Deep Learning at the Edge – Activity 3

Task 3: Image Classification on the Jetson Nano

Introduction

In this lab, we will go through the steps of creating an image classification project using a Convolutional Neural Network (CNN) that is deployed to the Jetson Nano developer kit using the Edge Impulse features.

Required Hardware

The Jetson Nano is an embedded Linux development kit featuring a GPU accelerated processor (NVIDIA Tegra) targeted at edge AI applications. You can easily add a USB external microphone or camera - and it's fully supported by Edge Impulse. You'll be able to sample raw data, build models, and deploy trained machine learning models directly from the Studio.

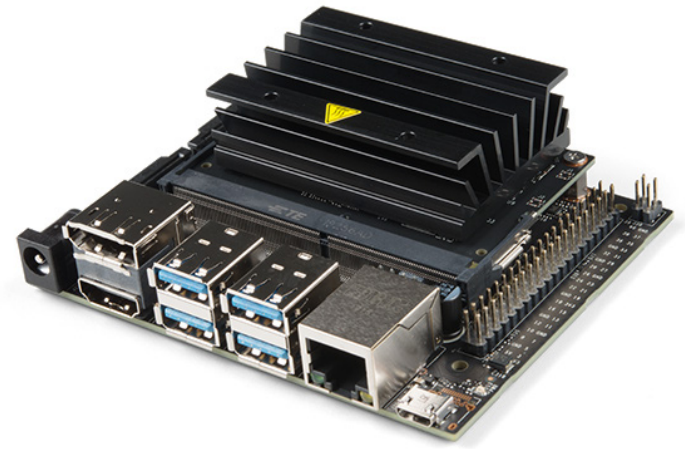


Fig. 1: Nvidia Jetson Nano

The Jetson Nano dev kit is already set up for you and can be accessed remotely using MobaXTerm software available on the host machine. The IP address for connecting remotely to the nano board is 192.168.55.1. Open a terminal connection using SSH and when prompted, enter the password that is 'nvidia**X**' for your Jetson device.

Later, we will connect the camera to your Jetson Nano and set up the software to connect to the Edge Impulse project specific to this lab.

In your Edge Impulse account, create a new project – **my-catvsdog-jetson-project**

1. Collect Data

We will start with a pre-made catvsdog image dataset. The dataset contains a separate set of train and test images:

- cat – train: 1000 images, test: 100 images
- dog – train: 1000 images, test: 100 images

Importing this dataset

You can import this dataset to your Edge Impulse project by:

1. Downloading the "**cat_dog.zip"** file from https://github.com/brenmul/deep_learning_lab/tree/main
2. Unzip the file in a location of your choice.
3. In your project, go to **Data acquisition** and click on the 'Upload icon'. Follow the instructions on the screen.

2. Upload Data

In your project in Edge Impulse. Head to the **Data Acquisition** page and then go to the **Upload data** option.

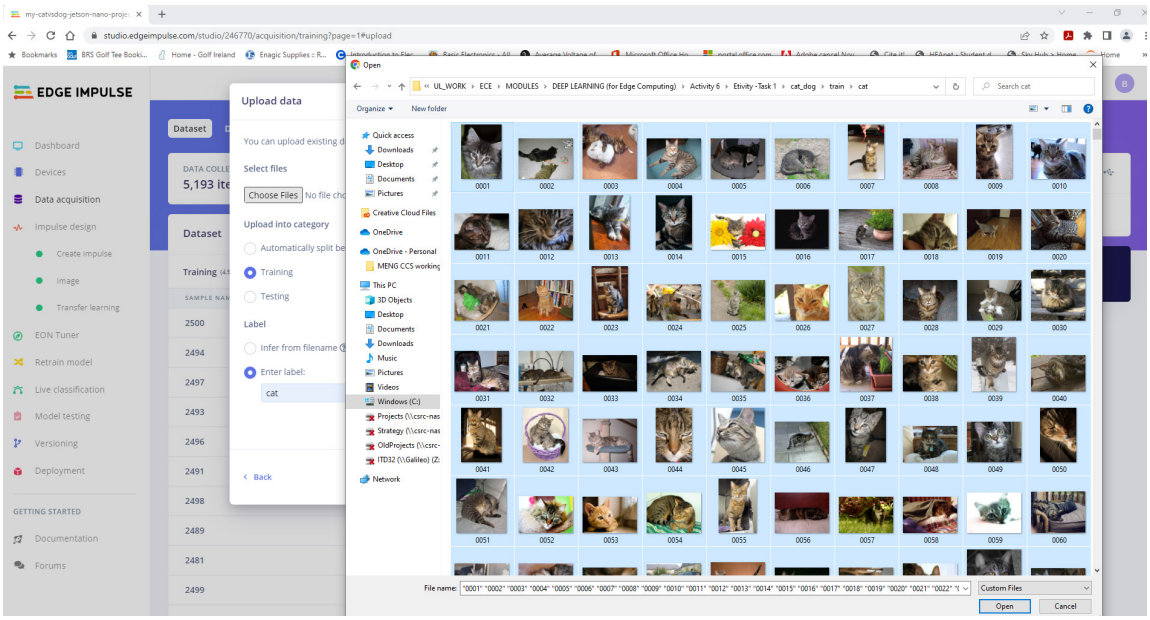
We must load the cat and dog images separately for the training and test sets. First, we load our cat and dog images into the training set.

