# TRAINING IMAGE CLASSIFICATION MODELS FOR i.MX DEVICES

Michael Pontikes
Systems and Applications Software Engineer

**JUNE 2022**

SECURE CONNECTIONS
FOR A SMARTER WORLD


PUBLIC

# TRAINING IMAGE CLASSIFICATION MODELS FOR i.MX DEVICES
**LEVEL:** ADVANCED

## OVERVIEW

- Artificial Intelligence and Machine Learning Overview
- eIQ® Toolkit Overview
- i.MX 8M Plus Overview
- Lab

## MICHAEL PONTIKES

- Systems and Applications Engineer
- Works with machine learning, ISP, machine vision use cases
- Works at NXP in Austin, TX
- Graduate of The University of Texas at Austin

# Artificial Intelligence and Machine Learning

SECURE CONNECTIONS
FOR A SMARTER WORLD

# eIQ® ML SOFTWARE DEVELOPMENT ENVIRONMENT SUPPORTS KEY APPLICATION DOMAINS

Support for IoT, Industrial, Networking Applications at the Edge

## VISION

Multi-camera observation

Active object recognition

Gesture control

## VOICE & SOUND

Voice processing

Alarm Analytics

## TIME SERIES DATA

Smart sense & control

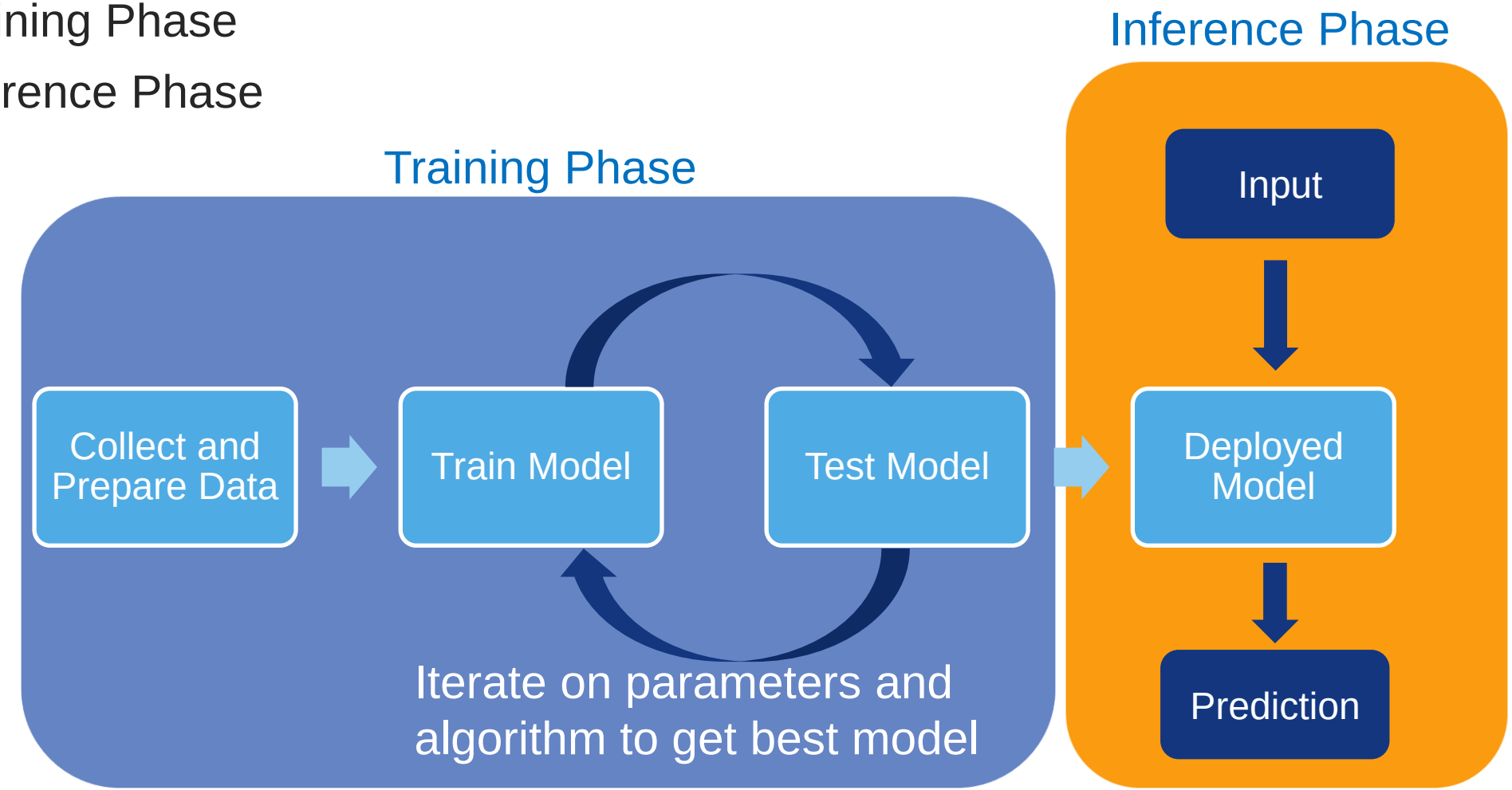
Anomaly Detection

Predictive Maintenance

Edge Processing for Privacy, Performance, Cost Factors

## MACHINE LEARNING PROCESS

1. Training Phase
2. Inference Phase



## MACHINE LEARNING MODELS

- Models are a mathematical representation of a real-world process
  - i.e. image recognition, speech recognition, etc.

