Intel Neural Compute Stick 2 with Transfer Learning

Using the AWS SageMaker Transfer Learning Example
and a Raspberry Pi robot with a camera

Intel Neural Compute Stick

- Formally the Movidius Neural Compute Stick
- A Neural Net Vision Processing Unit (VPU)
- Neural Compute Engine
- - a dedicated hardware accelerator for deep neural network inferences.
- Inference, not learning
- The NCS2 uses the Intel Movidius Myriad X VPU

Why a Neural Net Computer Stick?

- Development aid for the Intel Movidius Myriad VPU’s.
- The VPU’s are for low power edge devices, not for your desktop.
- Edge devices such as camera, drones, robots and other edge sensors.
- Specialised Inference at the edge devices allows for:
  - Quicker, low latency, decisions
  - Low bandwidth or no-bandwidth inference
  - Reduce the processing power requirements of smart edge devices.
  - No reliance on centralised servers to keep them running.
  - Sensitive information doesn’t have to leave the device. Good for sensitive solutions such as children's toys and security cameras.

SageMaker

- Amazon SageMaker is a fully managed machine learning service.
- Quickly and easily build and train machine learning models, and then directly deploy them into a production-ready hosted environment.
- Provides an integrated Jupyter authoring notebook instance.
- Common machine learning algorithms
- Support for bring-your-own-algorithms and frameworks.

