PetWave: Interactive Robotic Arm for Human and Dog Interaction

Zhao Zhenhai,Yangxue,Wang Bohan, Zhang Zihao, Wei Yutong, Hong Yuying and Li Yunzhe

***Abstract*—This project aims to develop a virtual environment enabling effective interaction between a robotic arm and humans as well as dogs. By integrating advanced machine learning algorithms and the ROS framework, the project achieves accurate recognition of humans and dogs within the environment, facilitating simple interactive actions such as waving and petting. Utilizing the YOLO algorithm for real-time object detection and classification, experiments were conducted within the ROS Reality virtual reality framework. The paper synthesizes insights from three relevant studies, exploring the applications and advantages of the ROS framework, object detection algorithms, and long-range recognition models. Preliminary experimental results demonstrate successful recognition of humans and dogs, alongside the capability for simple interactions. In conclusion, this research innovatively combines a robotic arm with advanced algorithms, offering new perspectives and methods for efficient human-robot interaction.**

I. Introduction

The objective of this project is to create a simulated environment enabling interaction between a robotic arm, humans, and dogs. For example, it can wave in response to a person's presence and pat a dog's head upon detection.

The methodology encompasses the utilization of machine learning algorithms for object detection and classification, integration of a robotic arm model within the ROS framework, and deployment of path planning algorithms for navigation.This paper delineates the advancements and initial discoveries achieved until the milestone checkpoint. During subsequent phases of optimization and iteration, its recognition capabilities and interactive gestures are intended to be enhanced.

These enhancements may include recognizing cats and engaging in playful interactions to entertain them,as figure.1 , as well as assessing the proximity between the robotic arm and individuals, waving when they are distant and offering a handshake when they are nearby.

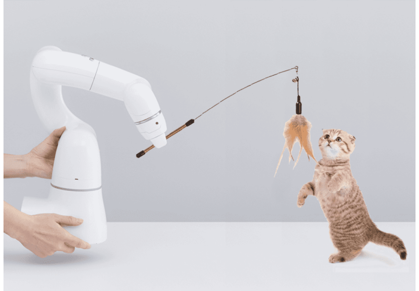


Fig. 1. Tease Cat Robot

II. Problem Statement

The objective of this project is to develop a virtual environment facilitating interactions between a robotic arm, humans, and dogs. Specifically, the aim is to enable accurate recognition of humans and dogs by the robotic arm within the environment, allowing for simple interactions such as waving to humans and patting dogs.

The project plan entails the study of YOLO algorithm principles and utilization of the COCO dataset for training a deep learning model capable of real-time detection and recognition, then procuring a controllable robotic arm within the ROS framework and integrate the image recognition algorithm with the robotic arm for simulation experiments conducted in the Gazebo simulator.

The anticipated outcome is for the robotic arm to accurately detect and differentiate between humans and dogs within the Gazebo simulator and interact with them based on predefined actions.Evaluation of the project will be based on the accuracy of human and dog detection, the precision of robotic arm interaction gestures, and the response time of the robotic arm in recognition and interaction.

III. Literature Review

[1] David Whitney; Eric Rosen; Daniel Ullman; Elizabeth Phillips; Stefanie Tellex ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1-9, doi: 10.1109/IROS.2018.8593513.*

This paper aims to introduce ROS Reality, which is an interface enabling remote operation of ROS-enabled robots using consumer-grade virtual reality (VR) and augmented reality (AR) hardware. The paper describes the system architecture and applications of ROS Reality, detailing how the system facilitates VR remote operation and integrates consumer-grade VR and AR hardware with the ROS system, allowing users to view and control robots over the internet.

This paper provides some insights for the interaction between the robotic arm and users in the project. Referring to the system architecture and methods in this paper, one can understand how to integrate with the ROS system and utilize consumer-grade VR and AR hardware for remote operation and user interface.

[2] Shaoqing Ren; Kaiming He; Ross Girshick; Jian Sun Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *in IEEE Transactions on Pattern Analysis and Machine Intelligence,*

vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

This paper introduces a novel object detection method called the Region Proposal Network (RPN), which shares full-image convolutional features within the object detection network, thereby achieving nearly cost-free region proposals. This paper can provide an effective way to reduce the computational cost of region proposals and improve the speed and accuracy of object detection. Considering the application of this method to the project can enhance the robotic arm's recognition and interaction capabilities with humans and dogs.

[3] Felix Gunawan; Chih-Lyang Hwang; Zih-En Cheng ROI-YOLOv8-Based Far-Distance Face-Recognition. *2023 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, Taiwan, 2023, pp. 1-6, doi: 10.1109/ARIS59192.2023.10268512.*

This paper introduces a model for far-distance face recognition utilizing ROI-YOLOv8, trained on custom datasets with various augmentation levels. The model adopts a two-stage recognition approach, initially employing a pre-trained YOLOv8 for human detection followed by ROI-YOLOv8 for face recognition. Evaluation metrics such as mAP50 are utilized to assess performance, with the trained model effectively recognizing faces at distances of 30 to 35 meters with high confidence. The methodology and insights provided by this study can inform the design and training of a recognition system for the robotic arm, particularly in scenarios necessitating far-distance detection and recognition of humans and dogs.

IV. Technical Approach

A. Robot Operation System

ROS, short for Robot Operating System, is a flexible framework and toolkit for robot development. It provides a series of libraries and tools to assist developers in creating, managing, and deploying robot applications.

ROS provides a variety of usable robotic arm model packages, such as URDF models, MoveIt software package, etc. The Gazebo simulation platform can visualize the functions of robotic arms, facilitating qualitative evaluation of the results.

1) URDF: URDF (Unified Robot Description Format) is an XML syntax framework used to describe the structure and attributes of robots. In ROS, URDF plays a crucial role, primarily used to describe the geometric shapes, link structures, joint types, and connection relationships of robots.

The process of leveraging URDF in ROS involves creating URDF files, utilizing XML syntax to describe the geometric structure, linkages, joint connections, and physical properties of robots. Subsequently, loading URDF models into the ROS environment can be accomplished using the roslaunch command or Python scripts. Initiating a simulation environment in ROS, such as the Gazebo simulation platform, loading URDF models, and conducting simulation and analysis to assess the robot's performance are crucial steps. Moreover, utilizing loaded

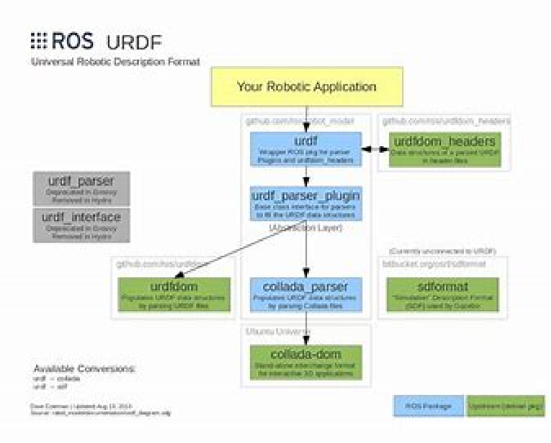


Fig. 2. URDF

2) MoveIt: MoveIt is an integrated package designed for robot motion planning and control within the Robot Operating System (ROS). It offers functionalities including motion planning, manipulation control, 3D perception, kinematics, and navigation algorithms. The typical process of utilizing MoveIt within ROS involves assembling the robot's URDF model, configuring parameters using the MoveIt! Setup Assistant tool, integrating controller plugins, and controlling robot motion through MoveIt's API or GUI, enabling developers to efficiently develop and deploy robust robot motion control applications.

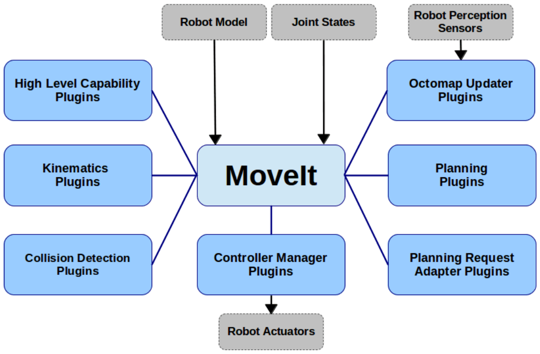


Fig. 3. MoveIt

3) Gazebo: Gazebo, an open-source robot simulation software widely utilized in robotics, offers a comprehensive suite of functionalities. These include facilitating the creation of intricate robot models using basic geometric shapes or imported CAD/Blender drawings, constructing diverse simulation scenes with objects from libraries or custom-built structures, simulating a wide array of sensors such as cameras and LiDARs, enabling the incorporation of real-world physical properties like gravity and friction into models, and providing a realistic physics simulation engine. The workflow typically involves users creating or importing models, constructing scenes, adding sensors, defining physical properties, conducting simulations, and analyzing results. In essence, Gazebo serves as a versatile platform for designing, testing, and validating robot systems through realistic simulation environments and rigorous testing procedures.

B. Image Recognition Algorithm

1) Faster-RCNN: Faster R-CNN is a deep learning model specifically designed for the task of object detection. It decomposes the task of object detection into two distinct subtasks: region proposal generation and region classification. This decomposition is facilitated by the incorporation of two essential components, namely the Region Proposal Network (RPN) and Fast R-CNN. The process encompasses several stages, including feature extraction, candidate box generation, non-maximum suppression, Region of Interest (RoI) mapping, classification, and bounding box regression,as shown as fig.4 This methodology not only maintains high detection accuracy but also enhances processing speed, rendering it highly applicable in diverse domains including video surveillance, autonomous driving, and medical image analysis.

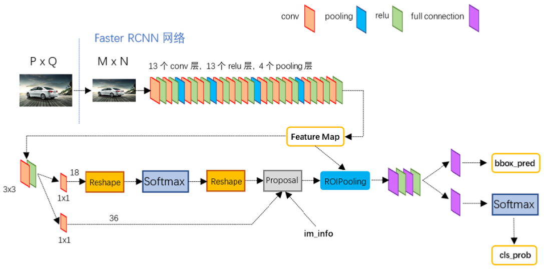


Fig. 4. Faster-RCNN

2) SSD: SSD (Single Shot MultiBox Detector) is a target detection algorithm that treats the detection task as both a regression and classification problem. By applying convolutional sliding windows on multi-scale feature maps, it predicts the positions and categories of targets, achieving a balance between speed and accuracy. Its workflow involves feature extraction with a base network, constructing a multi-scale feature map pyramid, detection with convolutional sliding windows, non-maximum suppression, and post-processing,as shown as fig.5. SSD excels in real-time capability, multi-scale detection, and accuracy, making it suitable for applications such as video surveillance and autonomous driving.

3) YOLO: YOLO(You Only Look Once)is a popular object detection algorithm known for its speed and accuracy.The core principle of YOLO (You Only Look Once) is to treat the object detection task as a single end-to-end regression problem. It employs a single neural network to predict the bounding boxes and class probabilities of objects within an image. Compared to traditional object detection methods, YOLO can directly output the positions and class probabilities of all objects in a single run, making it faster.

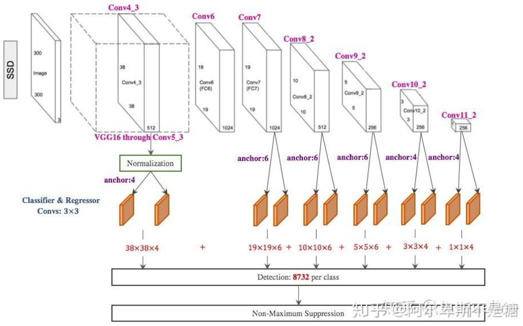


Fig. 5. SSD

1.Approach:

Whole Image Prediction: YOLO feeds the entire image into a neural network and outputs bounding boxes and class probabilities in a single forward pass.

Grid Segmentation: The image is divided into fixed-size grids, with each grid responsible for predicting the presence of objects, along with their bounding boxes and classes.

Feature Extraction: Deep Convolutional Neural Networks (CNNs) are applied to each grid cell to extract features.

Regression Output: The CNN outputs multiple bounding boxes and class probabilities for each grid cell.

Non-Maximum Suppression (NMS): Overlapping bounding boxes are merged using the NMS algorithm to reduce duplicate detections.

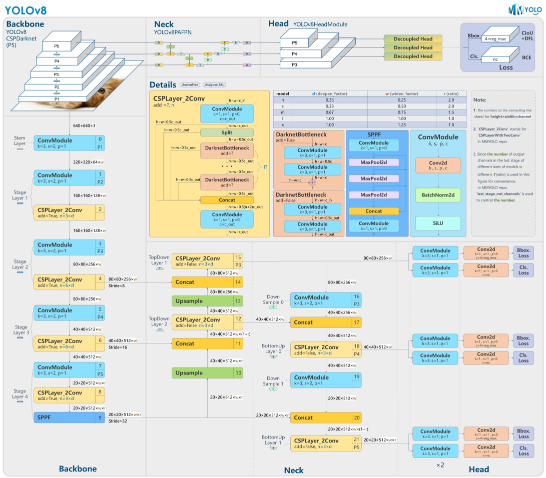


Fig. 6. YoloV8 Approach

2.Workflow:

Input Image: Feed the image to be detected into the YOLO model.

Feature Extraction: Extract image features using CNN.

Prediction: For each grid cell, predict bounding boxes and class probabilities using CNN.

NMS: Apply non-maximum suppression to filter out overlapping bounding boxes.

Output: Output the final bounding boxes and class probabilities.

3.Significance:

Object Detection: Primarily used for object detection tasks in images, capable of detecting multiple objects and providing their positions and classes.

Real-time Performance: Due to its efficient design, YOLO performs exceptionally well in real-time applications such as autonomous driving, surveillance, etc.

Single-stage Detection: Compared to two-stage detection methods like Faster R-CNN, YOLO only requires a single forward pass to complete the detection task, making it more efficient.

Multi-scale Detection: YOLO can handle objects of different sizes, making it suitable for various scenarios.

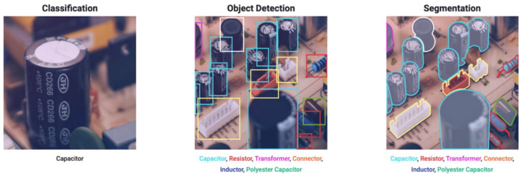

Classification
Object Detection
Segmentation
Capacitor
Capacitor, Resistor, Transformer, Connector, Inductor, Polyester Capacitor
Capacitor, Resistor, Transformer, Connector, Inductor, Polyester Capacitor

Fig. 7. Yolo Significance

V. Preliminary Results

A. ROS

To ensure the robot has the ability to wave to human and pat dogs, an open-sourced model is used as shown in the fig.8.

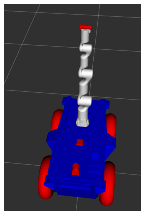


Fig. 8. The robot

The model includes a car base as well as a robotic arm with multi-degree of freedom.

With the use of ROS, advantage is taken of Moveit to control the robot arm and plan the moving path, as shown as fig.9

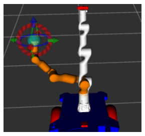


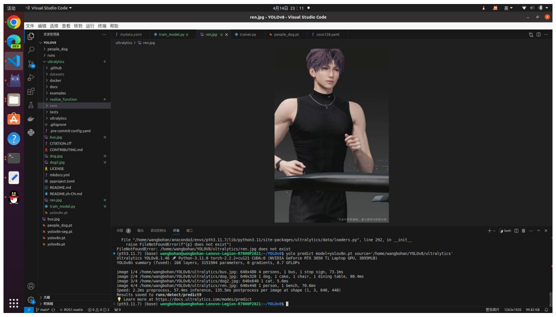
Fig. 9. The control of robot

The path of the robot arm can be effectively planned through this method, thus ensuring the specified actions are correctly performed by the robotic arm in gazebo.

