

# A programming introduction to AI models in computer vision

# Applications of computer vision

Optical character recognition: python tesseract

3D model building

Warehouse logistics

Motion capture

Surveillance

Biometrics

# Helpful references

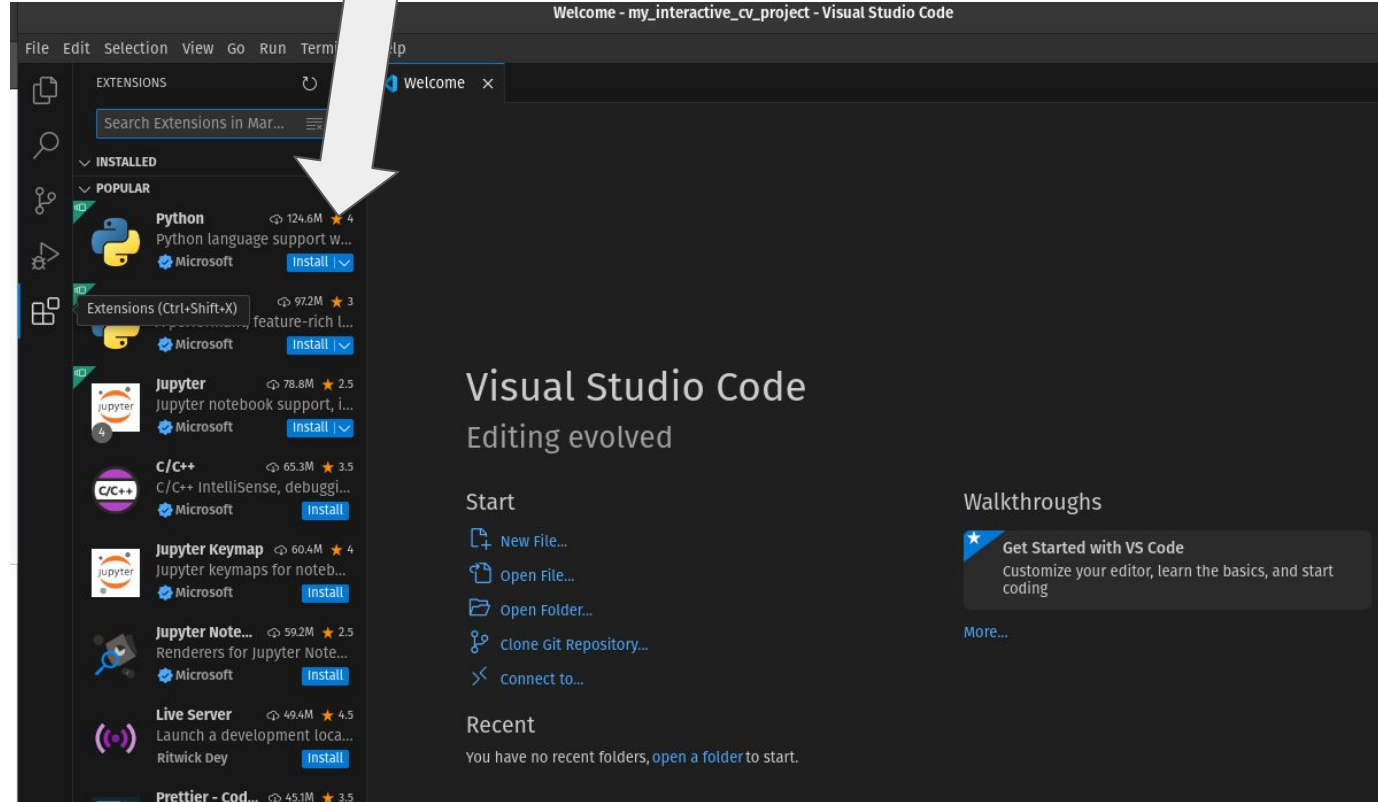
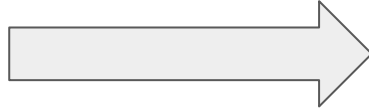
1. [VS Code quick start guide for Python](#)
2. [OpenCV-Python tutorials](#)
3. [Models prediction with Ultralytics YOLO](#)

