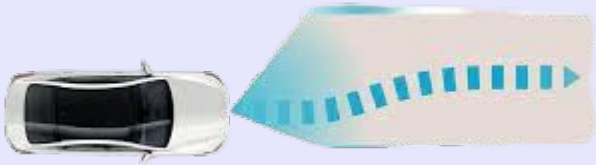
Lane Following Contributions on Autonomous Navigation

IoTeam

Challenging Task Context

Computer Vision System implementation enabled for the detection and tracking of lanes using a stereoscopic camera; as a mean to contribute in the autonomous navigation required tasks to carry on the AMR TEC2 platform.



Order of the Day

01. Project Management

Applied tools and process reinforcement Strategies.

02. Support Systems

Manufactured structures for utilized equipment and circuits (PCBs).

03. Live Data Monitoring

Via Amazon Web Services and Grafana usage.

04. Obstacle Detection

Pure YOLO [COCO dataset] implementation for object identification.

05. Pothole Detection

Trained YOLO implementation for negative obstacles identification.

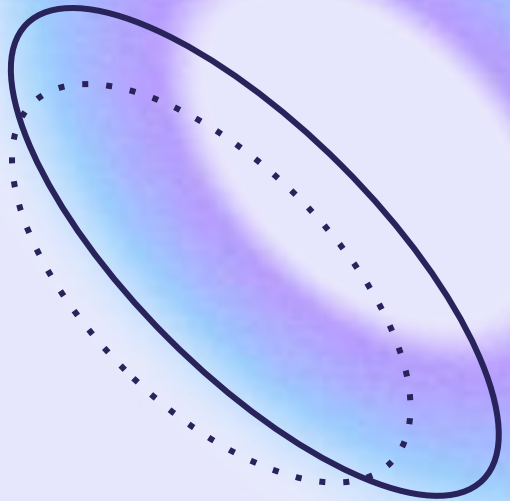
06. Lane Detection

Pix2Pix AI lane detection, image processing,control tasks, and integration proposal for autonomous navigation.

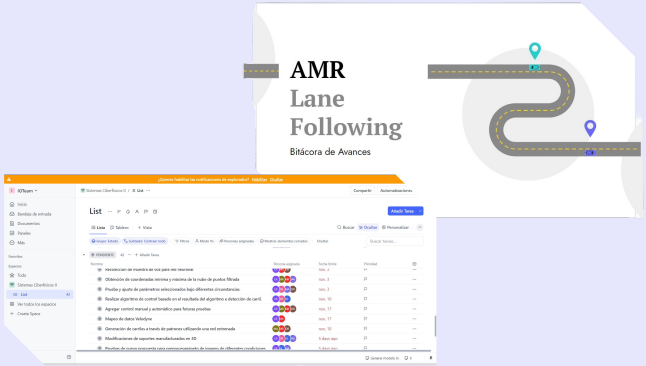
01.

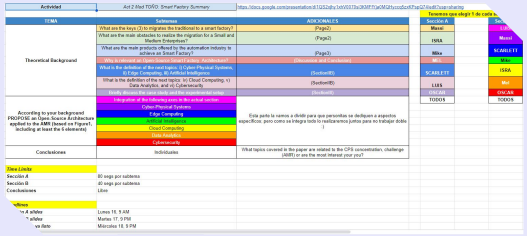
Project Management Tools

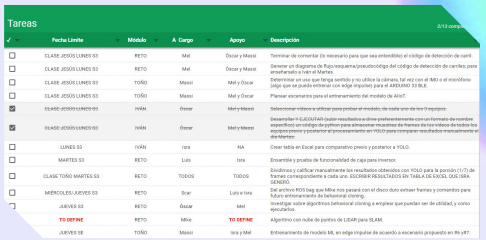
Module 04



Work teams for the year ahead







Class 1

Pre-programmed activities.

Evidenced through project binnacle.

Click-Up based organization.

Class 2

External-to-project activities.

Internal organization format.

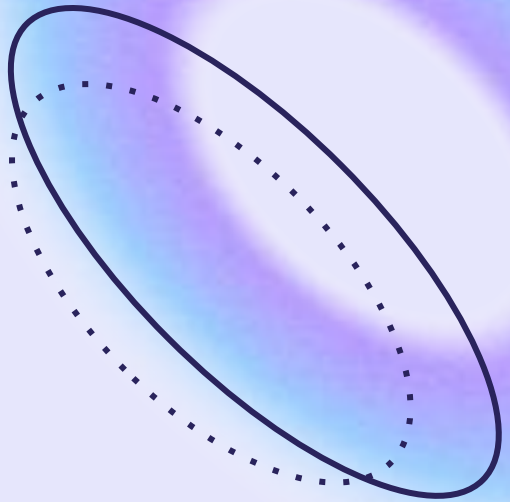
Class 3

Brief tasks.