Make sure to select *Training* from the **Upload into category** and specify the label as '**cat**' before selecting all the cat images from the train folder.

Click **Open** and then click **Upload Data**.

(Note: You might be asked if this is an object detection project – select No.)

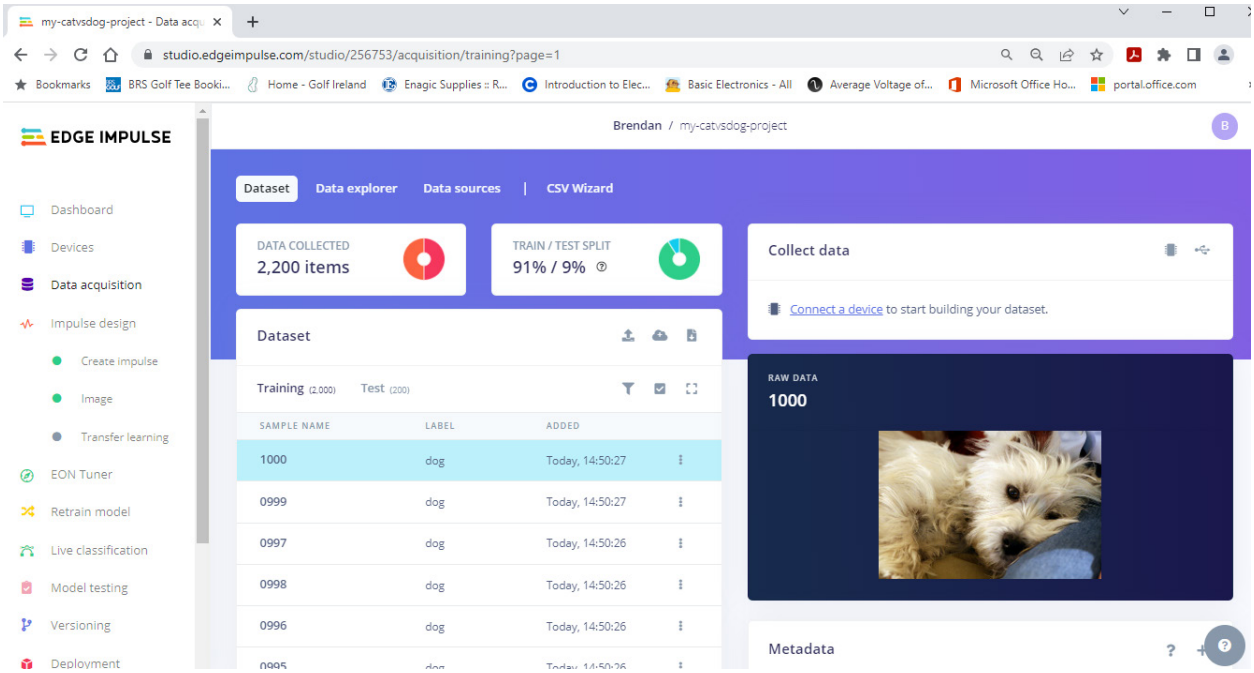
Deep Learning at the Edge – Activity 3



Repeat this process for the dog images in the training folder, making sure to specify *Training* and the *label* text as '**dog'**.

Repeat this process of loading the cat and dog images in the test folders for the *Testing'* category too.