# Setting up your development environment

1. Installing Python <https://www.python.org/downloads/>
2. Installing VS Code [https://code.visualstudio.com/docs/?dv=linux64\\_deb](https://code.visualstudio.com/docs/?dv=linux64_deb)

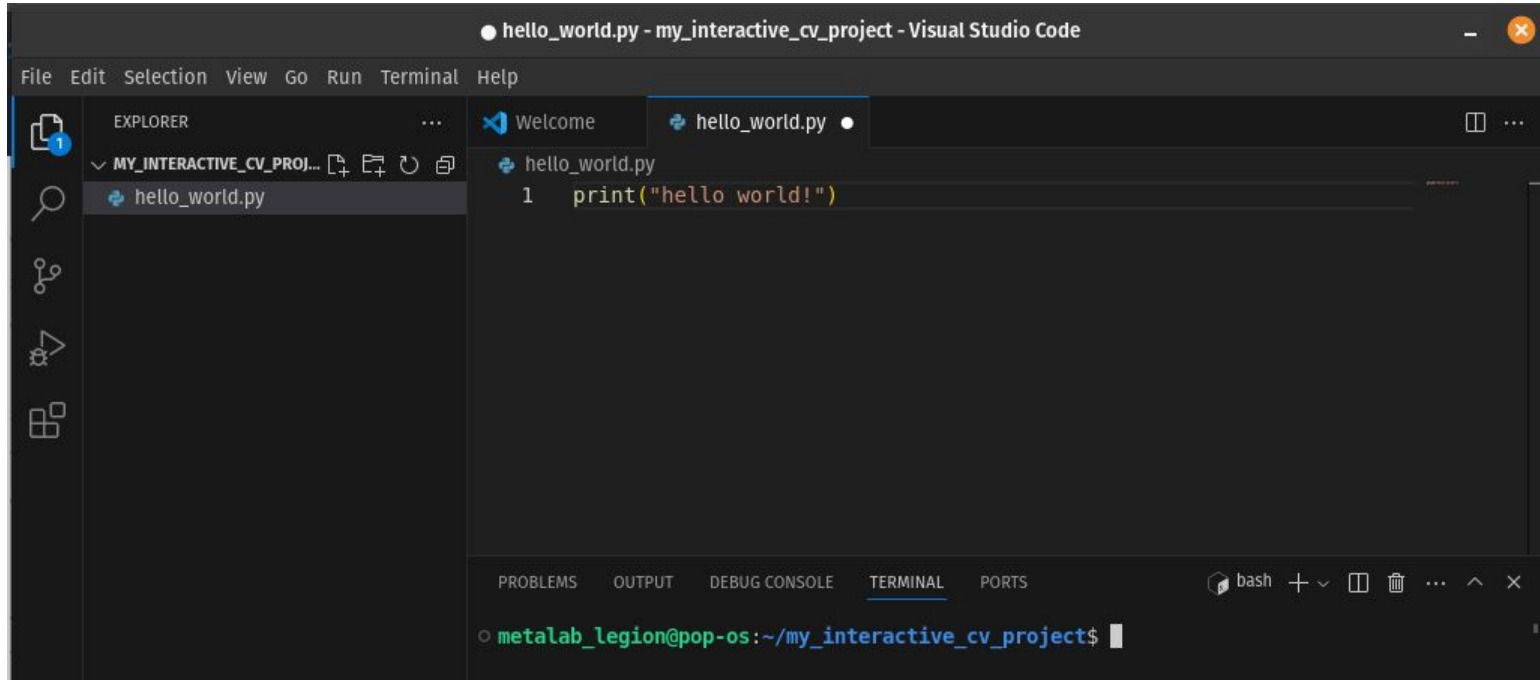
# VS Code environment

Let's install the  
python extension!

