Triton Smart Getting Started Guide

August 4, 2026
Ver. 1.4

Introduction

LUCID's Triton Smart (TRS123S) is able to perform inference based on the AI model loaded on to it. This tutorial covers the **classification** model, which accepts an image as an input, and guesses what is in the image based on its training.

This document shows you how to do the following:

- Clone a Google Cloud Platform (GCP) VM image containing the tools needed to create an AI model.
- Train and convert the AI model using tools on this VM.
- Upload the AI model on to the Triton Smart.
- Use the AI model on the Triton Smart.

Prerequisites

- Triton Smart camera (TRS123S)
- FW version 1.2.0.0
- ArenaView MP version 1.0.80.56 or higher.

Additionally, users should have a basic understanding of the Linux command line.

Basic workflow

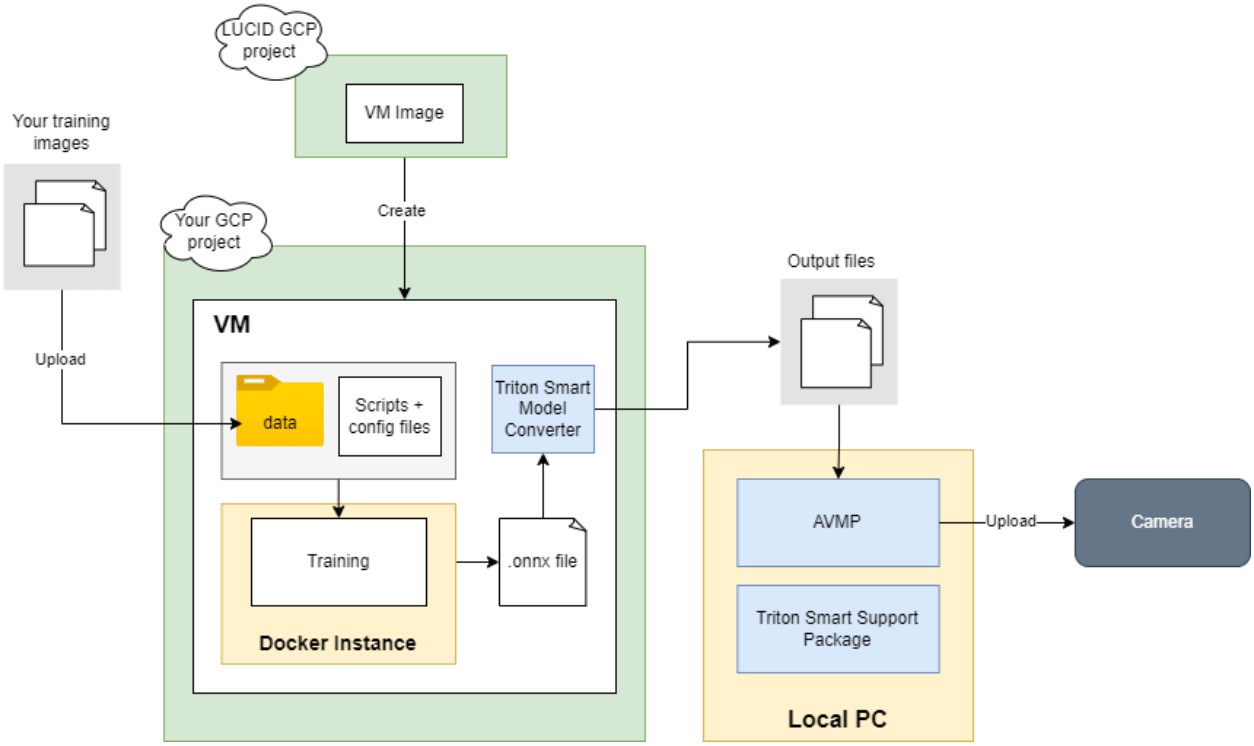


Figure 1: Basic workflow for the classification model.

This tutorial explains how to use a **classification** model to identify the content of an image based on the categories it was trained to recognize.

The first section instructs you on how to run the scripts and training data (images of cats and dogs) that are included in the VM to create a classification model. The second section tells you how to create a classification model from your own training data.

