

Clasificación y detección de señales de tráfico en imágenes

Este proyecto se va a dividir en fases. No es necesario completar todas las fases, sino que cada grupo puede elegir hasta dónde llegar. La forma de resolver el problema tampoco es única y se espera que utilicéis todas las técnicas que habéis aprendido para resolverlo.

Fase 1: clasificación de señales

El dataset de partida es un conjunto de imágenes dividido en 43 clases. Cada clase corresponde a una señal de tráfico. Por cada tipo de señal se disponen de varias muestras diferentes.

Se propone generar un modelo de clasificación que reciba una imagen de entrada y trate de clasificarla en cada una de las 43 clases existentes. Disponéis de algo más de 1200 imágenes, que podréis dividir en train/test para su análisis.

1.1 ¿Cómo entrenar un modelo basado en imágenes?

Hasta ahora es posible que no os haya tocado generar de entrenamiento para sistemas de clasificación supervisadas. Y menos con imágenes como entrada. La primera posibilidad que podríamos hacer para transformar un conjunto de 1200 imágenes en un dataset de entrenamiento podría ser: leer la imagen, transformarla a un vector de NM píxeles e ir concatenando todo hasta construir un dataset de $1200 \times NM$. Sin embargo, esta no suele ser la mejor aproximación. El paso intermedio que se suele realizar es aplicar un algoritmo de extracción de características. Es decir, un algoritmo que reciba una imagen y nos devuelve un conjunto de valores que describa dicha imagen y pueda ser de utilidad para discriminar entre una clase u otra. Por ejemplo, el histograma podría ser una primera opción. Busca información sobre algoritmos de extracción de características. Por ejemplo, HOG es una de las mejores aproximaciones para este problema.

Otra cosa que tendréis que pensar es en la unificación de tamaños de las señales, previa a la extracción de características. Como las imágenes de señales pueden ser de diferentes tamaños, deberéis transformar todas las imágenes a una misma resolución.

1.2 ¿Cómo afectan las condiciones de la imagen en la clasificación?

Algunos algoritmos de extracción de características tienen en cuenta las condiciones de iluminación o contraste de las imágenes, pero otros no. ¿Cómo afecta a la clasificación el realizar un preprocesamiento sobre las imágenes, por ejemplo, una ecualización del histograma previa a la extracción de características? ¿Y aplicar un suavizado sobre las imágenes? ¿Obtenemos mejor o peores clasificadores? ¿Merece la pena utilizar información de color?

Aquellos grupos que se queden en la fase 1 deberán hacer un análisis sobre estos (y otros) aspectos. Se valorará: algoritmos de procesamiento realizados a las imágenes, división del dataset en train y test, algoritmos de clasificación utilizadas, algoritmos de extracción de características, análisis de resultados...

Fase 2: detección de señales

En esta segunda fase nos olvidamos de la clasificación y nos centramos en resolver un problema de detección. Es decir, dada una imagen cualquiera, el programa debe localizar las posibles señales que encuentre. Para resolver esto, vamos a volver a entrenar un clasificador, pero esta vez centrado en detectar qué es una señal y qué no es una señal. Como mínimo, disponéis de un dataset de 1200 imágenes de señales. Ahora deberíais generar un conjunto lo suficientemente grande y variado de imágenes que no contengan señales. De esta manera, estaremos especializando nuestro clasificador en saber qué es y qué no es una señal.

En el dataset de imágenes generales disponéis de un buen conjunto de imágenes reales con escenas de carretera donde aparecen no solo señales, sino aceras, edificios, árboles, coches, etc. Utiliza alguna de estas imágenes para generar un conjunto de pequeños parches (de un tamaño parecido a las imágenes de señales) que contenga objetos de cualquier tipo: ventanas, coches, carretera, árboles...