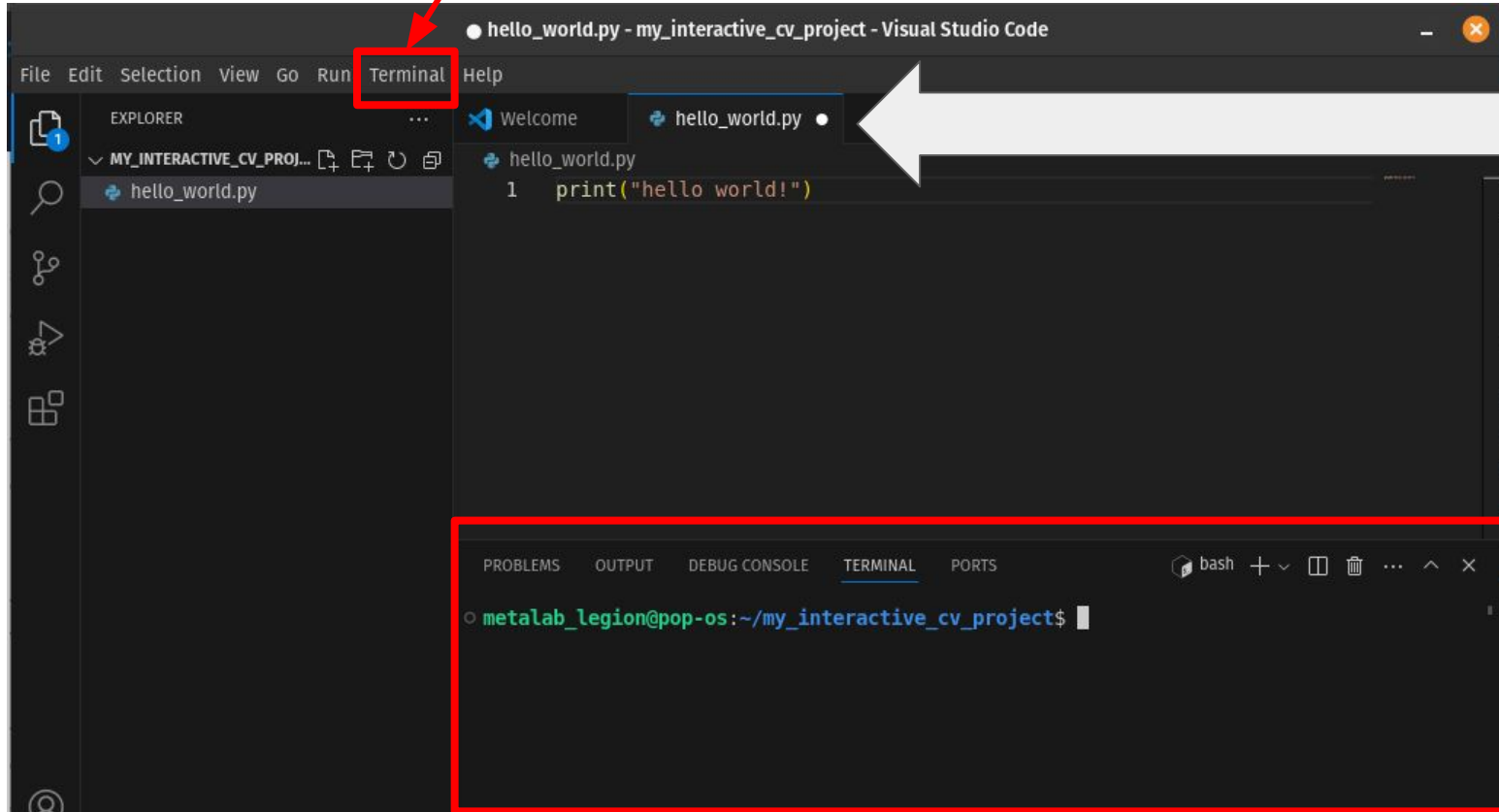


# hello\_world.py

1. Let's create a folder called "my\_interactive\_cv\_project"
2. We'll create a file named "hello\_world.py" in it



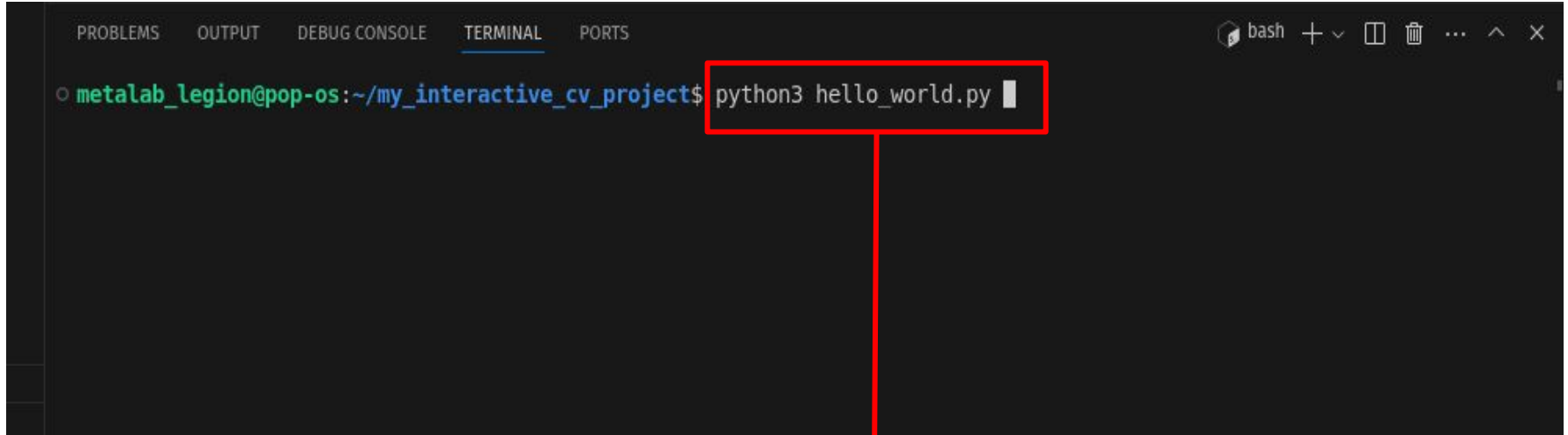
Click here to  
open a terminal



Changes  
must be  
saved!

