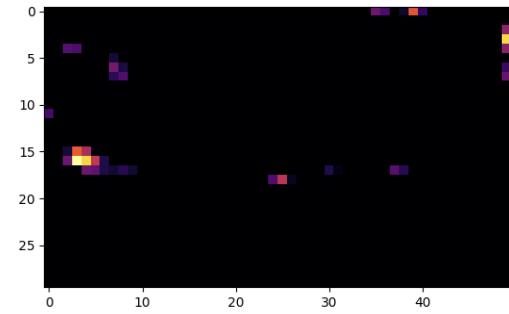
### 4.1. Extractor de características

Como antes se ha mencionado se genera un mapa de  $30 \times 50$  con 512 características, de cada una de las imágenes del dataset GTSRB, gracias a la red neuronal convolucional VGG-16. Algo a destacar es que estas características de las imágenes son a nivel teórico para nosotros los bordes, tamaños, texturas, patrones entre otros. Pero a nivel máquina, lo que puede visualizarse como características son figuras sin sentido.

Por ejemplo tomemos una imagen del dataset GTSRB y veamos un mapa de característica.



(a) Imagen extraída del dataset



(b) Un mapa de características de la imagen

Figura 27: Extracción de un mapa de característica

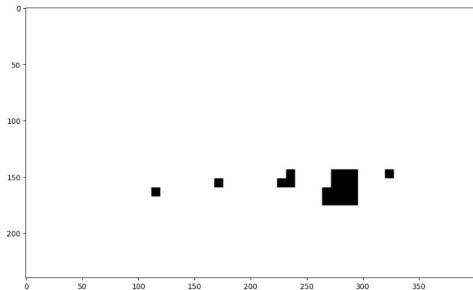
### 4.2. Region Of Interest

Como ya se ha mencionado, la capa ROI Pooling recibe mapas de características de dimensiones  $1 \times 1 \times 512$  generados por la red VGG-16 para cada imagen en el conjunto de datos GTSRB. Estos mapas representan abstracciones a nivel de bordes, tamaños, texturas y patrones. La capa ROI Pooling procesa selectivamente estas características y produce predicciones binarias que indican la presencia o ausencia de señales de tránsito en regiones específicas de la imagen.

Por ejemplo, si tomamos una imagen del dataset GTSRB con su respectiva matriz ROI resulta lo siguiente:



(a) Imagen original



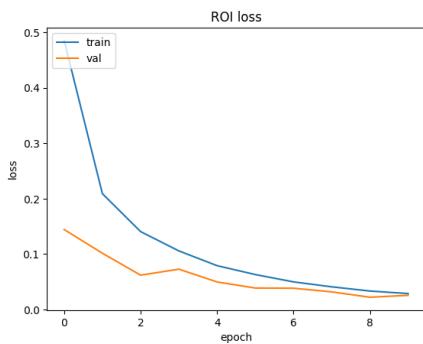
(b) Mapa de unos y ceros de la imagen

Lo que se puede observar entonces es una imagen blanca con pequeñas "manchas negras", que representan aquellas regiones de la imagen en donde puede haber características de interés (en este caso, posibles señales de tránsito).

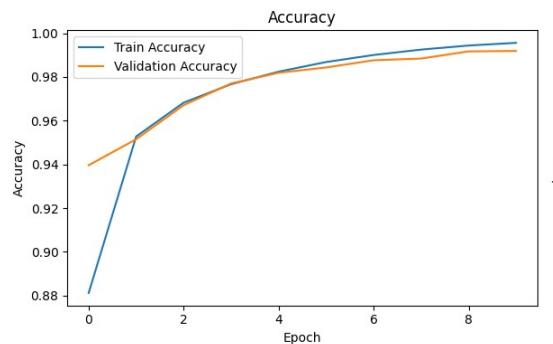
#### 4.2.1. Métricas

A continuación se verán reflejadas algunas métricas que se utilizaron para evaluar el modelo. Cabe recalcar que como el dataset se encuentra desbalanceado (muchos ceros y pocos unos), las métricas de accuracy y matriz de confusión suelen arrojar resultados bastante favorables. Es por ello que por lo general se presta mayor atención a la métrica referente a la pérdida del modelo.

A partir de lo expuesto a continuación por los gráficos de pérdida y accuracy del modelo, puede observarse que el modelo no está sobre entrenado (overfitting) ya que las curvas de entrenamiento y validación convergen a valores muy similares luego del entrenamiento.



(a) Pérdida del modelo



(b) Accuracy del modelo

Para la métrica **matriz de confusión** se ha separado del dataset de entrenamiento un cierto porcentaje de datos para la validación del modelo.

Podemos observar que en el eje X nos encontramos con las **Etiquetas Predichas** por el modelo y en el eje Y con las **Etiquetas Reales**.

La **matriz de confusión** nos dice que el modelo predice correctamente que se han detectado 15655 0's y 147 1's, sin embargo también hay 7 **falsos negativos** y 94 **falsos positivos**.

