

Universidad de Sevilla
Escuela Politécnica Superior de Sevilla



Trabajo Fin de Grado en Ingeniería Electrónica Industrial

**Aforador de vehículos en carretera mediante *OpenCV*
sobre plataforma local.**

Autor: Javier Sánchez-Barbudo Martín

Tutor: Antonio García Delgado

Junio de 2022

Aforador de vehículos en carretera mediante *OpenCV* sobre plataforma local.

ANEXO III: CUERPO DE LAS FUNCIONES EMPLEADAS EN EL ALGORITMO

Departamento de Tecnología Electrónica

Escuela Politécnica Superior de Sevilla

Universidad de Sevilla

Junio de 2022

1. Función de detección (*detect*):

```
def detect(subtraction_left, subtraction_right):
```

Esta función se encarga de analizar los contornos existentes en el fotograma tratado y obtener las *bounding box* adecuadas para asignarlas a los seguidores. También realiza las llamadas a la función de cálculo de distancias.

Recibe: Dos objetos, estos son las regiones de interés de cada carril. A estas regiones de interés se les ha aplicado la sustracción de fondo y las operaciones morfológicas de apertura y cierre.

Retorna: Ningún objeto.

2. Función de seguimiento (*tracking*):

```
def tracking(roi_lane_left_end, roi_lane_right_end):
```

Esta función se encarga de actualizar la detección, con lo cual crea, actualiza y finaliza los seguidores en uso. También realiza las llamadas a las funciones de representación y escritura en fichero.

Recibe: Dos objetos, estos son las regiones de interés de cada carril. En este caso, a las regiones de interés no se les ha aplicado ningún tratamiento.

Retorna: Ningún objeto.

3. Función de escritura en fichero (*writefile*):

```
def writefile (IncrLeft ,IncrRight, contador_veh_izq,  
              contador_veh_der):
```

Esta función se encarga de la apertura y cierre del fichero, escritura en este y obtención de las marcas temporales asociadas a los vehículos aforados.

Recibe: 4 objetos, los dos primeros son banderas o *flags* que indican a que carril se debe dirigir la escritura en fichero. Los restantes son los contadores asociados a los vehículos aforados en cada carril.

Retorna: Ningún objeto.

4. Función de representación (*draw*):

```
def draw (box_izq_1, box_izq_2, box_der_1, box_der_2, FlagRSI1, FlagRSI2,  
         FlagRSD1, FlagRSD2):
```

Esta función se encarga de la representación visual de las *bounding box* proporcionadas por los seguidores en uso, representación de los identificadores asociados a los seguidores empleados y la representación del número de vehículo en seguimiento.

Recibe: 8 objetos, los 4 primeros son las *bounding box* proporcionadas por los seguidores en uso. Los 4 parámetros restantes son banderas o *flags* que indican a que carril y seguidor se dirige la representación.
Retorna: Ningún objeto.

5. Función de cálculo de distancias (*distance*):

```
def distance (deteccion_izq, deteccion_der, box_izq_1_anterior,  
             box_izq_2_anterior, box_der_1_anterior, box_der_2_anterior):
```

Esta función se encarga de obtener los centroides de las *bounding box* con las que trabaja, calcular la distancia euclídea entre las *bounding box* proporcionadas por la función de seguimiento y calcular la distancia euclídea entre las *bounding box* proporcionadas por la función de seguimiento y las últimas *bounding box* proporcionadas por los seguidores en uso.

Recibe: 6 objetos, los dos primeros son las *bounding box* obtenidas en la función de detección. Los 4 objetos restantes son las últimas *bounding box* proporcionadas por los seguidores en uso.

Retorna: Dos objetos, ambos son listas. Ambas listas están compuestas por las distancias calculadas durante la ejecución de la función (en caso de no calcularse, la distancia se mantiene en cero).