Terminal is  
opened

# Using the terminal



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
metalab_legion@pop-os:~/my_interactive_cv_projects$ python3 hello_world.py
```

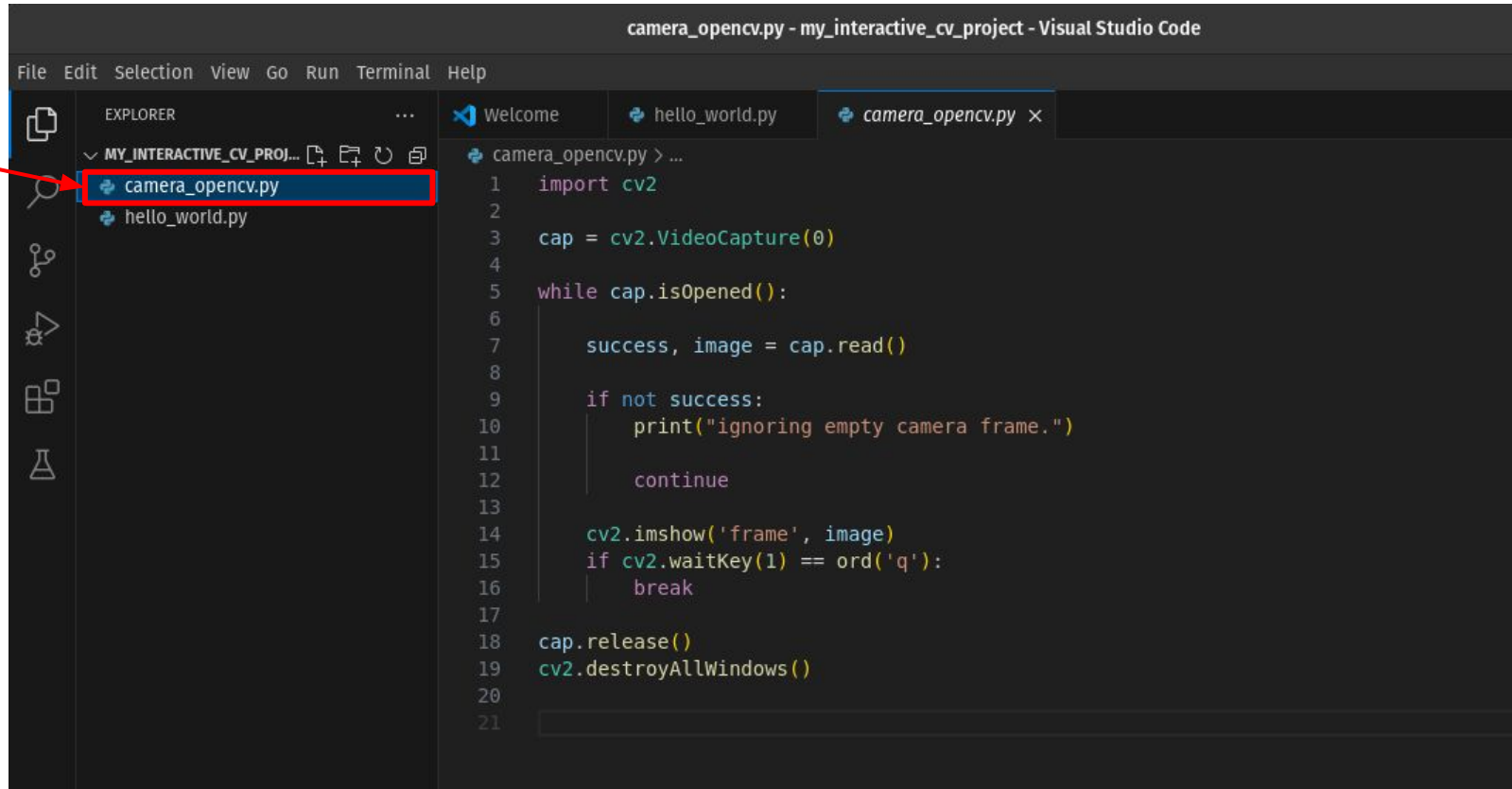
WARNING: Python command may differ based on your OS (try writing “python” rather than “python3”, for example)

Press enter on your keyboard after writing this down



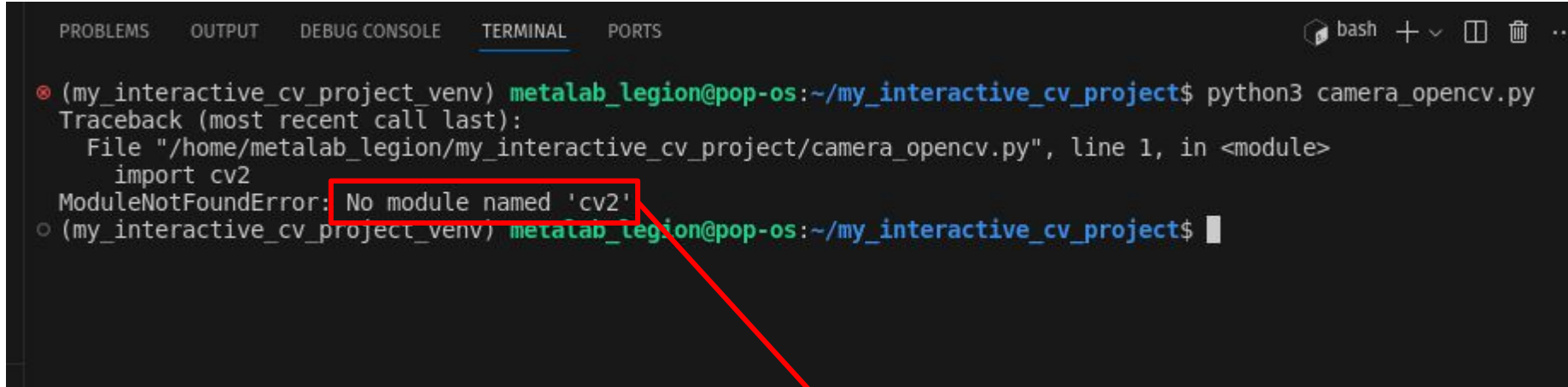
# Let's open a camera with python!

Create a file called  
camera\_opencv.py



# Let's open a camera with python..??

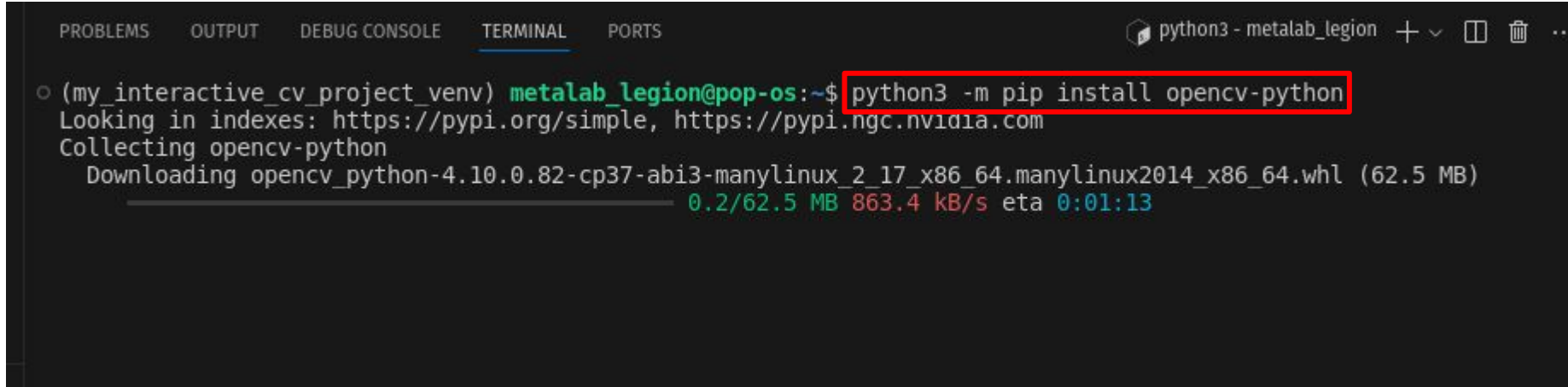
If we try running the script camera\_opencv.py, we'll probably run into the following error:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash + v [] [X] ...
(my_interactive_cv_project_venv) metalab_legion@pop-os:~/my_interactive_cv_project$ python3 camera_opencv.py
Traceback (most recent call last):
  File "/home/metalab_legion/my_interactive_cv_project/camera_opencv.py", line 1, in <module>
    import cv2
ModuleNotFoundError: No module named 'cv2'
(my_interactive_cv_project_venv) metalab_legion@pop-os:~/my_interactive_cv_project$
```