The model training workflow recommended by LUCID involves training a model on Google Cloud Platform (GCP), downloading the model to your local PC, and then using Triton Smart with this local PC.

Download the Triton Smart Support Package from the Triton Smart product page for compiled Windows applications for displaying the inference results.

Getting started with the Triton Smart

LUCID provides access to a VM image via GCP. Once you clone this image, you will have all the tools you need to train your own AI models for the Triton Smart.

Creating a Google Cloud Platform account & creating a VM on GCP

To use the VM, you will use Google Cloud Platform (GCP).

Create a GCP account. To do this, follow the instructions on the Google Cloud Platform page. Create a VM from the custom image made available from by LUCID:

To do this, open the cloud shell by clicking the button in the top right corner of the GCP console.

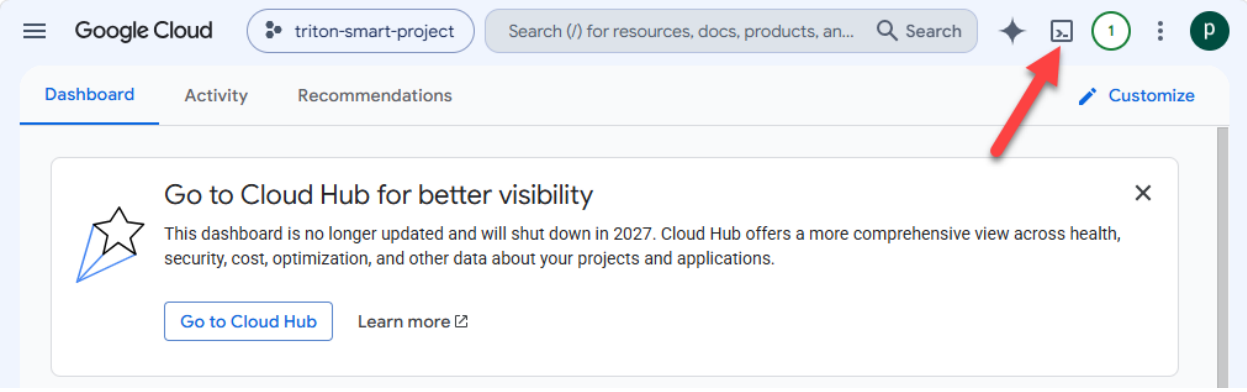


Figure 2: The GCP console.

The Cloud Shell appears in the browser.

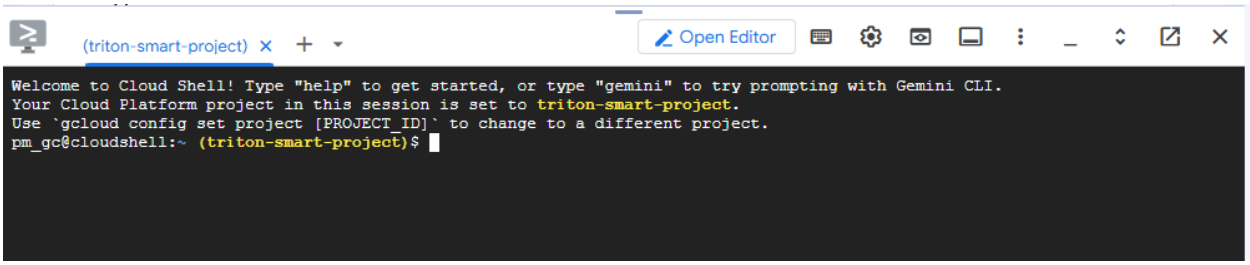


Figure 3. The Google Cloud Shell interface.

Create an VM instance based on the image available from LUCID. Choose a new instance name. The project is the current GCP project, which is displayed in the top ribbon.