- Basically, an extremely complicated math function that gives a “smart” output value for a given input

- Machine learning models look at data to create rules that can be applied to new never-seen-before data


## INFERENCE PHASE

- Inference is using a model to perform a prediction on new data
- Inference time depends on framework and model

Two possibilities:

**Inference on the Cloud**

- Require network bandwidth
- Latency issues
- Cloud computing costs

**Inference on the Edge**

- **Increased privacy and security**
- Faster response time and throughput
- Lower Power
- Don’t need internet connectivity

# AI TERMINOLOGY

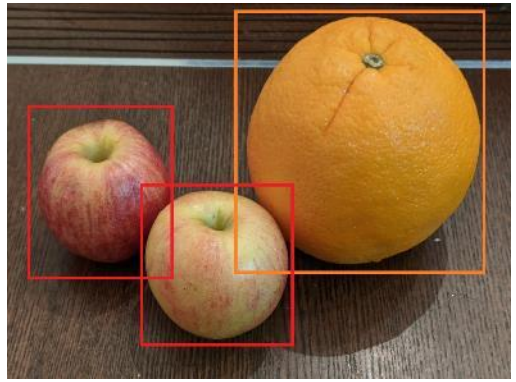
## Image Classification

Apple

## Classification with Localization

Apple

## Object Detection



Apples and Orange

## Instance Segmentation

Apples and Orange

## Facial Detection

This is a human's face

## Facial Recognition

This is NXP CEO Kurt Sievers' face

Complexity

## WHAT PROCESSOR DO I NEED?

- ML inferencing is mostly doing millions of multiply and accumulate math calculations, which any MCU or MPU could do
  - No special hardware or module is required for inferencing
  - However specialized ML hardware accelerators, high core clock speeds, and fast memory can drastically reduce inference time
- Determining if a specific model can run on a particular device is based on:
  - How long will it take the inference to run.
    - The same model will take longer to run on a less powerful device
    - The maximum acceptable inference time is very application dependent.
  - Is there enough Flash memory to store the weights, model itself, and inference engine
  - Is there enough RAM to store the intermediate calculations and output

## WHAT MODEL DO I NEED?

- Many different models could be used to accomplish the same goal
  - Determining the “best” model for a particular application requires trial and error
- End application has large effect on the required model complexity

  Model classifying images into 2 categories with similar lighting and position **vs**

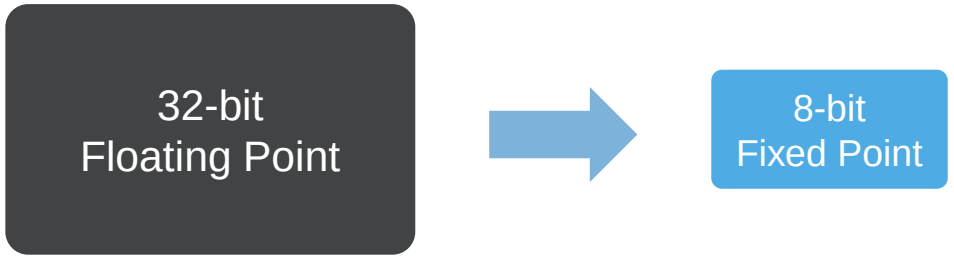
  Model classifying images into 1000 categories in variety of lighting and positions and backgrounds

- Can try to optimize a model for specific application to reduce hardware requirements
  - Trade-off is this takes ML expertise and time to save BOM costs

- Example models:
  - Image classification research models: Mobilenet, CIFAR10
  - Object Detection research models: Mobilenet SSD, Inception

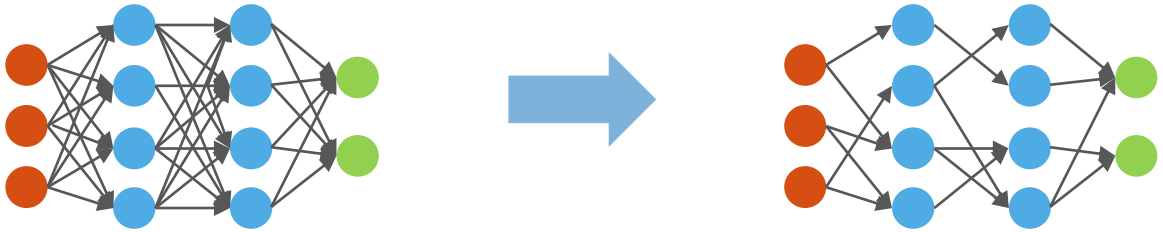
## Quantization

- Transform 32-bit floating point weights → 8-bit fixed point weights
  - Reduces weights data size by 4x
  - Fixed point math quicker than floating point
  - Usually, little loss of accuracy