We need to install the module cv2!

# Installing opencv-python



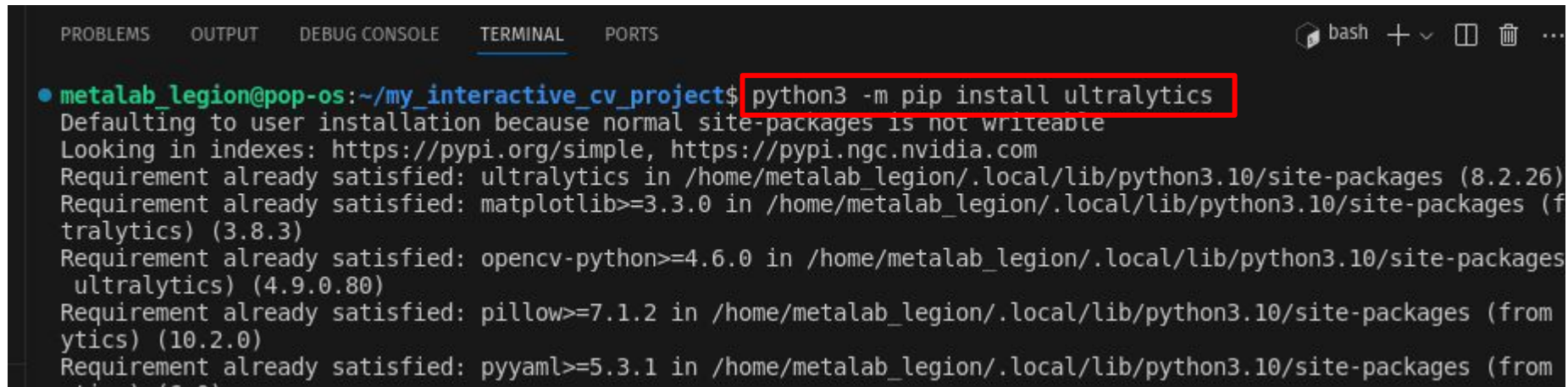
The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal title bar says 'python3 - metalab\_legion'. The prompt is '(my\_interactive\_cv\_project\_venv) metalab\_legion@pop-os:~\$'. The command 'python3 -m pip install opencv-python' is entered and highlighted with a red box. The output shows the package being collected and downloaded. A progress bar is visible for the download of 'opencv\_python-4.10.0.82-cp37-abi3-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (62.5 MB)'. The progress bar