ECP5 Object Counting Quick Start Guide

Application Note

FPGA-AN-02006-1.1

June 2019

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CKPT	Checkpoint
FPGA	Field-Programmable Gate Array

1. Introduction

This document provides a quick guide on how to train a machine and create a frozen file for the Lattice Machine Learning development using the Lattice's Embedded Vision Development Kit. It assumes that the reader is familiar with the basic Lattice FPGA design flow and mainly focuses on the Machine Learning part of the overall development process. For detailed instructions of the design flow described in this document, refer to the Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

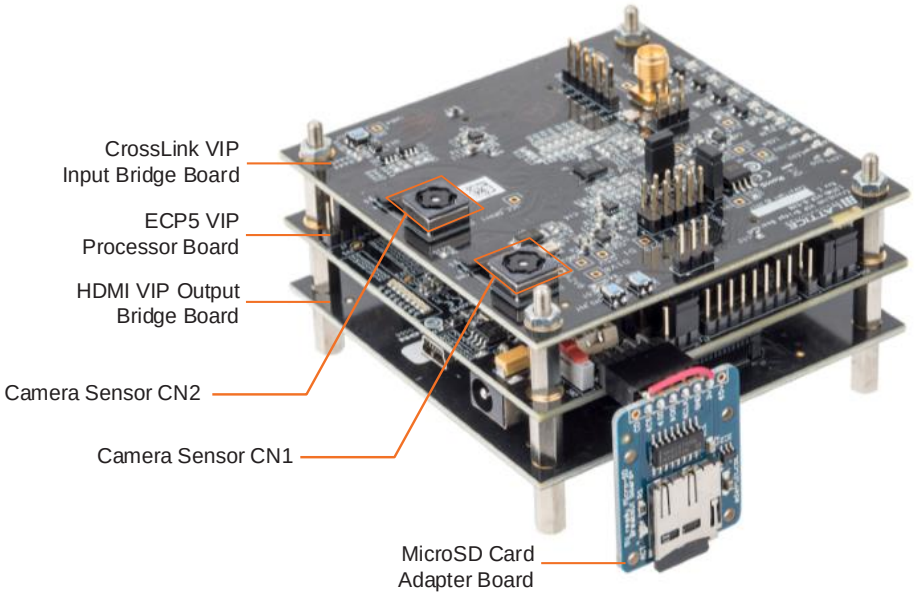

Figure 1.1. Lattice EVDK with MicroSD Card Adapter Board

1.1. Design Process Overview

The design process involves the following steps:

- Setting up the basic environment
- Preparing the dataset
- Training the Machine
- Creating the frozen file (*.pb)
- Creating the binary file with Lattice sensAI™ 2.0 program
- Programming the binary and bitstream files to VIP board and SD card

