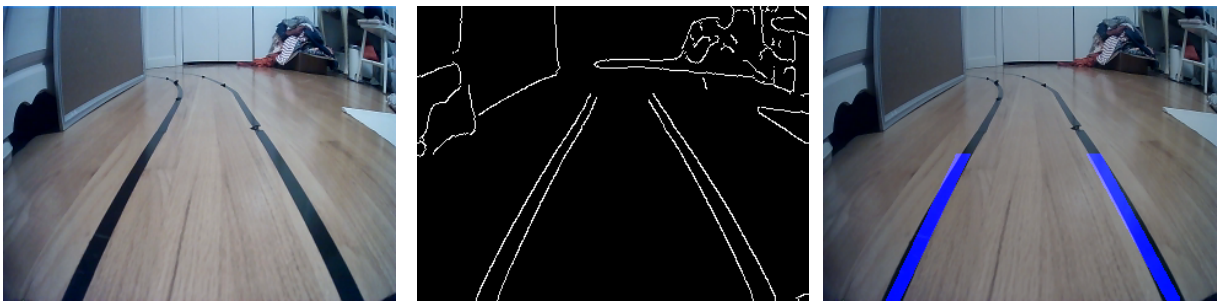
CS5330 CS 5330 Pattern Recognition and Computer Vision

Mini-project 6: Lane Detector

Driving the Line: A Vision-Based Lane Detector Adventure!

Project Overview:

The goal of this mini-project is to develop a lane detection system for identifying lane lines on a road using the Hough Line Transform. The system will be implemented using a given starter code (WebCamSave.py) and will include necessary preprocessing and post-processing steps to accurately detect the two-lane lines (left and right) for driving a car. The detected lanes will be highlighted with red lines in the output. The project also has an optional requirement to submit a recorded video showing the system's performance under real road conditions.



Steps to Complete the Project:

1. **Setup the environment**
   - **Create a folder at Group Project GitHub Repo**: Use your group project repository on GitHub, including a README file with the project description, team members, and setup instructions. Use a folder named "*mini-project6*" within the repository for this project.
   - Set up the development environment by installing necessary libraries, including OpenCV.
   - Familiarize yourself with the *WebCamSave.py* starter code, which is used to capture video frames from a webcam.

2. **Preprocessing**:
   - Convert each frame to grayscale to simplify the image data.
   - Apply Gaussian Blur to reduce noise and make edge detection more accurate.
   - Use Canny Edge Detection to identify the edges in the frame, which are likely to represent lane lines.

3. **Region of Interest (ROI):**
   - Define a region of interest in the frame where the lanes are expected to be, typically a trapezoid covering the bottom half of the image.
   - Mask out everything outside this region.

4. **Hough Line Transform**:

- Implement the Hough Line Transform on the masked edges to detect lines.
- Extract and separate the left and right lane lines based on the slope of the detected lines.

5. **Post-processing**:
   - Average the positions of the detected lines to create smooth lane lines.
   - Draw the detected lane lines on the original frame using red lines.
   - Merge the lane lines with the original frame.

6. **Testing with Input Files:**
   - Test the system on the provided input files (videos or images) to ensure it correctly detects and highlights the left and right lanes.

7. **Optional - Real Road Condition Testing**:
   - Record a video under real road conditions.
   - Run your lane detection system on this recorded video and analyze the performance.
   - [Cautions]
     - Do not test alone (one for driver, one for tester, and one for a camera from behind seat)
     - Do not test on a highway.
     - Do not test over 30 miles per hour.
     - Do not test over 20 seconds.

8. **Project Documentation**
   - Document the code and provide a README file explaining how to set up and use the application, including how to detect lines and the algorithms used for the detection.

Submission:

- **GitHub Repo**: Upload the complete project code to the GitHub repository within the "*mini-project6*" folder, including a README file with setup instructions, usage guide, and a description of the project.
- **Video Capture and Upload**: Record a video demonstrating the execution of your application for each input files and recorded video, showcasing the implemented lane detection. Upload all videos to a cloud service such as YouTube or Google Drive, ensuring the video is publicly accessible. Include the link to the uploaded video in the README file of the GitHub repository.

Grading Criteria:

The assignment will be graded based on the following:

1. **Functionality (15 points)**
   - Successful implementation of real-time frame capture and lane detection (8 points):
   - Accurate preprocessing including edge detection (2 points):
   - Correct implementation of the Hough Line Transform and lane separation (3 points):
   - Successful visualization of lane lines on frames (2 points)

2. **Code Quality (6 points)**
   - Clean, readable, and well-documented code (3 points):
   - Proper use of OpenCV functions and Python language features (3 points)

3. **User Interface and Experience (5 points)**
    - Allows for user control such as starting, stopping, or pausing the lane detection process. Smooth and responsive interaction with arguments (2 points)
    - Clear and informative display (3 points):

4. **Project Documentation (4 points)**
    - Comprehensive README file. Includes setup instructions, usage guide, and an explanation of the lane detection process (2 points)
    - Inclusion of a sample video demonstrating the application (2 points):

Optional - Real Road Video Submission (5 bonus points)

- Real-world testing and performance analysis (5 points):
- Successfully submits a video showing the lane detection system working under real road conditions, with accurate lane identification and performance commentary.