shows '0.2/62.5 MB' downloaded, with a speed of '863.4 kB/s' and an estimated time of 'eta 0:01:13'.

```
(my_interactive_cv_project_venv) metalab_legion@pop-os:~$ python3 -m pip install opencv-python
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting opencv-python
  Downloading opencv_python-4.10.0.82-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (62.5 MB)
    0.2/62.5 MB 863.4 kB/s eta 0:01:13
```

Let's try running camera\_opencv.py again!

# Let's run a human pose estimation model on our computer!

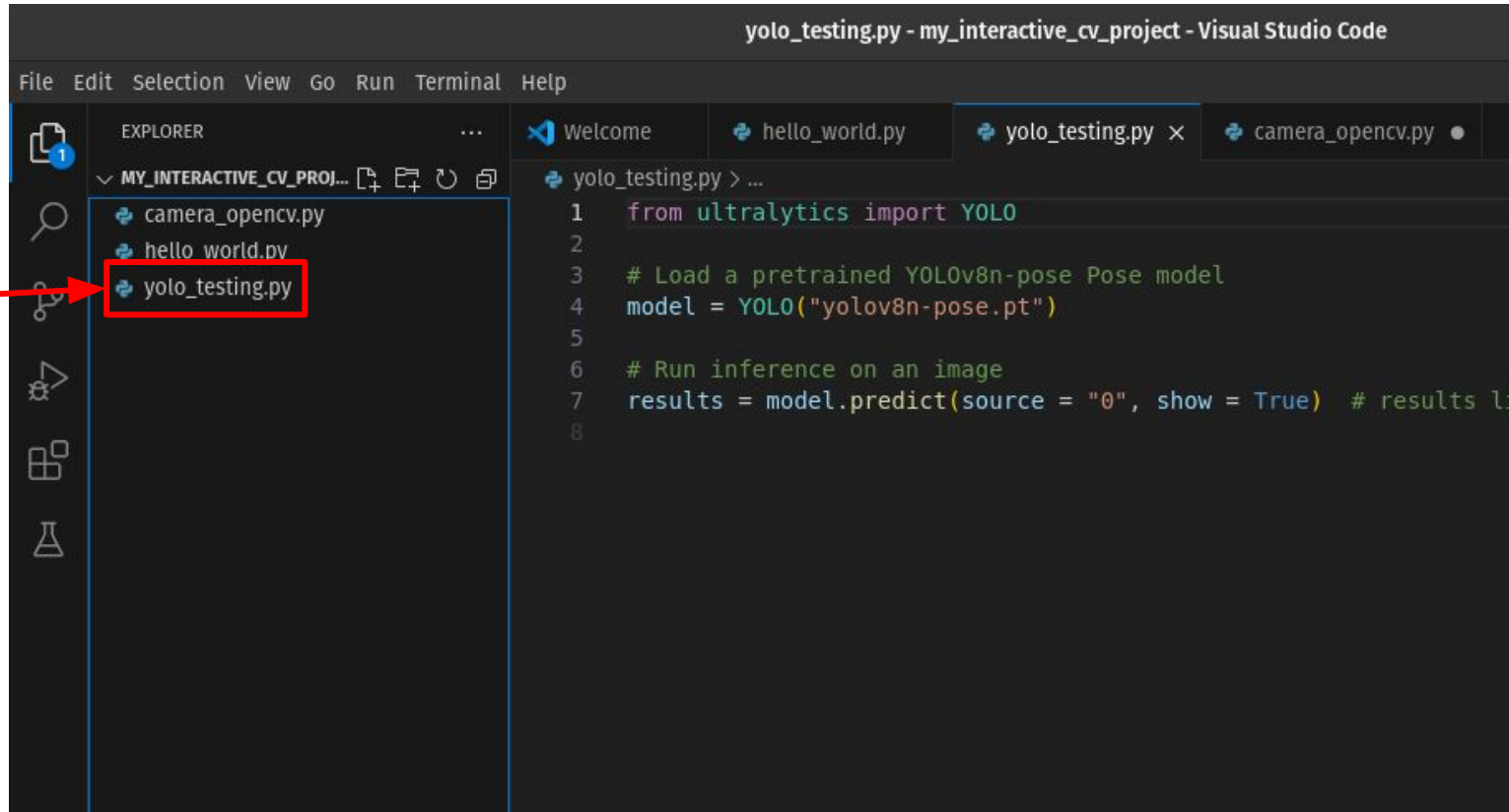
First, we'll install the Ultralytics python package