*gcloud compute instances create [NEW_INSTANCE_NAME] *

*--project=[DESTINATION_GCP_PROJECT] *

*--zone=us-west1-a *

*--machine-type=n2-standard-16 *

*--image=lucidvisionlabs-smart-classification-v1 *

--image-project=lucidvisionlabs-share

You can modify the VM instance configuration when creating the VM instance from the custom image, but we recommend that you use the following configuration (the command above uses this configuration)

- n2-standard-16 (16 vCPUs, 64 GB Memory)
- 250GB Storage

If the zone (uswest-1-a) does not work, choose another zone. You can see the list of all zones by running the following command in the cloud shell:

gcloud compute zones list

Using SSH

To use SSH to enter the VM:

1. From the GCP console, click the hamburger icon on the top left corner.
2. Open Compute Engine > VM Instances.
3. Select the SSH dropdown for the VM instance that you just created, and click "Open in browser window".
   After some authorization requests, the SSH-in-browser window opens.

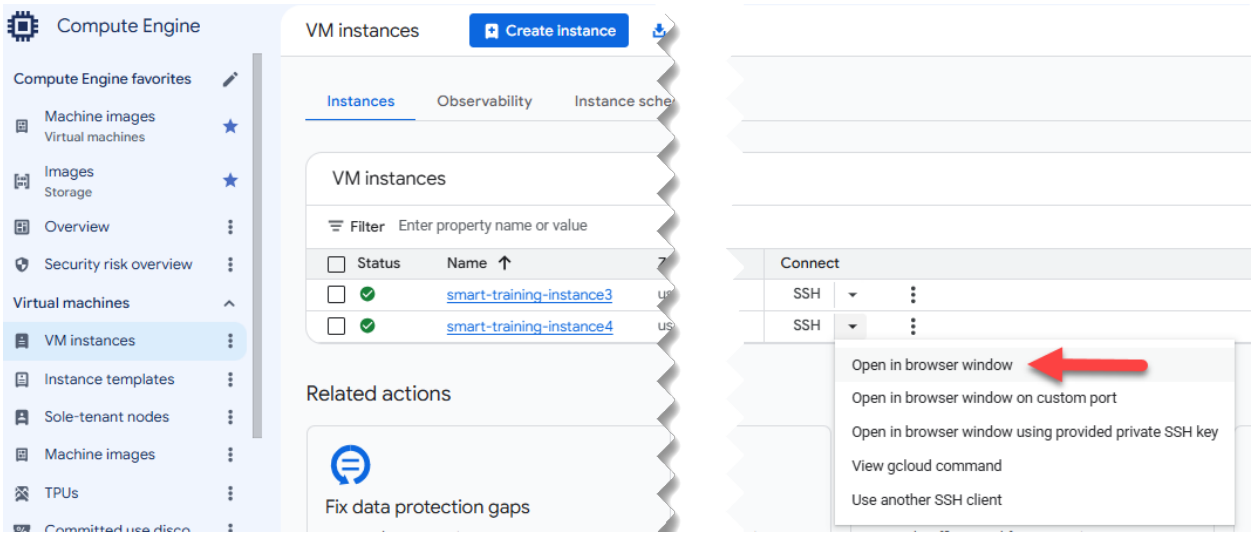


Figure 4: Opening the SSH window.

By default, the SSH-in-browser console in GCP tries to login to the VM using the Google account name as the username. However, the training sample is located under /home/apps_gc', so you need to change the username.

To do this, click the gear icon on the top right of the SSH screen, and change the Linux Username to "apps_gc".