## Pruning

- Remove low importance weights and biases from a neural network
  - Recommended to retrain model after pruning



# TENSORFLOW LITE AND DELEGATES

## Tensorflow Lite (TFLite)

- A mobile library for deploying models on mobile, microcontrollers and other edge devices.
  - Smaller footprint then the entire TensorFlow Library
  - Model training is mainly done off site, then the model file is saved to the device

## Delegate

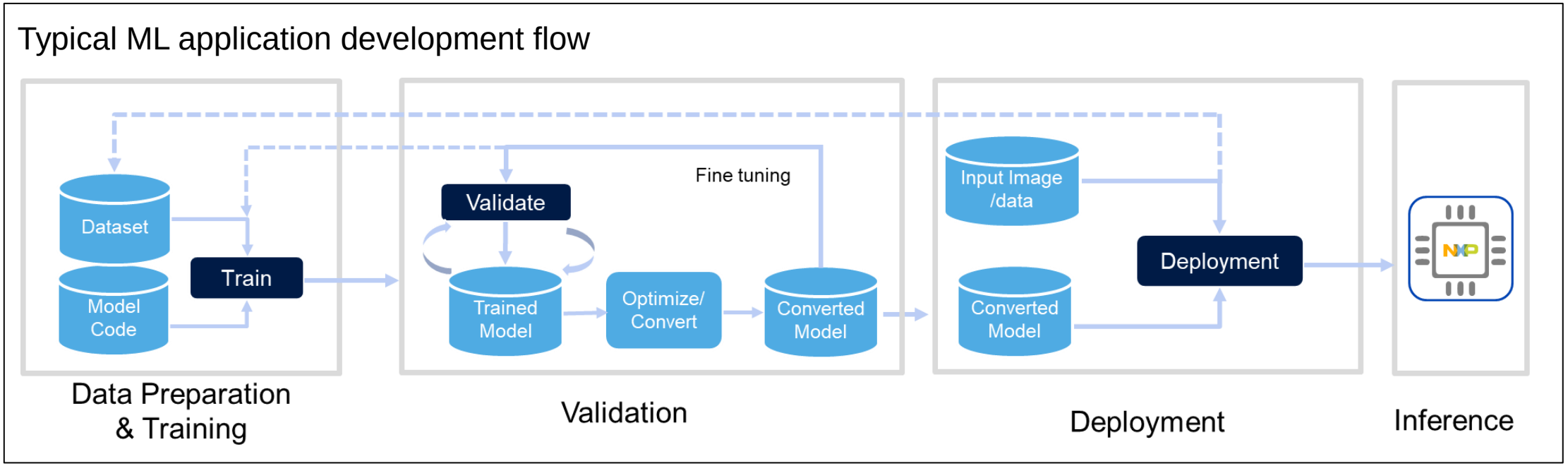
- Delegates enable hardware acceleration by leveraging on-device accelerators such as the GPU and NPU
  - NXP provides an OpenVX delegate to run inferences on the NPU

# eIQ

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

# eIQ® MACHINE LEARNING SW DEVELOPMENT ENVIRONMENT




NXP's **eIQ ML Software** provides a collection of development tools, utilities and libraries for building ML applications using NXP MCUs and applications processors (MPUs).

eIQ ML software can be leveraged as part of a user's existing flow or can be used for the complete flow depending on the ML application targeted.

The user can use external preferred tools and utilities from their existing flow and leverage as much or as little of the eIQ Toolkit as they need.

# eIQ® ML SW DEVELOPMENT ENVIRONMENT

**Embedded Developers**

**Data Scientists**

**ML Experts**

**eIQ Toolkit**

**eIQ Portal**

**eIQ Marketplace**

**Bring Your Own Data Workflow**

Data Curation → Model Selection → Model Training, Optimization, Quantization → Model Validation

**Bring Your Own Model Workflow**

PyTorch, ONNX →  → 

**Solutions and Services from NXP and NXP Eco-System Partners**

- ML Applications
- Optimized Models
- Optimization Tools and Modules
- Development tools
- Datasets
- Training
- Sensor solutions
- ….