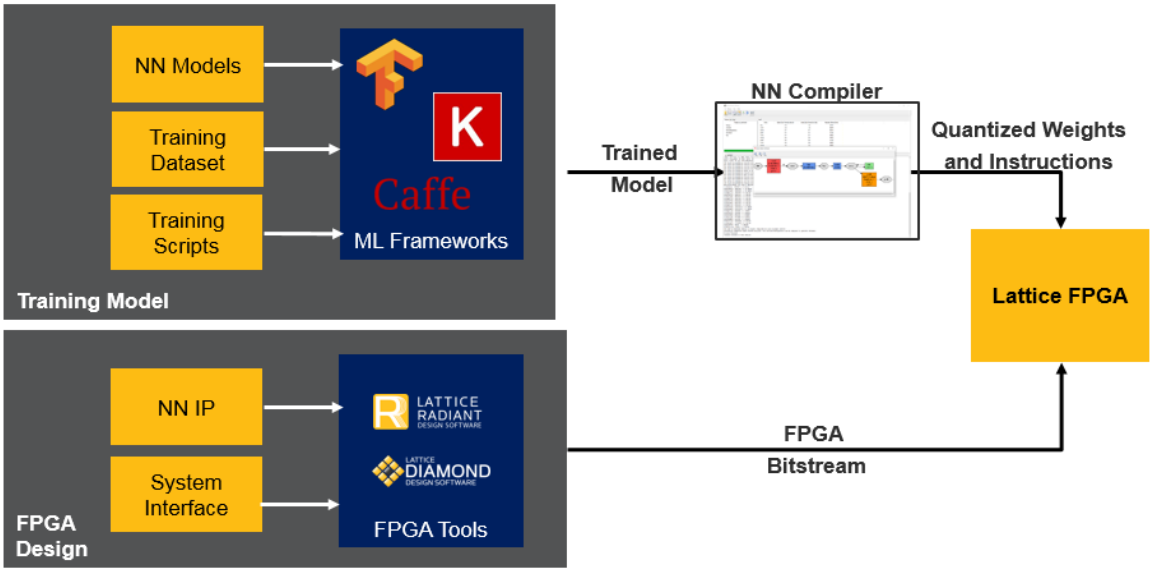

Figure 1.2. Lattice Machine Learning Design Flow

2. Machine Training and Creating Frozen file

2.1. Verifying TensorFlow and Tool Environment

Check if Tensorflow and your tool environment is installed correctly. For the detailed procedure in creating the basic environment on PC, refer to the Setting up the Basic Environment section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

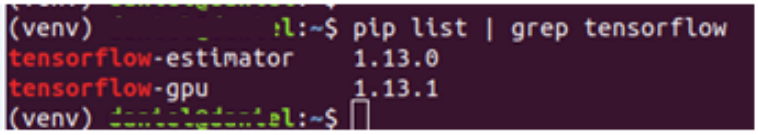

Figure 2.1. Tensorflow Installation Check

2.2. Preparing the Dataset

Prepare the image and label data (KITTI format). The image size for this design is 224 x 224 pixels.

For the detailed procedure in preparing the dataset, refer to the Preparing the Dataset section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

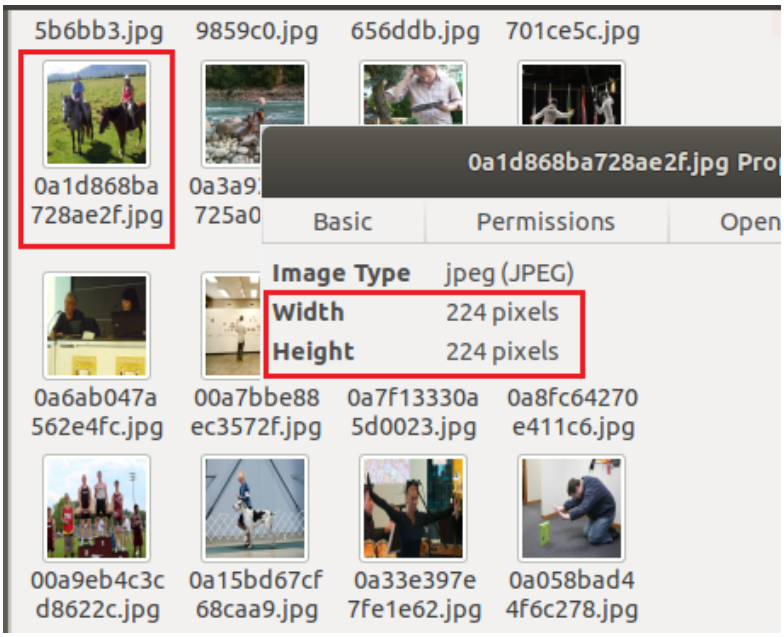

Figure 2.2. Dataset Image Size Check

2.3. Training the Machine

For the detailed procedure in machine training, refer to the Training the Machine section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

To train the machine:

1. Check the training dataset path in the training script file *train.sh*.

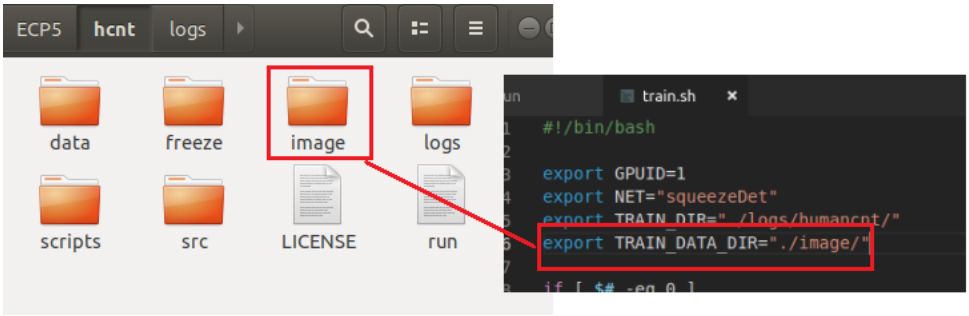

Figure 2.3. Dataset Folder Path Check

2. Check the subdirectory and training dataset.

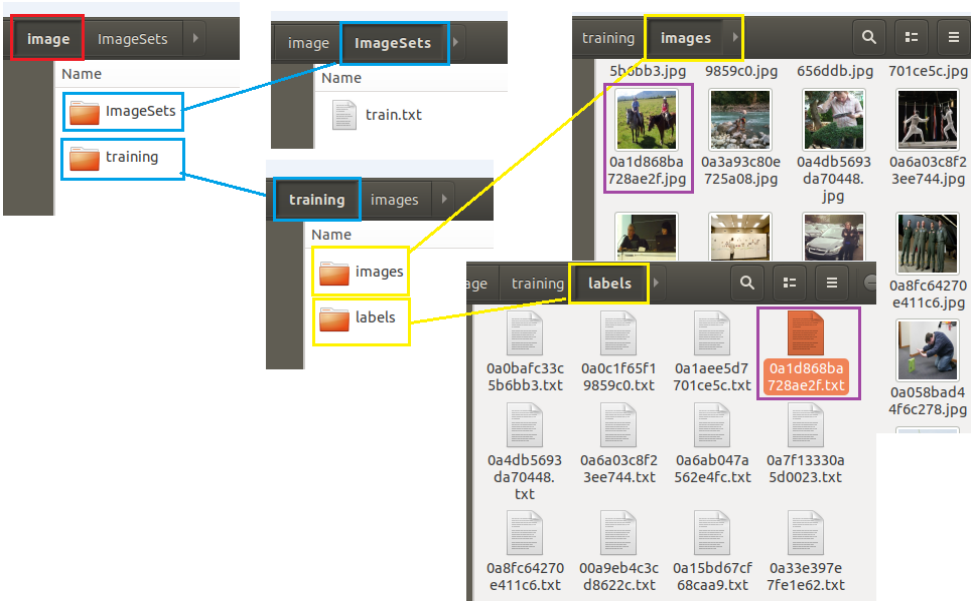

Figure 2.4. Dataset List, Image, and Label Data Path

3. Check the arguments of the *run* command.

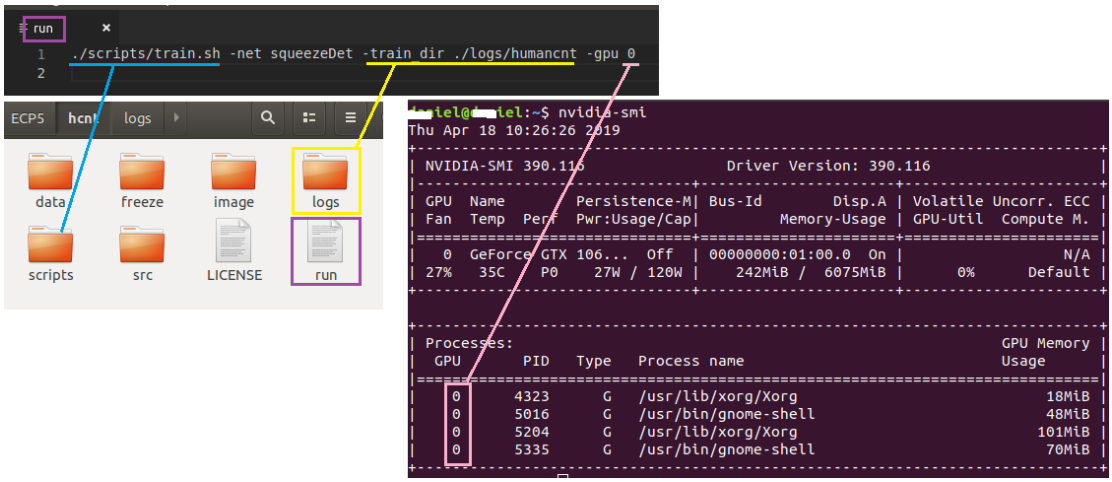

Figure 2.5. Run Script File

4. Run machine training. In the command prompt, execute *./run* command.