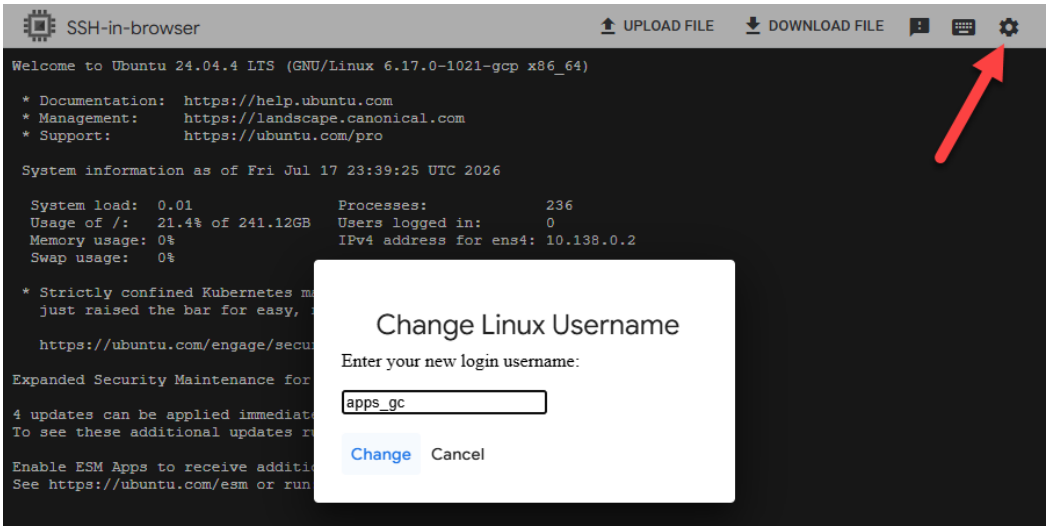


Figure 5: Changing the Linux username in the SSH-in-browser.

Running the sample code to create the model

The VM comes with scripts for creating a model. To build the model, follow the steps below. (No file modification is required for this section, which uses the training images included on the VM.)

1. To run the docker instance, ensure that you are in the aitrios-rpi-training-samples-r3 folder, and run:
   source docker-run.sh
2. Next, run:
   imx500-zoo mobilenet_v2_cat_dog.ini
3. This process creates two models:
   a) mobilenet_v2_cat_dog.onnx
   b) mobilenet_v2_cat_dog_quantized.onnx
4. You will use the quantized model in the steps that follow.

   Note: You can safely disregard the message "ERROR:Model Compression Toolkit:Found duplicate qco types."

Convert the model

This step is required to make the model usable on the Triton Smart.

1. In the TritonSmartModelConverter folder, run
   ./TritonSmartModelConverter
2. At the prompt, enter
   /conv
3. Enter the path the input model. This will be:
   /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/mobilenet_v2_cat_dog_quantized.onnx
4. For the Input Format, enter RGB
5. Enter the path to the input directory. Example:
   /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/
6. Once you get the message “Output saved to /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat/_dog/”, you can exit using:
   /exit
   Once the steps above are successfully completed, you will have a network.lpk file and a network_info.txt file.

```
apps_gc@smart-training-instance3:~/TritonSmartModelConverter$ dir
README.txt  TritonSmartModelConverter  TritonSmartModelConverter_v140.exe  tools
apps_gc@smart-training-instance3:~/TritonSmartModelConverter$ ./TritonSmartModelConverter
  Welcome to Triton Smart Model Converter.
  Please enter a command to begin.
  For help, type '/help'
  To exit, type '/exit'

/conv
Path to input model: /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/mobilenet_v2_cat_dog_quantized.onnx
Input format [RGB/BGR/Y/BayerRGB]: RGB
Path to output directory: /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/
Running conversion. It may take a few minutes.
    Processing... (1/3)
    Processing... (2/3)
    Processing... (3/3)
Output saved to /home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/
/exit
Closing application
apps_gc@smart-training-instance3:~/TritonSmartModelConverter$
```

Figure 6: The converter tool in action. Once it says the output is saved, type /exit to return to the prompt.

Download the files from the VM

Use the Downloads file feature of the SSH-in-browser to download the two files. You need to specify the full absolute path. If you ran the example above, you would download the following:

- */home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/network.lpk*

- */home/apps_gc/aitrios-rpi-training-samples-r3/samples/model/mobilenet_v2_cat_dog/network_info.txt*