**Applications running on EdgeVerse processors**

**eIQ Pipelines**

VISION PIPELINE

VOICE/AUDIO PIPELINE

ANOMALY (TIME SERIES) PIPELINE

PyTorch, ONNX

**eIQ inference**

eIQ inference with…

| TFLite-Micro | Glow | DeepViewRT™ | TFLite | Glow | DeepViewRT™ |
|---|---|---|---|---|---|
| Arm® Cortex®-M, DSP, NPU | | | Arm Cortex-A, GPU, DSP, NPU | | |

# eIQ ML SOFTWARE DEVELOPMENT ENVIRONMENT INFERENCE ENGINE OPTIONS

## NXP eIQ® ML Software Development Environment

**Inference Engines and Libraries for Neural Network Model Deployment**

DeepViewRT

TensorFlow Lite for Microcontrollers

GLOW

| Arm® Cortex®-M | | DSP |
|---|---|---|
| i.MX RT1050<br>i.MX RT1060<br>i.MX RT1064<br>i.MX RT1160<br>i.MX RT1170 | i.MX RT600<br>i.MX RT500* | i.MX RT600<br>i.MX RT500* |

Microcontroller Compute Engines

DeepViewRT

TensorFlow Lite

ONNX RUNTIME

| DSP | GPU | NPU | Arm® Cortex®-A |
|---|---|---|---|
| i.MX 8M Plus | i.MX 8M Plus<br>i.MX 8M<br>i.MX 8M Nano | i.MX 8M Plus | i.MX 8M Plus<br>i.MX 8M<br>i.MX 8M Nano<br>i.MX 8M Nano UL<br>i.MX 8M Mini |