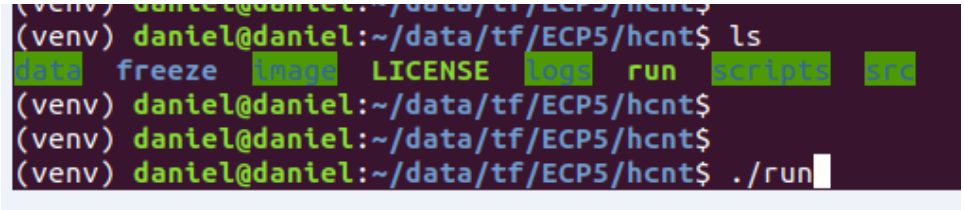

Figure 2.6. Execute the script

5. Run TensorBoard and execute *tensorboard –logdir='./logs'* command.
6. Open a new command prompt and web browser (such as Chrome).

   Note: Check *http::<>:<>* port name on your terminal. It is different.

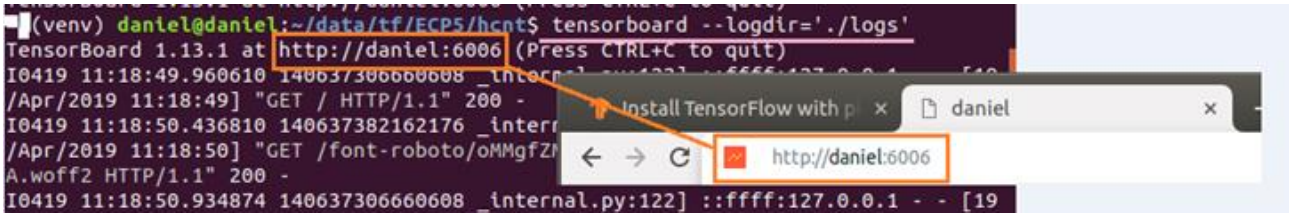

Figure 2.7. Execute TensorBoard

7. Check training status (White Box by label dataset, Green Box by machine inference) in *TensorBoard IMAGES* menu, check if the Green box detects persons.