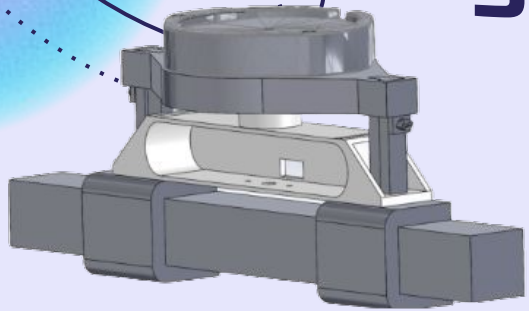
To do list internal format.

02.
Supporting Systems Manufacturing

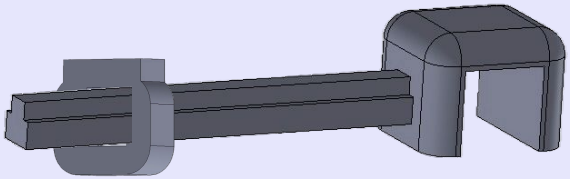
Module 01

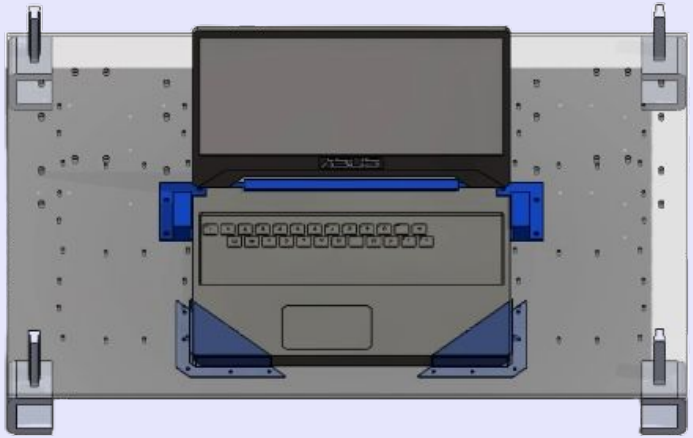


Supporting Structures CAD Modelling

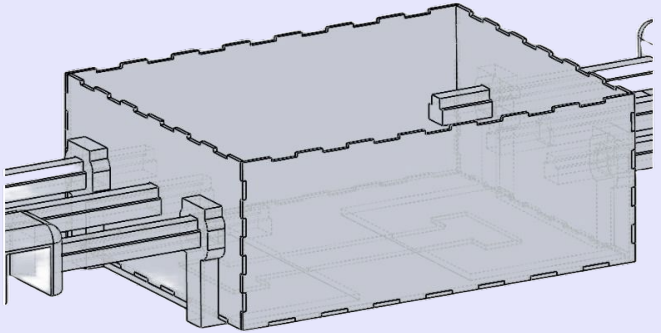


ZED 2 Cam + Velodyne Puck LiDAR



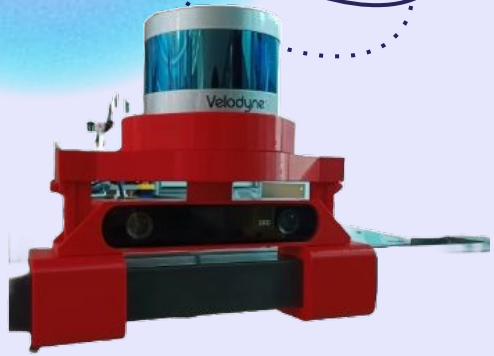


