**Project Report: Object Recognition using HuskyLens**

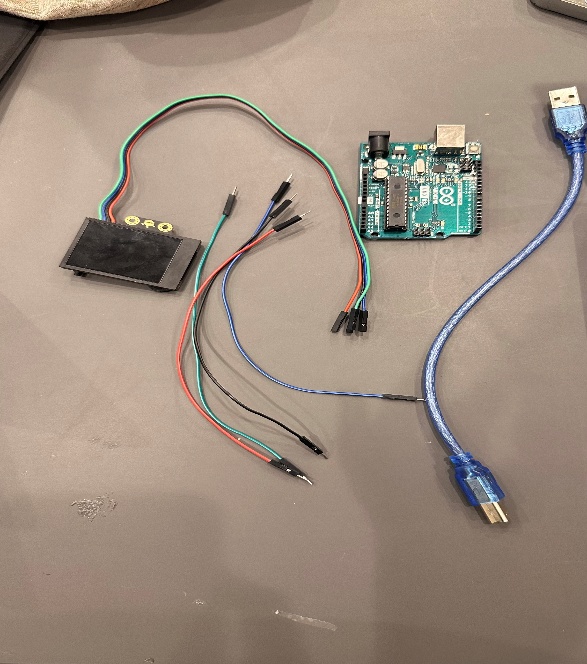
**1. Introduction** HuskyLens is an AI-powered vision sensor designed for machine learning applications. It can recognize objects, track them, and classify them with minimal programming effort. In this project, we utilized HuskyLens for **Object Recognition**, allowing it to detect and identify objects in real-time.

**2. Objectives**

* Implement **Object Recognition** using HuskyLens.
* Integrate the sensor with an Arduino board.
* Process and display detection results.
* Control a robotic system based on object recognition results.

**3. Hardware and Software Used**

* **Hardware:**
  + HuskyLens AI Camera
  + Arduino Board (Uno, Mega, or Leonardo)



* **Software:**
  + Arduino IDE
  + HuskyLens Library
  + C++ for Arduino programming

**4. Implementation Steps**

1. **Setup HuskyLens:**
   * Connect HuskyLens to the Arduino via **I2C communication**.
   * Ensure that HuskyLens is in **Object Recognition Mode**.
   * Train the sensor to recognize specific objects.
2. **Write the Arduino Code:**
   * Initialize HuskyLens in **I2C mode**.
   * Request object recognition data.
   * Control a robotic system based on detected objects.
3. **Testing and Debugging:**
   * Validate the detection accuracy of HuskyLens.
   * Adjust parameters like **width levels** and **motion control**.
   * Display recognition data using Serial Monitor.

**5. Results**

* The system successfully detected and recognized trained objects.
* The robotic system responded accurately based on object detection.
* The project demonstrated real-time object tracking and movement control.

**6. Challenges and Solutions**

* **Detection Accuracy:** Sometimes, the recognition was not precise. This was improved by adjusting training conditions and ensuring proper lighting.
* **I2C Communication Errors:** These were resolved by checking wiring connections and setting the correct protocol type in HuskyLens settings.

**7. Conclusion** This project successfully implemented **Object Recognition** using HuskyLens, demonstrating its capability in AI-driven object detection. The results show potential applications in robotics, automation, and AI-based vision systems.

**Code:**

#include "HUSKYLENS.h"

#include "DFMobile.h"

DFMobile Robot (7,6,4,5); // initiate the Motor pin

HUSKYLENS huskylens;

//HUSKYLENS green line >> SDA; blue line >> SCL

void setup() {

Serial.begin(115200);

Robot.Direction (HIGH, LOW); // initiate the positive direction

Wire.begin();

while (!huskylens.begin(Wire))

{

Serial.println(F("Begin failed!"));

Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General Settings>>Protocol Type>>I2C)"));

Serial.println(F("2.Please recheck the connection."));

delay(100);

}

huskylens.writeAlgorithm(ALGORITHM\_OBJECT\_TRACKING); //Switch the algorithm to object tracking.

// while (true)

// {Robot.Speed (200-50,200+50);

// delay(2000);

// Robot.Speed (0,0);

// delay(2000);

// Robot.Speed (200,200);

// delay(2000);

// Robot.Speed (0,0);

// delay(2000);

// Robot.Speed (200+50,200-50);

// delay(2000);

// Robot.Speed (0,0);

// delay(2000);}

}

int widthLevel = 50;

int xLeft = 160-40;

int xRight = 160+40;

bool isTurning = false;

bool isTurningLeft = true;

bool isInside(int value, int min, int max){

return (value >= min && value <= max);

}

void printResult(HUSKYLENSResult result);

void loop() {

int32\_t error;

int left = 0, right = 0;

if (!huskylens.request()) Serial.println(F("Fail to request objects from HUSKYLENS!"));

else if(!huskylens.isLearned()) {Serial.println(F("Object not learned!")); Robot.Speed (0,0);}

else if(!huskylens.available()) Serial.println(F("Object disappeared!"));

else

{

HUSKYLENSResult result = huskylens.read();

if (result.width < widthLevel){

widthLevel = 65;

if (isInside(result.xCenter, 0, xLeft)){

if (isTurningLeft){

if (!isTurning){

Robot.Speed (200-50,200+50);

}

}

else{

if (isTurning){

isTurning = false;

isTurningLeft = !isTurningLeft;

}

Robot.Speed (200-50,200+50);

}

}

else if (isInside(result.xCenter, xLeft, xRight)){

if (isTurning){

isTurning = false;

isTurningLeft = !isTurningLeft;

}

Robot.Speed (200,200);

}

else if (isInside(result.xCenter, xRight, 320)){

if (isTurningLeft){

if (isTurning){

isTurning = false;

isTurningLeft = !isTurningLeft;

}

Robot.Speed (200+50,200-50);

}

else{

if (!isTurning){

Robot.Speed (200+50,200-50);

}

}

}

}

else

{

widthLevel = 55;

isTurning = true;

if (isTurningLeft){

Robot.Speed (0,200);

}

else{

Robot.Speed (200,0);

}

}

printResult(result);

}

}

void printResult(HUSKYLENSResult result){

if (result.command == COMMAND\_RETURN\_BLOCK){

Serial.println(String()+F("Block:xCenter=")+result.xCenter+F(",yCenter=")+result.yCenter+F(",width=")+result.width+F(",height=")+result.height+F(",ID=")+result.ID);

}

else if (result.command == COMMAND\_RETURN\_ARROW){

Serial.println(String()+F("Arrow:xOrigin=")+result.xOrigin+F(",yOrigin=")+result.yOrigin+F(",xTarget=")+result.xTarget+F(",yTarget=")+result.yTarget+F(",ID=")+result.ID);

}

else{

Serial.println("Object unknown!");

}

}