Una vez generado un buen dataset, ya seréis capaces de entrenar un clasificador que coja una pequeña imagen (por ejemplo 100x100) y nos diga si contiene una señal o no. ¿Cómo haremos ahora la detección en imágenes que son mucho más grandes? Utilizando un algoritmo de ventana deslizante

2.1 Algoritmo de ventana deslizante

Un algoritmo de ventana deslizante se parece mucho a los algoritmos de filtrado que hemos trabajado en clase. Básicamente consiste en fijar un tamaño de ventana, por ejemplo, 100x100, e ir haciéndolo “deslizar” por toda la imagen como si fuese una convolución. En cada paso podríamos tomar esa ventana 100x100 y preguntarle al clasificador si dicho parche contiene una señal o no. En caso en que sí, ya tendríamos generado nuestro detector de señales

Evidentemente, no todo es tan sencillo. Debes tener en cuenta:

- el tamaño de la ventana es determinante. Puedes empezar por un tamaño fijo, 100x100.
- El deslizamiento de la ventana no puede ser de 1 en 1, ya que generaríamos demasiadas ventanas y ralentizaría mucho el programa
- Es fácil que una señal aparezca dentro de varias ventanas, por lo que tendremos múltiples positivos por cada señal.

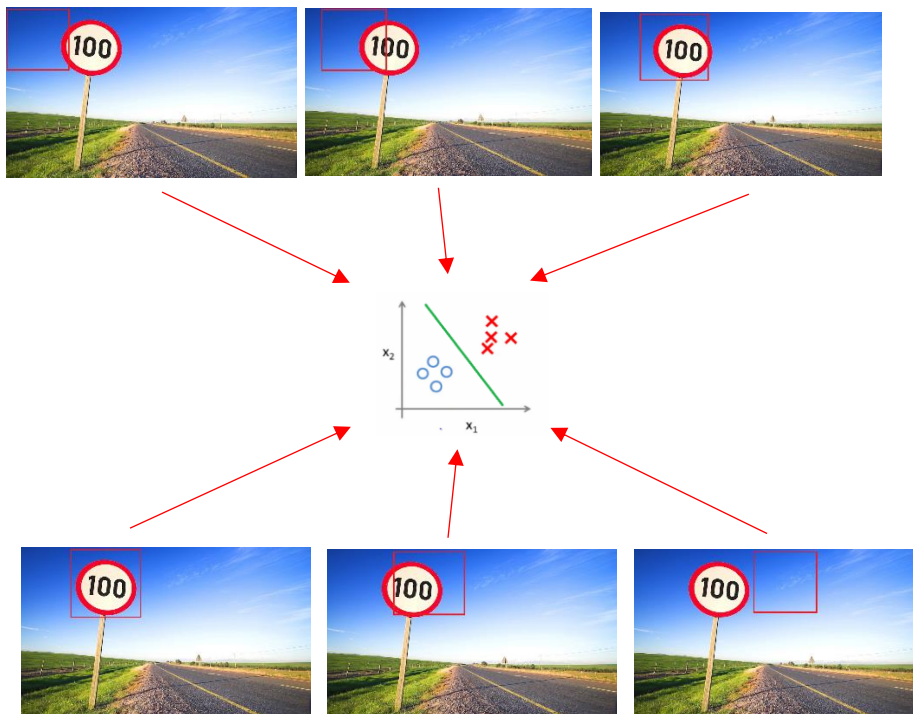
En esta fase se valorará: generación del dataset, algoritmo de clasificación utilizados, métodos de extracción de características, implementación del algoritmo de ventana deslizante, adaptación del algoritmo a múltiples escalas, supresión de no máximos para eliminación de múltiples positivos,...

Ejemplo visual:

Objetivo final:



Cómo funciona el algoritmo de ventana deslizante



Fase 3: localización y clasificación

La última fase sería una combinación de ambas, donde probablemente tengamos que entrenar dos clasificadores diferentes: uno para detectar dónde hay una señal y luego otro que, a partir de aquellas imágenes que detecte, nos diga qué señal es.