Applications Processor Compute Engines

* i.MX RT500 is supported only with TensorFlow Lite for Microcontrollers

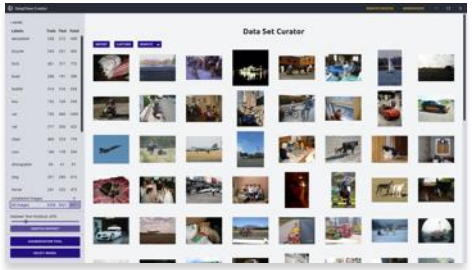
** Planned

*** Additional support for devices not listed can be available or requested

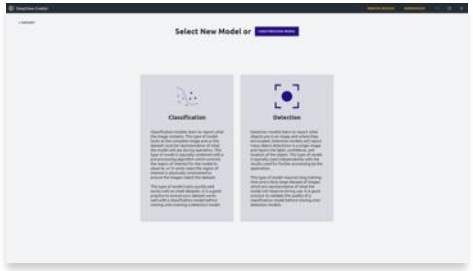
# eIQ Toolkit

SECURE CONNECTIONS
FOR A SMARTER WORLD

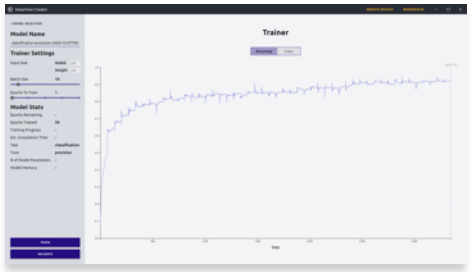
# **Bring Your Own Data** Workflow




Data Curation




Model Selection




Model Training, Optimization, Quantization


Model Validation

# **Bring Your Own Model** Workflow

PyTorch

ONNX

Caffe2

Public or Proprietary Model

Model Conversion, Optimization, Quantization

On Target Profiling and Production

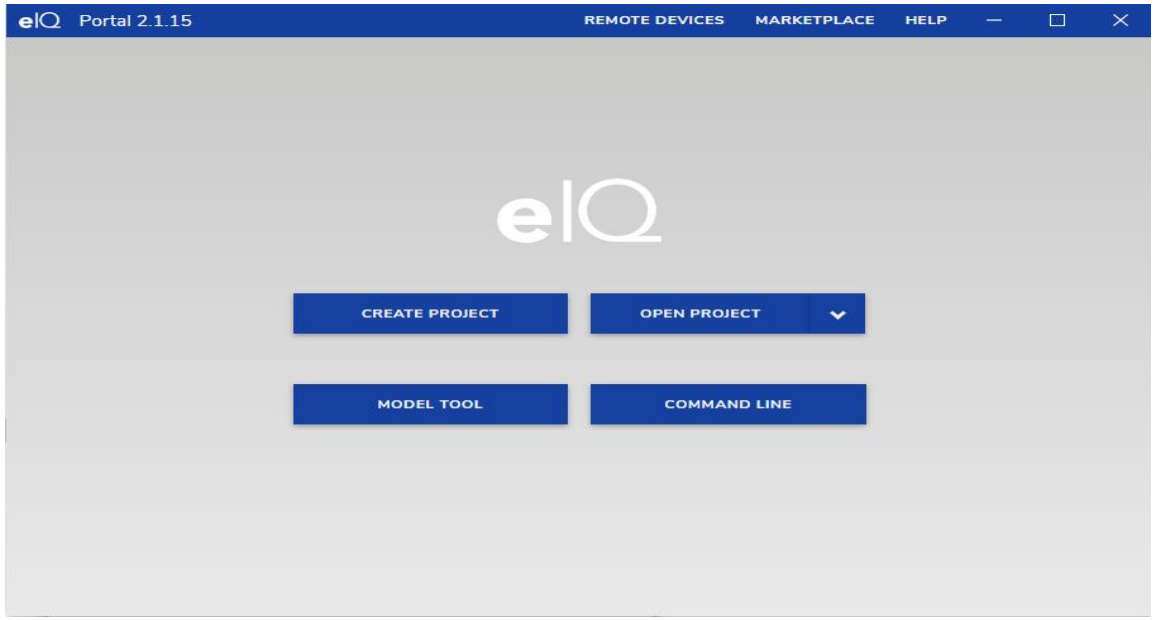
# eIQ TOOLKIT

eIQ Toolkit can be found at www.nxp.com/eiq/toolkit


- eIQ Toolkit package includes the eIQ Portal GUI as well as command line tools for model conversion and creation
- The eIQ Toolkit consists of three key components:
  - eIQ Portal
  - eIQ Model Tool
  - eIQ Command-line Tools

# eIQ PORTAL

- Create, train, and validate models using an intuitive GUI interface on your Window PC.
- Current release support Window PC and Linux.
- Output compatible with DeepViewRT, ONNX, and TensorFlow Lite inference engines
- Support Model validation and Profiling