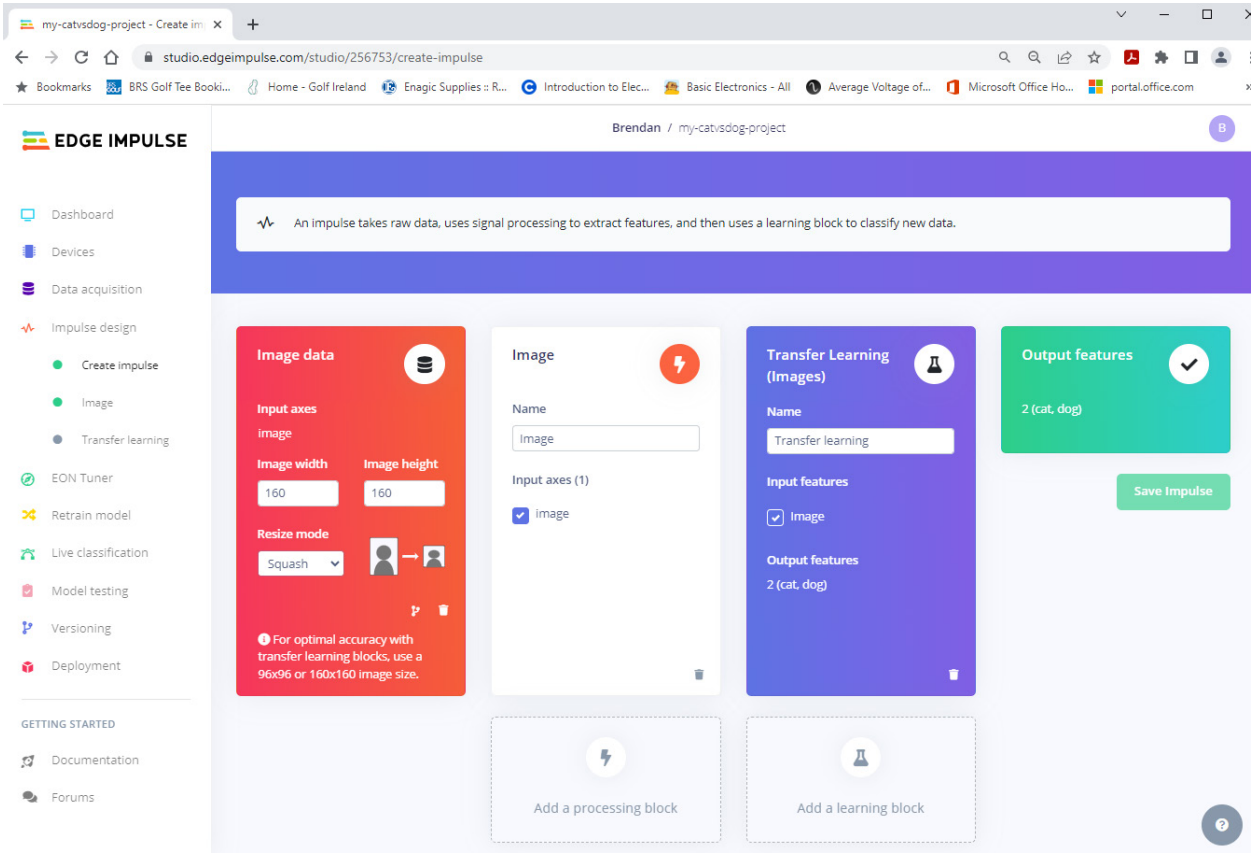
Finally, click on the **Data acquisition** link to go back to the Data Acquisition page. Here, make sure that all the images are present and that they are divided between the training and test sets correctly (there should be 2000 image samples in the training set and 200 images in the test set).



