Yolov5 Hand Gesture Walkthrough

Contents

Preparations

In this section, we will set up the running environment for our Yolov5 python software application. To run our script, we will need to download the main python files that allow an Operating System (OS) to run any python project. Integrated Development Environments (IDE) are optional but highly recommended. We can use any IDE that is compatible with Python, but for this guide we will use PyCharm as it is very user friendly, powerful and allows for the easy creation of python virtual environments. We recommend using virtual environments in our python projects since it creates stand alone projects that do not break compatibility with older, previously created python projects.

Install Python

To run Yolov5 we will need to install any python version greater than, or equal to Python 3.7. Posted below are the links to install python version 3.7 to 3.10:

- Python 3.7
- Python 3.8
- Python 3.9
- Python 3.10

While you can use any version listed above, we recommend you use Python 3.9 as python 3.10 is only a month old, meaning there could be some bugs with this version. Select the version that matches your Operating System (OS). If you are using a Windows OS, then we suggest installing “Windows x86-64 executable installer”.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		e19e75ec81dd04de27797bf3f9d918fd	26724009	SIG
XZ compressed source tarball	Source release		6ebfe157f6e88d9eabfbaf3fa92129f6	18866140	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	16ca86fa3467e75bade26b8a9703c27f	31132316	SIG
Windows help file	Windows		9ea6fc676f0fa3b95af3c5b3400120d6	8757017	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	60d0d94337ef657c2cca1d3d9a6dd94b	8387074	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	b61a33dc28f13b561452f3089c87eb63	28158664	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	733df85afb160482c5636ca09b89c4c8	1364352	SIG
Windows x86 embeddable zip file	Windows		d81fc534080e10bb4172ad7ae3da5247	7553872	SIG
Windows x86 executable installer	Windows		4a2812db8ab9f2e522c96c7728cfcccb	27066912	SIG
Windows x86 web-based installer	Windows		cdbfa799e6760c13d06d0c2374110aa3	1327384	SIG

Python Virtual Environment

If you are using PyCharm, a virtual environment will be setup automatically. If you are not using an IDE, or are using a text editor, you can setup a virtual environment in the file path of your choice by running the following two commands in the terminal:

For Windows:

1. To Setup the virtual environment:
    a. python3 -m venv [Virtual Environment Name]
        i. Example : python3 -m venv aircraftdefect_venv
2. The virtual environment only needs to be created once. It will have to be activated/run each time we want to run our aircraft script. To activate the virtual environment run the following script:
    a. .\[Virtual Environment Folder Name]\Scripts\activate
        i. Example: .\aircraftdefect_venv\Scripts\activate

For Mac/Linux:

1. To Setup the virtual environment:
    a. python3 -m venv [Virtual Environment Name]
        i. Example : python3 -m venv aircraftdefect_venv
2. The virtual environment only needs to be created once. It will have to be activated/run each time we want to run our aircraft script. To activate the virtual environment run the following script:
    a. .\[Virtual Environment Folder Name]\bin\activate
        i. Example: .\aircraftdefect_venv\bin\activate

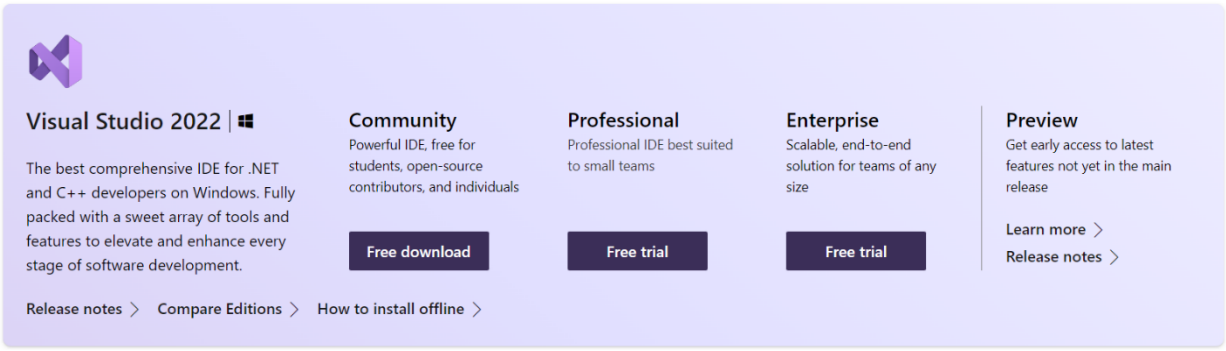
To deactivate the virtual environment, you can either turn off the computer, or type “deactivate” in the top directory of the virtual environment pathway.

Package Installations

After we have installed our Python version and our python virtual environment, we will now install all the required dependencies so our trained model can run. Our model can run without using a GPU but using a GPU will make our model run 40-20 times faster. This will be a significant advantage if this model needs to process videos. Skip to “Yolov5 installation” if you do not have a NVDIA GPU chip. If you do have a GPU chip, we will need to install CUDA and PyTorch to have our model run a GPU. Unfortunately, we will also need to install Visual Studio 2022 as Visual Studio comes with Microsoft C++ Build tools which will prevent build errors when we run our PyTorch installation command.