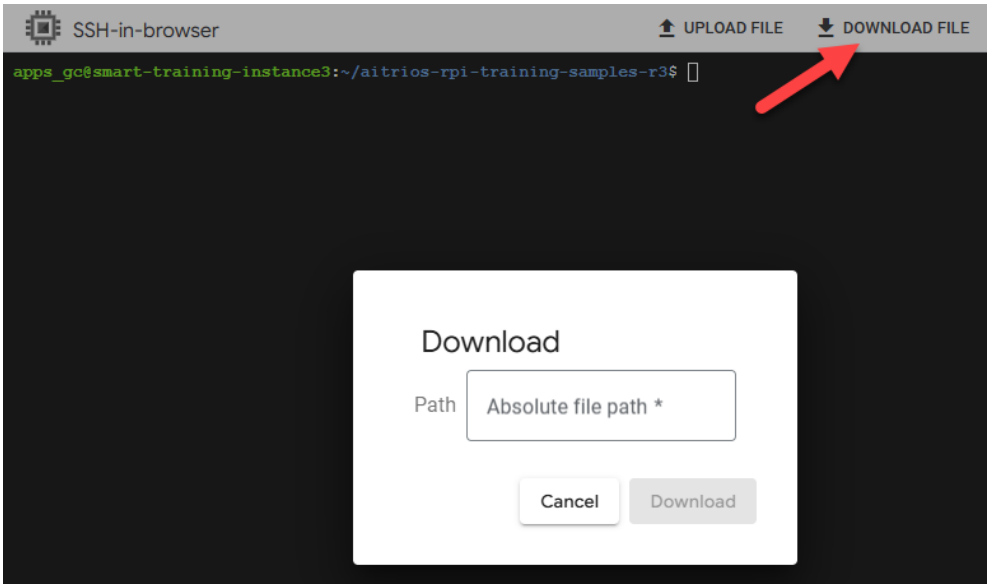
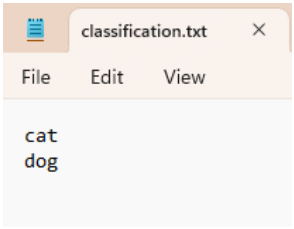


Figure 7: Downloading the files from the VM.

Note: If you have problems downloading a file, start a new shell and try again.

Load the model on to the Triton Smart

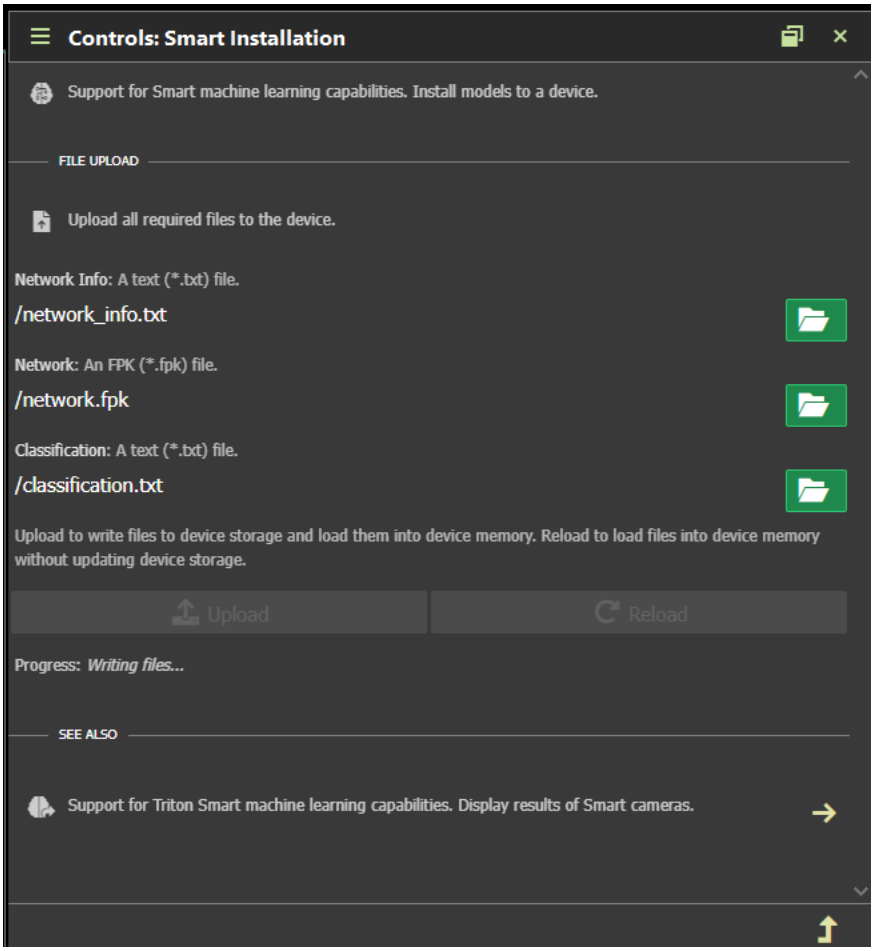
1. Connect the camera to your computer.
2. Open ArenaView MP.
3. Turn on the camera.
4. On your computer, transfer the two files from the VM to your local computer.
5. [If you have ArenaView MP v1.0.80.56] Rename network.lpk to network.fpk. Later versions of AVMP don't require not require a rename.
6. Create a file called classification.txt consisting of "cat" and "dog". The entries need to be in alphabetical order.