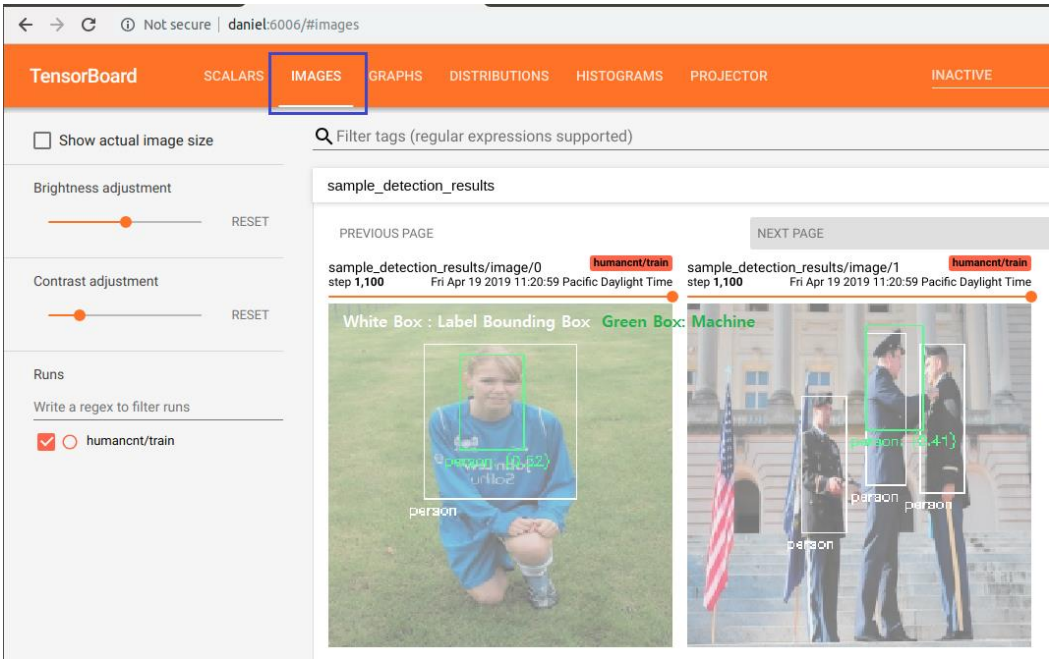

Figure 2.8. TensorBoard Interface

8. Check if the checkpoint and meta data are generated at **/logs/humancnt/train/** log directory.

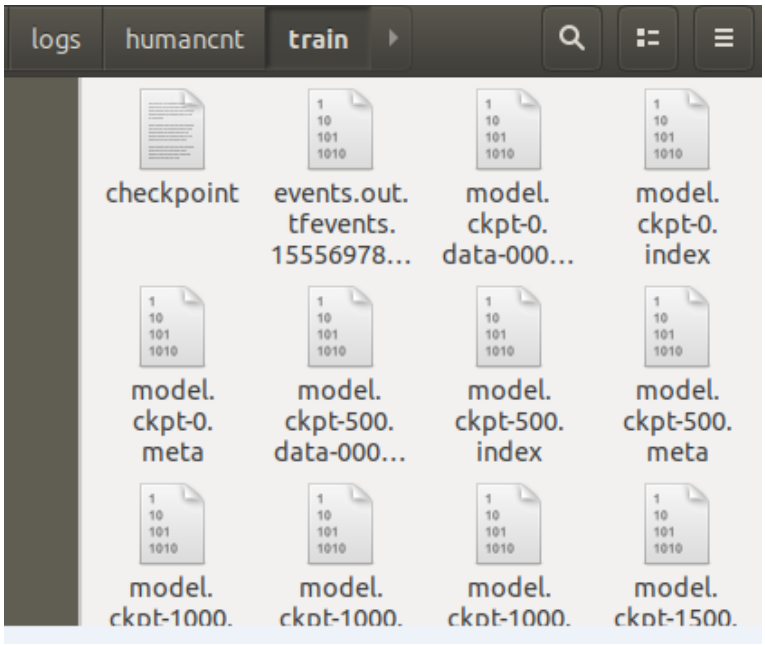

Figure 2.9. Checkpoint Data Files at Log Folder

2.4. Generating Frozen (*.pb) File

To generate Fronzen file(*.pb):

1. Find the lastest checkpoint data at log path folder.

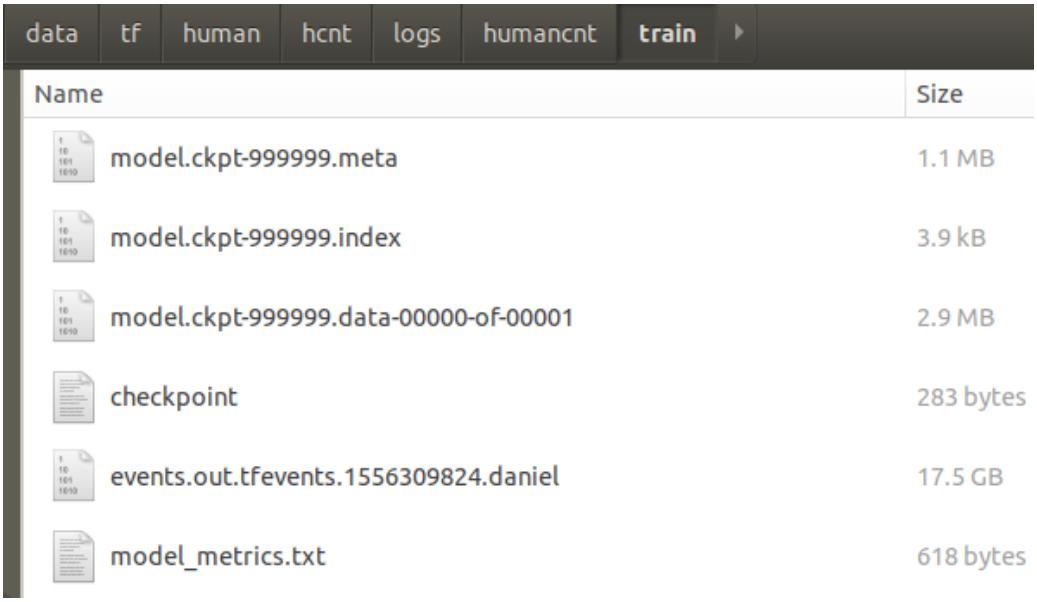

Figure 2.10. Latest Checkpoint Data Files

2. Run *genpb.py*.
3. Execute *python src/genpb.py –ckpt_dir='./logs/humancnt/train/'* command. For the detailed procedure in creating .pbtxt file, refer to the Creating Frozen File section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

