

Detector de peatones: código de ejemplo versión 2.0

En esta segunda versión del código del detector de peatones hemos añadido dos nuevas funcionalidades:

1. Ejecutar el detector sobre un conjunto de imágenes y mostrar los resultados sobre todas las imágenes.
2. Evaluar el resultado de la detección sobre un conjunto de imágenes, mostrando las gráficas con la evaluación del rendimiento del detector.

1. Instalación y configuración

Para poder utilizar esta nueva versión del código, simplemente tenéis que descargar y descomprimir el fichero `Pedestrian_Detector_v2.0.zip` que podéis encontrar en la página del curso, en la sección dedicada al código del detector. Este fichero contiene el código en Python y los modelos pre-entrenados.

Para evaluar el detector podéis utilizar las mismas bases de datos que ya os proporcionamos con la primera versión del código. Seguid las instrucciones que hay en la página para descargar las bases de datos de imágenes y configurar el entorno Python, en caso que no lo hayáis hecho previamente.

Esta segunda versión se puede descargar y ejecutar de forma independiente a la primera versión. De todas formas, os recomendamos que os familiaricéis con el código de la primera versión siguiendo las instrucciones y ejercicios propuestos que podéis encontrar en la página del detector.

2. Cambios al código de ejemplo

En esta nueva versión del código se han añadido tres ficheros de código y se han añadido algunos parámetros al fichero `config.py`:

- **test_folder.py:** este módulo aplica el detector sobre un conjunto de imágenes que se encuentran en una subcarpeta determinada, especificada en el fichero `config.py`. El modelo del clasificador debe estar pre-calculado en la subcarpeta `Models`, como resultado de ejecutar el módulo `train_model.py`. El resultado de la detección se guarda en la subcarpeta `Results`.
- **show_results.py:** este módulo toma todos los resultados guardados en la subcarpeta `Results` y genera, para cada uno de las imágenes de test, una nueva imagen, superponiendo a la imagen original los rectángulos con el resultado de la detección.
- **evaluate_results.py:** a partir de los resultados guardados en la subcarpeta `Results`, genera la gráfica con la evaluación del rendimiento que muestra la curva *tasa de error (miss rate) vs. FPI*, tal como se ha explicado en los vídeos.

3. Cambios en el fichero de configuración `config.py`

En esta sección vamos a explicar los parámetros nuevos que se han incluido en el fichero `config.py` en esta segunda versión del código:

- **testFolderPath:** directorio que contiene las imágenes que se van a utilizar como test.
- **annotationsFolderPath:** directorio que contiene la anotación con la localización correcta de las imágenes de test, para poder realizar la evaluación.
- **resultsFolder:** subcarpeta donde se guardan los resultados de las detecciones y las imágenes que se generan con las ventanas detectadas.

4. Datasets

Os proporcionamos tres bases de datos con imágenes de peatones que podéis utilizar para entrenar y ejecutar el detector:

- **Pedestrians:** base de datos completa con imágenes de peatones preparadas para poder realizar el aprendizaje de los modelos. Contiene imágenes positivas y negativas (extraídas del conjunto de entrenamiento de la base de datos de INRIA) ya recortadas todas al mismo tamaño para poder extraer los descriptores y entrenar los modelos. La podéis utilizar para hacer un aprendizaje completo del detector.
- **Pedestrians_dummy:** es un subconjunto de la base de datos anterior, con pocas imágenes positivas y negativas de peatones. Se puede utilizar para hacer un aprendizaje rápido del detector y comprobar cómo pueden influir los diferentes parámetros. Evidentemente, el rendimiento del detector entrenado con esta base de datos no será óptimo.
- **INRIA:** base de datos con imágenes de test. Contiene imágenes completas (extraídas del conjunto de test de INRIA), algunas de las cuales incluyen peatones mientras que otras no. Son las imágenes que podéis utilizar para comprobar el funcionamiento del detector.