Detección de Elementos de Tráfico en Tiempo Real con Deep Learning

Trabajo fin de grado

Grado en Ingeniería Informática

Autor: *Gonzalo Gómez Nogales*

Tutor: *Alberto Sánchez Campos*

Universidad Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

Contenido

Introducción

Existen varias formas de enfocar la detección de objetos en la actualidad. ¿Por qué **deep learning**?:

- Antiguamente se utilizaban métodos y algoritmos de la **visión artificial** como los clasificadores de distancia euclídea, bayesianos, etc, y herramientas como la transformada de Hough o el algoritmo MSER para localizar bordes y formas (P.I.) en imágenes. Pero de esta forma, a pesar de que los resultados eran buenos, no se podían conseguir en **tiempo real**.
- Hasta hace aproximadamente **cinco** años esto era la forma usual de construir detectores, pero en la actualidad, gracias al **deep learning** y en concreto a los avances en la investigación de **redes neuronales convolucionales** se pueden obtener resultados de forma prácticamente **instantánea**.

Objetivos del TFG

1. Investigar y aprender las bases del **deep learning** y sus conceptos más avanzados (detalles).

2. Crear un conjunto de datos personalizado y etiquetado manualmente para resolver un problema conocido como es la detección de objetos de tráfico.

3. Acercarse a la comprensión completa de la detección de objetos en tiempo real implementando un algoritmo más básico como **sliding windows**.

4. Implementar un detector de elementos de tráfico en tiempo real funcional, con diversas aplicaciones posibles y con capacidad de adaptarse a diferentes situaciones.

Desarrollo I

Conjuntos de Datos I

Para conseguir los datos se colocó una cámara en el salpicadero de un coche y se realizaron varios trayectos para captar diversas situaciones en diferentes momentos del día y en **FullHD**.

Se han decidido detectar **9** clases de objetos y se han etiquetado **3172** imágenes.

Los requisitos para elaborar un conjunto de datos con los que entrenar una **red multiclase** son diferentes a los necesarios con un **detector en tiempo real**:

- Para entrenar la red multiclase, las imágenes deben contener únicamente el elemento a clasificar, ya que si aparecen varios, solo se escogerá uno.
- Por otro lado, para el detector final es necesario contar con la imagen completa con todos los elementos visibles, y con un fichero de texto con las regiones donde aparezcan, etiquetadas.

Desarrollo I

Conjuntos de Datos II

Desarrollo II

Ventanas Deslizantes I

El algoritmo consiste en recorrer la imagen a detectar desde la esquina superior izquierda hasta la inferior derecha con recuadros de tamaño prefijado que de forma iterativa clasifican la región que contienen mediante una **red neuronal multiclase**.

Para diferenciar las zonas en las que no aparece ningún objeto relevante se ha añadido la clase **none** en el conjunto de datos usado durante el entrenamiento de los modelos para implementar este algoritmo.

Para que se produzca una detección es necesario **localizar** el elemento en la imagen y **clasificarlo**. En este caso, la clasificación la realiza una red neuronal seleccionada tras una investigación con varias candidatas, y la localización se obtiene gracias a las ventanas que se van moviendo y cuando se clasifica algún elemento relevante en su interior se guardan como detecciones completas.

Desarrollo II

Ventanas Deslizantes II

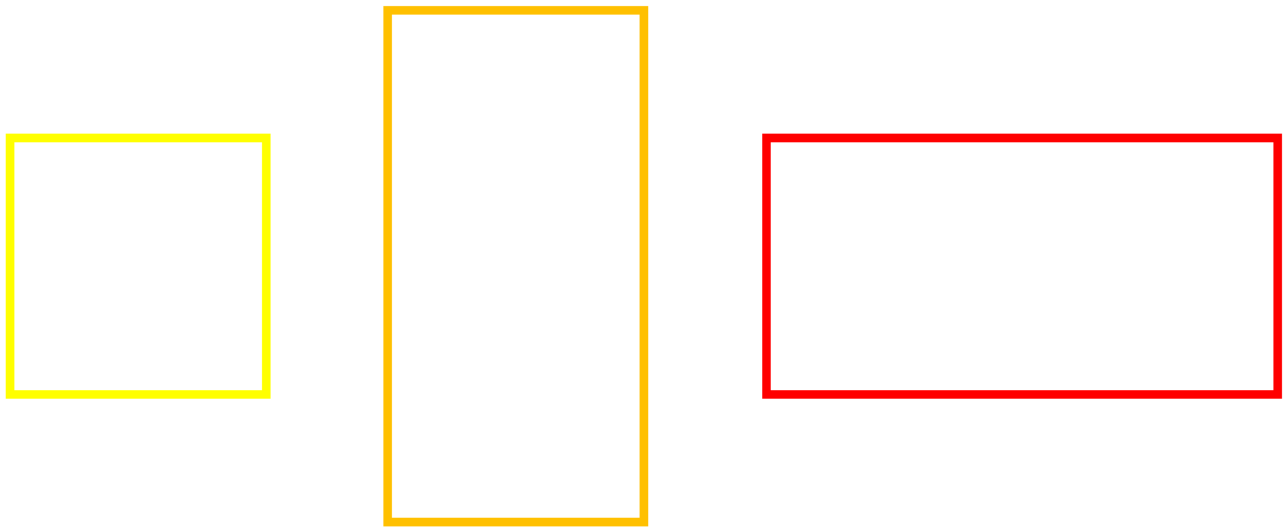
Tabla de mejores entrenamientos obtenidos durante la búsqueda del mejor clasificador para implementar el algoritmo:

Resultados x Arquitectura	v1	v2	VGG16	Inception v3	ResNet 50
Precisión en Entrenamiento	0.991	0.995	0.930	0.998	**0.999**
Precisión en Validación	0.968	0.979	0.918	0.989	**0.992**
Pérdida en Entrenamiento	0.052	0.045	0.235	0.012	**0.002**
Pérdida en Validación	0.124	0.080	0.282	0.045	**0.032**

Desarrollo II

Ventanas Deslizantes III

Para abarcar las diferentes formas y perspectivas en las que se pueden presentar los elementos de tráfico se han seleccionado las siguientes ventanas en 3 escalas cada una:



(a) (b) (c)

Desarrollo II

Ventanas Deslizantes IV

Desarrollo III

Detector en Tiempo Real I

Para implementar el detector en tiempo real se ha escogido la red neuronal **RetinaNet** como base, y se han incluido en el proyecto los pesos de uno de los últimos entrenamientos de la misma con el conjunto de datos COCO realizado por investigadores de Google en sus centros de supercomputación, así como su librería de detección de objetos para poder realizar modificaciones y conseguir de esta forma adaptar su conocimiento anterior (mediante **transfer learning** y **fine tuning**) al nuevo escenario con elementos de tráfico.

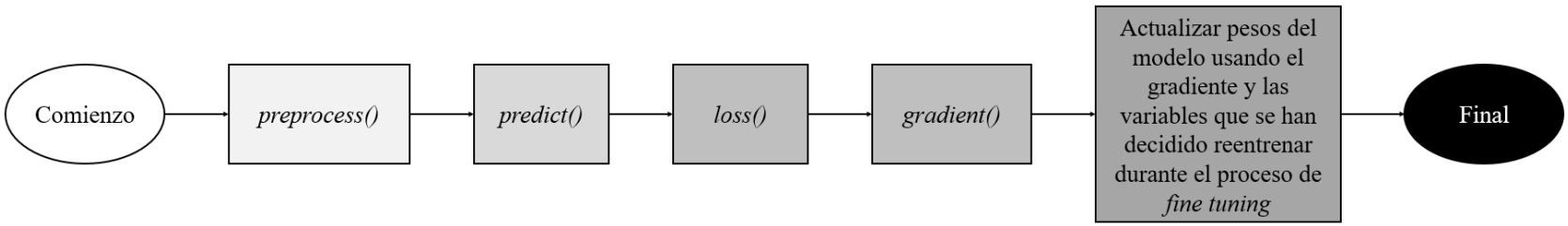
RetinaNet es un detector de disparo único (SSD) lo que significa que procesa una imagen de una pasada sin necesidad de realizar más modificaciones al resultado obtenido tras salir de la red, el modelo devuelve tanto las regiones donde aparecen los objetos detectados como sus clases.

Desarrollo III

Detector en Tiempo Real II

Aplicar tanto **transfer learning como fine tuning** a esta red que cuenta con una estructura tan compleja, requiere implementar el bucle de entrenamiento completamente, a diferencia de lo que sucedía con las redes vistas anteriormente, las cuales permitían el uso de la función *fit()* para realizar el entrenamiento encapsulado.

A continuación se presentan las fases por las que pasa cada imagen del lote de datos seleccionado para que se produzca el aprendizaje durante el entrenamiento:



Resultados I

Ventanas Deslizantes

4 imágenes nuevas creadas a partir de videos de 5 segundos seleccionados esta tarde. puestas en formato esq sup izq, esq sup der, esq inf izq y esq inf der.

Resultados II

Detector en Tiempo Real

Enlace a vídeo en YouTube montado con los 4 escenarios seleccionados, urbano, interurbano, lluvioso y nocturno.

Conclusiones

- El investigar durante más de un año sobre el **deep learning** y sus aplicaciones, enfocado mayormente en la detección de objetos, ha servido para aprender a buscar información sobre temas que se encuentran en desarrollo en la actualidad.
- Gracias a este proyecto se observa claramente la diferencia entre realizar predicciones recibiendo datos **unidimensionales** (regiones de interés) frente a **multidimensionales** (imágenes completas) como entrada, y se explica cómo se relacionan entre ellos para ensamblar modelos que combinan ambos.
- Finalmente, se concluye que para obtener buenos resultados en imágenes de alta resolución se debe contar con modelos que coincidan con la resolución utilizada o que se aproximen lo máximo posible. En el problema de la detección de objetos, cuanto mayor es la resolución del conjunto de datos, se obtiene mayor precisión en las predicciones.

Trabajos Futuros

Finalmente, se numeran algunas aplicaciones que se podrían implementar en el futuro a partir del trabajo realizado en este proyecto:

1. Creación de un modelo completo desde cero, que se entrene con un conjunto de datos mucho más extenso que el actual con cientos de miles de imágenes, para que el usar **transfer learning** no fuera necesario.
2. Una vez entrenado el detector y teniendo la arquitectura se puede transferir su aprendizaje a otros ámbitos.
3. Integración del detector en un vehículo para orientar el proyecto a la conducción autónoma, y realizar ajustes para perfeccionar su funcionamiento desempeñando esta tarea.

Detección de Elementos de Tráfico en Tiempo Real con Deep Learning

Trabajo Fin de Grado

Grado en Ingeniería Informática – Curso 2021-2022

Autor: *Gonzalo Gómez Nogales*

Tutor: *Alberto Sánchez Campos*

Universidad Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática