Posture Detection

Feature Name: Posture Detection

Version	1.0.0
Date	27 Oct 2023
Author(s)	George Wang

Overview

The ***openpose_node*** node under the ***openpose_controller*** package will be able to detect the posture of the human body. It is also able to detect the keypoints of a person's hands with the --hand flag, and facial keypoints with the --face flag.

Prerequisites

- Python 3.x installed
- ROS2 environment
- Install the following packages via apt-get:
    - GStreamer: https://gstreamer.freedesktop.org/documentation/installing/on-linux.html?gi-language=c
    - opencv: python3-opencv
    - python development package: python3-dev
- Install CUDA (11.7) and cuDNN (8.5.0):
    - CUDA: Download from website: https://developer.nvidia.com/cuda-11-7-1-download-archive
    - cuDNN: Download from website: https://developer.nvidia.com/cudnn
- Additional openpose prerequisites: https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/installation/1_prerequisites.md
    - Ignore "sudo pip install numpy opencv-python"

Usage

1. Setting up the Workspace
    a. Navigate to the ROS2 workspace

    ```
    cd openpose_ws
    ```

    b. Build packages

    ```
    colcon build --symlink-install
    ```

    c. Source the setup file

    ```
    source install/setup.bash
    ```

2. Launch the ***openpose_node*** node
    a. After successfully built, ***openpose_node*** ROS2 node can be launched via the following command:

    ```
    ros2 run openpose_controller openpose_node
    ```

3. Check detected objects
    a. A window will show the camera with keypoints attached
    b. The keypoint values can also be inspected through ***/vision/face_keypoints, /vision/hand_keypoints, /vision/body_keypoints*** with the command:

    ```
    ros2 topic echo /vision/<face/hand/body>_keypoints
    ```

Published Topic Information

- Individual keypoints within the message will include:
    - x coordinate

        - y coordinate
        - confidence score
- Format for hand and face keypoints can be found here: https://github.com/ArtificialShane/OpenPose/blob/master/doc/output.md
- Format for body keypoints can be found underneath "Keypoint Ordering in C++/Python": https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_02_output.html

Release Notes

Features:

- Integrate Openpose functionality within a ROS2 node
- Display keypoints visually on top of camera feed
- Publishes keypoints towards ROS2 topics

Known Issues:

1. cudnn runs into an error:

```
status == CUDNN_STATUS_SUCCESS (3 vs. 0) CUDNN_STATUS_BAD_PARAM
```

Notes:

- When following openpose instructions, ignore the pip install for opencv-python, use apt-get install python3-opencv instead
- For WSL2:
    - Need to install usbipd-win and rebuild the linux kernel in order to connect to camera through wsl2
    - Find camera id and attach with command in powershell:

```
usbipd wsl list
usbipd wsl attach --busid <camera id>
```

    - Confirm camera attachment and give permissions to camera with:

```
lsusb
ls -al /dev/video*
sudo chmod 777 /dev/video0
```

    - Change the number 0 to the appropriate index
- To look at openpose flags, consult: https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/include/openpose/flags.hpp
- Models for openpose can be downloaded here: https://github.com/CMU-Perceptual-Computing-Lab/openpose/issues/1602
    - Insert models downloaded under openpose/models/
        - face: pose_iter_116000.caffemodel
        - hands: pose_iter_102000.caffemodel
        - pose/body_25: pose_iter_584000.caffemodel

Future Improvements:

1. Attempt to separate the openpose wrapper and the camera node, using publishers and subscribers
2. Attempt to publish the message in an ndarray format, rather than string format