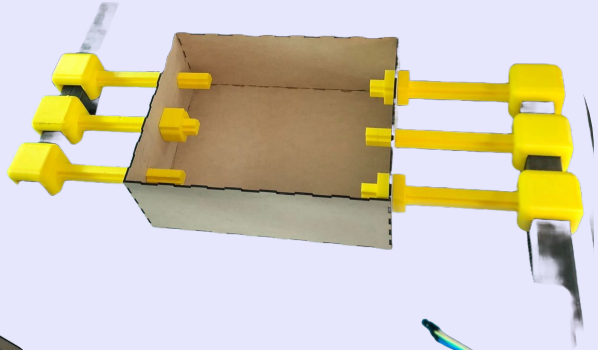
Processing Unit

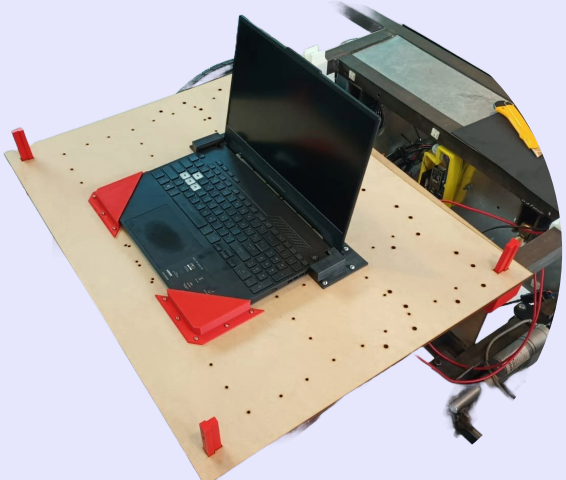


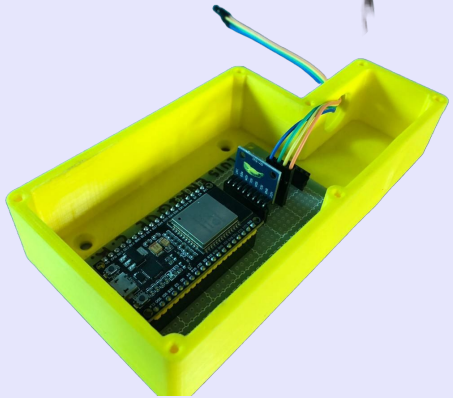
Current Inverter

Supporting Structures Manufacture







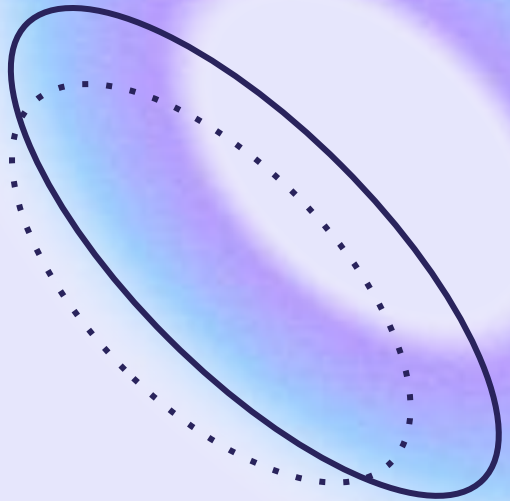


*Developed changes to show later.**

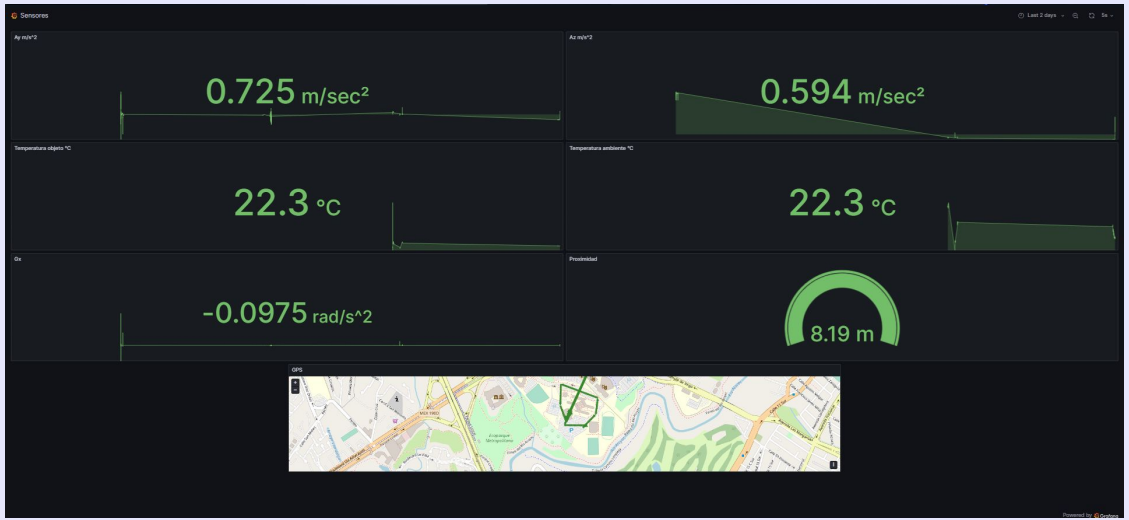
03.