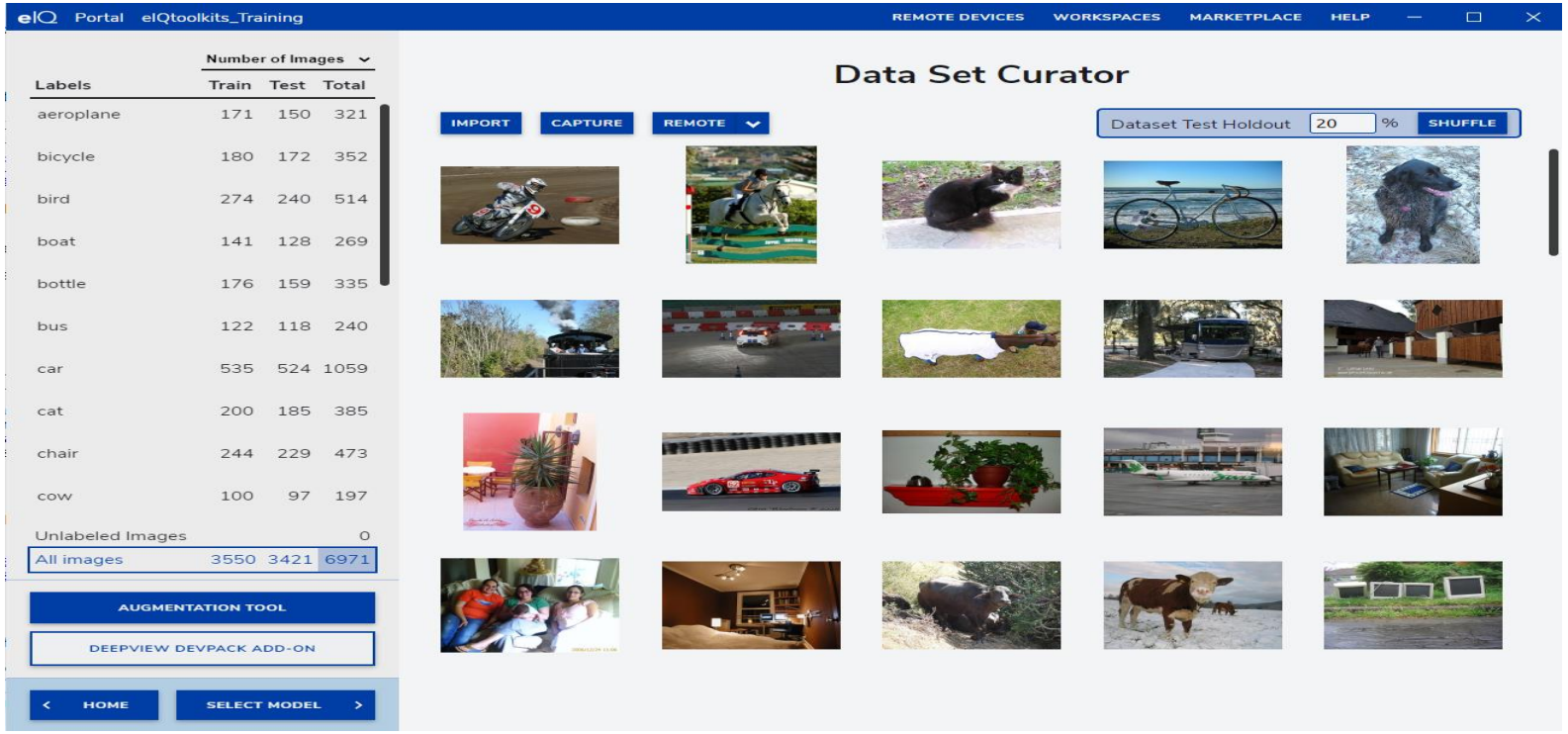
## eIQ PORTAL – IMPORT IMAGES

- Methods for importing images into eIQ Portal:
  - Using the eIQ Portal GUI directly
  - Using DeepView Importer command line tool to import datasets in the VOC format
  - Using Python script to import images based on directory structure
  - Using Python script to import pre-created datasets from TensorFlow

- All methods have the images and associated labels stored in a .eiqp file that can be opened by eIQ Portal
- The **eiqp** file is essentially an SQL database.
- **eiqp** file can grow large with large datasets

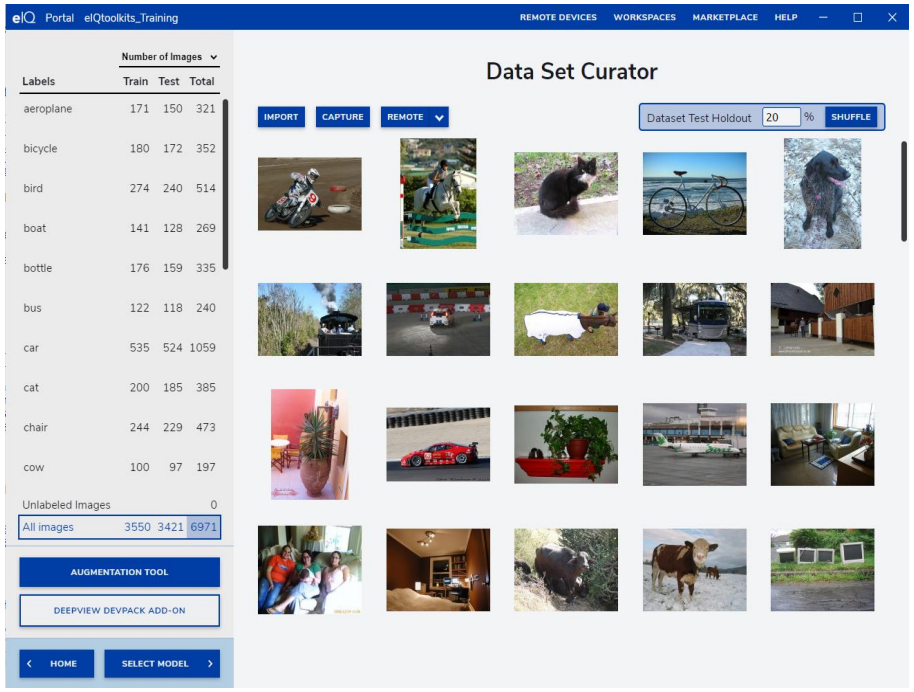
# eIQ PORTAL – IMPORT IMAGES

- Once imported, can easily see how images are labeled and the Training/Test categories
- Can also see if have any unlabeled images