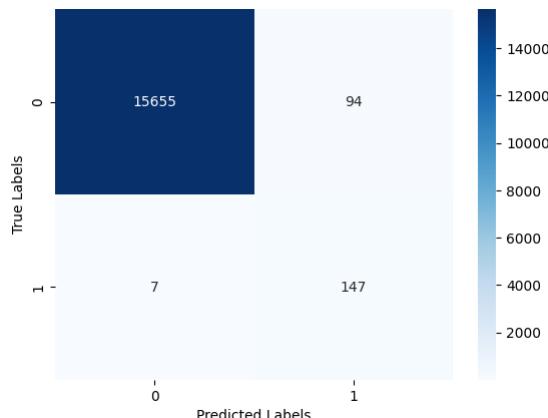


Figura 30: Matriz de Confusión

#### 4.3. Modelo de Detección y Clasificación de Señales

El modelo de detección de señales de tránsito es un sistema completo hecho para procesar imágenes originales y realizar la identificación y clasificación de señales de tránsito presentes en la escena.

Además, la entrada del modelo consiste en la imagen original capturada, que puede contener diversas señales de tránsito en diferentes contextos y condiciones de iluminación.

Por otro lado, la salida final del modelo incluye la imagen original con las señales de tránsito recuadradas y etiquetadas, mostrando visualmente cómo resultan la detección y clasificación de las señales presentes en la escena.

Por ejemplo, así es como el modelo procesa una de las imágenes del dataset:



Figura 31: Procesamiento de una imagen

#### 4.4. Arquitectura del modelo total

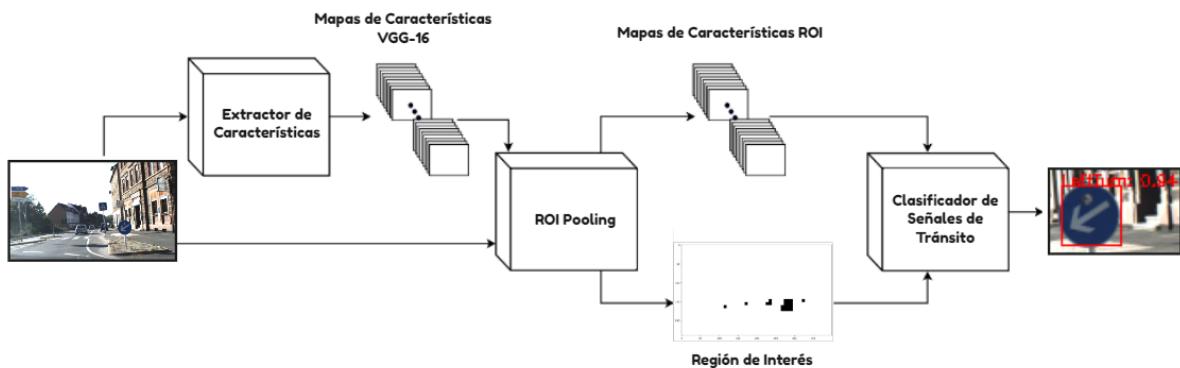


Figura 32: Implementación del modelo final

#### 4.5. Plataforma de implementación

La implementación del trabajo **Redes Neuronales: Aplicada a la detección de señales de tránsito** se llevó a cabo en la plataforma Google Colab. El link se encuentra en [1]

## Conclusión

En este proyecto de detección de señales de tránsito, hemos implementado una arquitectura que utiliza la red neuronal convolucional VGG-16 junto con la capa ROI Pooling. A través de este enfoque, hemos logrado identificar regiones de interés en las imágenes, interpretando estas áreas como posibles señales de tránsito.

La capa ROI Pooling, ha sido de utilidad en el procesamiento selectivo de estas regiones, generando predicciones binarias sobre la presencia de señales. La inclusión de un componente de clasificación ha proporcionado etiquetas asociadas con las señales detectadas.

Los resultados obtenidos indican que la arquitectura es capaz de abordar la tarea de detección y clasificación de señales de tránsito. Sin embargo, y aunque los resultados obtenidos son alentadores, reconocemos la necesidad de mejorar la robustez de nuestro modelo ROI, especialmente en situaciones donde las señales son demasiado cercanas entre sí. Un modelo más sofisticado podría abordar estos desafíos, mejorando la capacidad de discernir señales adyacentes y optimizando la precisión general del sistema.

Adicionalmente, consideramos que el rendimiento de nuestro modelo podría beneficiarse significativamente de entornos de ejecución más potentes. Con recursos computacionales adicionales, podríamos procesar imágenes de mayor resolución, lo que permitiría la detección de señales a mayores distancias.

## Bibliografía

- [1] ALFONSO MOUTON - ESTEBAN SUAREZ - SANTIAGO MEDINA. Implementación: Señales de Tránsito, 2023. [https://colab.research.google.com/drive/1dFNiqWg-2gy6FNCu02\\_yY1tvwy1tcgVI?usp=sharing](https://colab.research.google.com/drive/1dFNiqWg-2gy6FNCu02_yY1tvwy1tcgVI?usp=sharing)*Haz click aquí.*
- [2] LEANDRO, B. Fundación Tarpuy. Filminas de clase: Arquitecturas de redes neuronales aplicadas a segmentación y reconocimiento de objetos en imágenes.
- [3] ROHITH, G. R-CNN, Fast R CNN, Faster R-CNN, YOLO — Object Detection Algorithms, 2018. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>*Haz click aquí.*