Live Data Monitoring

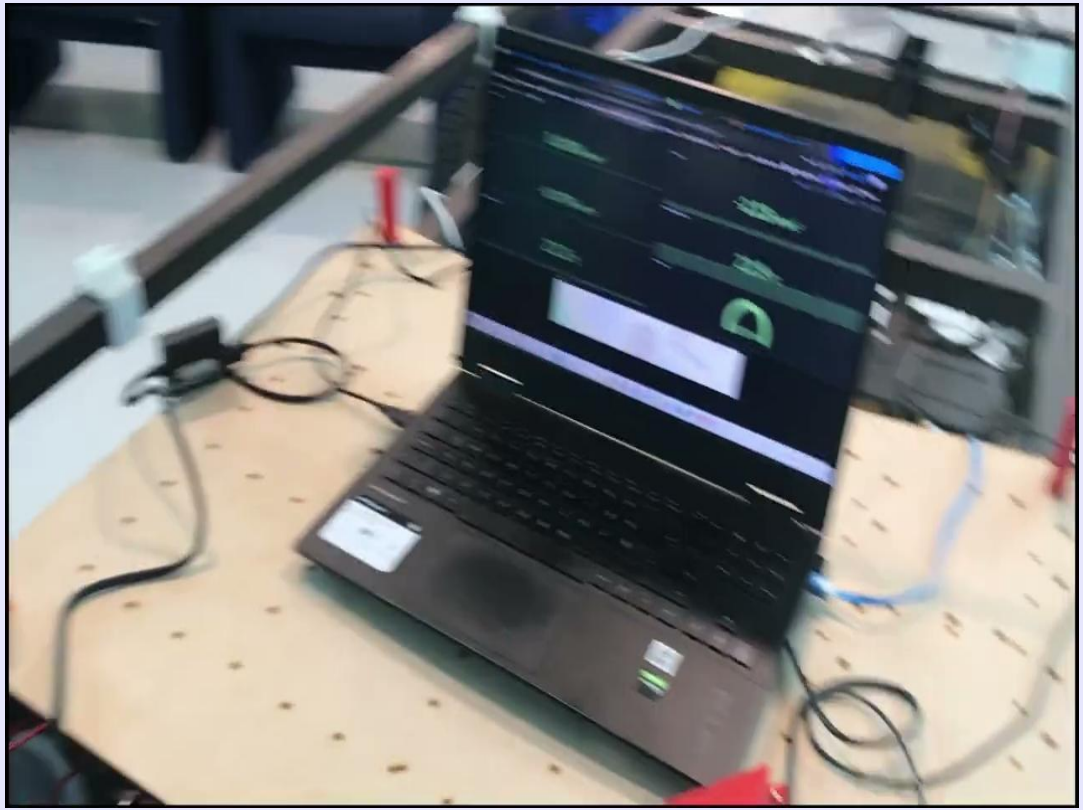
Module 02



Live Data Monitoring



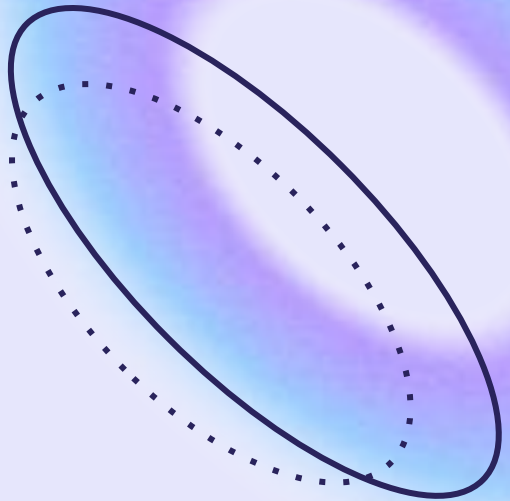
Demonstration



04.

Obstacle Detection Using YOLO

Module 03



Desktop software

Based on pre-trained COCO dataset.

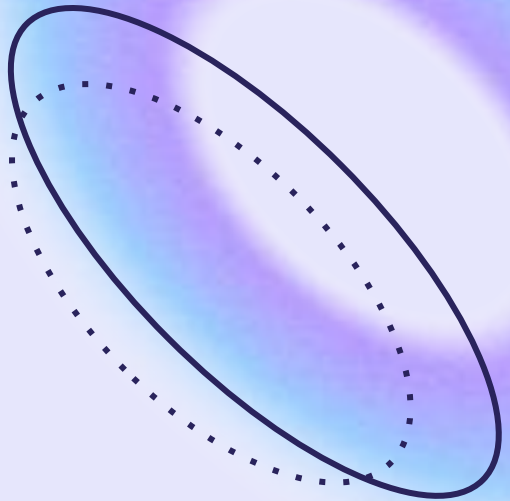
Presented:

- 91.51% Accuracy.
- 0.0944 Standard Deviation

05.

Pothole Detection Using YOLO

Challenge Project

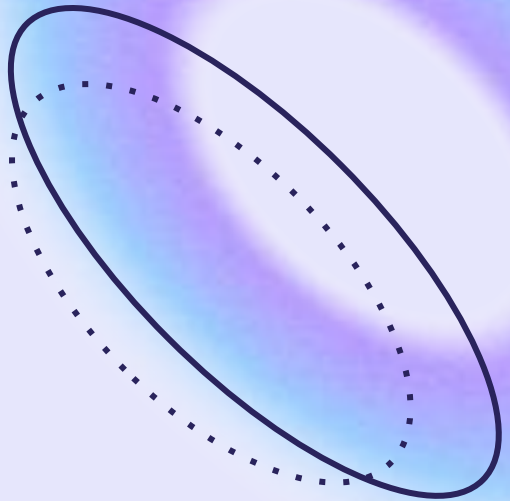


Pothole Detection Demonstration

06.

Lane Detection

Challenge Project



Autonomous-Manual Control

```
if(sbus_rx.Read()){
  sbusFlag = true;
  data = sbus_rx.data();
  Serial.println(data.ch[7]);
  //Serial.println(data.ch[8]);
  if (data.ch[4]<980){
    valorFreno = 179;
    valorEnable = 0;
  }else if (data.ch[4]>1000){
    valorEnable = 1;
    if (data.ch[7]<980){
        record();
        //alto();
        control_reversa();
        control_gear();
        control_direccion();
        control_freno();
        delay(100);
    }else if (data.ch[7]>1000){
      read_serial();
      control_freno();
    }
  }
}
```

```
void read_serial(){
  if(Serial.available()>0){
      commandBuffer = Serial.readString();  //Leer puerto ser
      int arrayLength = length_string(commandBuffer);
      if (arrayLength == 2) {
      commandBuffer.trim();
      // Assuming elements are separated by spaces
      String thirdElement = extractElement(commandBuffer, 2);
      valorDirEx = thirdElement.toInt();
      if (valorDirEx > -1 && valorDirEx < 101) {
        valorDir = valorDirEx;
        delay(100);
      }
    }else if (arrayLength == 0) {
      valorDirEx =commandBuffer.toInt();
      if (valorDirEx > -1 && valorDirEx < 101) {
        valorDir = valorDirEx;
        delay(100);
      }
    }
  }
}
```

