

Procesamiento de datos Line-Detector

Mario

August 15, 2022

Puse el código de Jupyter y las etiquetas en un repo en github.
<https://github.com/iMarioSL/Lineas-Carril>

1 Procesamiento de datos.

Cuando se realizó el etiquetado de las imágenes se obtuvieron las coordenadas (x_1, y_1) y (x_2, y_2) de los diferentes tipos de líneas, es decir, líneas de carril, línea del borde izquierdo, borde derecho, y la línea de emergencia.

La tarea de la red neuronal será determinar la probabilidad de ocurrencia de la imagen, es decir, si dicha línea aparece o no, para lo cual definimos la variable p_c como variable de probabilidad de ocurrencia, la cual solo toma los valores de 0 y 1, es decir, cuando la línea no aparece en la imagen y cuando sí aparece.

El objetivo es tener los datos en un vector como el que se ve a continuación:

$$y = [p_c, c_1, c_2, c_3, c_4, x_1, y_1, x_2, y_2, w, h]^T \quad (1)$$

Donde:

1. p_c : Probabilidad de ocurrencia
2. c_1 : Línea de carril
3. c_2 : Línea de emergencia
4. c_3 : Borde izquierdo
5. c_4 : Borde derecho
6. x_i : Coordenada x de la línea
7. y_i : Coordenada y de la línea
8. w : Ancho de imagen
9. h : Altura de imagen

Las etiquetas c_1, c_2, c_3 y c_4 toman los valores de 0 y 1, esto depende de que tipo de línea se tenga, notemos que solo uno de estas etiquetas puede tomar el valor de 1, y el resto el valor de 0.

Cuando se realizó el etiquetado de las imágenes, se marcaron líneas con las etiquetas c_1, c_2, c_3 y c_4 . Se estableció el criterio de que a lo más se etiquetaran 4 líneas de carril, 1 de borde izquierdo, 1 de borde derecho, y 1 de línea de emergencia, de modo que tenemos a lo más 7 líneas por imagen, así que a cada imagen se le asigna un vector y descrito en (1). Para este caso se tienen 65 entradas, incluyendo los parámetros w y h .