Digital Image Processing
Complex Engineering Problem (End Term project)
(CLO4 -> PLO5)
Real-time Image Analysis for Self-Driving Capabilities
Submission Deadline: 11 May 2025
11:59 P.M

Note: Students should score 50% in OBE specific questions to ensure their accumulated scores towards respective PLOs are above 50%

Advancements in artificial intelligence have significantly enhanced self-driving technology, largely due to the superior performance of deep learning models that have made fully autonomous vehicles a reality. Nonetheless, classical image processing techniques can also deliver robust performance and operate in pseudo real-time, offering a viable alternative to deep learning approaches. In this term project, you will develop the visual component of a self-driving car optimized to run on a Raspberry Pi 5, demonstrating that high-performance, cost-effective autonomous navigation is achievable using traditional methods.

Dataset:
The dataset that we will be using has been collected from the College. You are free to capture more videos/images if you see fit to test and refine your algorithm. The videos can be downloaded from the following link: https://drive.google.com/file/d/1dJYyjc6u08ob8WTFlGNBBppujia_SzGz/view?usp=sharing

In this applied project, you are tasked with **designing and developing** a robust self-driving solution using classical image processing techniques that runs entirely on a Raspberry Pi equipped with a Pi Camera. The system should address the following core functionalities:

- **Stop/Move Decision:** Analyze the environment to determine whether the car should stop or continue moving, integrating obstacle detection as well.
- **Directional Control:** Establish the appropriate direction of movement—left, right, forward, or backward—by interpreting lane geometry and any dynamic obstacles in the car's path.
- **Lane Detection:** Reliably detect road lanes using classical techniques such as edge detection, adaptive thresholding, and Hough transforms, with further refinements like polynomial fitting for curved roads.
- **On-Device Processing:** Ensure that all image processing and decision-making steps are executed in real time on a Raspberry Pi, optimizing computational efficiency through region-of-interest (ROI) restriction, down sampling, and parallel processing where applicable.
- **Output Display and Debugging:** Provide real-time output on a laptop via terminal, including not only the final driving decisions but also intermediate outputs (e.g., processed frames, detected edges, and lane markings) for debugging and system tuning.

This would require you to do a complete study of the problem and read relevant literature. After getting requisite knowledge about the problem, you need to use all the knowledge which you have gathered during this course along with your other mathematics and programming courses to build the complete solution.

In addition, you can add more features to your solution such as streaming from the Pi to the laptop and any other features that you see fit.

Segmentation:

Segmentation is the process of categorizing every pixel in an image into meaningful classes, effectively separating objects from the background. This technique is crucial for accurately identifying obstacles in the vehicle's path. By determining which pixels belong to potential hazards, the system can make reliable decisions about whether to stop or continue moving, ensuring safer navigation.

Decision Making:

Once you have extracted the information from the image, you can then make your decision based on it. This will include whether you should keep moving forward or whether you should stop. In addition, there might be places where you can turn left or right. You should also be able to determine whether an obstacle is far away or close enough to stop or not.

Performance Metrics

The effectiveness of your developed solution will be assessed using the following key performance metrics:

1. Real-Time Performance:
   Evaluate the system's ability to process video frames quickly by measuring the frames per second (FPS).
2. Accuracy:
   Assess the precision of the system in detecting various elements within the video feed, such as lane markings, obstacles, and traffic signals.
3. Robustness:
   Test the system under varying environmental conditions, such as changes in lighting, weather, and road scenarios.

Submission:

Your submission on LMS should have the following:

- Make a zip file including
  - A report containing description about your solution, flow diagram and some sample outputs.
  - All code files.
- All code and associated files are to be submitted via a public GitHub account as well.

Evaluation Rubrics:

Trait	Unsatisfactory (0-3)	Satisfactory (4-6)	Good (7-8)	Exceptional (9-10)
Understanding of Problem 25%				
Problem Understanding Weight = 15%	There is no knowledge of the problem depth.	The depth of the problem is very weak.	The depth of problem is understood to a sufficient level.	The depth of the problem is exceptionall y understood.
Review Weight = 10%	No effort made to cover related work of the selected problem.	Related work covered is very sketchy.	Related work of the problem is covered sufficiently except for the latest developments.	Related work of the problem is covered to an excellent extent.

Presentation and Demonstration Skills 25%				
Demonstration Weight = 25%	The demonstration is poor with no organization and fails at the cross questionin g session.	The demonstration is unorganized and cross questioning session is sketchy.	Good skills while demonstrating project and fairing fairly good in cross questioning	Excellent command of project is shown with exceptional question answer session .
Coding and Documentation				
Specifications Weight = 30%	The program is producing incorrect results.	The program produces correct results but does not display them correctly .	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications	The program works and meets all of the specifications.
Readability Weight = 10%	The code is poorly organized and very difficult to read.	The code is readable only by someone who knows what it is supposed to be doing	The code is fairly easy to read.	The code is exceptionally organized and very easy to follow.
Report (Documentation ` Weight = 10%	The documentation is simply embedded in the code and does help the reader understand the code.	The documentation is simply embedded in the code with some simple header comments separating routines.	The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the	The documentation is well written and clearly explains what the code is accomplishing and how.

Comment of CEP attributes

WP1	Depth of knowledge required	WK4 (Specialist Knowledge) WK5 (Engineering Design) WK6 (Engineering Practice) WK7 (Engineers & Society) WK8 (Research Literature)
WP3	Requires abstract thinking	Require abstract thinking and originality in analysis to choose between the different analysis techniques for generation of best representative attributes in order to perform required grading
WP7	Interdependence	Need to relate multiple concepts learned in other courses like programming, data structures, probability etc