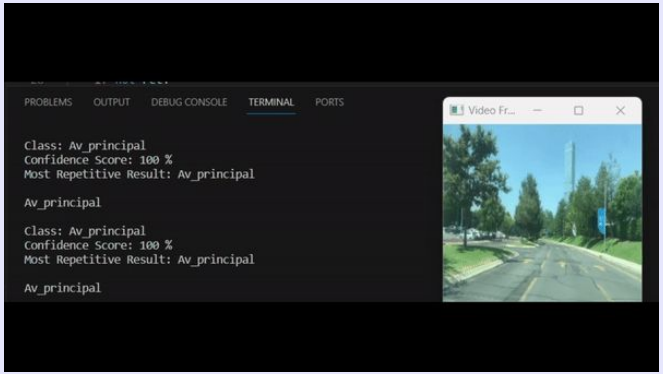
```
void control_freno(){
   if(data.ch[0]<980){
        valorFreno = map(data.ch[0],980,172,140,179);
        valorVel = 0;
      } else if(data.ch[0]<1000 && data.ch[0] > 980){
        valorVel = 0;
        valorFreno = 140;
      } else if(data.ch[0]>1000){
        valorVel = map(data.ch[0],980,1811,50,170);
        valorFreno = 140;
      }
}
```

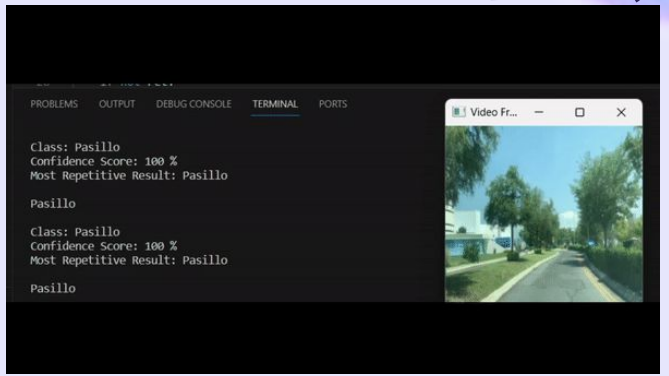
```
void control_direccion(){
  valorDir = map(data.ch[1],172,1811,100,0);
}
```

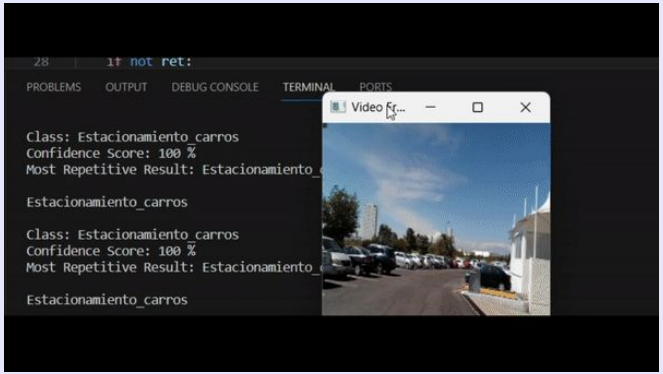
```
void control_reversa(){
  if(data.ch[3]<980){
        valorReversa = 1;
      } else if(data.ch[3]>1000){valorReversa = 0;}
}
```

```
void control_gear(){
  if(data.ch[2]<980){
        lowGear = 0;
        highGear = 1;
      } else if(data.ch[2]>1000){
        lowGear = 1;
        highGear = 0;
      }
}
```

