

Guide to Training A Custom DetectNet Model On the NVIDIA Jetson NANO

John Purpura

August 2021

Table of Contents

- I. Data Collection
- II. Data Annotation
- III. Model Training

Preface:

- Make sure you have built the Jetson-inference library from source:
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md>
- Make sure you complete the mounting swap so your Jetson has enough memory to train:
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-transfer-learning.md#mounting-swap>
- Make sure you have the Python SSD repo downloaded:
<https://github.com/dusty-nv/pytorch-ssd>
- Make sure you complete the setup steps:
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-ssd.md#setup>

1) Data Collection

- Data collection can be in any form, as long as the data set contains images in the form of .jpg
- There are three main types of image data collection
 - i) Screenshotting
 - Taking screenshots of images from a virtual simulation or another virtual source
 - The 2021 XForce team used screenshotting to collect image data from the Gazebo robot sim for ROS implementation
 - ii) Live Camera Capture
 - This involves using a camera connected to the Jetson to collect images from a live camera stream and annotating the images
 - A tutorial of how to collect data using the live camera stream can be found here:

<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-collect-detection.md>

- It is recommended to go through this tutorial even if you are not using the live camera capture tool, as it goes through correct directory formatting for training
- iii) Pre-captured Data Sets from a personal camera or open-source images

- This involves using data downloaded from a personal camera or downloading images from online open source image repositories
- An easy way to download open source image data can be found here:

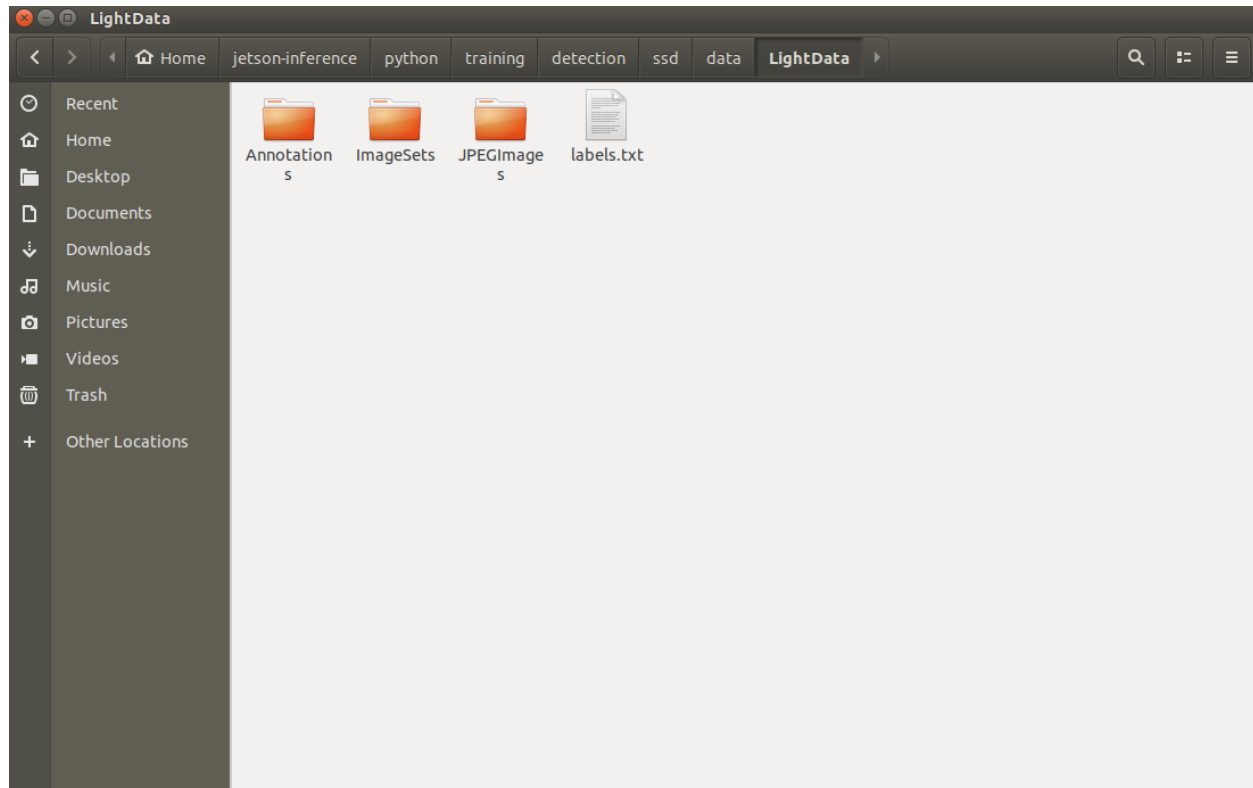
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-ssd.md#downloading-the-data>

- Make sure you have all your image data inside the *jetson-inference/python/training/detection/ssd/data* directory
- Inside *data* you can name the custom data file whatever you want, ****this file will eventually contain four files: *Annotations*, *ImageSets*, *JPEGImages*, *labels.txt*****
- Make sure the folder containing the raw images is called *JPEGImages*

2) Data Annotation

- Once your raw image data set is collected, go to cvat.org to begin annotating your images with custom bounding boxes and labels
- This tutorial gives a great step by step guide on how to use CVAT for object detection annotation purposes: <https://youtu.be/OMgQ2JzOAWA>
****watch up until the 8:06 mark****
- After you have exported your data in PASCAL VOC format from CVAT, move the *Annotations* file and the *ImageSets* file into the *jetson-inference/python/training/detection/ssd/data/<customName>* directory where your *JPEGIMAGES* file already is containing the raw image data
- Create a *labels.txt* file or use the one created in the CVAT export and add it to this directory

- Your new directory should look like this:



3) Model Training

- To train a model using your custom annotated data, you must use the *train_ssd.py* script which is contained in the jetson inference ssd repo downloaded earlier
- To run it from your terminal:
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-ssd.md#training-the-ssd-mobilenet-model>
 - i) The data folder name used in your terminal command line should be the custom name you created earlier. For example, the name that would be used from the figure above would be 'LightData' which contains the four data files
 - ii) The model file name can be whatever you want the model to be called, and this will be saved to the models directory
 - iii) You can specify the batch size (recommended 2), number of workers (recommended 1), and number of epochs (recommended 30)
- Once your model has been trained, you must export to .onnx to use it in ADetect_xy.py or in ROS:
<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-ssd.md#converting-the-model-to-onnx>
- Your model is now ready to be used