7. In ArenaView MP, open **Controls > Smart Installation**.
8. In the Smart Installation panel, enter the location of the following files:
    a. network.lpk / (or network.fpk depending on the version)
    b. network_info.txt
    c. classification.txt

9. Ensure that the camera is not streaming, and then click the Upload button.

   The message “Writing files” appears as ArenaView MP updates the camera.



Using the model

The best way to see the model in action is to use the Triton Smart Support Package, which is available on the Triton Smart product page.

1. In Windows, unzip the file in a folder.
2. Before starting, ensure that the camera is not streaming (e.g., in ArenaView MP).
3. To use the pre-compiled programs, open the bin folder and double-click Disp_Mobilenet.exe
   The program should detect the Triton Smart, and start streaming from it.
   Point the image at a picture of a dog. The model’s detection score displays on the image.

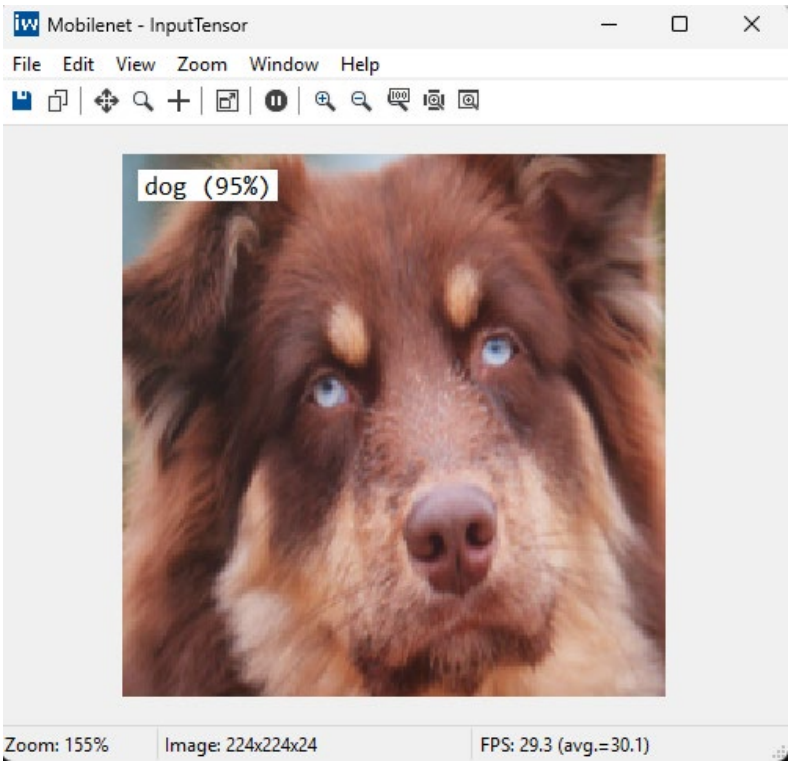


Figure 8: Disp_Mobilenet.exe in action. The image appears grainy because it is an input tensor (a scaled-down image that the Triton Smart sends to its onboard AI).

Note:

- The program works by connecting to the Arena SDK.
- The C++ source code for this application is available from the support package folder.