Scenario Case Detection

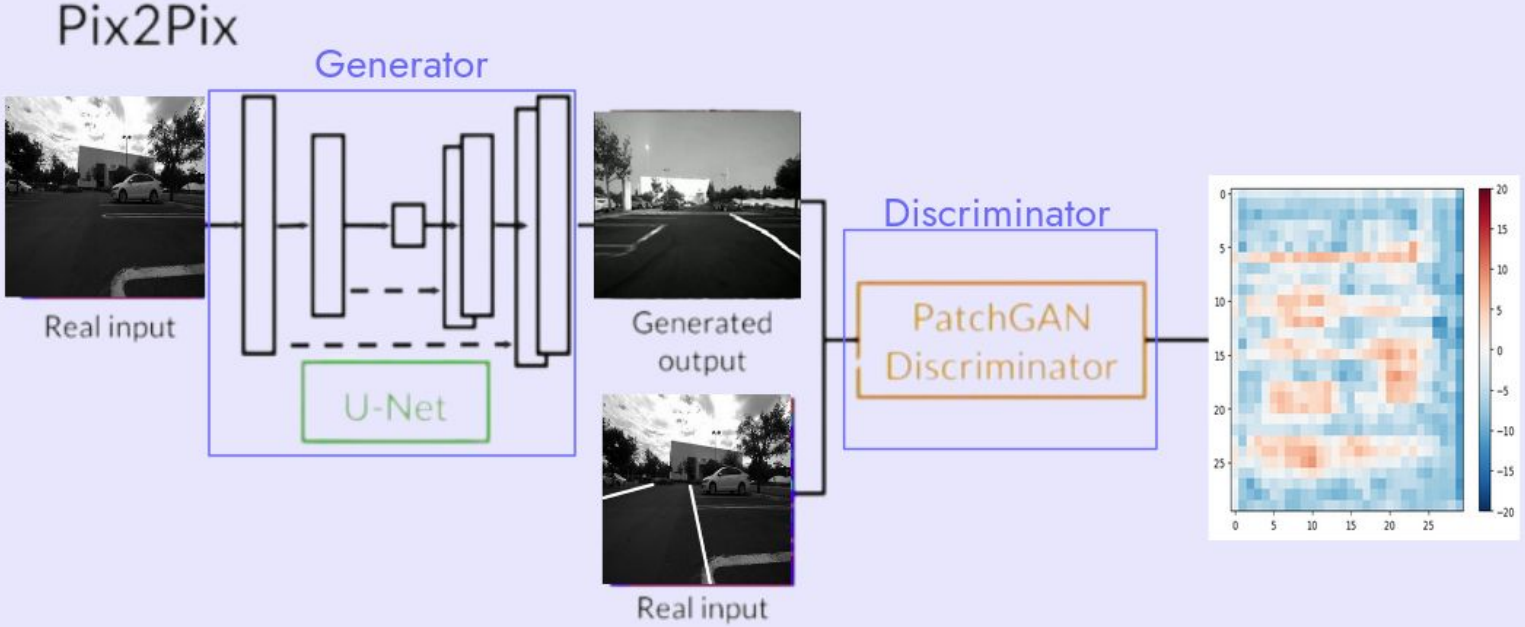






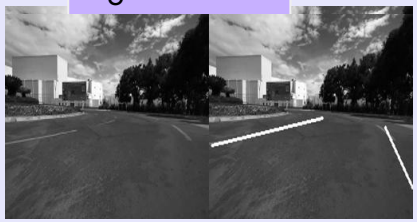
Implementation of Pix2Pix Architecture

Pix2Pix Architecture

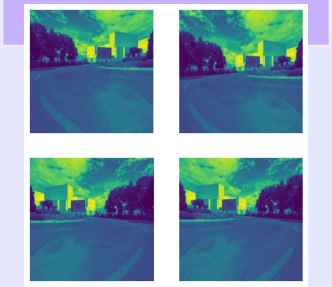


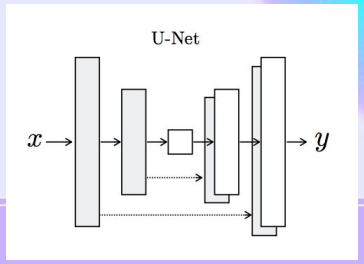
Implementation

Data set generation



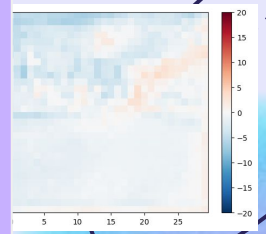
Data preprocessing (random jitter and crop)



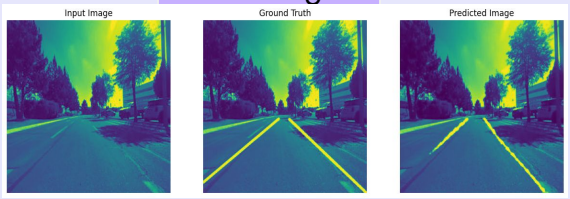


Generator:
Encoder (Convolution, Batch normalization, Leaky ReLu)
Decoder(Transposed convolution, Batch normalization, Dropout, ReLU)

Discriminator:
Encoder (Convolution, Batch normalization, Leaky ReLu)



