Project Report for ENR506: Robotics Winter 2023

#Project Name: Object Detection using a CNN Model and AI Kit OakD camera.

#Student Names and Enrolment numbers:

Meet Vyas (AU2040084)

Akhil Anand (AU2040106)

1. **Project Abstract:** The aim is to use a Convolutional Neural Network (CNN) which is fed with a dataset of objects and outliers of a significant size so that the model can be trained to a sufficient amount which removes overfitting. YOLOv8 which is the latest state of the art image detection model helps in achieving that. 3-D SLAM is integrated into the model using Spectacular AI. The whole setup is then controlled using Arduino motor drivers for functionality of the autonomous robot.

1.1 Project Objectives:

- To make a functional model that detects objects defined as classes in real-time.
- Integrate the model with a real time 3D SLAM (Simultaneous Localization and Mapping) environment.
- Apply it to the physical model of a mobile robot.

1.2 Literature Search:

Three recent papers in the fields of computer vision and robotics were examined. The first paper proposes RELAX, an approach for attribution-based explanations of representations for deep learning algorithms. The authors claim that RELAX is the first method that can explain what influences the learned representation, and it can model the uncertainty in its explanations, which is essential to produce trustworthy explanations. The paper provides theoretical interpretations of RELAX and conducts a user study to assess how well the proposed approach aligns with human intuition. The second paper proposes a feature extraction model based on LIDAR scans to classify the nodes of indoor structures. The authors suggest that the model can be used as an efficient tool for patrolling, and their experiments in a real scenario with a differential robot demonstrate that the model is able to establish the graph automatically and with precision. Finally, the third paper presents an innovative vision system based on YOLOv5 for floor-cleaning robots. The vision system is built to detect dirty spots on the floor, and it can control the robot's speed and how much water and detergent is spent according to the detected dirt. The authors suggest that implementing a robust vision system in floor-cleaning robots can optimize their navigation and reduce power, water, and chemical products' consumption.

2. Prototype preparation:

Sr No.	*Material*	*Making*	*Usage*
1	Luxonis OAK D Camera	Ready Made	Main component of the assembly which is used to host and deploy neural networks.
2	L298N motor drivers	Ready Made	Motor Drivers are used to control and
3	12 Volt battery	Ready Made	The battery is used to power the motor drivers
4	Arduino Mega	Ready Made	The arduino mega is used to communicate between the laptop, camera and the motor drivers.
5	Chassis	Laser Cutted	We built the chassis by cutting a 10mm Acrylic Sheet and then laser cutting the mounting holes on it.
6	Clamps	Ready Made	We used the clamps to mount the motors onto the chassis
7	Laptop	Ready Made	We used an HP Omen 15 laptop for the project
8	Mechanics Kit	Ready Made	We used the mechanics kit to create mounts for the arduino mega, motor drivers, and the camera.
9	Motor Bracket	Ready Made	We used the motor brackets to change the diameter/pitch of the motors from 6mm to 4mm to fir into the mecanum wheel mounts which were 3D printed.
10	Mounts	3-D Printed	We 3-D printed mounts to attach the mecanum wheels to the robot chassis.
11	Bolts	Ready Made	We used the bolts to tie down the clamps and the motor drivers to the chassis.
12	12 Volt DC Gear Reduction	Ready	We used these motors as they provide

	Motors	Made	rotation speeds up to 1350 rpm with variable speed control
13	Mecanum Wheels	Ready Made	We used mcannum wheels to parallel steer, but were unable to fully utilise the functionality

Preparation:

*F*or our robot assembly, we started by preparing the chassis by cutting a 10mm acrylic sheet to the desired size and laser cutting mounting holes on it. We then 3-D printed mounts for attaching the mecanum wheels to the robot chassis and attached them using bolts.