Notes by Brendan Mullane

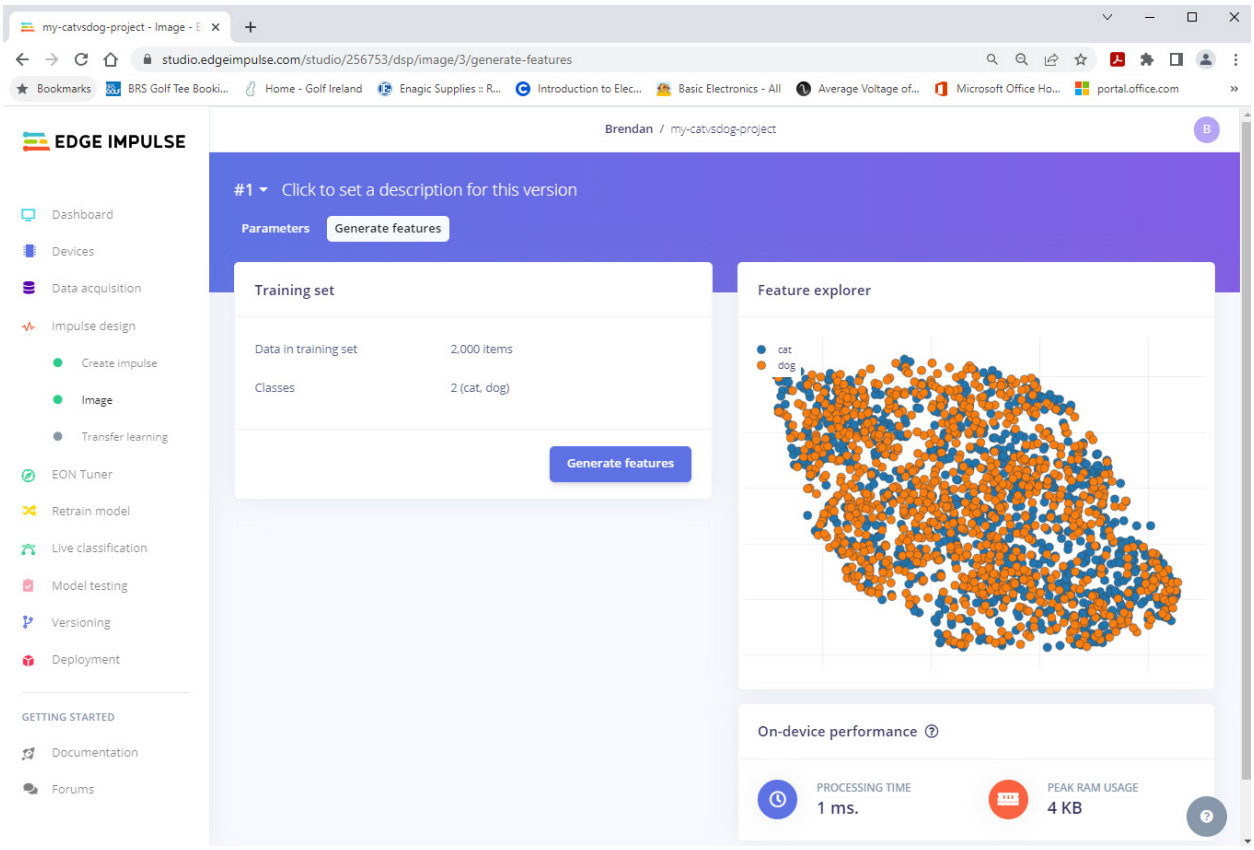
3. Feature Extraction

Navigate to the **Impulse design** page of your project. Add an **Image** processing block and a **Transfer Learning** (Images) block.



Click **Save Impulse.**

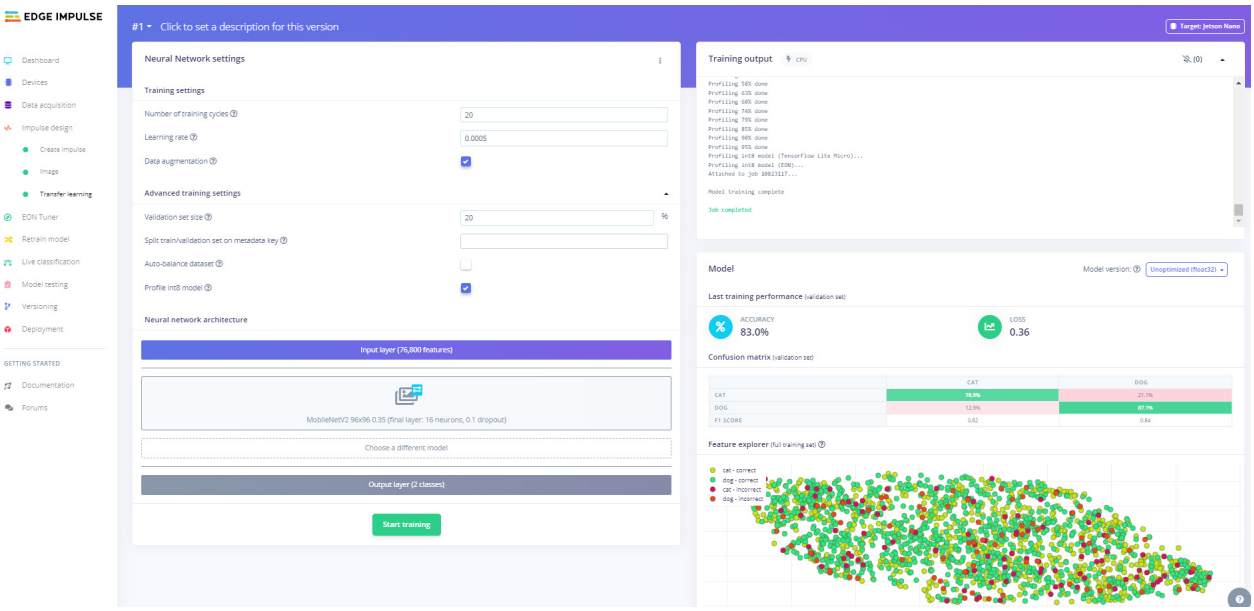
Go to the **Image** page and make sure the Image Parameters option is set to RGB, then click on the **Generate Features** tab and wait a moment while your image samples are analysed. When it's done, look at the **Feature explorer** to see if you can identify separation among your classes. These features do not look like they are separated well!