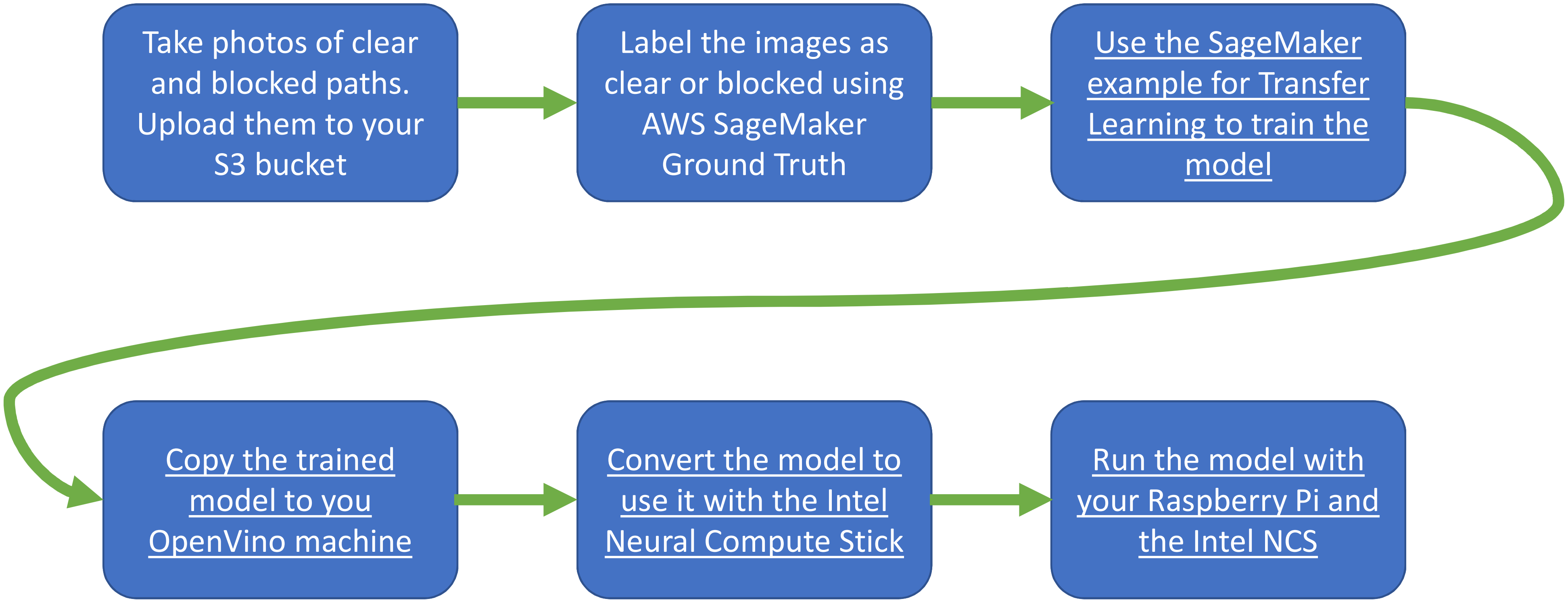
Transfer Learning

- **Transfer learning** takes a model trained for one purpose (e.g. detecting cars in a picture) and applying it to another related problem (detecting trucks).
- In this example, we will use the Caltech-256 object category dataset model
- We will use the AWS SageMaker transfer learning example
- Uses the Apache MXnet deep learning library to perform transfer learning to determine if the path in front of the robot is blocked or not.
- This is not a recommendation. This example is for learning the process rather than a best approach.

Getting Started

- You’ll need the following to perform this process:
  - Raspberry Pi
  - Camera for the Raspberry Pi (Raspberry Pi Camera or USB Camera)
  - AWS Account for SageMaker and an S3 bucket
  - Machine with OpenVino installed. The Raspberry Pi version doesn’t include the tools to convert the models. (https://software.intel.com/en-us/openvino-toolkit)
  - A Robot for the Raspberry Pi to control (if you want to test it on a robot)

Process Overview



Take photos of clear and blocked paths. Upload them to your S3 bucket

Label the images as clear or blocked using AWS SageMaker Ground Truth

Use the SageMaker example for Transfer Learning to train the model

Copy the trained model to you OpenVino machine

Convert the model to use it with the Intel Neural Compute Stick

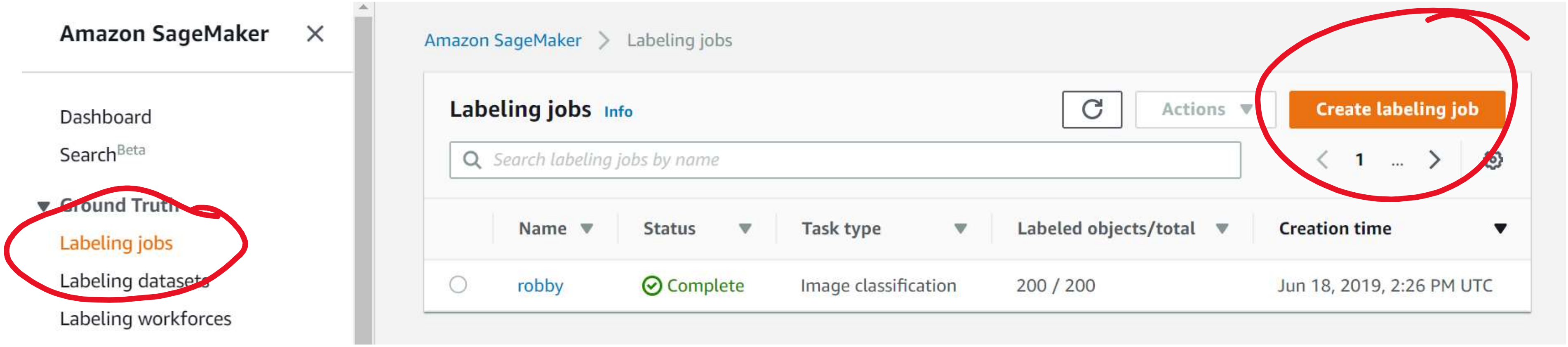
Run the model with your Raspberry Pi and the Intel NCS

Taking lots of photos

- You’ll need at least 200 photos (more is better) of blocked and free paths for the robot. You can use you mobile phone.
- The images will need to be cropped square and reduced to 224 x 224 pixels.
- Upload them to the S3 bucket you will use for training and put them in a folder called images.