Training your own classification model

In the steps below, you will train your own classification model with your own images. You will:

- Capture images for training.
- Upload the training images to your cloned VM.
- Update the script files for your model.

Preparing image files

1. Capture images for training. Images taken using the Triton Smart work the best.
2. Put the images into the following folder structure. Randomly divide your images in between train_data and valid_data. 70–85 % of your data should be under train_data, and the remainder under valid_data. Under each of these two folders,

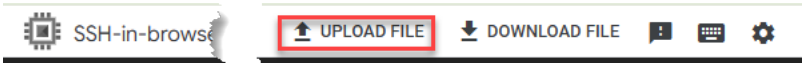
there should be folders with each of your classifications.

Example:
|---*fork_knife_spoon_classification*
|---*train_data*
|---*fork*
|---*knife*
|---*spoon*
|---*valid_data*
|---*fork*
|---*knife*
|---*spoon*

3. Zip the entire file.

Uploading your image files to the VM

1. To upload the zip file of images that you prepared above, click the Upload File button.



2. Move the zip file to the data folder

Example:
mv fork_knife_spoon_classification.zip /home/apps_gc/aitrios-rpi-training-samples-r3/samples/data/

3. Unzip the files.

You should have the following pattern:
aitrios-rpi-training-samples-r3
|--*samples*
|--*data*
|---*fork_knife_spoon_classification* (subfolders follow the pattern above)

Updating the script files

Note that each path described below is under the **aitrios-rpi-training-samples-r3** folder.

src/imx500_zoo/datasets/card_classification.py

Create an appropriately named copy of this file (e.g., fork_knife_spoon_classification.py).

- Comment out the DOWNLOAD_DATASET definition:
DOWNLOAD_DATASET = ...

- Change the data loader class name to an appropriate one

  (e.g., CardClassification to ForkKnifeSpoonClassification.)
- Comment out the download function
- Change below path so that it points at your data.
  - *self.data_path in setup()*
- Delete the following command:
  - *self.download(self.data_path) in setup()*

