VISION-BASED ALGORITHM USING PLC CONTROLLER

[1a] Mr.S.Sarveswaran

[2a]K. Lingesarajan, [2b]S. Aravindan, [2c]MK. Bhuvan Shankar

[2a]M. Pravinkumar, [2b]S. Tamilselvan, [2c]PK. Vishva

[1] Assistant Professor, Department of Robotics and Automation,

Sri Ramakrishna Engineering College, Coimbatore – 641022, India.

[2] Students, Department of Robotics and Automation,

Sri Ramakrishna Engineering College, Coimbatore – 641022, India.

Abstract:

In the production sector, automation plays a vital role in enhancing efficiency and output. One of the most prevalent uses of automation is in pick-and-place operations. This article presents a pick-and-place robot that employs a vacuum gripper and is based on the Cartesian system, using a Programmable Logic Controller (PLC) system. The robot is constructed with three linear axes, namely X, Y, and Z, that employ the Cartesian coordinate system to move the robot's end-effector to the desired position. The base of the robot is stationary, while the linear axes are attached to it. The end-effector is fitted with a vacuum gripper that generates a vacuum between the gripper and the object being picked, allowing the robot to transfer it to the desired location. The PLC system governs the robot's movements and the vacuum gripper's function. It is programmed using ladder logic, which sends signals to the motor controllers to control the robot's movements. Furthermore, the PLC system manages the vacuum pump and the suction cups of the gripper. The operator inputs the desired coordinates using a human-machine interface (HMI), which communicates with the PLC system.

In this project, we propose a method for ball detection using the YOLOv5 object detection framework. Our approach involves training a YOLOv5 model on a dataset of annotated images of balls in different environments and conditions. We also employ data augmentation techniques to improve the robustness of the model. Our results show that the YOLOv5 model can achieve high accuracy in detecting balls in various settings, including outdoor and indoor environments, and with different lighting conditions. We also compare our approach with other state-of-the-art ball detection methods and show that our method outperforms them in terms of accuracy and speed. Our work demonstrates the effectiveness of using the YOLOv5 model for ball detection and its potential for use in a range of applications, such as sports analysis and robotics.

KEYWORDS:
Cartesian robot, Programmable Logic controller, Pneumatic cylinder

1. Introduction

1.1 Project Objectives

The primary goal of this project is to develop a software program that enables the robot to retrieve items from the conveyor and deposit them into the container on the opposite end, utilizing PLC technology and Cartesian coordinate systems while moving its axes.In addition to the main objective, one of the project's objectives is to be equipped to handle any technical issues that may arise while continuing with another team's project. As a result, a thorough understanding of the machinery and its proper configuration will be required.

1.2 Project Scope

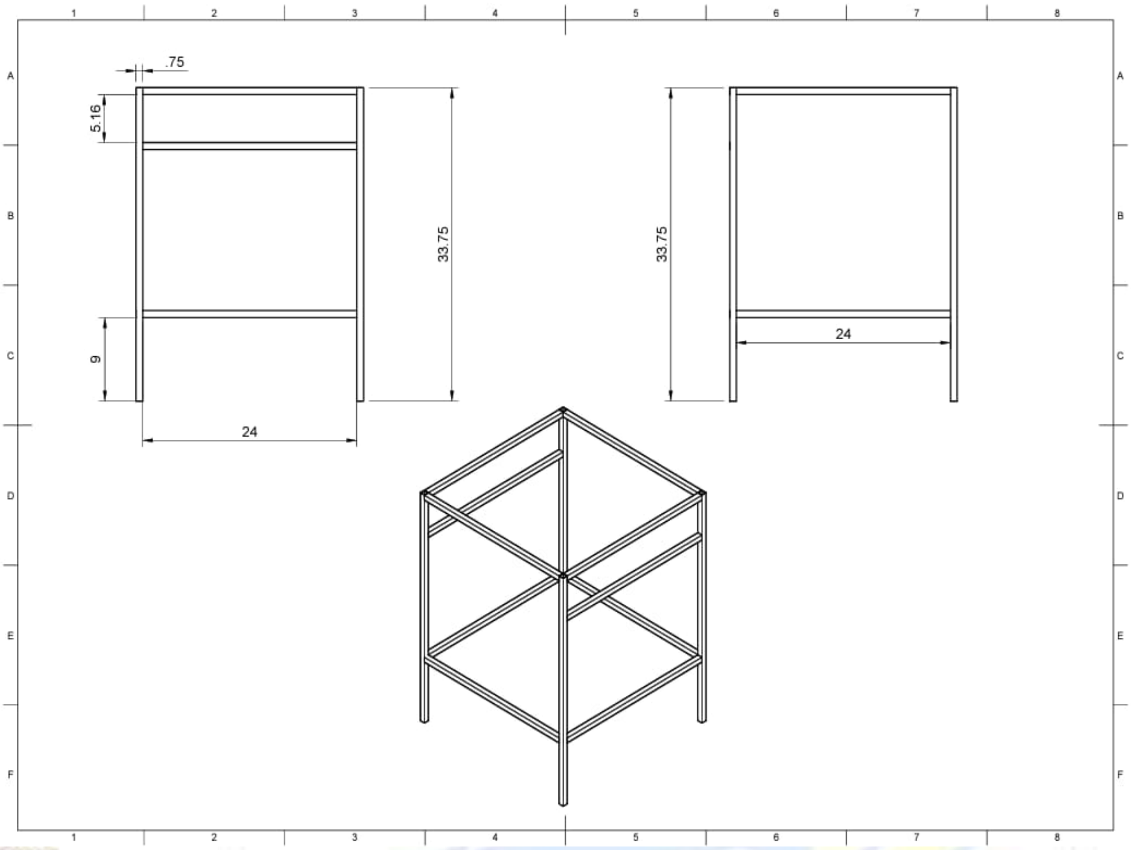
This is a substantial project that encompasses a wide range of tasks. As previously stated, a separate group of students was responsible for the robot's vision system, component selection, construction, and mechanical work, while our batch students were responsible for the PLC system.The configuration of the components in the PLC system is critical to ensure that the process runs smoothly. In the manufacturing industry, automation is critical in increasing efficiency and productivity. One of the most common applications of automation is pick-and-place operations. This report discusses a Cartesian-based pick-and-place robot that employs a vacuum gripper to retrieve and deposit objects. The robot is controlled by a PLC (Programmable Logic Controller) system, which provides precise control and allows for customization of the robot's movement.

Project Aim

The primary goal of this project is the creation of a programme that enables the robot to remove an objects from a conveyor, pick it up with a PLC system using Cartesian coordinate technology, and store it in a bin at the other end.

In addition to the core goal, this project also intends to prepare the team to handle any potential technical issues that can arise when they work on another team's project. It will therefore require a thorough knowledge of the equipment and its proper configuration.

Design of Frame:



Components:

1. Structure:

The structure of the robot is provided by the frame, which holds all the components together. The material used for the frame depends on the size and payload capacity of the robot, including options such as aluminium, steel, or other materials.

2. Movement Devices:

Actuators are responsible for providing the robot with movement. In pick and place robots, some commonly used actuators are:

- Pneumatic Cylinders: These actuators use compressed air for linear movement, making them ideal for high-speed assembly lines where speed is crucial.

3. Interaction Tool:

The part of the robot that interacts with the object being picked and placed is called the end effector. Depending on the application, it could be a gripper, suction cup, or other specialized tool. The end effector is attached to the robot's arm and is controlled by the actuators.

Stepper motor:

- Nema 23 Stepper motor 10kg/cm , Holding torque - 10 Kg/cm, 4 wire bipolar,Step angle - 1.8 °/Step,
- Rated Voltage: 2.3V, Phase current- 2Amp, Phase Resistance (Ohms): 1.13, Phase inductance (mH) 3.6.
- Shaft 6mm Round, Shaft length: 20.5mm, Compatible with TB6600 and other 2-phase stepper drivers
- Frame Size 57mm x 57mm, Mounting direction - horizontal and vertical both
- Stepper Motor Use for 3D printer, DIY CNC , Arduino project, Robotics, Router Engraving machine, XY plotters, Pick and Place Robot

Cartesian robot :

A Cartesian-based pick and place robot with a vacuum gripper using a Programmable Logic Controller (PLC) system is a popular automation system used in various manufacturing processes. It is a versatile system that can perform repetitive tasks with high accuracy and speed, making it ideal for pick and place operations.

The system consists of a Cartesian robot that moves along the X, Y, and Z axes, a vacuum gripper that picks and places objects, and a PLC system that u-0als from sensors and other devices to control the robot's movement and the vacuum gripper's operation.

The operation of the system involves the following steps:

1. Object Recognition:
Sensors recognize the presence of items in the assigned area, prompting the PLC system to activate the robot's motion.

2. Item Retrieval:
The robot travels on the X, Y, and Z axes to the item's location, and the vacuum gripper picks up the item.

3. Motion Control:
The robot transports the item to the desired location, guided by the PLC system.

4. Release Procedure:
The vacuum gripper releases the item at the designated location, and the robot returns to its starting point.

The PLC system oversees and regulates each stage of the process, ensuring its precision and effectiveness. It receives input from sensors and other devices, enabling it to modify the robot's motion and the vacuum gripper's operation as necessary.

In conclusion, a Cartesian-based pick and place robot with a vacuum gripper utilizing a PLC system is a remarkably efficient and precise automation system utilized in various manufacturing operations. Its capacity to execute repetitive tasks with high precision and speed makes it an ideal solution for pick and place operations.

Hardware components:

1. Stepper motor
2. Stepper motor drive
3. Solenoid Valve
4. Suction cup
5. SMPS
6. Push Button
7. Conveyor
8. Timing belt
9. Ball & Bearing
10. Emergency Stop
11. PLC
12. Relay Module
13. Limit Switch
14. Air hose
15. Connectors
16. Terminals
17. Timing Pulley
18. 230V Indication lamp
19. ON/OFF Rotary Switch
20. MCB

Stepper motor calculation:

Motor Selection For belt and pully,

Total Load to carry is equal to 3kg.

Torque = Force x Radius

Friction Force = μ x (R/r)

(R - normal load ⇒ 3 x 9.81 ⇒ 29.43 N

r - radius = 3 mm ⇒ 0.003 m

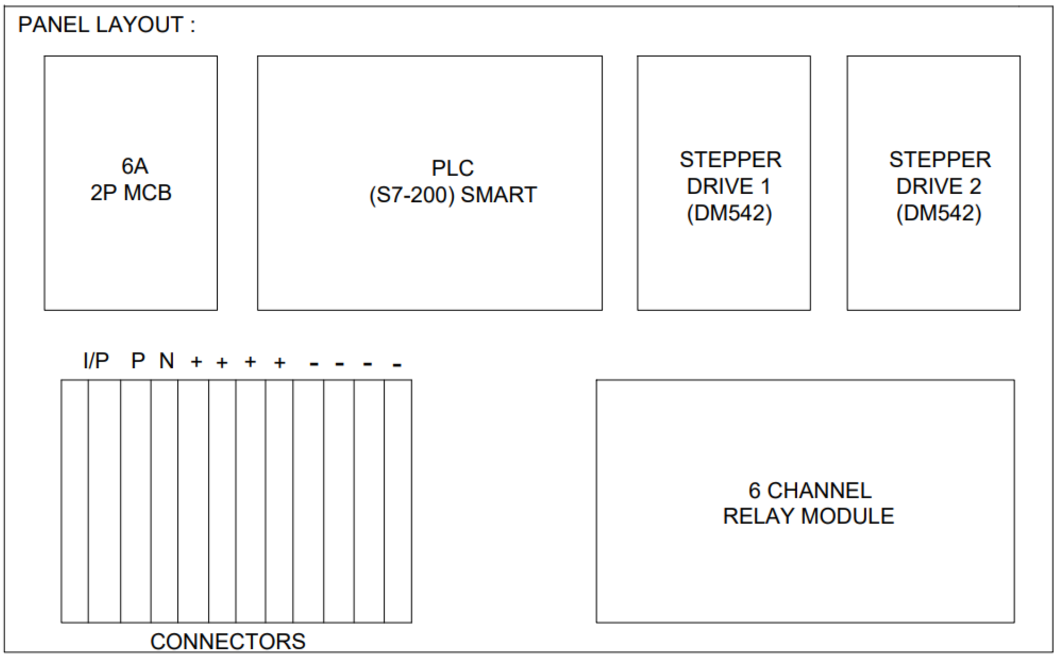
μ - 0.4 coefficient of friction for Mild Steel

Friction Force = 0.4 x (29.43/0.003)

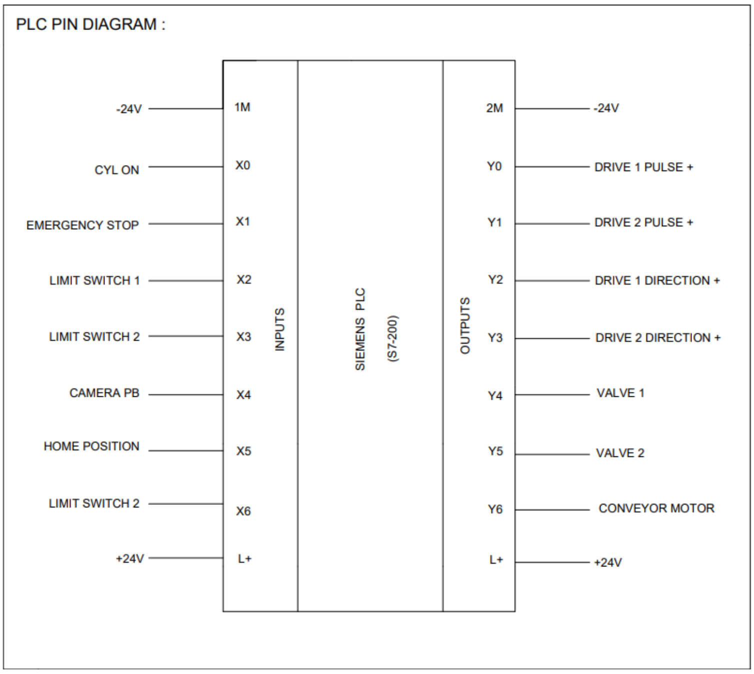
=3924 N

Torque = 3924 x 0.003, Torque = 11.772 Nm.

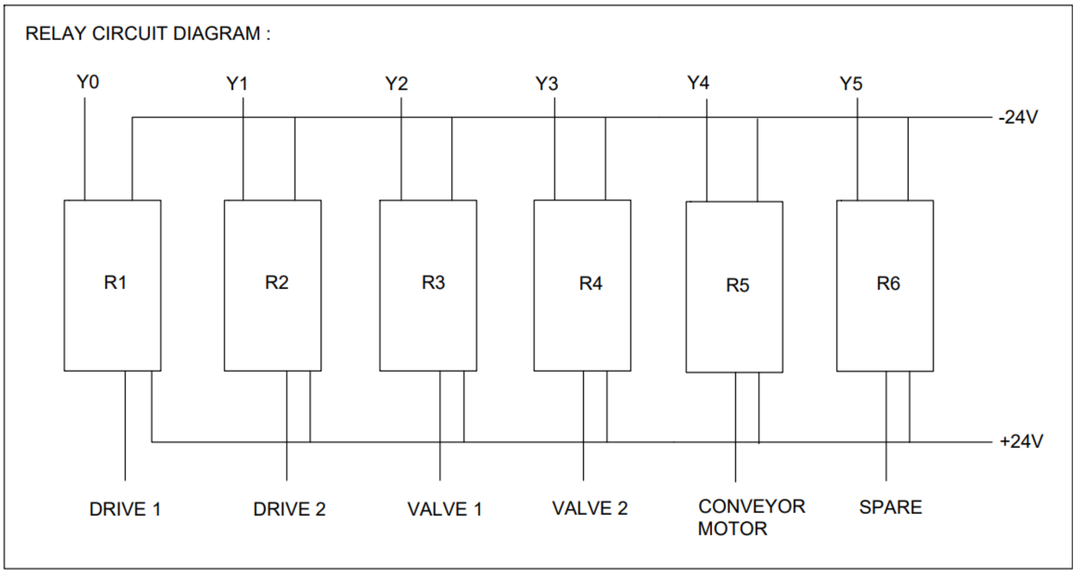
Panel layout:



PLC PIN DIAGRAM:



RELAY CIRCUIT DIAGRAM:

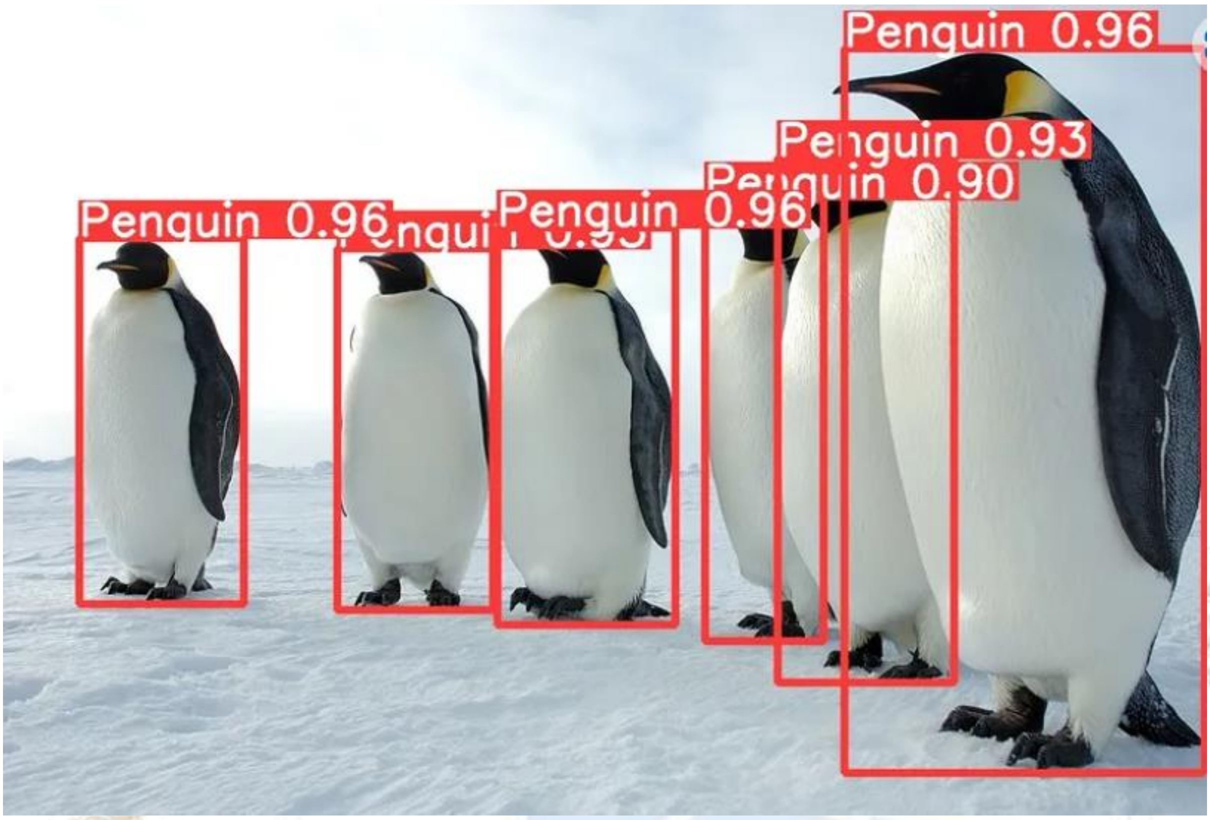


VISION ALGORITHM

YOLOv5 Object Detection:

yolov5 is a state-of-the-art object detection algorithm that builds on the success of previous versions of YOLO (You Only Look Once) and is based on deep convolutional neural networks. YOLOv5 is known for its fast and accurate object detection, which makes it an excellent choice for real-time applications.

One recent study focused on the comparison of different object detection algorithms, including YOLOv5, for pedestrian detection in crowded scenes. The study found that YOLOv5 achieved higher accuracy than other algorithms while maintaining a fast processing speed, making it an ideal choice for real-time applications that require reliable pedestrian detection.



Another study explored the use of YOLOv5 for the detection of potholes in road images. The study found that YOLOv5 achieved high accuracy and fast processing speed compared to other object detection algorithms, making it a promising candidate for the automated detection of potholes in road images.

TRANSFER LEARNING:

Transfer learning is a powerful technique that has been widely used in deep learning to improve the performance of models on different tasks. In transfer learning, a pre-trained model on one task is fine-tuned on a new, related task. This process involves freezing the pre-trained weights and retraining the model on the new task with a smaller learning rate.

One recent study explored the use of transfer learning to improve the performance of YOLOv5 on the detection of COVID-19 on chest X-rays. The study found that transfer learning significantly improved the performance of YOLOv5, reducing the number of false positives and improving the accuracy of COVID-19 detection.

Another study focused on the use of transfer learning to improve the performance of YOLOv5 on the detection of faces in low-resolution images. The study found that fine-tuning a pre-trained YOLOv5 model on a dataset of low-resolution face images significantly improved the performance of the model, reducing the number of false positives and improving the accuracy of face detection.

DATA COLLECTION:

One dataset that can be used for object detection is the COCO (Common Objects in Context) dataset. The COCO dataset is a large-scale object detection, segmentation, and captioning dataset that contains over 330,000 images with more than 2.5 million object instances labeled with 80 different object categories.

This dataset is widely used in object detection research and provides a diverse range of object categories and variations in object scale, orientation, and background.

To collect the necessary data from the COCO dataset, you can download the images and annotations from the official website (http://cocodataset.org/#download).

The annotations are provided in JSON format and contain information about the object categories, bounding boxes, and segmentation masks for each object instance in the image.
It is important to carefully review the annotations to ensure that they are accurate and of good quality.

Alternatively, you can use pre-processed datasets such as the Py Torch implementation of COCO or the OpenCV implementation of COCO that provide ready-to-use datasets with pre-defined splits for training, validation, and testing.

Once you have collected the necessary data, you can use it to train and evaluate your YOLOv5 object detection model using transfer learning in OpenCV and Python Regenerate response.

MODEL TRAINING:

Model training is a crucial step in developing an object detection system using YOLOv5 and transfer learning in OpenCV and Python. Transfer learning is a popular approach to improve the performance of object detection models by leveraging pre-trained models and adapting them to a new task. In this section, we will discuss the steps involved in training the YOLOv5 model using transfer learning on preprocessed data.

Choose a Pre-Trained Model:

The first step is to choose a pre-trained YOLOv5 model that is trained on a large-scale dataset such as COCO. You can download the pre-trained models from the official YOLOv5 GitHub repository **(https://github.com/ultralytics/yolov5).** The pre-trained models are available in different sizes (small, medium, large) and can be selected based on the available computing resources and the requirements of the task.

Prepare the Data for Training:

The preprocessed data should be prepared for training by creating data loaders that can efficiently load the data in batches. The data loaders should be configured to apply data augmentation techniques such as random cropping, flipping, and rotation to increase the variability of the data.

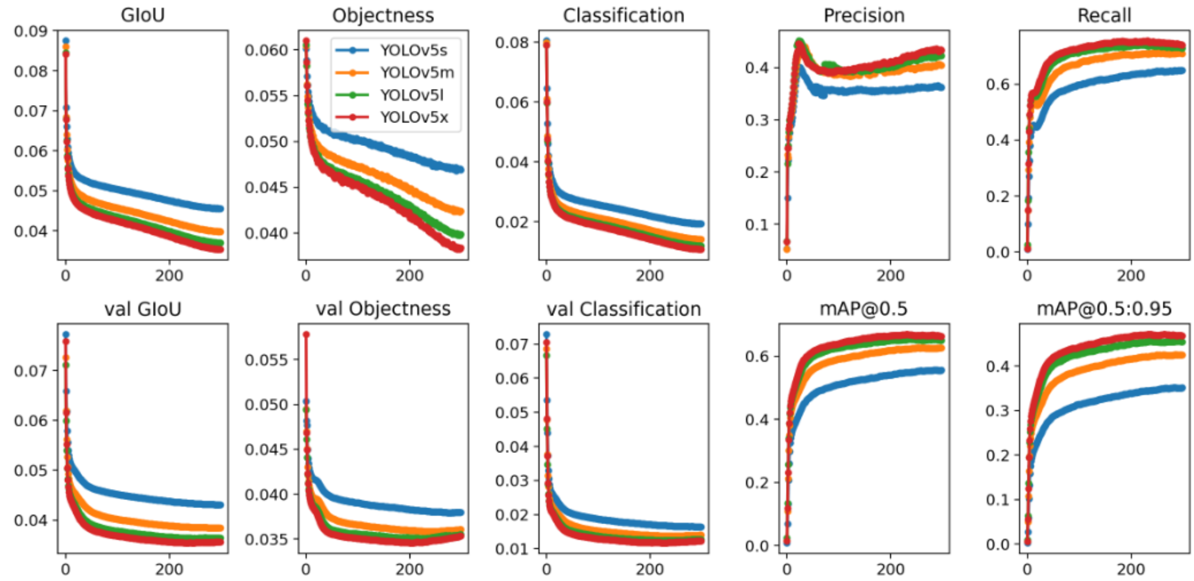
Configure the Training Parameters:

The training parameters should be configured to optimize the performance of the model. These parameters

include the learning rate, batch size, number of epochs, and weight decay. It is important to tune these parameters carefully to avoid overfitting or underfitting the model. The above given is an example model for **cat** and **dog.**

Train the Model:

Once the data and training parameters are prepared, the YOLOv5 model can be trained on the preprocessed data using transfer learning. During training, the model is updated iteratively using backpropagation to minimize the loss function.



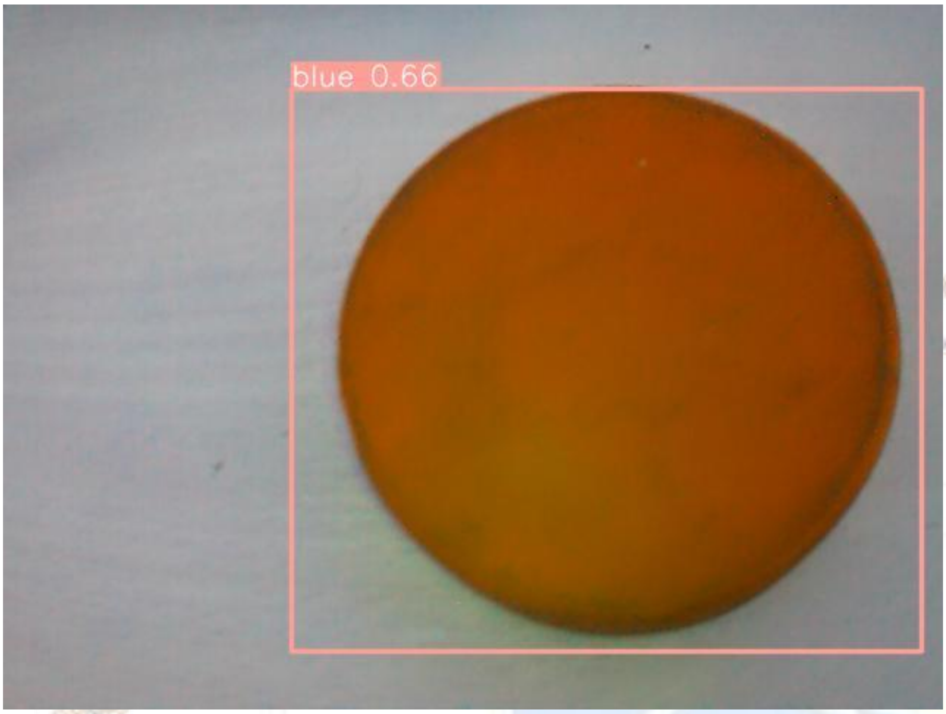
The loss function measures the difference between the predicted bounding boxes and the ground truth bounding boxes. The training progress can be monitored using metrics such as average precision, recall, and F1 score.

Evaluate the Model:

After training, the model can be evaluated on the validation set to assess its performance. The evaluation metrics such as mean average precision (MAP) can be used to measure the accuracy of the model. The model can be fine-tuned further by adjusting the training parameters or applying different data augmentation techniques.