Label the images using SageMaker Ground Truth

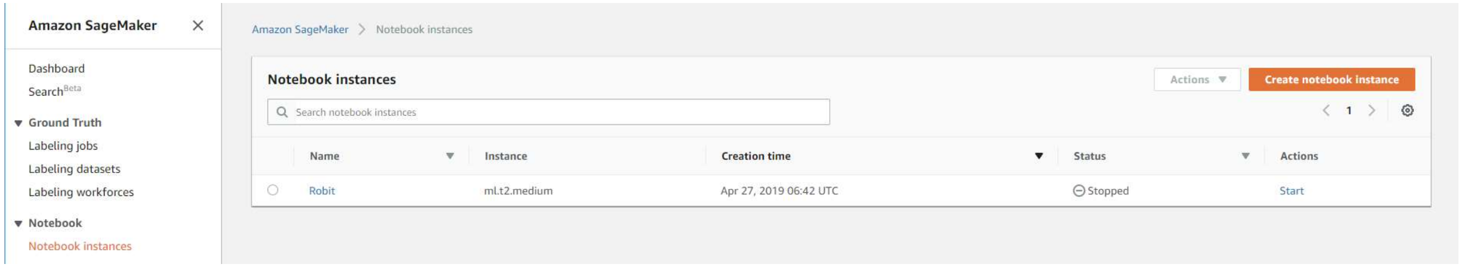
- Open up AWS SageMaker and create a Labelling Job in Ground Truth.
- Follow the instructions to label the job.



	Name	Status	Task type	Labeled objects/total	Creation time
○	robby	Complete	Image classification	200 / 200	Jun 18, 2019, 2:26 PM UTC

Open Notebook Instances

Click on Notebook Instances

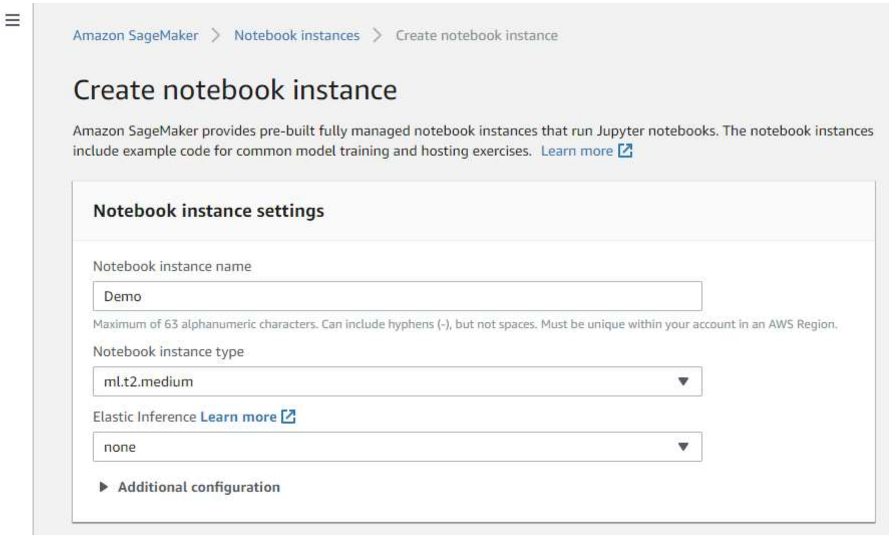


Notebook instances

	Name	Instance	Creation time	Status	Actions
○	Robit	ml.t2.medium	Apr 27, 2019 06:42 UTC	Stopped	Start

Create a Notebook

Create a new Notebook



Amazon SageMaker > Notebook instances > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. Learn more

