



Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de Imágenes 1 (PDI)

Informe Trabajo Práctico N°3

Integrantes:

Sol Kidonakis

Betsabé Gomez

Claudia Leguiza

Análisis y Desarrollo del Problema

Ejercicio 1: Detección de Carriles en Videos de Rutas

Descripción del Problema: El objetivo es desarrollar un algoritmo para detectar los carriles en videos de rutas mediante el procesamiento de imágenes.

Métodos Utilizados:

1. Preprocesamiento del Video:

- **Conversión a Escala de Grises:** Se convierte cada frame del video a escala de grises utilizando `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`. Esto simplifica el procesamiento al reducir la imagen a una sola dimensión de intensidad de píxeles.
- **Detección de Bordes con Canny:** Se aplica el detector de bordes Canny con `cv2.Canny(gray, 50, 150)` para identificar cambios significativos en la intensidad de los píxeles. Los bordes detectados son esenciales para identificar los límites de los objetos en la imagen.

2. Definición de Región de Interés (ROI):

- La función `region_de_interes(img, vertices)` se utiliza para definir y aplicar una máscara sobre la imagen. Los vertices definen un polígono que delimita la región de interés donde se esperan encontrar los carriles. El proceso se realiza en tres pasos:
 - **Inicialización de la Máscara:** `mask = np.zeros_like(img)` crea una matriz de ceros del mismo tamaño y tipo que la imagen de entrada.
 - **Creación de la Región de Interés:** `cv2.fillPoly(mask, [vertices], 255)` rellena el polígono definido por vertices en la máscara mask con el valor 255 (blanco), creando así una región blanca sobre fondo negro.
 - **Aplicación de la Máscara:** `cv2.bitwise_and(img, mask)` aplica la máscara a la imagen original utilizando una operación AND bit a bit. Esto significa

que solo los píxeles dentro del área blanca de la máscara se conservan en la imagen original, mientras que el resto se vuelve negro.

3. Transformada de Hough para Detección de Líneas:

- Se utiliza la transformada de Hough probabilística (`cv2.HoughLinesP`) para detectar segmentos de líneas rectas en la región de interés. Esta técnica permite identificar líneas que podrían representar los bordes de los carriles en la imagen.

4. Filtrado y Selección de Líneas:

- Se filtran las líneas detectadas según criterios específicos como la pendiente y la posición en la imagen. Esto ayuda a eliminar líneas irrelevantes y seleccionar aquellas que probablemente formen parte de los carriles en la carretera.
- Se realiza un filtro adicional para descartar líneas horizontales y verticales que no corresponden a los carriles, mejorando así la precisión de la detección.

Ejercicio 2: Generación de Videos con Carriles Resaltados en Azul

Descripción del Problema: El objetivo es generar videos donde los carriles detectados estén resaltados en color azul para una mejor visualización y análisis.

Métodos Utilizados:

1. Dibujo de Líneas sobre el Video:

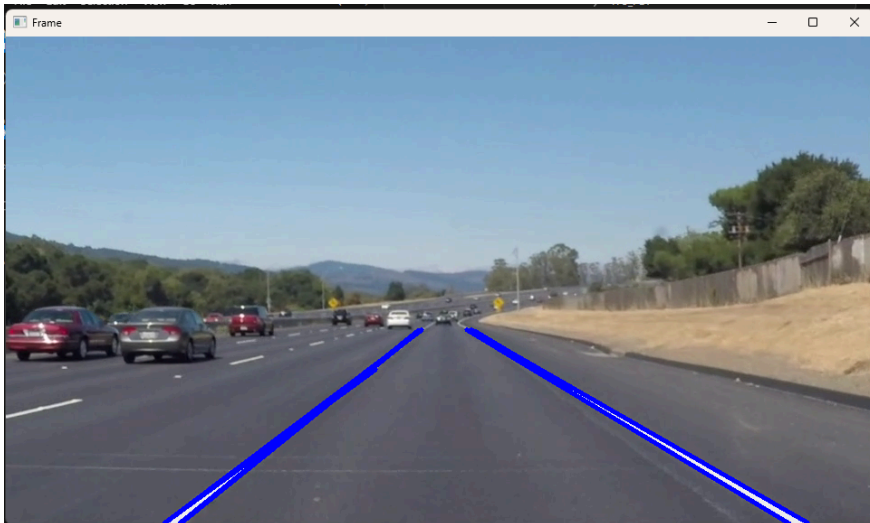
- La función `dibujar_lineas(img, lines, color=(255, 0, 0), thickness=3)` se implementa para dibujar líneas sobre cada frame del video procesado. Las líneas que representan los carriles detectados se dibujan en color azul ((255, 0, 0) en el espacio BGR de OpenCV) para resaltarlas visualmente sobre el fondo.

2. Procesamiento de Video y Visualización:

- Se procesa cada frame del video utilizando los métodos descritos en el Ejercicio 1.
- Se utilizan las funciones `region_de_interes`, `cv2.HoughLinesP` y el filtrado de líneas para detectar y seleccionar los carriles en cada frame del video.

- Las líneas seleccionadas se dibujan sobre cada frame utilizando `dibujar_lineas`, resaltándolas en color azul para indicar los carriles detectados.
- Se genera un nuevo video de salida donde los carriles resaltados en azul son visibles, facilitando la evaluación visual de la precisión y efectividad del algoritmo.

Salida:



Conclusión

Ambos ejercicios están diseñados para abordar el problema de detección y visualización de carriles en videos de carretera de manera

sistemática y efectiva. El enfoque combina técnicas avanzadas de procesamiento de imágenes como Canny y la transformada de Hough con métodos de filtrado y dibujo para lograr una detección robusta y una visualización clara de los carriles identificados.