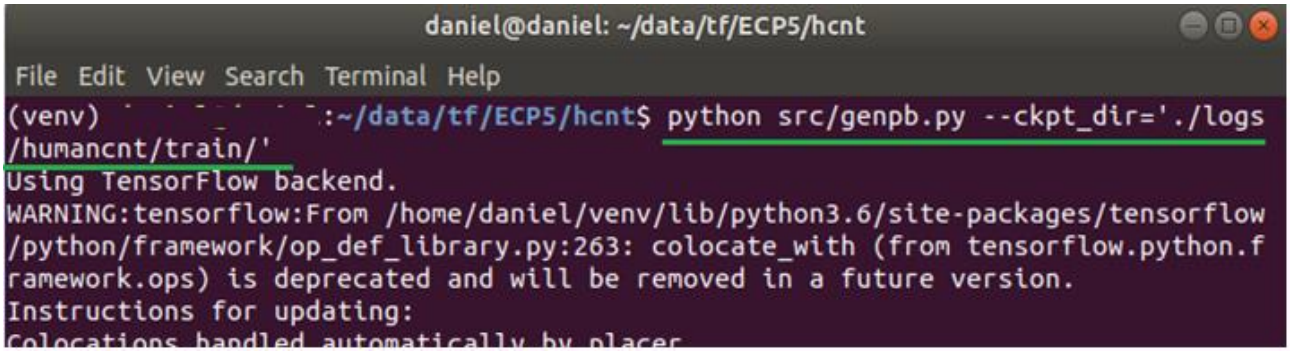

Figure 2.11. Create *.pbtxt File

4. Check if *graph.pbtxt* is generated.

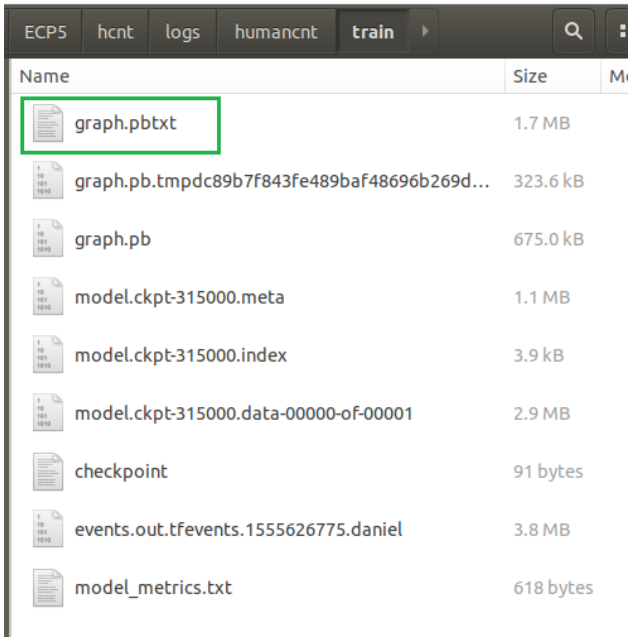

Figure 2.12. Check *.pbtxt File

5. Copy the files of the *train* folder to the **/freeze/model/** directory**.** Rename graph.pbtxt to *model.pbtxt*.

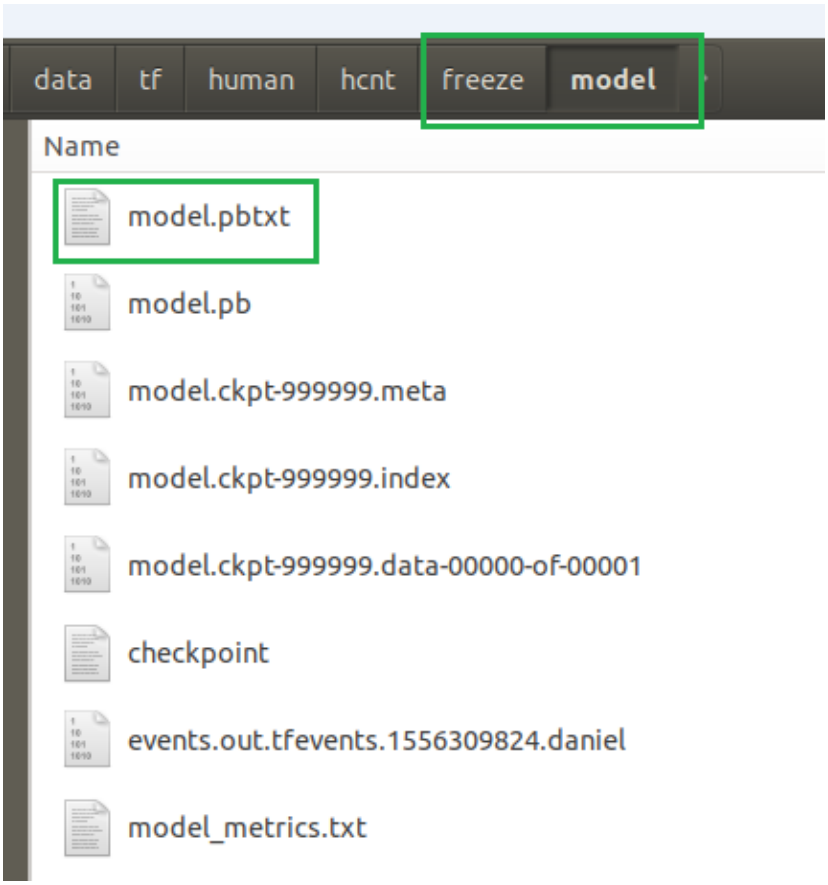

Figure 2.13. Rename and Copy Checkpoint Files

6. Go to the **/freeze/** path at the command prompt and run *trainckpt2interencepb.py***.**
7. Execute *python trainckpt2interencepb.py* command. For the detailed procedure in creating .pbtxt file, refer to the Creating Frozen File section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

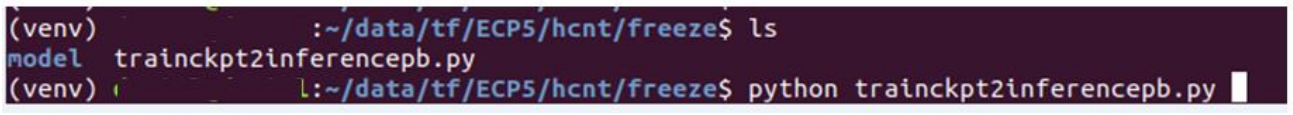

Figure 2.14. Running trainckpt2inferencepb.py

