Clasificador de señales de tráfico.

Objetivos.

- Aprender a obtener una descriptor de imagen basado en LBP.
- Aprender a entrenar clasificadores supervisados usando la interfaz de OpenCV.
- Aprender a utilizar un clasificador para clasificar una imagen.

Descripción.

Esta práctica consiste en, partiendo un conjunto de imágenes (dataset) de señales de tráfico anotado con un número predefinido de clases de señales, entrenar un clasificador para que “aprenda” a reconocer las señales para, posteriormente, utilizarlo para clasificar nuevas imágenes de señales proporcionadas. En la Figura 1. se muestran algunos ejemplos de señales de tráfico.

Figura 1. Dataset German Traffic Signs. Ejemplo de imágenes de señales de tráfico utilizadas en la práctica. El dataset original puede ser descargado [aqui].

Los pasos necesarios a realizar son:

1. **Extracción de características.** (Implementado en lbp.cpp) Dada una imagen de una señal, se extrae un descriptor de la misma basado en el LBP pudiendo elegir distintas configuraciones de rejilla (esto ya estará hecho en una práctica anterior).
2. **Ciclo Entrenamiento/validación:** (implementado en train.cpp y metrics.cpp).

    a. Entrenamiento: Usando los descriptores de las imágenes pertenecientes a la partición del dataset para entrenar, entrenamos un clasificador usando una configuración de sus hiper parámetros.
    b. Validación: Posteriormente usando los descriptores de las imágenes pertenecientes a la partición del dataset para validar, comprobamos el rendimiento del clasificador y los hiper parámetros. Usaremos como métrica la exactitud (*accuracy*) del clasificador.
    c. Los pasos a y b se realizan en secuencia para distintas configuraciones de los hiper parámetros quedándonos siempre con la configuración proporciona la mejor métrica de exactitud *(sugerencia: puedes utilizar un script de línea de comandos para automatizar este proceso.)*
3. **Test.** (Implementado en train.cpp y metrics.cpp) Usando los descriptores de las imágenes pertenecientes a la partición del dataset para test, comparamos el rendimiento del clasificador (con la mejor configuración de hiper parámetros según el ciclo entrenamiento/validación) con otros clasificadores. Como métricas para comparar resultados, usaremos la exactitud y el ratio de reconocimiento por clase promedio (*mean recognition rate*) y el ratio de reconocimiento de cada clase.
4. **Test nueva imagen (opcional).** De manera opcional, se implementará un programa que carga una imagen, permite seleccionar una ROI donde haya una señal de tráfico (de algunas de las clases entrenadas) y clasificarla.

Para simplificar el problema, se proporcionará un subconjunto del dataset original.

Qué entregar.

Al menos se realizarán todos los pasos anteriores usando el clasificador “Vecino más cercano” y se debe entregar:

- El código fuente correspondiente a los realizado.
- En la carpeta “models/” el mejor modelo conseguido para cada uno de los clasificadores probados.
- En el fichero “LEEME.txt” ejemplos “verbatim” sobre cómo ejecutar la fase de test usando el programa “train” con cada uno de los modelos entrenados.
- Una memoria (guardada en “models/memoria.pdf”) donde se explique los resultados de la experimentación relativos al ciclo entrenamiento/validación y la fase final de test que haya realizado. Se valorará el uso de recursos gráficos (tablas/gráficos de línea o barras).

IMPORTANTE: El dataset no deberá meterse en la entrega.

Evaluación.

Se ha evaluado el descriptor LBP con posibles configuraciones distintas de rejilla usando el clasificador k-NN validando el hiper parámetro k.	50%
Lo anterior más validar usar uLBP y	10%

normalizar las imágenes (mi/max, mean/stdev).	
Lo anterior más validar un clasificador SVM con kernel Lineal.	10%
Lo anterior más validar otros kernels SVM y sus hiper parámetros.	10%
Lo anterior más validar un clasificador Random Trees y sus hiper parámetros.	10%
Implementar el programa “test.cpp” que carga una imagen, selecciona una ROI con una señal y le aplica un modelo de los entrenados.	10%

(* La entrega fuera de plazo supondrá una penalización en la nota obtenida. Cada porcentaje se entiende “hasta” el indicado en función de lo realizado).