# 

# Trabajo Práctico 3

## Lucía Masciangelo, Francisco Rodríguez y Barros, Julieta Texier

## 02-07-2024

## Procesamiento de Imágenes I

## Tecnicatura en Inteligencia Artificial

## Requisitos previos para correr los ejercicios:

1. Clonar el repositorio <https://github.com/franciscoryb1/PDI-TUIA-RodriguezYBarros-Texier-Masciangelo>
2. Acceder a la carpeta: cd .\TP3
3. Tener python instalado en el entorno a utilizar. Se puede descargar desde <https://www.python.org/downloads/>
4. Instalar las librerías que serán usadas en los ejercicios:
   * OpenCV: pip install opencv-python
   * Matplotlib: pip install matplotlib
   * numpy: pip install numpy

## Ejercicio 1: Detección de carril

El objetivo de este ejercicio, es poder detectar las líneas del carril en el que va un auto, en una ruta.

Para poder llevar a cabo esto, en primer lugar, importamos las librerías y creamos una función para mostrar las imágenes (**imshow**).

Creamos una función llamada **drawlines**, que toma como argumentos un frame, el ancho y el alto del video. Tomamos un frame del video para poder trabajarlo como una imagen, y luego la generalizamos al video. Esta función consta de varios pasos:

* Aplicamos transformaciones morfológicas a la imagen:
* Recortamos la imagen para quedarnos sólo con la parte que nos interesa (la sección del carril en que va el auto)
* La pasamos a escala de grises
* La binarizamos con un umbral de 200, esto nos deja en blanco sólo las líneas del carril
* Probamos con varios umbrales de **Canny** para detectar bordes. (elegimos los umbrales 0.2\*255 y 0.6\*255)
* Aplicamos gradiente morfológico con un kernel de (10,10) para obtener las líneas más gruesas.

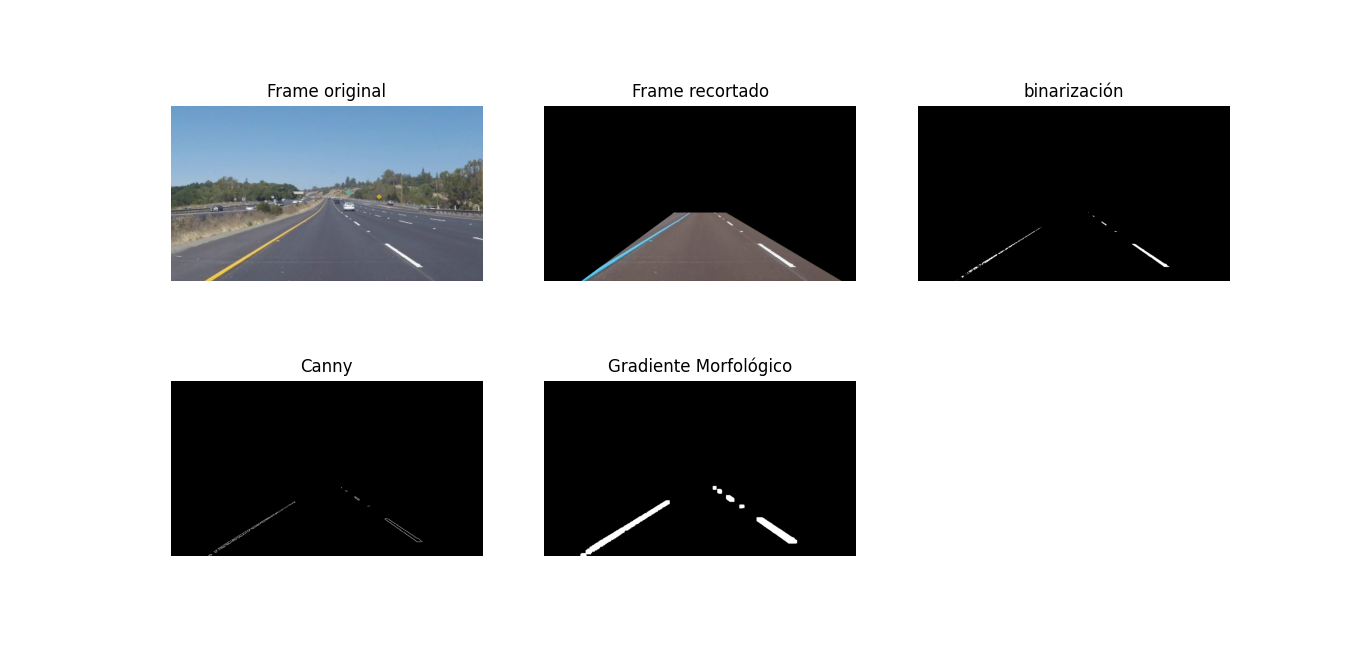
Obtuvimos los siguientes resultados:

**VIDEO 1**

Forma

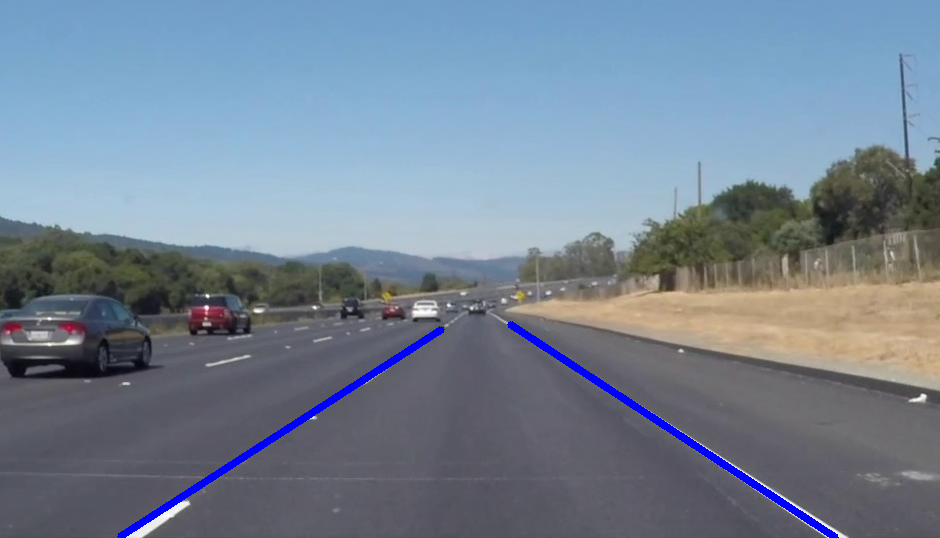
Descripción generada automáticamente

**VIDEO 2**



* Luego aplicamos la función **HoughLinesP** de cv2 con los siguientes parámetros: Rho = 1, Theta = np.pi/180, Threshold = 100, minLineLength = 50, maxLineGap = 10, para detectar las líneas de la imagen.
* La idea es que nos muestre solo una línea de cada lado del carril, entonces como nos detectaba muchas líneas procedimos a ordenarlas por el valor de x1 (ascendente) en el primer punto de cada línea y separarlas en dos listas:
* las que tenían un valor de x1 mayor a 500, en la lista de la derecha; y las que tenían un valor menor a ese, en la lista de la izquierda. (así nos aseguramos de que estén cada una de su lado del carril)
* Una vez hecho esto, para la línea de la izquierda: tomamos las coordenadas del punto 1 que tenían menor x1, con su respectivo y1 para el primer punto de la línea, luego volvimos a ordenar, pero esta vez por la coordenada y2 del segundo punto, para tomar las coordenadas con menor y2, y su respectivo x2, para el segundo punto de la línea.
* Para la línea de la derecha, hicimos lo mismo que para la izquierda.
* Con los puntos ya detectados, (los extremos de las líneas) de cada lado, trazamos la línea que marca el carril, para lograr este resultado:

**VIDEO 1**



**VIDEO 2**

Una carretera con coches

Descripción generada automáticamente

Una vez que logramos marcar las líneas del carril en un frame, llamamos a la función drawlines cuando ponemos a correr el video para que la aplique a todos los frames de mi video.

#yo esto que sigue no lo pondría

Como pasos intermedios, adjuntamos el resultado de las líneas del lado derecho y del lado izquierdo detectadas por separado

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

Diagrama

Descripción generada automáticamente con confianza media