Additionally, an attempt is made to add a six-dimensional force sensor to the end of the robot arm and set limitations so that the force acting on the model is within certain boundaries.

B. YOLO

Based on evaluations of real-time performance, simplicity, resource efficiency, and applicability, the decision was made to use the YOLO algorithm and forego the Faster R-CNN and SSD algorithms in the project. Initially, a model capable of recognizing images was trained,as shown as fig.10.



```
  File "/home/wangbohan/anaconda3/envs/pth3.11.7/lib/python3.11/site-packages/ultralytics/data/loaders.py", line 292, in __init__
    raise FileNotFoundError(f"{p} does not exist")
FileNotFoundError: /home/wangbohan/YOLOV8/ultralytics/ren.jpg does not exist
(pth3.11.7) (base) wangbohan@wangbohan-Lenovo-Legion-R7000P2021:~/YOLOv8$ yolo predict model=yolov8n.pt source='/home/wangbohan/YOLOv8/ultralytics'
Ultralytics YOLOv8.1.46 🚀 Python-3.11.8 torch-2.2.2+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 Ti Laptop GPU, 3895MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs

image 1/4 /home/wangbohan/YOLOv8/ultralytics/bus.jpg: 640x480 4 persons, 1 bus, 1 stop sign, 73.1ms
image 2/4 /home/wangbohan/YOLOv8/ultralytics/dog.jpg: 640x320 1 dog, 1 cake, 1 chair, 1 dining table, 80.4ms
image 3/4 /home/wangbohan/YOLOv8/ultralytics/dog2.jpg: 640x640 1 cat, 5.6ms
image 4/4 /home/wangbohan/YOLOv8/ultralytics/ren.jpg: 640x448 1 person, 1 bench, 70.6ms
Speed: 2.2ms preprocess, 57.4ms inference, 135.5ms postprocess per image at shape (1, 3, 640, 448)
Results saved to runs/detect/predict9
💡 Learn more at https://docs.ultralytics.com/modes/predict
(pth3.11.7) (base) wangbohan@wangbohan-Lenovo-Legion-R7000P2021:~/YOLOv8$
```

Fig. 10. recognizing images

Afterwards, a camera was integrated into the YOLO algorithm to perform real-time detection of classes captured by the camera,as shown as fig.11.

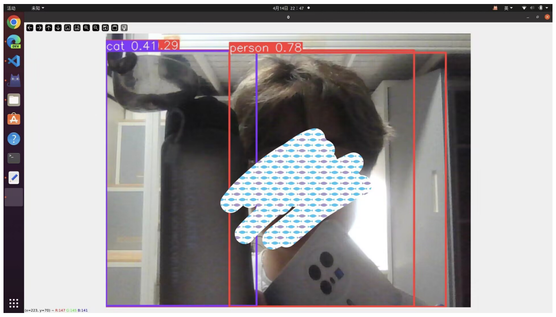


Fig. 11. real-time detection

Currently, a new dataset containing images of only humans and dogs has been re-collected, and a model with higher accuracy, capable of detecting only these two classes, has been traine, as shown as 12.

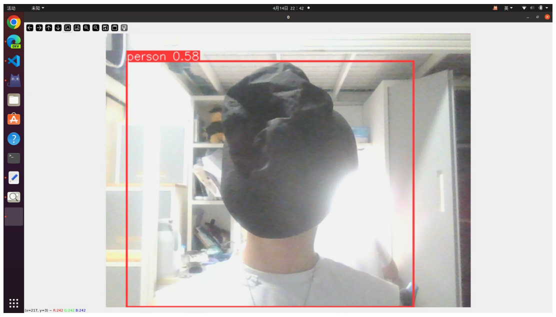


Fig. 12. higher accuracy

Subsequently, further datasets specific to the respective scenarios of humans and dogs will be collected to train corresponding models, aiming to further enhance the accuracy of the models.