Notebook instance settings

Notebook instance name

Demo

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

ml.t2.medium ▼

Elastic Inference Learn more

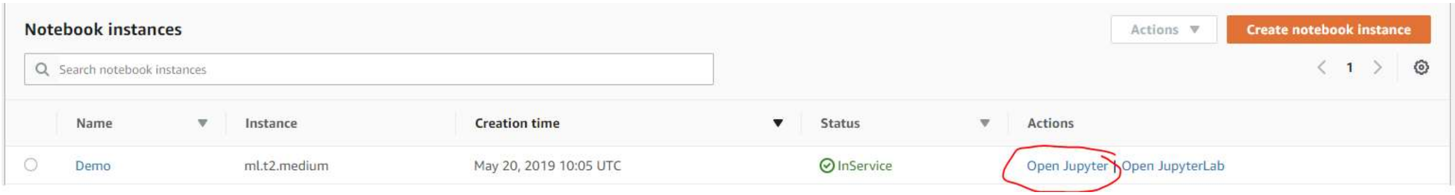
none ▼

▶ Additional configuration

Click Create notebook instance at the bottom of the page

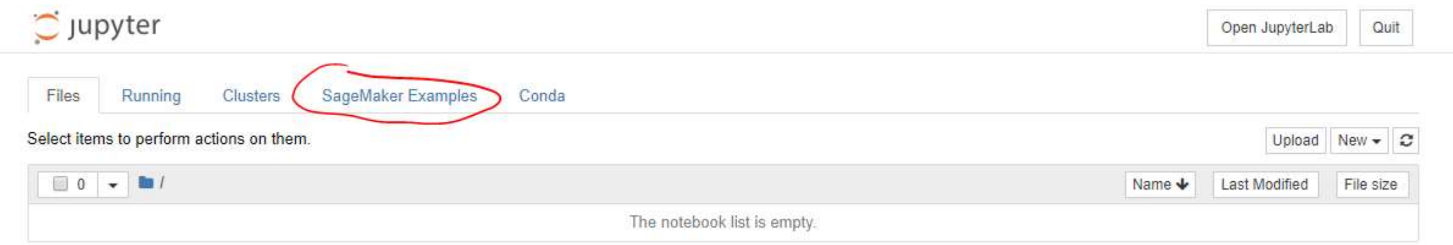
Open a Jupyter Notebook

Open the Jupyter Notebook



Notebook instances

	Name	Instance	Creation time	Status	Actions
○	Demo	ml.t2.medium	May 20, 2019 10:05 UTC	✓ InService	Open Jupyter | Open JupyterLab



Select items to perform actions on them.