Model Training



Qualitative analysis of the results

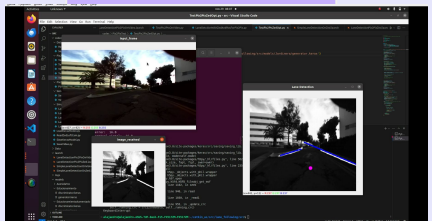
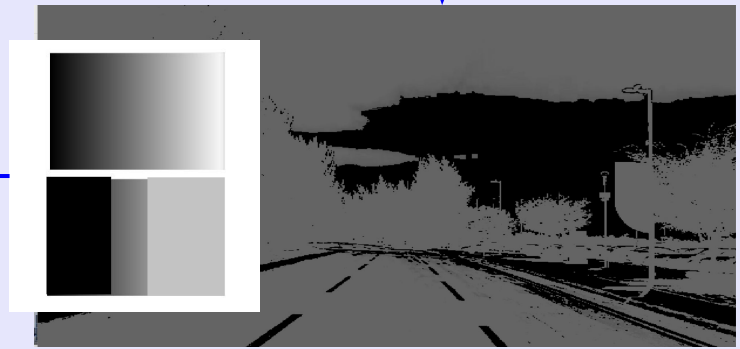
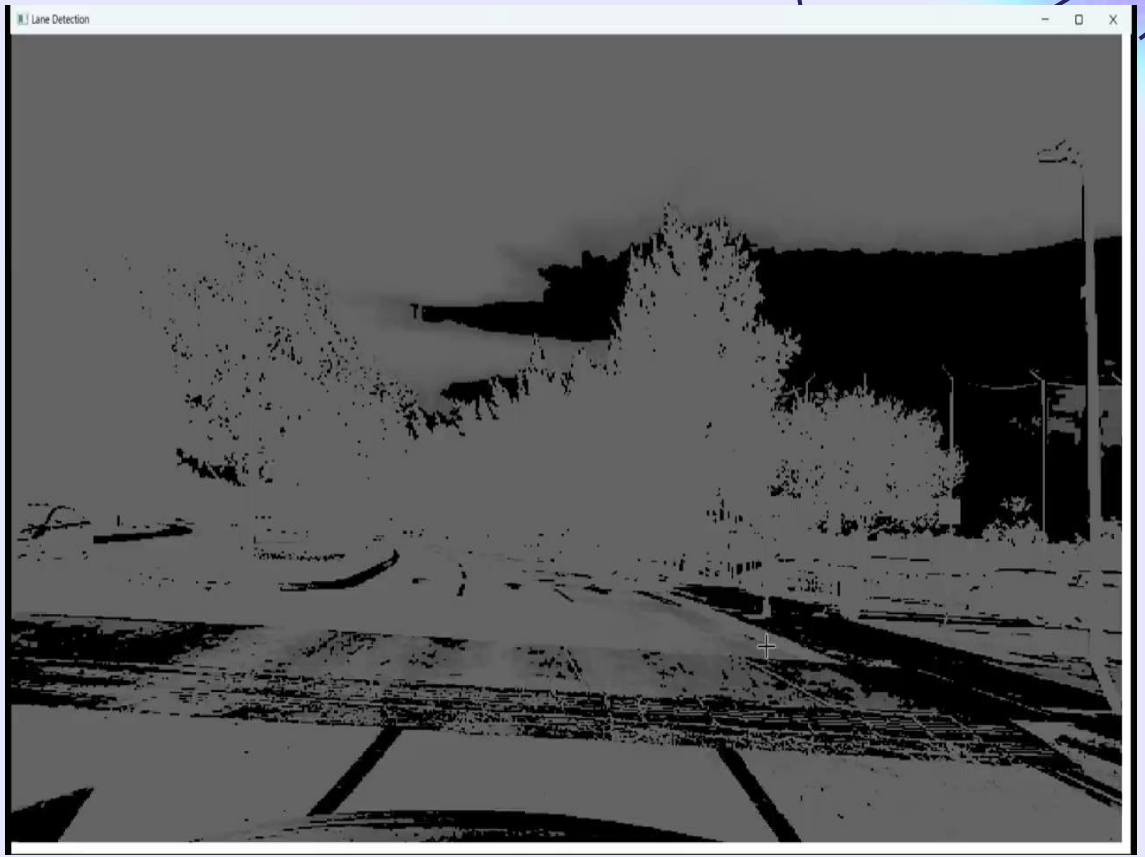