A screenshot of a terminal window with a dark background. The terminal title bar shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. On the right side of the title bar, there are icons for a terminal, a plus sign, a minus sign, a window, a trash can, and an ellipsis. The terminal content shows a command prompt for 'metalab\_legion@pop-os:~/my\_interactive\_cv\_project\$' followed by the command 'python3 -m pip install ultralytics', which is highlighted with a red rectangular box. Below the command, the terminal displays several status messages: 'Defaulting to user installation because normal site-packages is not writeable', 'Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com', and a series of 'Requirement already satisfied' messages for 'ultralytics (8.2.26)', 'matplotlib (3.8.3)', 'opencv-python (4.9.0.80)', 'pillow (10.2.0)', and 'pyyaml (5.4.1)'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash + - [] ...  
● metalab_legion@pop-os:~/my_interactive_cv_project$ python3 -m pip install ultralytics  
Defaulting to user installation because normal site-packages is not writeable  
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com  
Requirement already satisfied: ultralytics in /home/metalab_legion/.local/lib/python3.10/site-packages (8.2.26)  
Requirement already satisfied: matplotlib>=3.3.0 in /home/metalab_legion/.local/lib/python3.10/site-packages (from ultralytics) (3.8.3)  
Requirement already satisfied: opencv-python>=4.6.0 in /home/metalab_legion/.local/lib/python3.10/site-packages (from ultralytics) (4.9.0.80)  
Requirement already satisfied: pillow>=7.1.2 in /home/metalab_legion/.local/lib/python3.10/site-packages (from ultralytics) (10.2.0)  
Requirement already satisfied: pyyaml>=5.3.1 in /home/metalab_legion/.local/lib/python3.10/site-packages (from ultralytics) (5.4.1)
```

# Let's run a human pose estimation model on our computer!

Create a file named  
"yolo\_testing.py" →



```
Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8n-pose.pt to 'yolov8n-pose.pt'...  
100% |██████████| 6.51M/6.51M [00:05<00:00, 1.25MB/s]
```



What just happened!?!