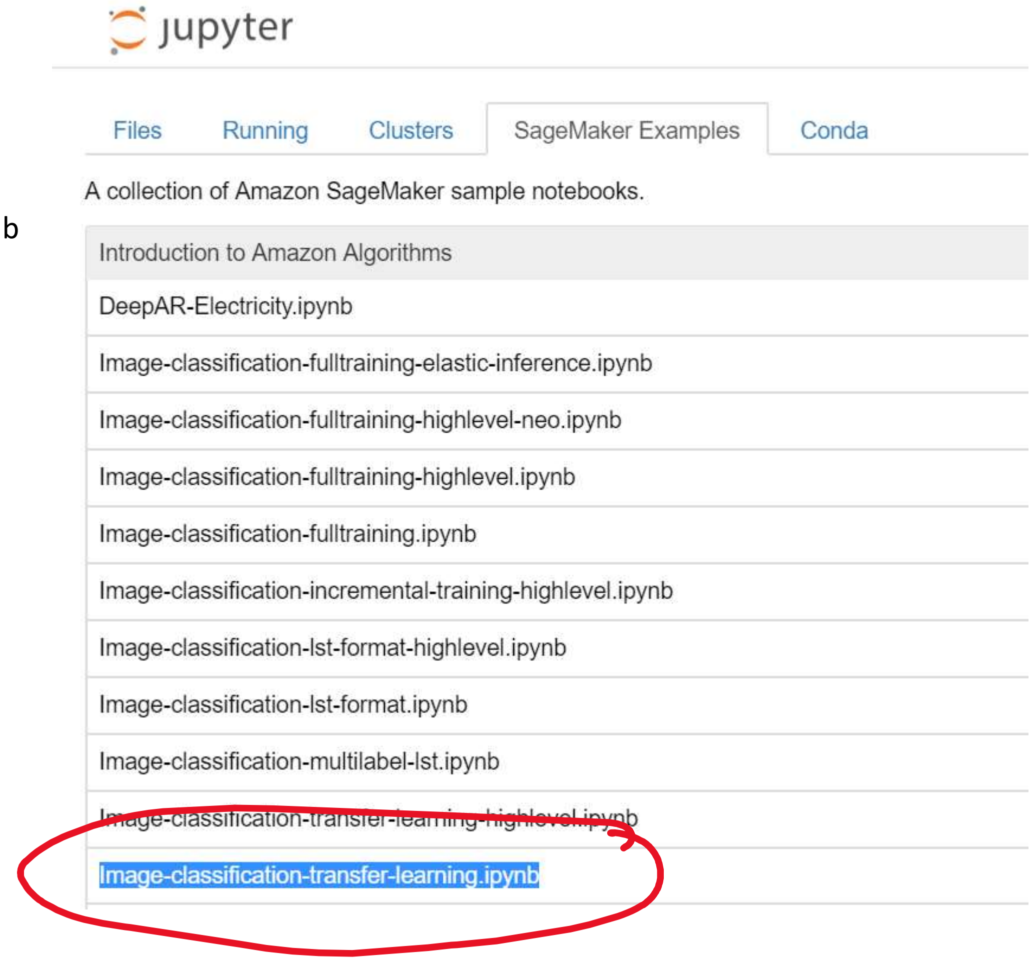
The notebook list is empty.

Open SageMaker Examples

Opening the Example

Use the example

Image-classification-transfer-learning.ipynb



A collection of Amazon SageMaker sample notebooks.

Introduction to Amazon Algorithms

DeepAR-Electricity.ipynb

Image-classification-fulltraining-elastic-inference.ipynb

Image-classification-fulltraining-highlevel-neo.ipynb

Image-classification-fulltraining-highlevel.ipynb

Image-classification-fulltraining.ipynb

Image-classification-incremental-training-highlevel.ipynb

Image-classification-lst-format-highlevel.ipynb

Image-classification-lst-format.ipynb

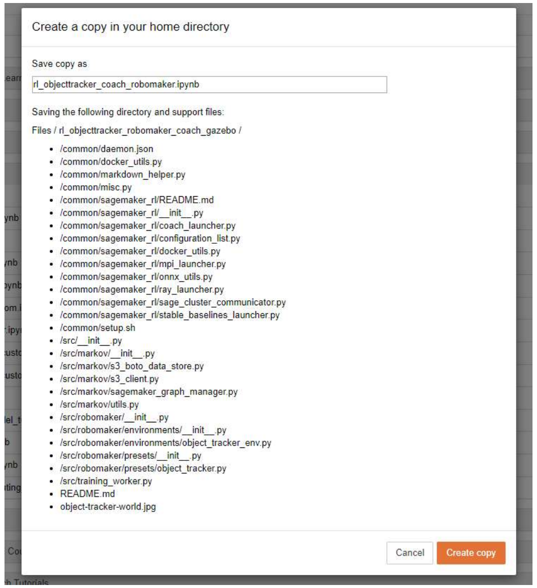
Image-classification-multilabel-lst.ipynb

Image-classification-transfer-learning-highlevel.ipynb

Image-classification-transfer-learning.ipynb

Create a copy of the example

Click [Create Copy]



Create a copy in your home directory

Save copy as

rl_objecttracker_coach_robomaker.ipynb

Saving the following directory and support files:

Files / rl_objecttracker_robomaker_coach_gazebo /

- /common/daemon.json
- /common/docker_utils.py
- /common/markdown_helper.py
- /common/misc.py
- /common/sagemaker_rl/README.md
- /common/sagemaker_rl/__init__.py
- /common/sagemaker_rl/coach_launcher.py
- /common/sagemaker_rl/configuration_list.py
- /common/sagemaker_rl/docker_utils.py
- /common/sagemaker_rl/mpi_launcher.py
- /common/sagemaker_rl/onnx_utils.py
- /common/sagemaker_rl/ray_launcher.py
- /common/sagemaker_rl/sage_cluster_communicator.py
- /common/sagemaker_rl/stable_baselines_launcher.py
- /common/setup.sh
- /src/__init__.py
- /src/markov/__init__.py
- /src/markov/s3_boto_data_store.py
- /src/markov/s3_client.py
- /src/markov/sagemaker_graph_manager.py
- /src/markov/utils.py
- /src/robomaker/__init__.py
- /src/robomaker/environments/__init__.py
- /src/robomaker/environments/object_tracker_env.py
- /src/robomaker/presets/__init__.py
- /src/robomaker/presets/object_tracker.py
- /src/training_worker.py
- README.md
- object-tracker-world.jpg

Run the Jupyter Notebook

- Make the following changes to the Jupyter notebook:

1. Set the bucket name to the name of your S3 bucket
   bucket='robby.robot’
2. Set the training parameters to match your sample size and number of classes (clear,blocked = 2):
   num_training_samples = 200
   num_classes = 2
   mini_batch_size = 100

- Follow the Instructions in the Jupyter Notebook:

Converting to run on the NCS

- Your model will be in a TAR compressed file in you s3 bucket at the following location except the timestamp highlighted will be different:
  DEMO-imageclassification/output/DEMO-imageclassification-2019-06-18-15-44-02/output/model.tar.gz
- Copy the model to the OpenVino machine and decompress the files. You should have three files but we only need the follow two:
  image-classification-0001.params << It may be a different number.
  image-classification-symbol.json
- Run the following command (assuming Linux install):

```
python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo_mxnet.py --input_model image-classification-0001.params --input_symbol image-classification-symbol.json --input_shape [1,3,224,224] --data_type FP16
```

- This will give you the following two files that you need to transfer to the Raspberry Pi:
  image-classification-0001.bin
  image-classification-0001.xml

Running on the Raspberry Pi

- You need to install OpenVino for Raspberry Pi and create a python script from one of the examples to test the model inference. https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_raspbian.html
- Run the python script with the two files you copied over as the trained model as input.
- Code used on the robot: https://github.com/ShaunPrice/AI-in-Robotics/tree/master/Code%20for%20Intel%20Neural%20Compute%20Stick%20on%20Waveshare%20Alphabot%202
- Video of the robot running. The robot was trained using images from a phone. The robots camera is a 160 degree wide angle but only the centre portion is used: https://www.youtube.com/watch?v=QptokSX7YzY

References

Suppliers of the Intel NCS2:

- RS Components: https://au.rs-online.com/web/p/processor-microcontroller-development-kits/1811851/
- Mouser: https://au.mouser.com/search/refine.aspx?Ntk=P_MarCom&Ntt=153249968

Intel OpenVino toolkit: https://software.intel.com/en-us/openvino-toolkit

AWS SageMaker:

https://aws.amazon.com/sagemaker/

Robot used in the demonstration:

https://www.waveshare.com/wiki/AlphaBot2-Pi

Videos

Transfer Learning by Saraj Raval:

https://www.youtube.com/watch?v=Ui1KbmutX0k

NVIDIA Jetbot Neural Network Based Collision Avoidance Demo - Uses transfer learning

https://www.youtube.com/watch?v=U3VJCSDqdG4

Articles

Comparison of Edge Compute Devices

https://medium.com/@aallan/benchmarking-edge-computing-ce3f13942245