Installing Visual Studio 2022

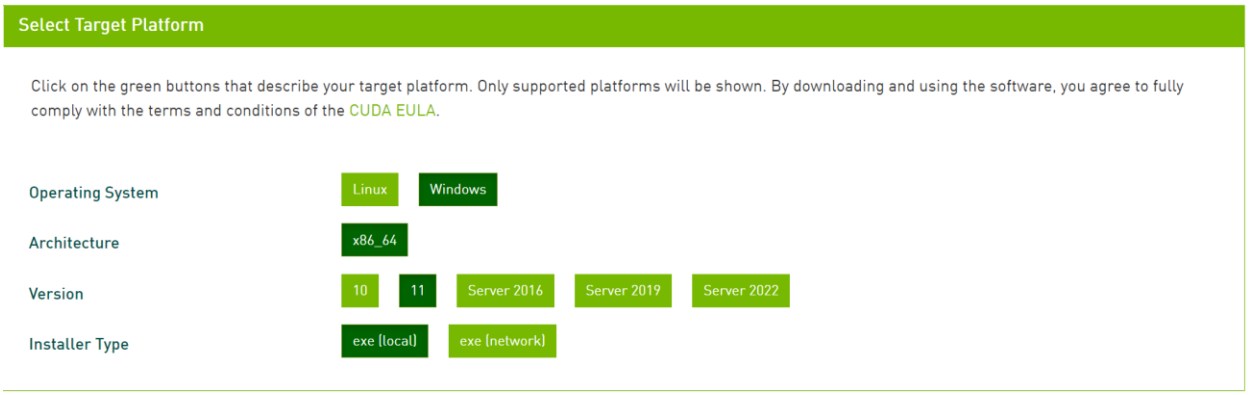
Visual Studio 2022 is a fantastic IDE that can be used for multiple different software applications. We will not need to use it to run our model once the installation of the IDE is complete, but it can be potentially used for future software projects. To install Visual Studio Code 2022, we recommend you going to the following link. At this page, you should download the Community edition (as its the only free version) for the OS you are using. By clicking on “Free Download” the executable file will automatically be downloaded to your downloads folder. Once the download is completed, follow the user guide that will pop up after clicking on the executable file to finish installing Visual Studio 2022.



Installing CUDA

CUDA is NVIDIA's parallel computing platform for GPUs. CUDA makes it possible to run parts of the computation in parallel, which significantly speeds up our model. Please use the NVIDIA CUDA toolkit page here to download CUDA by selecting the options that apply to your system. Once the installation is complete, it requires a system reboot. This step will only work if you have a NVDIA GPU chip installed in your hardware system. If you don't have a NVDIA GPU, then the model will have to be run using the CPU chip installed in the computer.

If you are using Windows, we recommend you use version "11". If you are using Linux, please select the architecture and distribution that your OS is running.



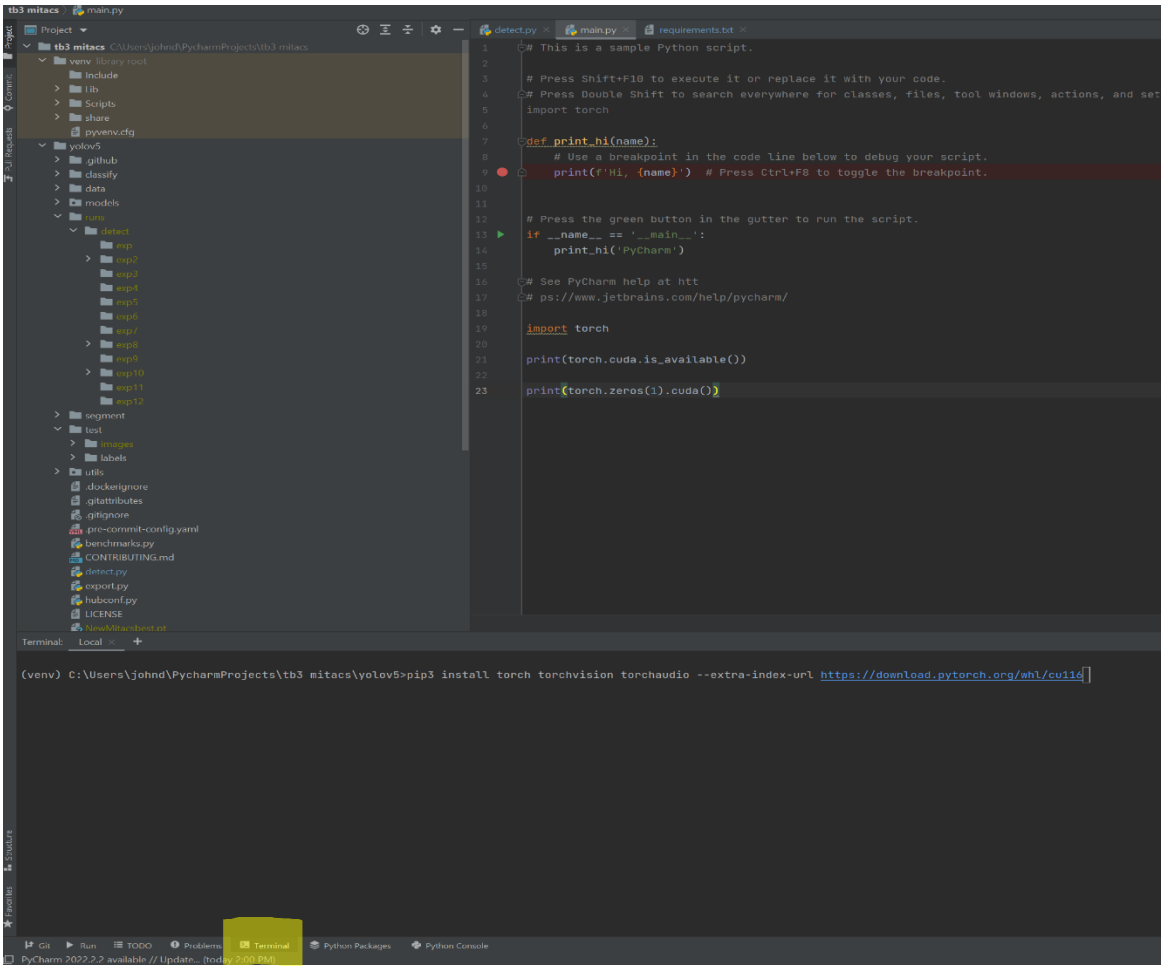
Installing PyTorch

PyTorch is an open-source machine learning framework heavily used in computer vision applications. YOLOv5 uses the PyTorch framework. PyTorch can be installed at this site listed here .

It is essential to select the right OS and package installer option based on your system. We must also choose the correct Python language and appropriate CUDA version so our installation can happen without any errors. Please note that running the detection on the CPU will be slower but will not reduce the quality of results. The figure below shows an example of a downloadable option for the Windows system using the pip package installer with CUDA version 11.6.