8. Verifiy and find the generated frozen file.

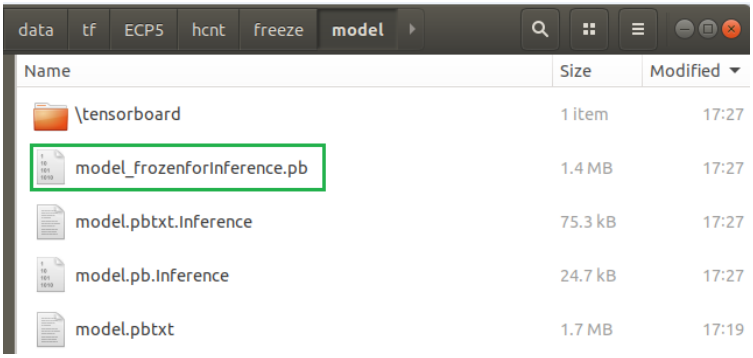

Figure 2.15. Check Frozen File

2.5. Generating the Binary File

For the detailed procedure in creating the binary file, refer to the Creating Binary File with SensAI section in Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

2.6. Programming the Bitstream and Binary files to VIP Board and SD Card

For the detailed procedure in flashing the bistream file to the VIP board and flashing the binary file to the SD card, refer to Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.1, April 2019

Section	Change Summary
All	Corrected link to Object Counting Using CNN Accelerator IP (FPGA-RD-02058).

Revision 1.0, May 2019

Section	Change Summary
All	Initial release.

www.latticesemi.com