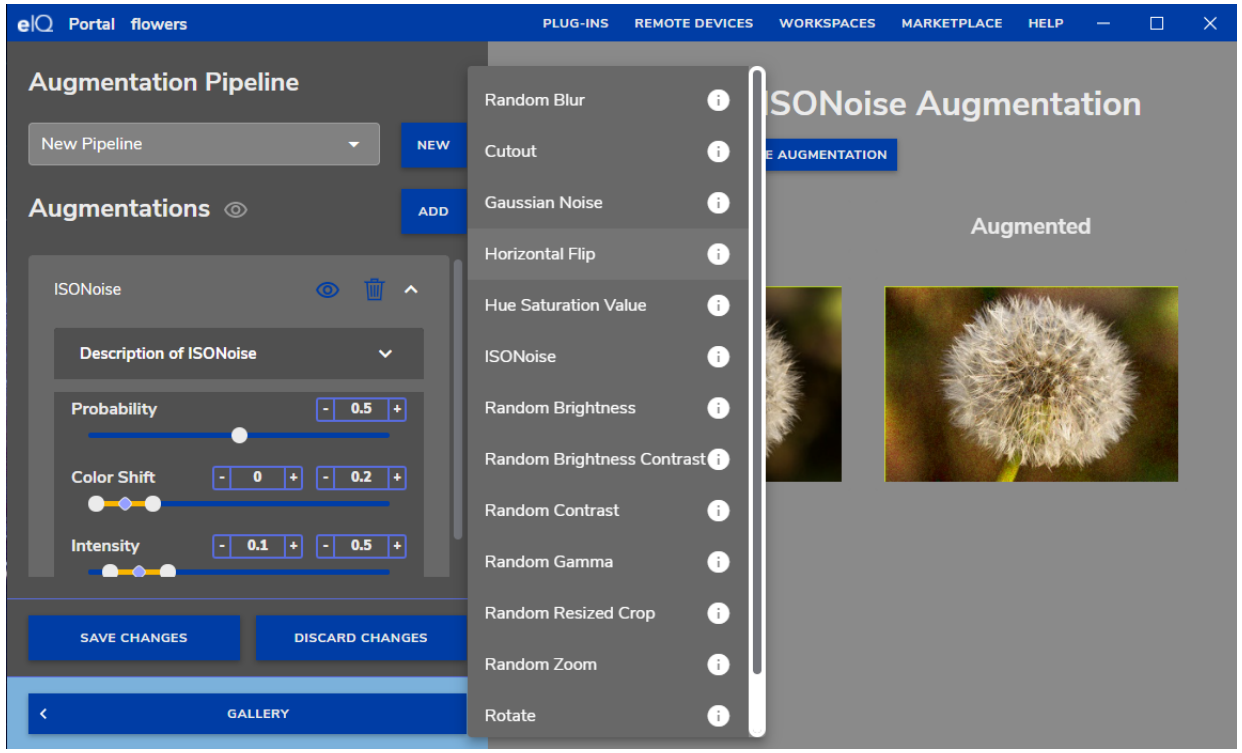
# eIQ PORTAL – DATA CURATION

- Support Data Curation.
  - Data curation is the organization and integration of data collected from various sources.
  - It involves annotation, publication and presentation of the data.
- Simple command line to import well known data set.
- Allow capture of new data set
- Support data labeling on whole image or part of image.
- Can divide into Train and Test categories
- Supports a wide variety of image formats
  - (JPG, PNG, GIF, BMP, etc.)



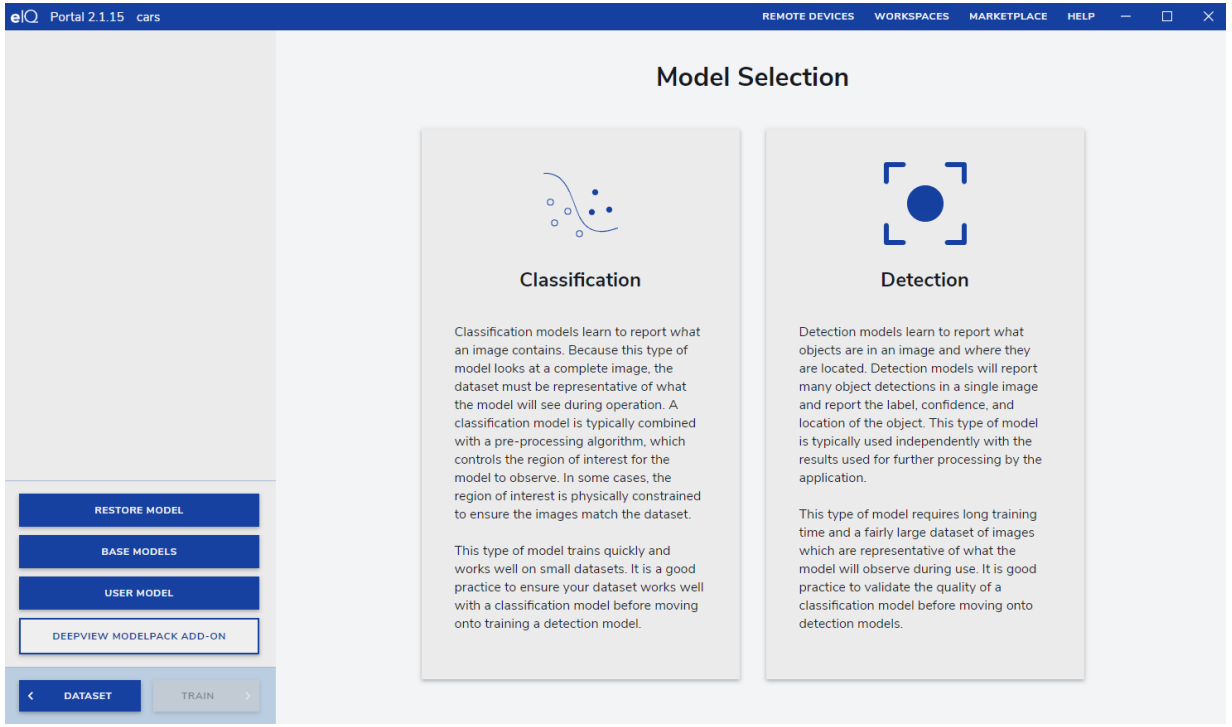
# eIQ PORTAL – DATA AUGMENTATION

- Support Data Augmentation.
  - Data augmentation is a commonly used strategy to significantly increase the diversity of data available for models training.
  - There are various inbuild ways to add new training data set without collecting new data
    - Horizontal/Vertical Flip
    - Random Light Noise
    - Random Cropping
    - …