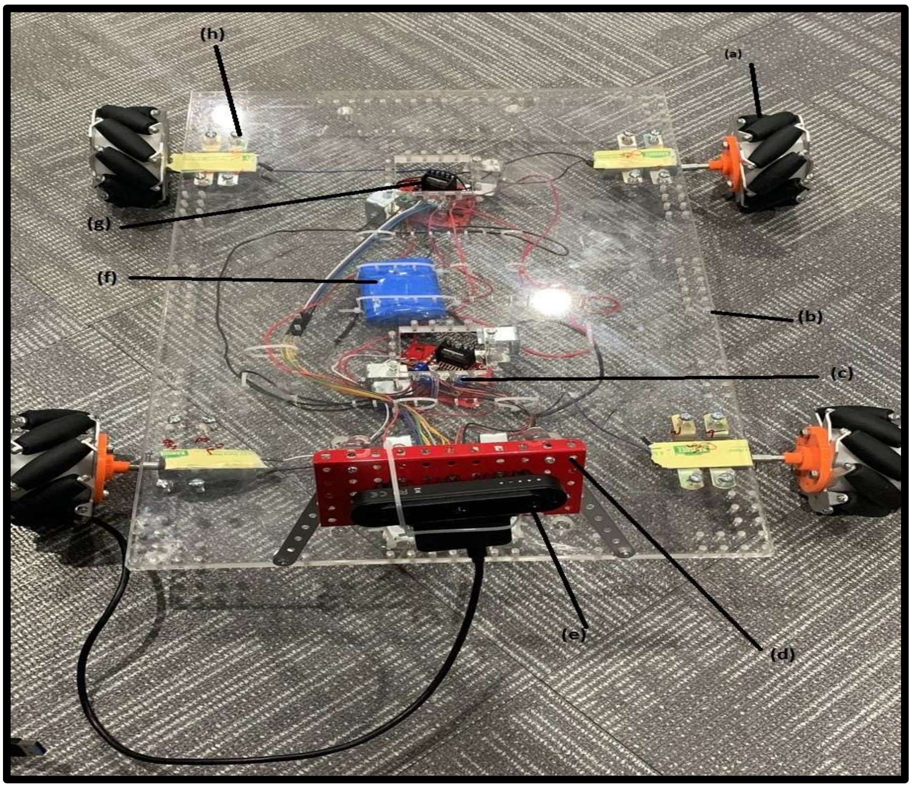
Next, we used the motor brackets to change the diameter/pitch of the motors from 6mm to 4mm to fit into the mecanum wheel mounts. We mounted the motors onto the chassis using clamps and bolts, and connected the L298N motor drivers to the motors using wires. The motor drivers were then attached to the chassis using bolts.

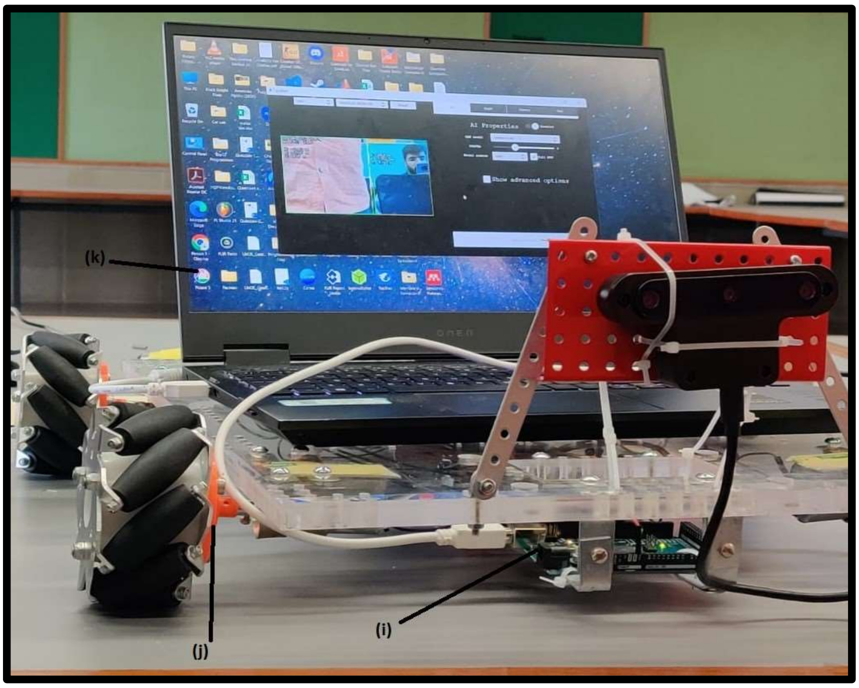
We then mounted the Luxonis OAK D Camera onto the chassis using the mechanics kit, and attached the Arduino Mega to the chassis using the same kit. We connected the L298N motor drivers to the Arduino Mega using wires.

To power the robot, we connected a 12 Volt battery to the L298N motor drivers. We also connected the Arduino Mega to a laptop using a USB cable and programmed it to control the motors and the Luxonis OAK D Camera.

Once all the components were properly connected and programmed, we powered on the robot by turning on the 12 Volt battery. We were then able to use the laptop to control the robot and deploy neural networks using the Luxonis OAK D Camera.

3. Annotated pictures of the prototype:





(a) Metal fabricated mecanum wheels.
(b) 10 mm thick Acrylic Chassis.
(c) Motor Driver for front wheels.
(d) Mount for the camera.