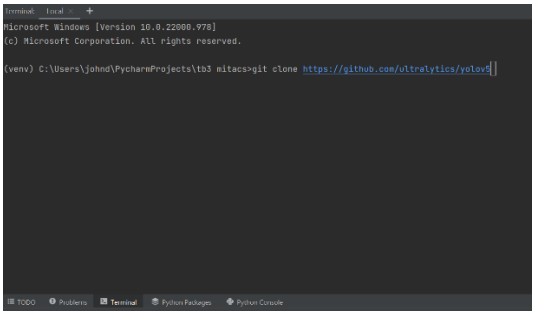
PyTorch Build	Stable (1.12.1)	Preview (Nightly)	LTS (1.8.2)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 10.2	CUDA 11.3	CUDA 11.6	~~ROCm 5.1.1~~	CPU
Run this Command:	pip3 install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cu116			

Once the command above has been displayed by PyTorch, copy, and paste the “run this command” part into the terminal in your newly created PyCharm virtual environment as seen below. The terminal line will appear as a clickable tab at the bottom left of the PyCharm page. If you are not using an IDE, copy and run the “Run this Command” in the terminal where your virtual environment is setup.

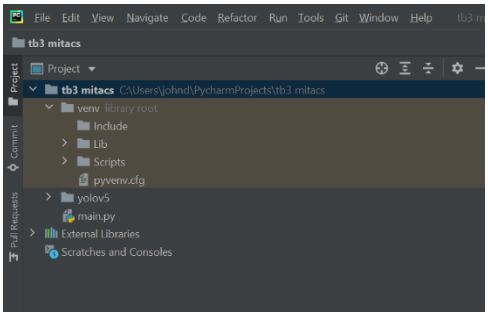


Yolov5 installation

To install Yolov5 please click on the Terminal Tab. Once the terminal tab is open, we just need to copy and paste “git clone https://github.com/ultralytics/yolov5” into the (venv) terminal line and then press enter as can be seen below:



Once installed the Yolov5 package folder will appear in the upper left-hand corner of the project file. If you are not using an IDE, the files will be installed in the project directory of your virtual environment.



Now go to the terminal and enter the following commands:

1. “cd yolov5”
2. “pip install -qr requirements.txt”

Once the above commands are completed, we can run our model.

How To run

To run our model, we just need to provide the directory path to the weight file, and the directory path to the photos or video folder we wish to run inference/detection on. The weights file (“weights.pt”) provided does not have to be placed in the PyCharm folder or where the folder of the detect.py script is located but we do recommend it as it makes the script easier to read.

Once the weights and photo directory path has been determined, in the PyCharm terminal, run the following command:

detect.py --weights path-to-weight-file/yolo5best07312022.pt --img 416 --conf 0.5 --source 0 --device 0

or:

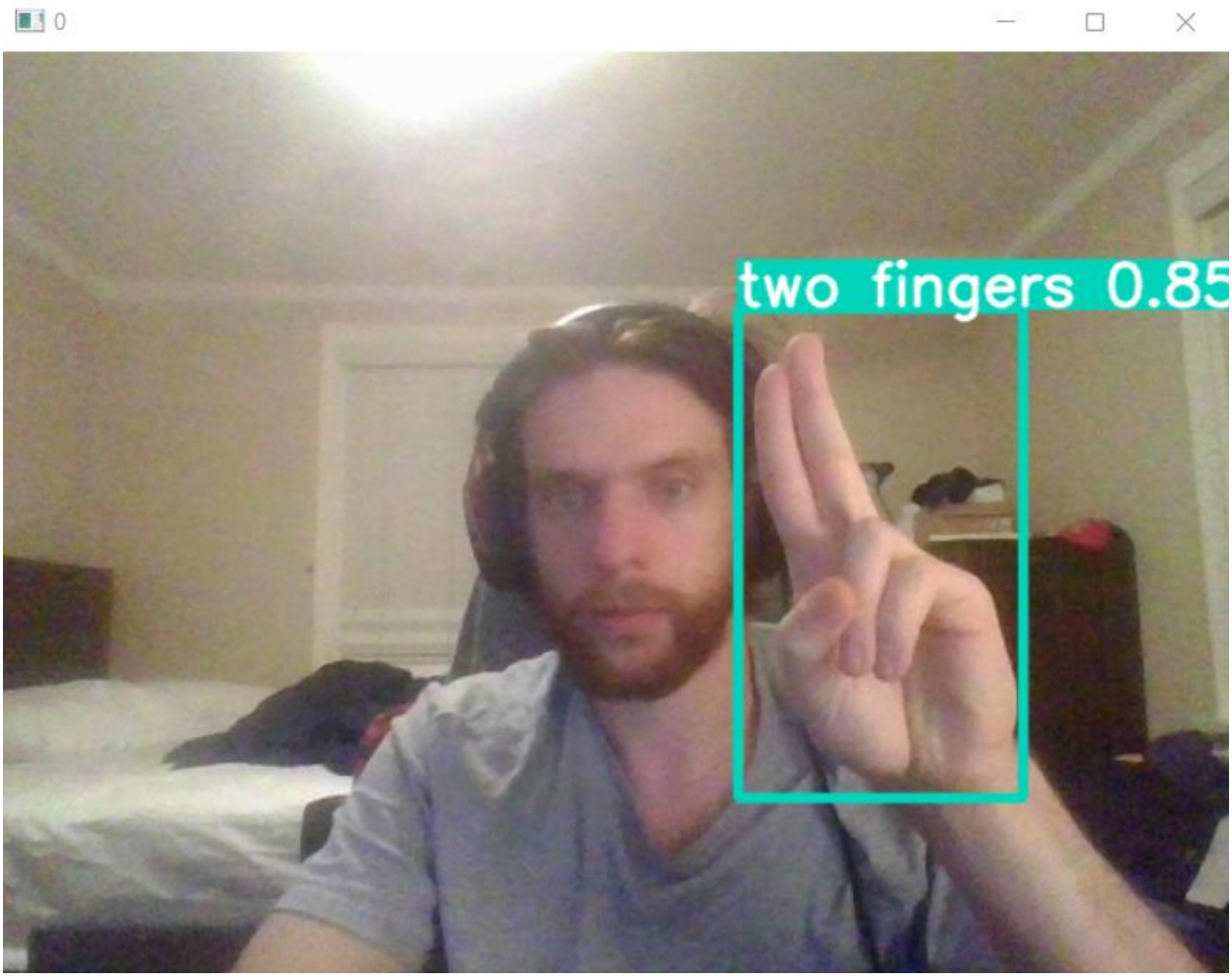
detect.py --weights path-to-weight-file/640yolo5best.pt--img 640 --conf 0.5 --source 0 --device 0

The first command line is for the smaller quicker model (image size 416) while the second command line is for a larger slightly more accurate model (image size 640).

