Documentation of installing OpenVino Demo 2024 and the Face Recognition Model on a Linux System

The following documentation is a guide on installing the latest demo of Openvino 2024 on a Linux system, additionally we will be specifically looking at making the Facial Recognition model run.

OpenVino 2024 does not have a facial detection model, so we will be using the facial recognition model instead, and the overall goal is to use this model to help in our 'AI Based Asset Tracking' project.

We will be using **Ubuntu 22.04** and **OpenVino 2024.1**, this documentation will show you how to install the Openvino demo and will be specific towards installing the facial recognition model.

Steps to Installing Openvino 2024:

Firstly, we will need to make a directory, the following command creates the opt/intel directory.

```
sudo mkdir/opt/intel
```

Next, we will need to go into our download directory.

```
cd ~/Downloads
```

Next, go to **step 4 of Installing OpenVino runtime website** (Open vino runtime on Install Linux from an archive file) and copy paste the following into the terminal:

- This downloads the file and extracts the file, putting into a folder for us. It will end up moving the folder into the directory we made earlier (/opt/intel).

```
curl -L
https://storage.openvinotoolkit.org/repositories/openvino/packages/2024.1/
linux/l_openvino_toolkit_ubuntu22_2024.1.0.15008.f4afc983258_x86_64.tgz --
output openvino_2024.1.0.tgz
tar -xf openvino_2024.1.0.tgz
sudo mv l_openvino_toolkit_ubuntu22_2024.1.0.15008.f4afc983258_x86_64
/opt/intel/openvino_2024.1.0
```

Next, we need move to that newly made directory.

```
cd /opt/intel/openvino_2024.1.0
```

And we will run a shell script in that directory.

```
cd /opt/intel/openvino_2024.1.0
sudo -E ./install_dependencies/install_openvino_dependencies.sh
```

(Optional step, but make sure to install the NumPy library, its only required for using the PythonAPI)

Now we go back one level down to this directory.

```
cd /opt/intel
```

Next, we put this in the terminal, the reason we do this is so we can make a symbolic link for the directory we just moved into, so for easier access or references in the future.

```
cd /opt/intel

sudo ln -s openvino_2024.1.0 openvino_2024
```

This is what our new symbolic link should look like:

```
source /opt/intel/openvino_2024/setupvars.sh
```

The source command is configuring the document.

OpenVino 2024 runtime is now installed.

Steps to gathering Openvino Demo files:

Firstly, make a directory for the openvino demos.

```
mkdir ~/ovdemos
```

Enter the newly made directory.

```
cd ~/ovdemos
```

Clone the demo files from the openVino GitHub.

The link to the GitHub repository for Openvino 2024:

```
git clone -- recurse submodules -b releases/2024/1 --depth 1
https://github.com/openvinotoolkit/open_model_zoo.git
```

This is cloning all the demo files and models of openVino, out of which we will be using the facial recognition demo.

Then enter the newly made directory.

```
cd ~/ovdemos/open_model_zoo/
```

The following command, will build all the demo files, giving us the ability to build the model to let us later implement the demos for use.

```
./demos/build_demos/sh
```

Go back to the home directory:

```
cd ~/
```

Then we are creating the virtual env for openVino with python:

```
python3 -m env openvino_env
```

Then we activate the environment:

```
source openvino_env/bin/activate
```

Python Library Installations

Make sure to install the following modules as shown below:

```
Python3 -m pip install -- upgrade pip
```

```
pip install openvino-dev==2024
pip install openvino-dev[extras]
pip install openvino-dev[tensorflow, onnx]
```

openvino-dev==2024

- this command installs all the openvino dev tools package, this allows us to use the omz_downloader and omz_converter commands.

openvino-dev[extras]

- this download is getting the extra packages for the frameworks of the openvino models.

openvino-dev[tensorflow2, onnx]

- this download is getting the extra packages for the frameworks of the openvino models.

(Optional)

If you want to download all the models you can run the command:

```
omz_downloader --all
```

The rest of this documentation document provides instructions on running the facial recognition demo.

Running the Facial Recognition demo:

Since we are focusing on the Facial Recognition demo, we can continue to that directory, which is the primary demo for the project:

```
cd ~/ovdemos/open_model_zoo/demos/face_recognition_demo/python
```

For specific facial recognition models only, we will be in the directory only and will do:

```
omz_downloader --list models.lst
```

Before you run the demo, there are some errors that could arise and the main reason for those errors were because of uninstalled libraries.

For some reason, these modules were not included with the initial installation with the demos from beforehand, so make sure to install them:

```
pip install opencv-python
pip install scipy
```

This command will finally run the facial recognition demo:

```
Python ./face_recognition_demo.py -i 0 -m_fd
~/ovdemos/open_model_zoo/demos/face_recognition_demo/python/intel/face-
detection-retail-0004/FP32/face-detection-retail-0004.xml -m_lm
~/ovdemos/open_model_zoo/demos/face_recognition_demo/python/intel/landmark
s-regression-retail-0009/FP32/landmarks-regressionretail-0009.xml -m_reid
~/ovdemos/open_model_zoo/demos/face_recognition_demo/python/intel/face-
reidentification-retail-0095/FP32/face-reidentificationretail-0095.xml --
verbose --output_resolution 1920x1080 -fg
”/home/<user>/ovdemos/open_model_zoo/demos/face_recognition_demo/python/fa
ce_gallery” -d_fd CPU -d_lm CPU -d_reid CPU
```

End of documentation.

- Samuel G. and Joseph Yu