(e) OakD Camera.
(f) 12 Volt Rechargeable battery.
(g) Motor Driver for back wheels.
(h) Clamps.
(i) Arduino Mega.
(j) Horizontal mount for the wheel.
(k) Laptop that runs the model (is advisable to be replaced by Jetson Nano)

Misc : Bolts, Nuts, Mechanical Kit that includes motor brackets.

4. Programming:

```
# Code to upload the neural network using python and depthai_sdk

from depthai_sdk import OakCamera

with OakCamera() as oak:

    color = oak.create_camera('color', resolution='1080p')

 nn = oak.create_nn('obstacle_detection', color)

    nn.config_nn(resize_mode='letterbox')

 visualizer = oak.visualize([nn, nn.out.passthrough], fps=True)

    oak.start(blocking=True)

with OakCamera() as oak:

    color = oak.create_camera('color')

    model_config = {

      'source': 'roboflow',

      'model': 'obstacle-detection-f3yxc/1',

      'key': 'mN2NdBpq63DRwJ0uy5ml'

    }

    nn = oak.create_nn(model_config, color)

    oak.visualize([nn.out.main, nn.out.spatials], fps=True)
```

```
  oak.start(blocking=True)
```

```
# Code to use the motor drivers

// Motor Driver Pins into Arduino Mega

int ena = 13;

int in1 = 12;

int in2 = 11;

int in3 = 4;

int in4 = 3;

int enb = 2;

int enc = 5;

int in5 = 6;

int in6 = 7;

int in7 = 8;

int in8 = 9;

int enf = 10;


void setup() {


  pinMode(ena, OUTPUT);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(enb, OUTPUT);
```

```
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enc, OUTPUT);
  pinMode(in5, OUTPUT);
  pinMode(in6, OUTPUT);
  pinMode(in7, OUTPUT);
  pinMode(in8, OUTPUT);
  pinMode(enf, OUTPUT);

  // STOP
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  digitalWrite(in5, LOW);
  digitalWrite(in6, LOW);
  digitalWrite(in7, LOW);
  digitalWrite(in8, LOW);
}


void goRight() // Makes the robot go right
{
```

```
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  digitalWrite(in5, HIGH);
  digitalWrite(in6, LOW);
  digitalWrite(in7, LOW);
  digitalWrite(in8, HIGH);
  analogWrite(ena, 255);
  analogWrite(enb, 255);
  analogWrite(enc, 255);
  analogWrite(enf, 255);

}

void goLeft()  // Makes the robot go left
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  digitalWrite(in5, LOW);
  digitalWrite(in6, HIGH);
```

```
  digitalWrite(in7, HIGH);
  digitalWrite(in8, LOW);
  analogWrite(ena, 255);
  analogWrite(enb, 255);
  analogWrite(enc, 255);
  analogWrite(enf, 255);

}
void goForward() // Makes the robot go forward
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  digitalWrite(in5, HIGH);
  digitalWrite(in6, LOW);
  digitalWrite(in7, HIGH);
  digitalWrite(in8, LOW);
  analogWrite(ena, 255);
  analogWrite(enb, 255);
  analogWrite(enc, 255);
  analogWrite(enf, 255);
}
```

```
void goBackward() // Makes the robot go backward
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  digitalWrite(in5, LOW);
  digitalWrite(in6, HIGH);
  digitalWrite(in7, LOW);
  digitalWrite(in8, HIGH);
  analogWrite(ena, 255);
  analogWrite(enb, 255);
  analogWrite(enc, 255);
  analogWrite(enf, 255);

}
void stopa() // Makes the robot stop
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
```

```
  digitalWrite(in5, LOW);
  digitalWrite(in6, LOW);
  digitalWrite(in7, LOW);
  digitalWrite(in8, LOW);
  }

// Loops the above controls with a delay of a second between each
void loop()
{
  goForward();
  delay(1000);
  goBackward();
  delay(1000);
  goLeft();
  delay(1000);
  goRight();
  delay(1000);
}
```

5. **Conclusion:** This project acts as a bridge towards making a fully functional and capable autonomous robot that is able to serve in real world tough conditions. It is still a few steps away from being fully functional and this model serves as a first semi-working prototype.

References

Roboflow.(n.d.).LuxonisOAKInference.Roboflow. https://docs.roboflow.com/inference/luxonis-oak

Luxonis.(n.d.).UsingCustomModelsLuxonisDocs. https://docs.luxonis.com/en/latest/pages/tutorials/first_steps/#using-custom-models

Luxonis. (n.d.). DepthAI SDK. https://github.com/luxonis/depthai-sdk