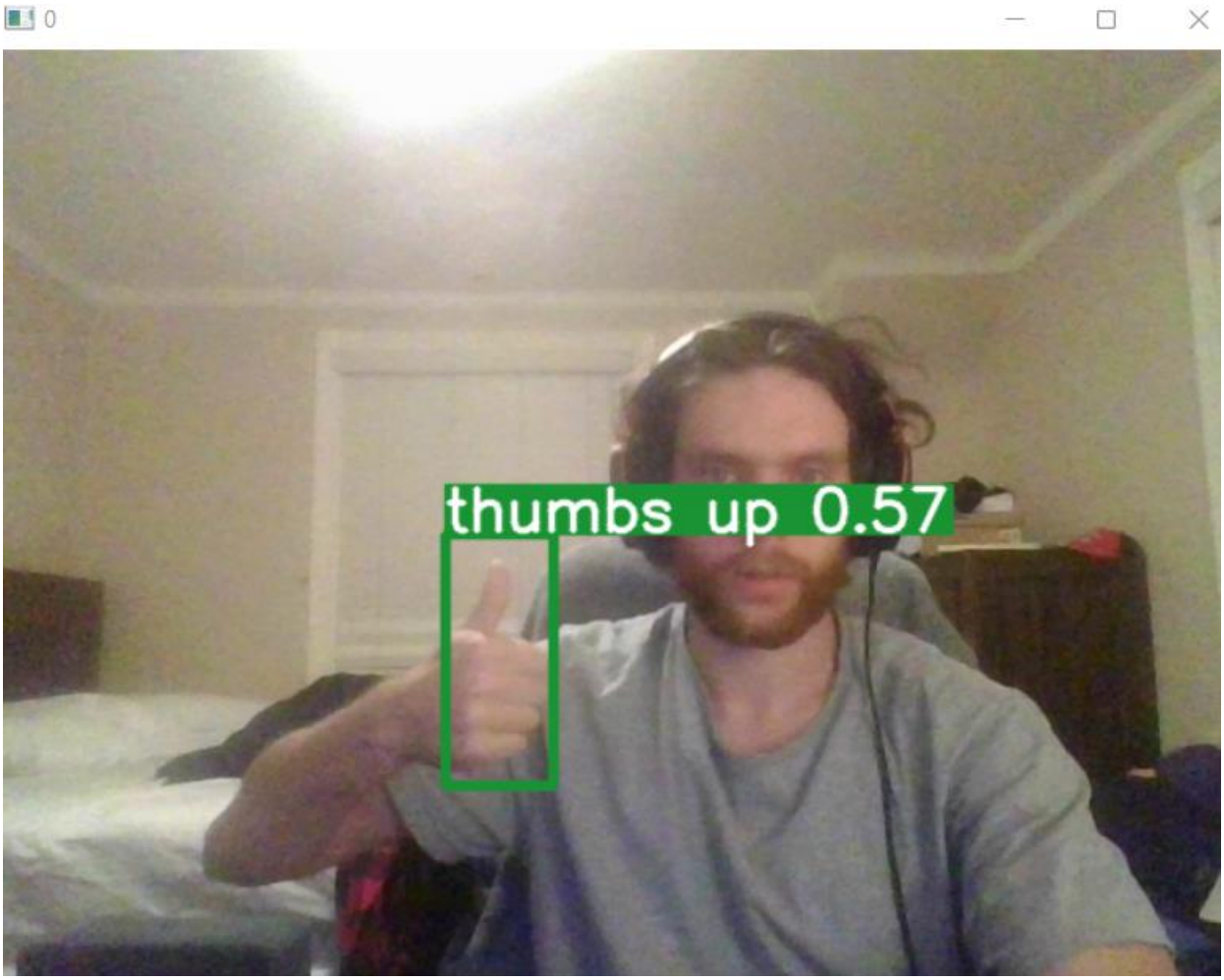
The parameters will be discussed in more depth in the next section, but we need at least four parameters (denotated by "--") to run our model. The four main parameters are briefly discussed below:

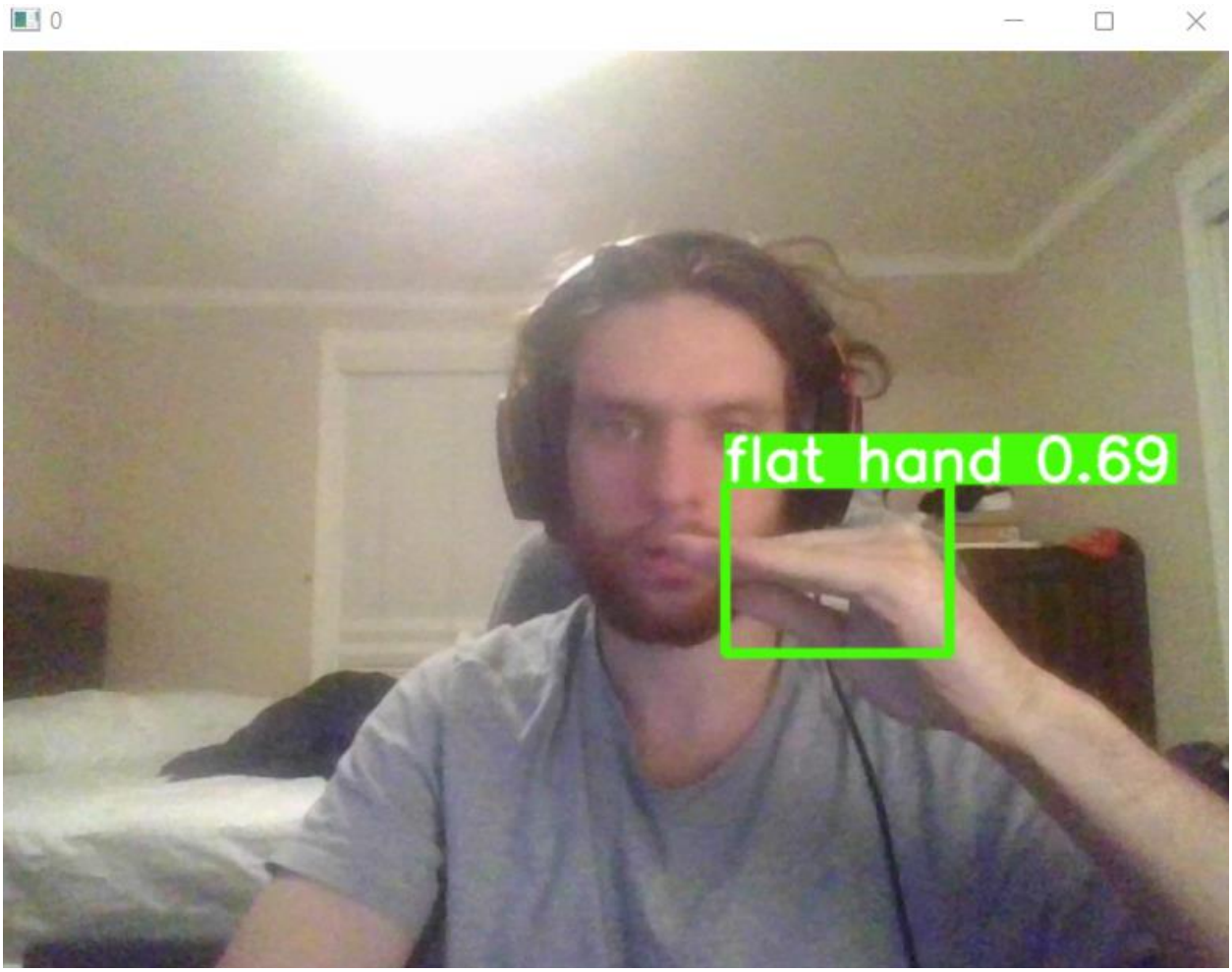
- --weights: This parameter is used to provide the path to the weights file ("weights.pt").
- --img: This parameter is used to change the image size (pixel size). We recommend leaving this parameter at 1088.
- --conf: This parameter is used to tell the Yolov5 model how confident the model must be to predict a defect or defects for a given image. This parameter ranges from 0.1-1.0 (10% to 100% confident).
- --source: This parameter is used to provide the path to the image path file.
- --device: This is to run the model on your webcam

A new screen will pop up after the above terminal commands are run. The new screen will show whatever your webcam is directed towards. An example of the model being run can be seen below. In addition, the 10 hand gestures can be seen below:

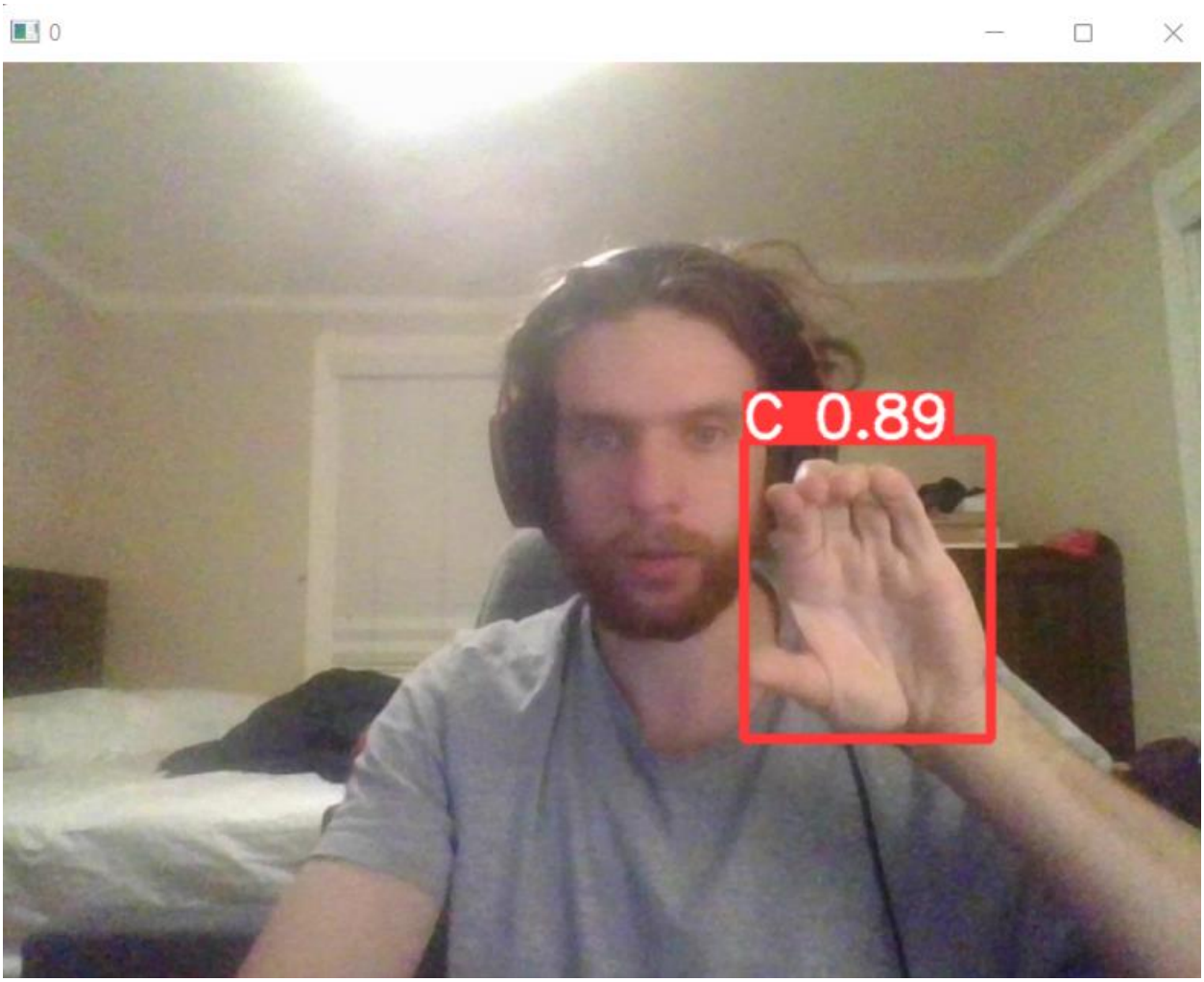


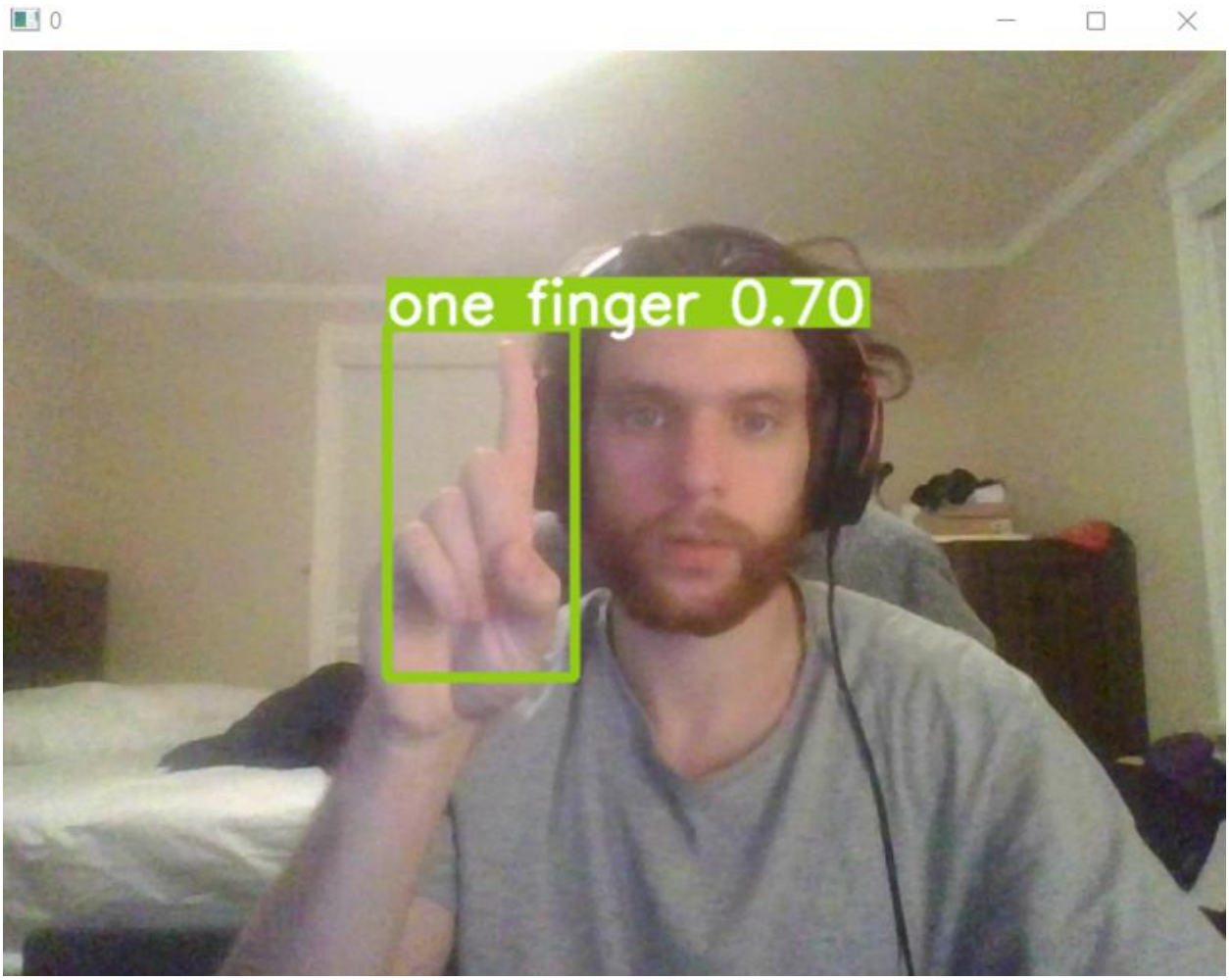

0
Five Fingers 0.83

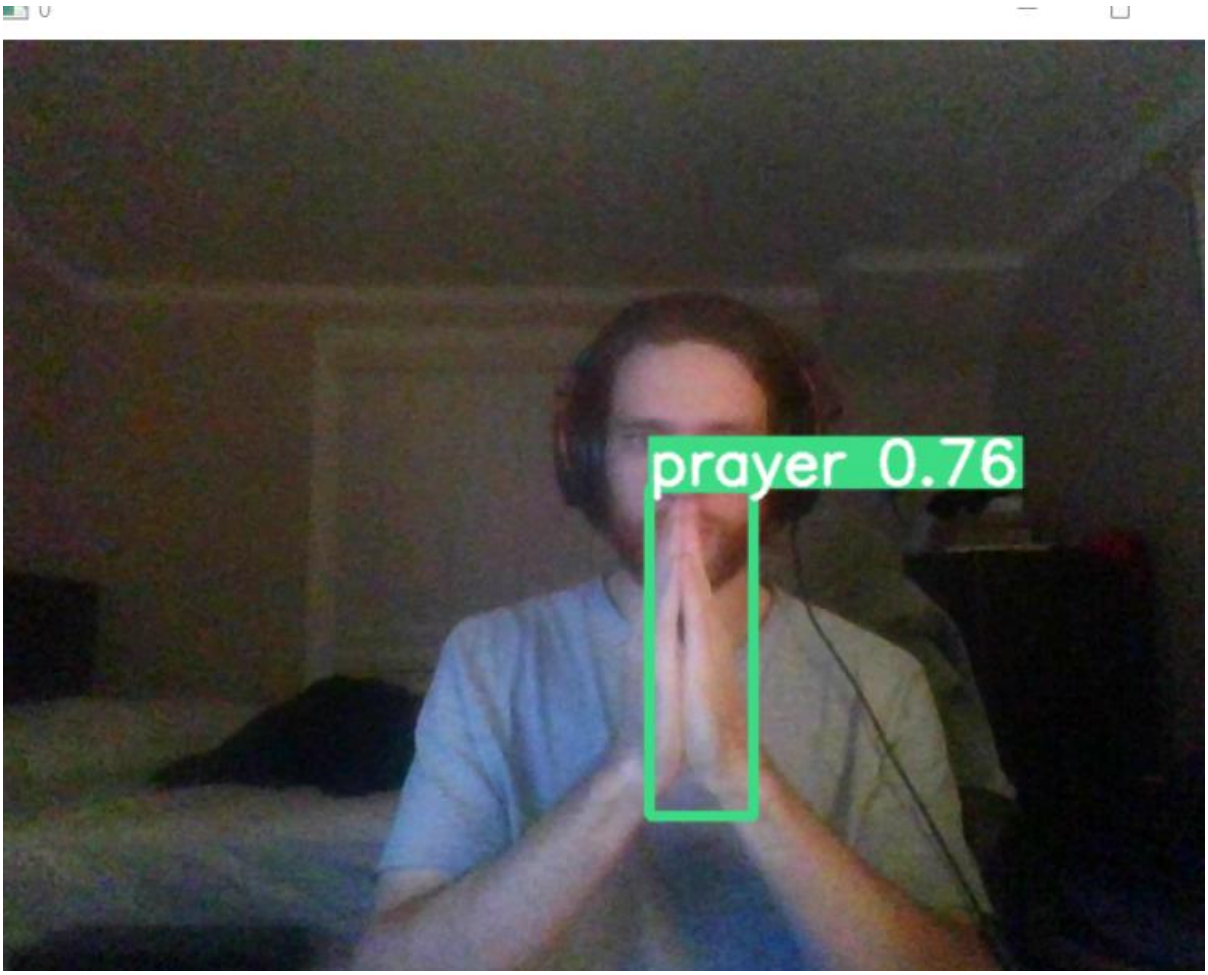

0
thumbs up 0.57

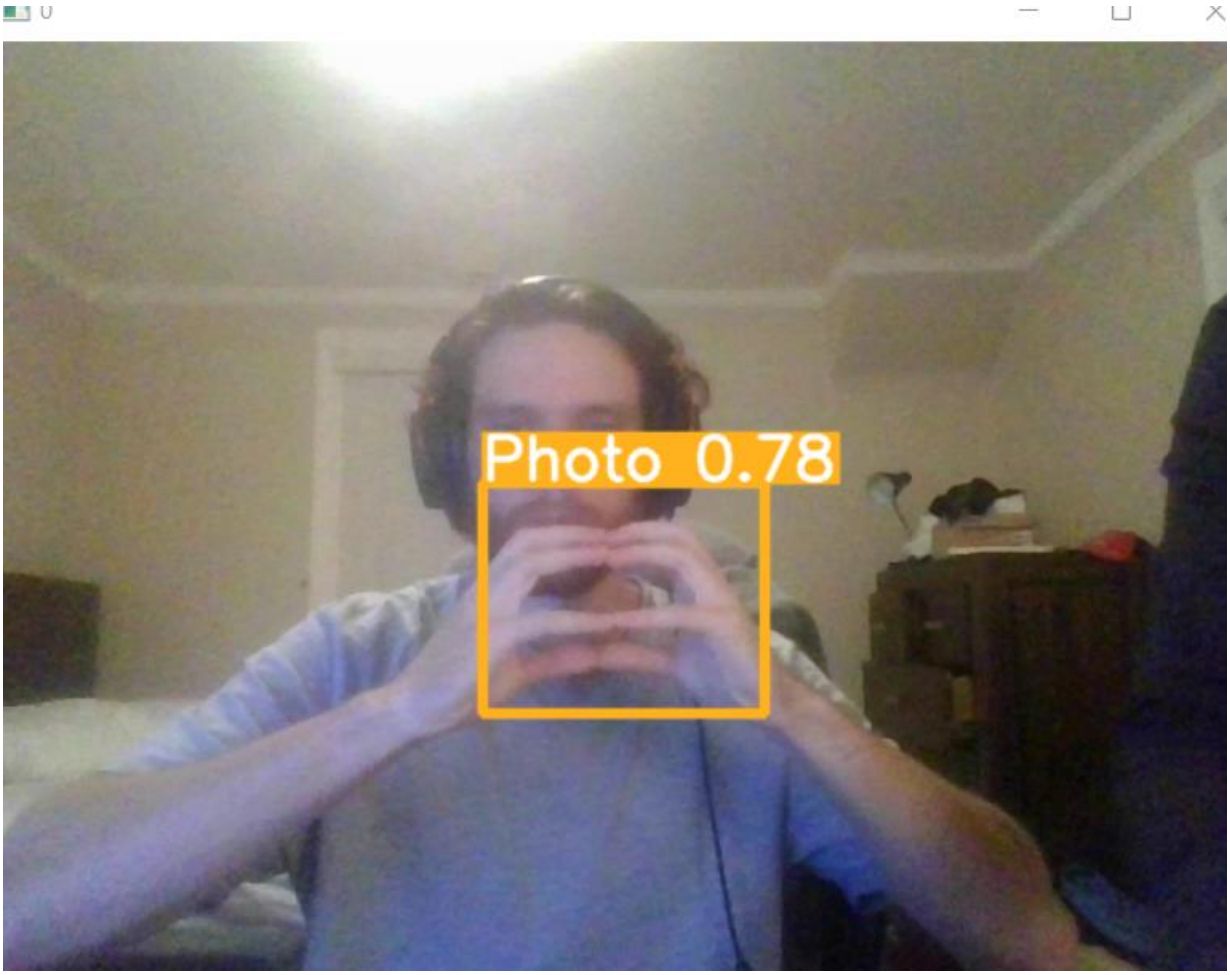

0
flat_hand 0.69

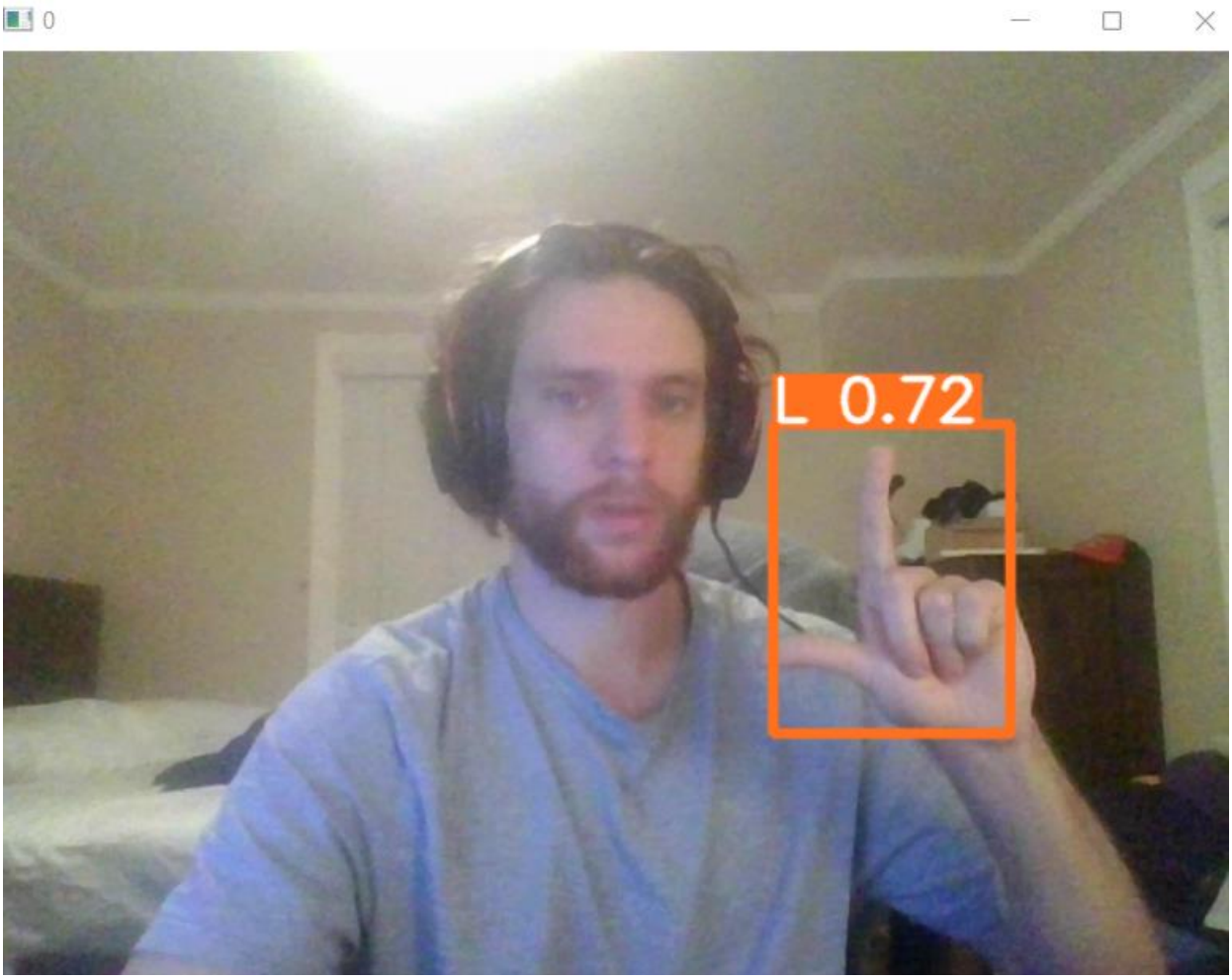

0
fist 0.92


0
C 0.89


0
one finger 0.70


prayer 0.76


0
Photo 0.78



Customization and Parameters

The script "yolov5/detect.py" creates a new directory "yolov5/runs/detect/exp{#}" for each run. "exp" may be followed by a number indicating the number of runs performed in total (e.g. exp2, exp10). However, by typing in "–exist-ok" as a parameter into our script, this will overwrite the existing "exp" folder. Please refer to the following items to customize how results are stored:

- “--weights” argument is used to set the path to the pre-trained weights file.
    - This argument will only be modified if a new model is trained, or the weights path is moved to a different location
- “--img” argument is used to set the input image size (1088*1088 pixels in our case). This argument can be modified to different image sizes, but we recommend using 1088.
- “--conf” argument is used to define the minimum confidence score for detection (0.4 confidence threshold in our case). This argument can be changed to any round decimal number from 0.1 to 1.0. Changing this parameter to a lower number will cause more detections with less accuracy and increasing this number will create less detections with more accuracy.

- “--save-txt” argument is used to save the normalized bounding box coordinates. This can be helpful to understand where the model is detecting certain defects in each photo or video. This parameter is not needed to run our model.
- “--save-conf” argument is used to save the confidence score for each detection in a txt file. This can be helpful if you want to modify the --conf argument, but this parameter is not needed to run our model.
- “--exist-ok” argument is used to overwrite the output folder from a previous run.
- “--nosave” argument is used to avoid saving images/videos outputs.
- “--device” argument is used to specify which device (GPU or CPU) the model should use. GPU is run by default. To run the model with a CPU device, add “--device cpu” in our script. To run the model using a GPU device, do not include this argument or put “--device 0” in our script.