4. Model Training

Navigate to the **Transfer Learning** page. Leave all the hyperparameters at their defaults, except for *the Number of training cycles, which* is set at 20. As the free Edge Impulse version limits us to 20 minutes of training time and since we have a relatively large set of images, we only train the model over 20 epochs. **Make sure** to select the Jetson Nano as your target device in the top right-hand corner. Then click **Start training**. When it is done, scroll down to view the *Confusion matrix* of the validation data.

Note that the default model that was chosen for this exercise is a MovbileNetV2 model, which is suitable for image classification on edge devices.

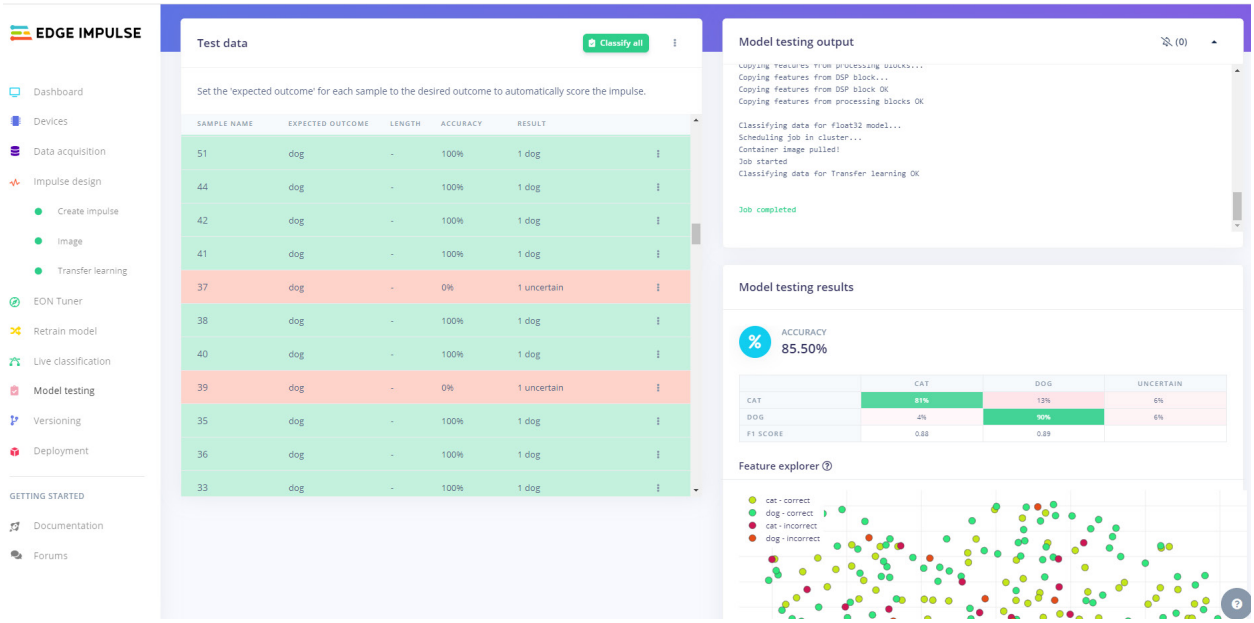


Note the *F1 scores* of each class and the *total accuracy*.

We have achieved a valid set accuracy of 83%, which is very good for training with just 1000 images. Feel free to try changing some of the hyperparameters and re-training your model to see if it improves the per-class accuracy. Note that there is no easy solution here: much of creating a better model is trial and error, and sometimes, you simply do not have enough (or the right kind of) data to train a good model. In those cases, it is back to the drawing board to gather data!