Test the Model:

Finally, the trained model can be tested on a separate test set to assess its performance on unseen data.



The test results can be visualized using metrics such as precision, recall, and F1 score.

In summary, training the YOLOv5 model using transfer learning on preprocessed data involves choosing a pre-trained model, preparing the data for training, configuring the training parameters, training the model, evaluating the model, and testing the model. This process requires careful tuning of various parameters and techniques to optimize the performance of the model.

Prepare and validate the test data:

The validation and test datasets should be prepared by creating data loaders that can efficiently load the data in batches. The data loaders should be configured to apply the same data augmentation techniques used during training to ensure consistency in the evaluation process.

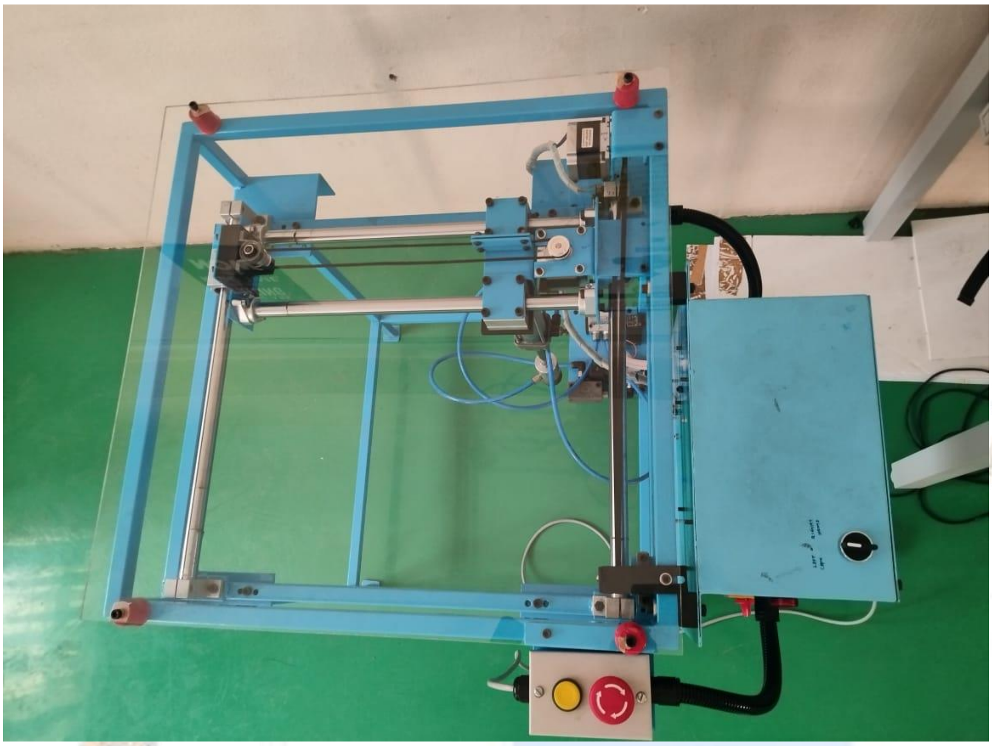
Fine-Tune the Model:

Based on the evaluation results, the model can be fine-tuned further by adjusting the training parameters or applying different data augmentation techniques. The fine-tuning process should be iterative, with frequent evaluation of the model's performance on the validation dataset.

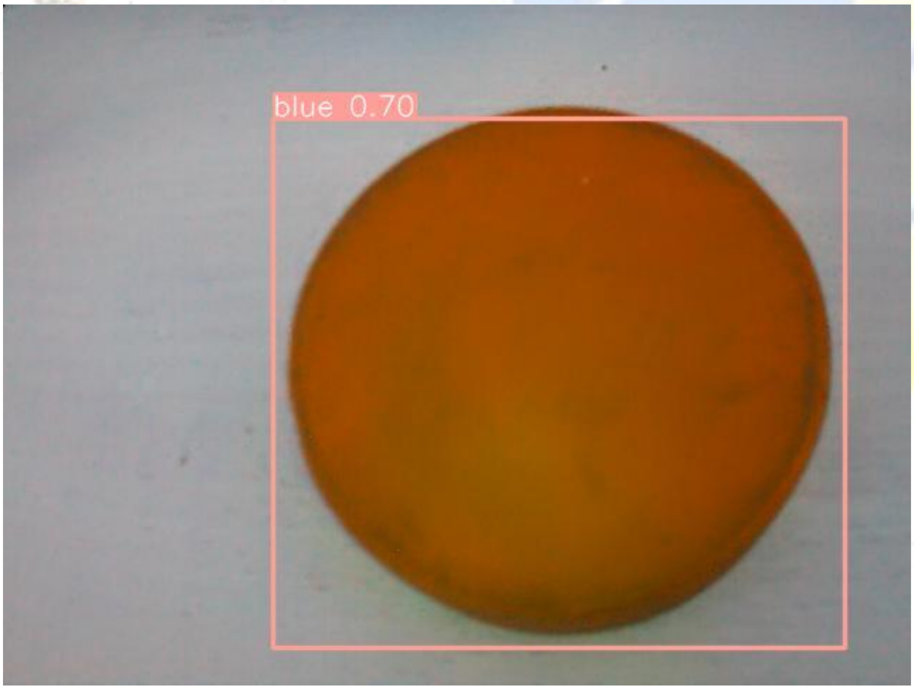
Test the Model:

Once the model is fine-tuned, it can be applied to the test dataset to assess its performance on unseen data. The test results should be interpreted carefully to ensure that the model is not overfitting the training data.

Final Prototype:



Final Prototype 1

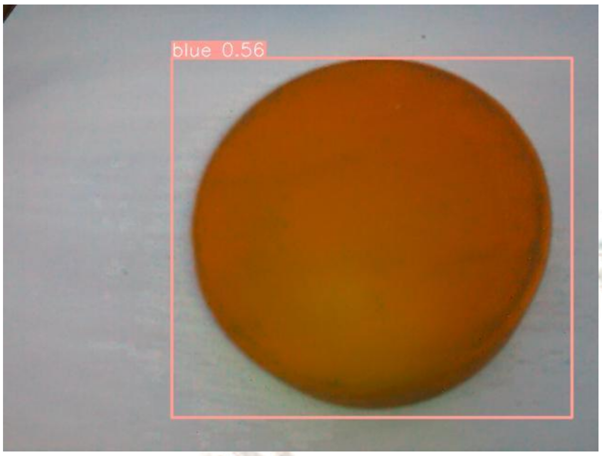


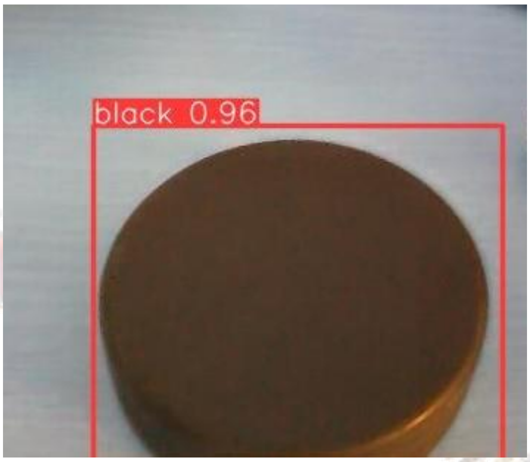
Final Prototype 2

FINAL OBSERVATION

In summary, evaluating the trained model's performance on the validation and test datasets involves choosing appropriate evaluation metrics, preparing the validation and test data, running inference on the validation data, interpreting the results, fine-tuning the model, and testing the model on the test dataset the evaluation process requires careful interpretation.

Presenting the results of the evaluation is a critical step in any object detection project using YOLOv5 and transfer learning in OpenCV and Python. The results provide insights into the accuracy of the model and help to identify areas for improvement. In this section, we will discuss the key aspects to consider when presenting the evaluation results.





Accuracy	Black 0.56	Blue 0.96
Execution time	0.29024 in sec	0.282918 in sec

REFERENCES

1) https://github.com/ultralytics/yolov5

2) Ali Ahmad El Souki, 2D Pick and Place robot, The Lebanese University Thesis, June 17, 2013.

3) Pablo Sánchez-Sánchez; Fernando Reyes-Cortés, Cartesian Control for Robot Manipulators, Manipulators Trends and Development, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5

4) S. Krishankant, Computer Based instrumentation Control, New Delhi: PHI Learning Pvt. Ltd, 2009.

5) Ma Yinyuan; Jiang Zhaoyuan, Task-Oriented Analysis and Design Method for Developing PLC Programs for Mechanical System Control, International Conference on Measuring Technology and Mechatronics Automation, IEEE Conference Publications, vol. 3, pp 726-729, 2010.

6) Reiner Dudziak, Dirk Mohr, Development Projects as an Integral Element Education of Mechatronics Engineers, IEEE Mechatronics-REM, 2012.

7) P.H Meckl; W.P Seering, Experimental Evaluation of Shaped Inputs to Reduce Vibration for a Cartesian Robot, Journal of Dynamic System, Measurement, and control,volume112, doi:10.1115/1.2896122.

8) Pattacini, U, Nori, F, Natale, L, An Experimental evaluation of a novel minimum-jerk Cartesian controller for Humanoid robots, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on 18-22 oct. 2010.

9) Wilson, William J, Relative end effector control using Cartesian position based visual serving, Robotics and Automation, IEEE Transactions on (Volume: 12, Issue: 5).