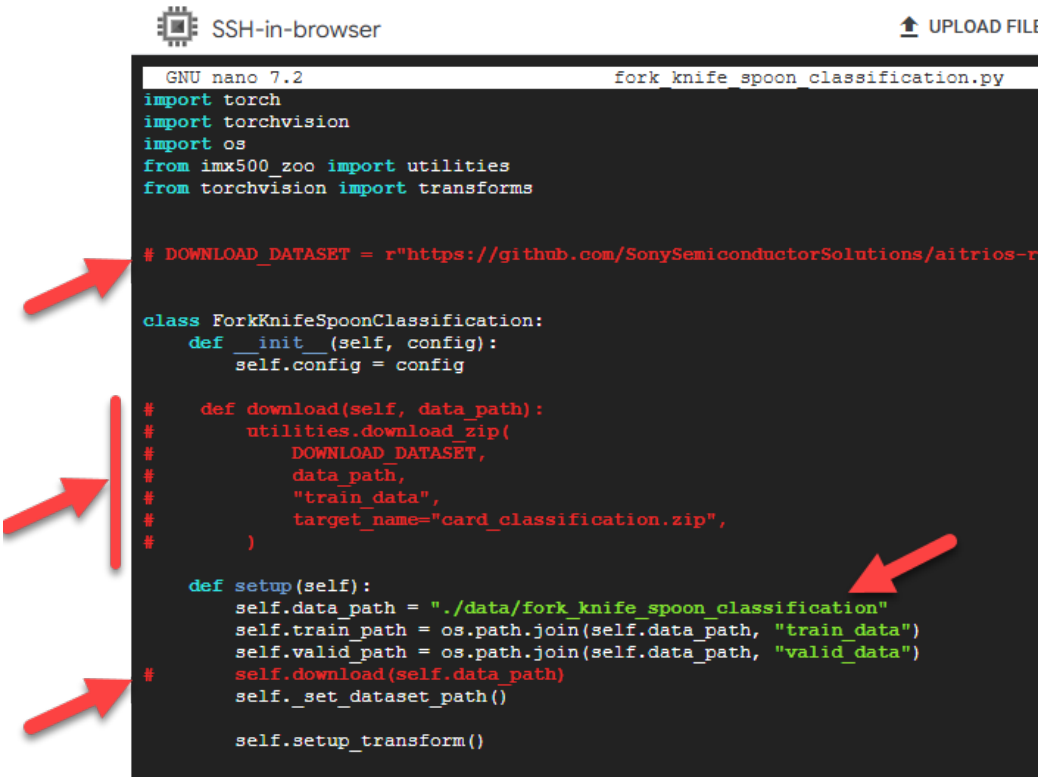


*Figure 9: Adjusting the *_classification.py file.*

src/imx500_zoo/datasets/__init__.py

Edit this file to include the data loader class name:

Example:
from imx500_zoo.datasets.fork_knife_spoon_classification import ForkKnifeSpoonClassification

samples/mobilenet_v2_card.ini

Create an appropriately named copy of this file (e.g., mobilenet_v2_fork_knife_spoon.ini).

- Change [MODEL] > NUM_CLASSES to show the number of classes.
- Change [DATASET] > NAME to the name of dataloader class in the Python file above.
- Ensure that [TRAINER] > BATCH_SIZE is less than the number of images in each subfolder of your training set.

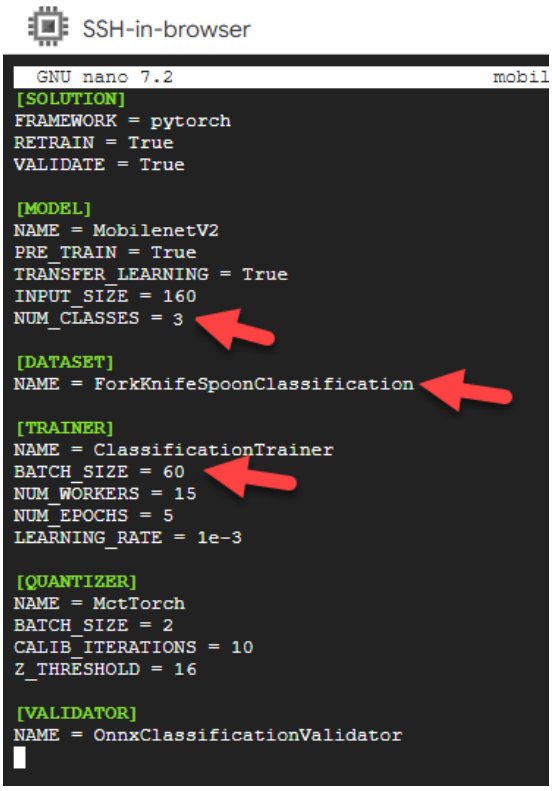


Figure 10: Adjusting the ini file.

- Follow the instructions in the section “Running the sample code to create the model” to create the model. Remember to specify the new copy of the .ini file that you created.
- Convert the model as shown in the section “Convert the model”.
- Download the model as shown in the section “Download the files from the VM”
- Upload the model to the camera as shown in “Load the model on to the Triton Smart”. Create a classification.txt that contains the name of the classes you are trying to detect (e.g., knife, fork, spoon).

Arena SDK support

For more information on using the Triton Smart with the Arena SDK, see the Triton Smart Support Package available on the Triton Smart product page.

FAQ

- Q: How many images do I need for my training setup?
  A: There is no fixed number of images needed. Generally, more controlled environments where the lighting and background are controlled and the object is always the same size and orientation will need fewer images. In less controlled environments where the objects are partially hidden or differently sized, more training images will be needed.

  For reference, the number of files used for training and validation for the cat/dog example is shown below:

Training data	cat	400 images
	dog	400 images
Validation data	cat	100 images
	dog	100 images

- Q: Do I have to annotate images before training a classification model?
  A: The classification model does not require annotation.

- Q: What if I want to train my model using another cloud platform?
  A: Contact LUCID Support.

- Q: What else can I do with the Triton Smart? Are there other types of models?
  A: Upcoming VM image releases will include other model types including object detection.