5. Testing

Head to the **Model testing** page and classify all the test data.



If you are happy with the test results, continue to the deployment step. 85.5% is good, but this performance indicates that we need to collect more data for this project or perhaps we just need to run the training for more epochs!

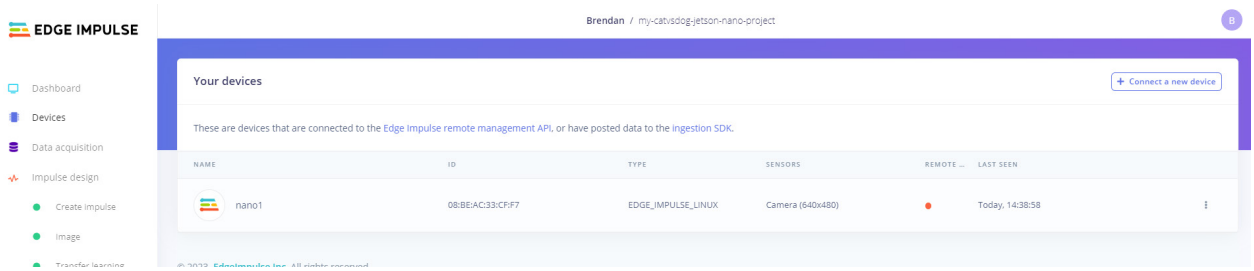
6. Deployment – Jetson Nano device

We will use the Edge Impulse CLI to deploy this model. First, connect your USB camera to the Jetson Nano and run the following command in the Linux terminal that you connected by SSH from the MobaXterm tool:

```
edge-impulse-linux --disable-microphone
```

This will start a wizard, which will ask you to log in, and choose an Edge Impulse project. If you want to switch projects run the command with --clean.

That is all! Your device is now connected to Edge Impulse. To verify this, go to your Edge Impulse project, and click **Devices**. The device will be listed here with the name you have chosen.

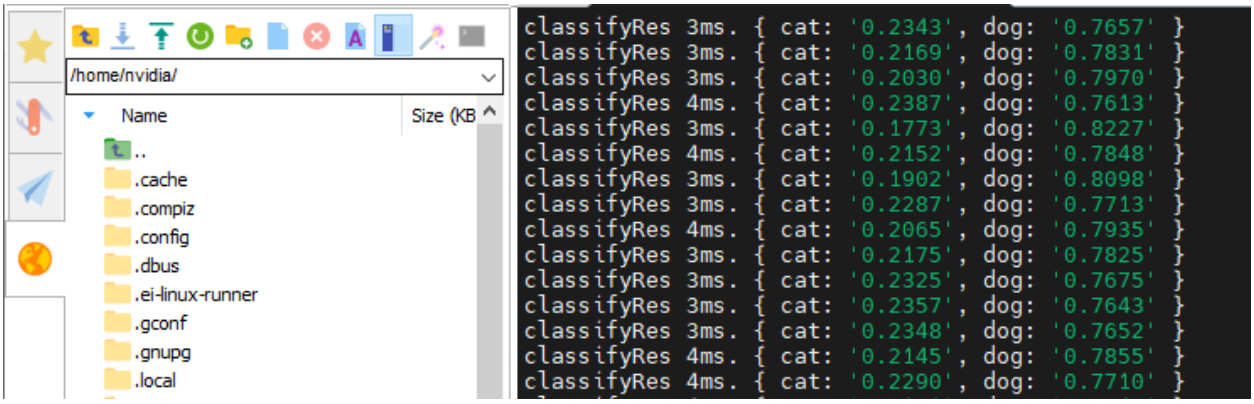


Next, to run your impulse locally, just connect to your Jetson again, and run in the same terminal:

```
edge-impulse-linux-runner
```

This will automatically compile your model with full hardware acceleration, download the model to your Jetson, and then start classifying.

It is as simple as that using Edge Impulse. Test out the model running on the Jetson, pointing the camera at different cat and dog images (pointing the camera at images from the internet is ok). You should see results like this screenshot below, which gives the probability of the cat or dog image as a value between 0 and 1.



7. Conclusion

Congratulations, I hope this has helped you get familiar with image classification and developing a model deployed to the Jetson Nano development kit! The Edge Impulse studio really makes it easy. In the next lab, we will use **Nvidia's Hello AI world - image classification tutorial** to train a model and then classify the results stand-alone on the Jetson Nano device without using the Edge Impulse platform.