Wired streaming from Go1 + real-time human detection on desktop

Created by	ⓖ guangzhl
Created time	@March 25, 2023 1:38 PM
Last edited by	Ⓚ Kangni Liu
Last edited time	@April 7, 2023 3:05 PM
Status	Done
Tags	Decision

Author: Kangni Liu

Records of reflashing SD card image

Last PR	Dependency broken after running apt-get update
Reflashed with image provided by Unitree Support	Robot worked well. Unexpected error: Wifi card couldn't be detected.
Reflashed with saved erroneous image	Failed
Reflashed with image provided by the slack community	Robot worked well. Wifi card couldn't be detected.
Decided to request a new wifi hardware from Unitree.	Switched to alternative goal: Real-time human detection with ethernet connection
Yesterday afternoon the wifi is back	We couldn't figure out why….

Decision on image processing pipeline

Goal	Receive camera stream from Go1 via ethernet and perform real-time human detection with yolov8 on Nightwalker desktop
Image transmission	Ethernet
Hardware	Go1 head camera only (without payload)
Last time	Previously, YOLOv7 was trained and tested on Colab. Because of the delayed payload, I was unable to work on human detection with thermal camera, so I can't

	take advantage of Dragoon’s dataset. YOLO trained with RGB images from Dragoon’s dataset has low accuracy. So I decided to simply use YOLO trained with Coco. The goal of this PR, configure wireless image transmission, can’t be reached because the wifi card usb couldn’t even be detected in the past week. So I switched to working on human detection via ethernet. Update on the day before PR: wifi card is detected. The hardware is not broken. It’s just contact issue.
step 1: yolov8	I installed yolov8 and trained a NN on my local computer. The NN can detect human on video and image.
challenge	Yolov8 is running on python environment, while the images from Go1 is received by a c++ I thought the image receiver program depends on other Unitree libraries (not ture, this program only needs opencv) so I decided to keep those two programs and bridge them(not necessary).
Solution 1	C++ program calling system commands, using the yolo program
Solution 2	Implement Publisher and subscriber mechenism in ROS.
I chose Solution 2	Because in the future this pipline could be used for publishing human locations.
Challenge from Solution 2	I merged the code of image receiver and ros publisher but camera could not be reached. So I installed opencv-python and opencv. The reason is that GStreamer is not being integrated in opencv-python. The tutorials are attached. I successfully built up a publisher in C++ and a subscriber in Python running YOLOv8.
Solution 3 (Final solution)	I realized that since only opencv is needed, why not rewrite the C++ program in python. I merged the image receiver with yolo.

Complete YOLO v8 Custom Object Detection Tutorial | Windows & Linux

A complete YOLO v8 custom object detection tutorial with two-classe custom dataset. I cover how to annotate custom dataset in YOLO format, setting up environment for YOLO v8, and train custom object detection model. Although I

https://www.youtube.com/watch?v=gRAyOPjQ9_s

How to build OpenCV with Gstreamer

Ubuntu

https://galaktyk.medium.com/how-to-build-opencv-with-gstreamer-b11668fa09c

OpenCV Python with Gstreamer Backend

Hi everyone, I've been playing around with getting gstreamer functionality in opencv and thought I'd prefer to use gstreamer as an opencv VideoCapture/VideoWriter backend rather than using the python gstreamer

https://discuss.bluerobotics.com/t/opencv-python-with-gstreamer-backend/8842