Image Processing

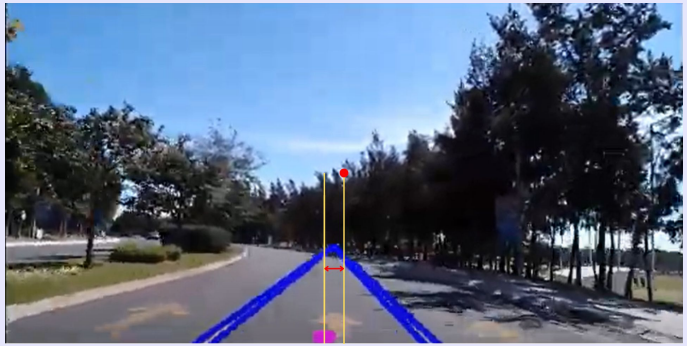


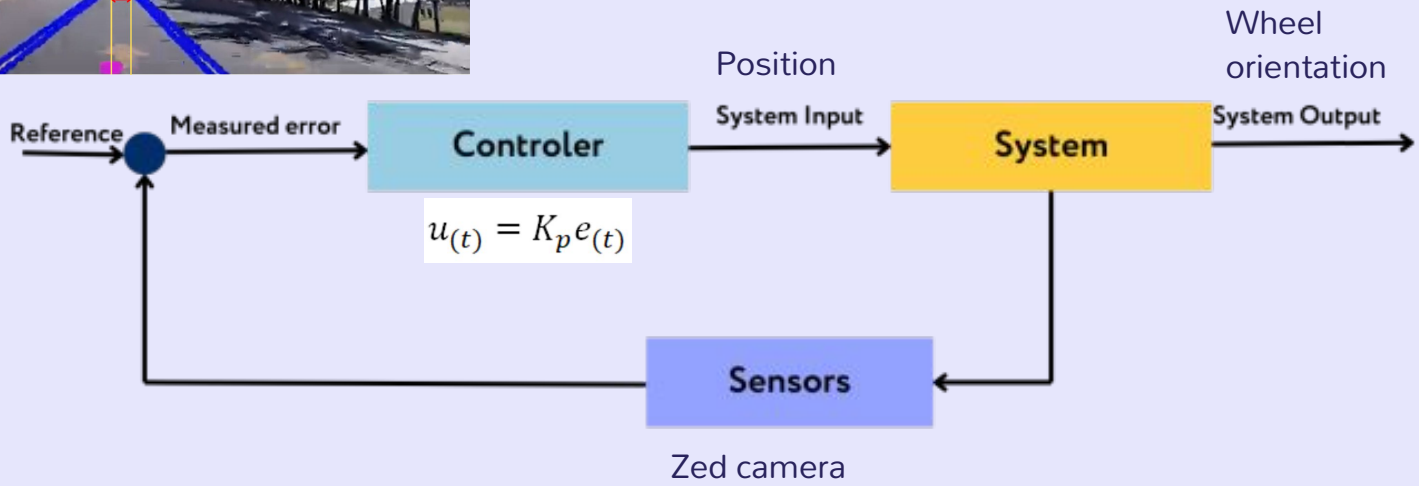
Canny Filtering



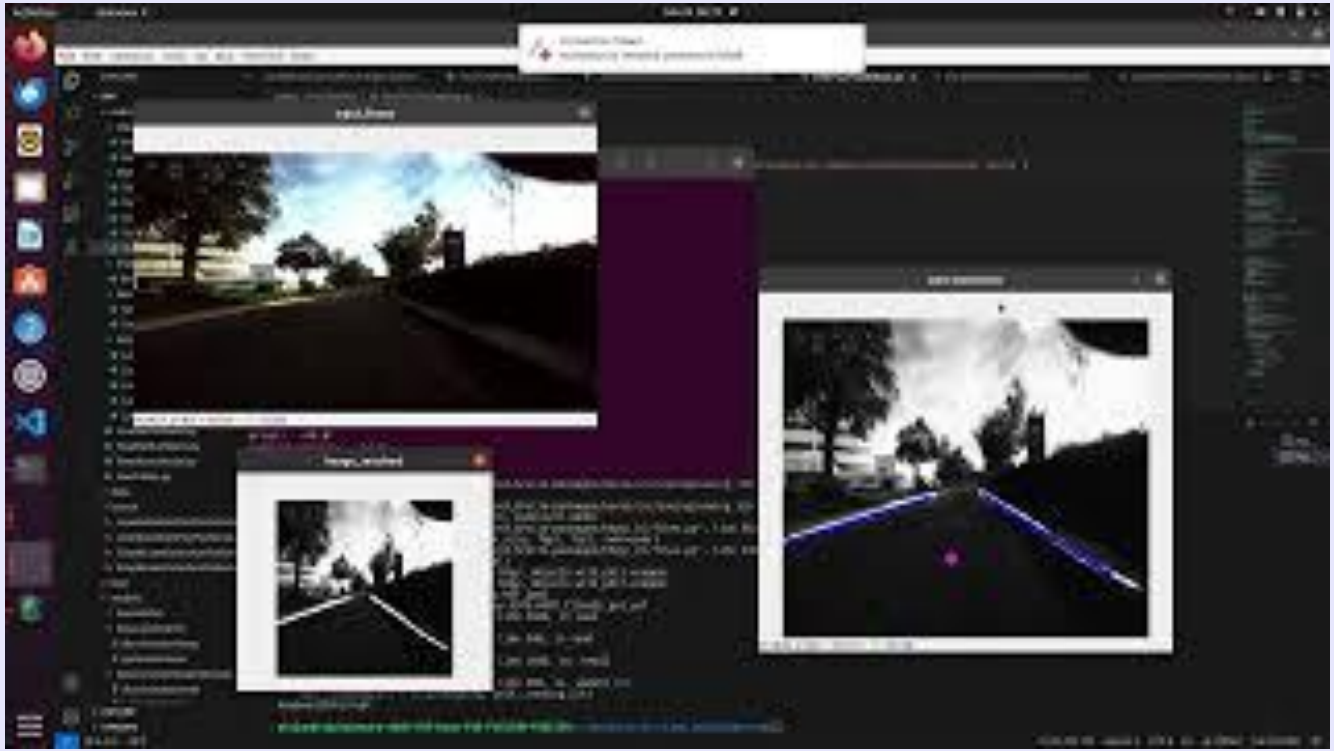
Lane Detection

Proportional Controller





Results



Results

Progress & Achievements Log Access

Regards!

- Luis Enrique Camaños Rebollo - A01732055
- Miguel Alejandro Martínez Ayala - A01734990
- Melanie Alarcón García - A01734924
- Lilian Scarlett Díaz Romero - A01734788
- Israel Lezama López - A01734758
- Óscar Emiliano Ramírez Díaz - A01660338
- Massimiliano Tuccella Ramírez - A01734774

Photo by H. González