# eIQ PORTAL – SELECT MODEL

- Can choose between two main types of visual models:
  - Image Classification – Simpler model to analyze entire image and provide estimate on main object in image
  - Object Detection – More complex model that can identify bounding box for specified classification

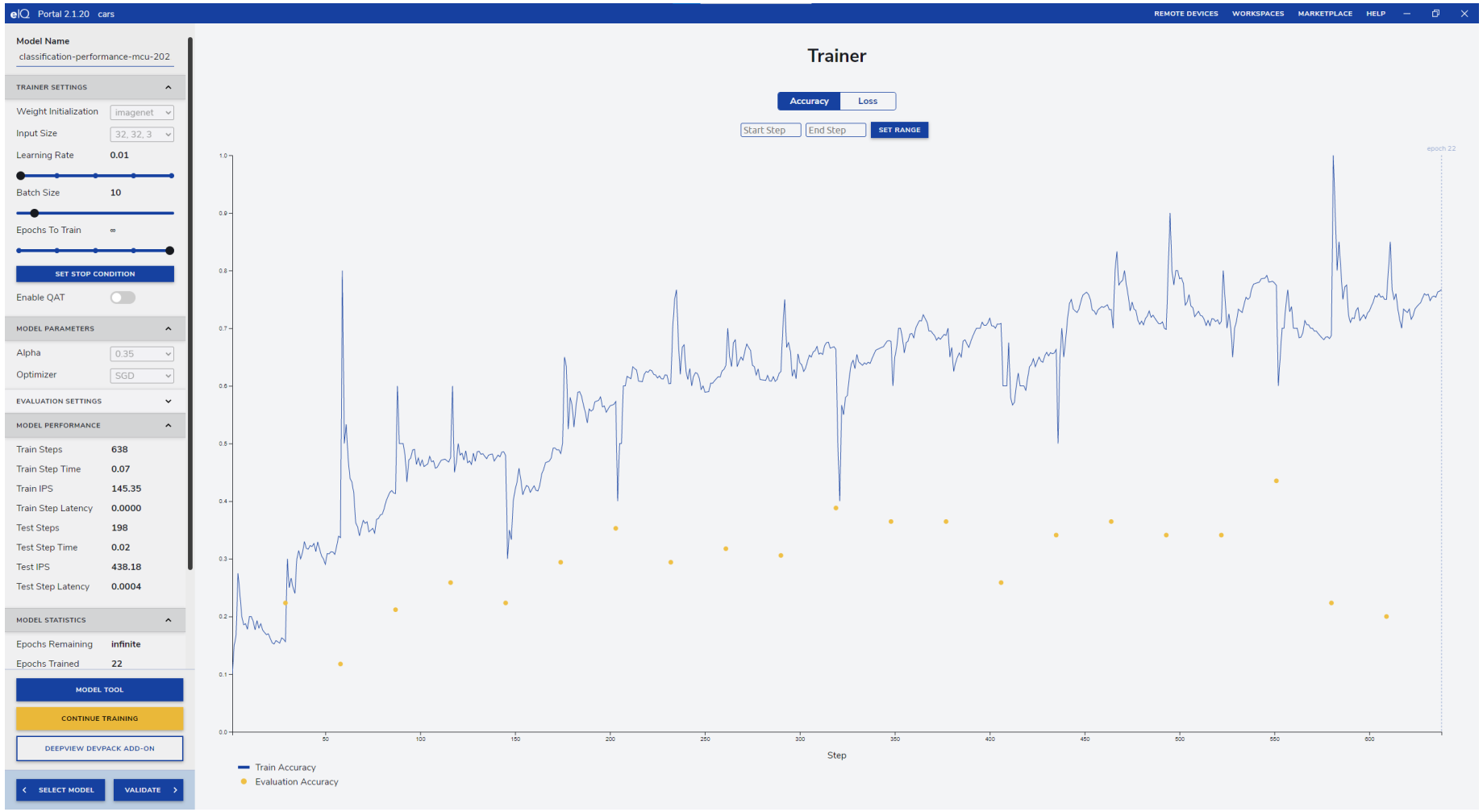


## eIQ PORTAL – TUNE

- The model wizard will select between two pre-built models:
  - Mobilenet v2 (Classification)
  - Mobilenet SSD (Detection)
- For Classification: choosing between Performance, Balanced, and Accuracy will adjust the alpha from 0.35, 0.5, and 1.0 respectively.
- For Detection:  choosing between Performance and Balanced or Accuracy will adjust the scale between small and large
- In the current release, the target selection (MCU/CPU/GPU/NPU) does not make a difference
  - Future releases will target specific hardware engines
- These options are all adjustable in the training phase

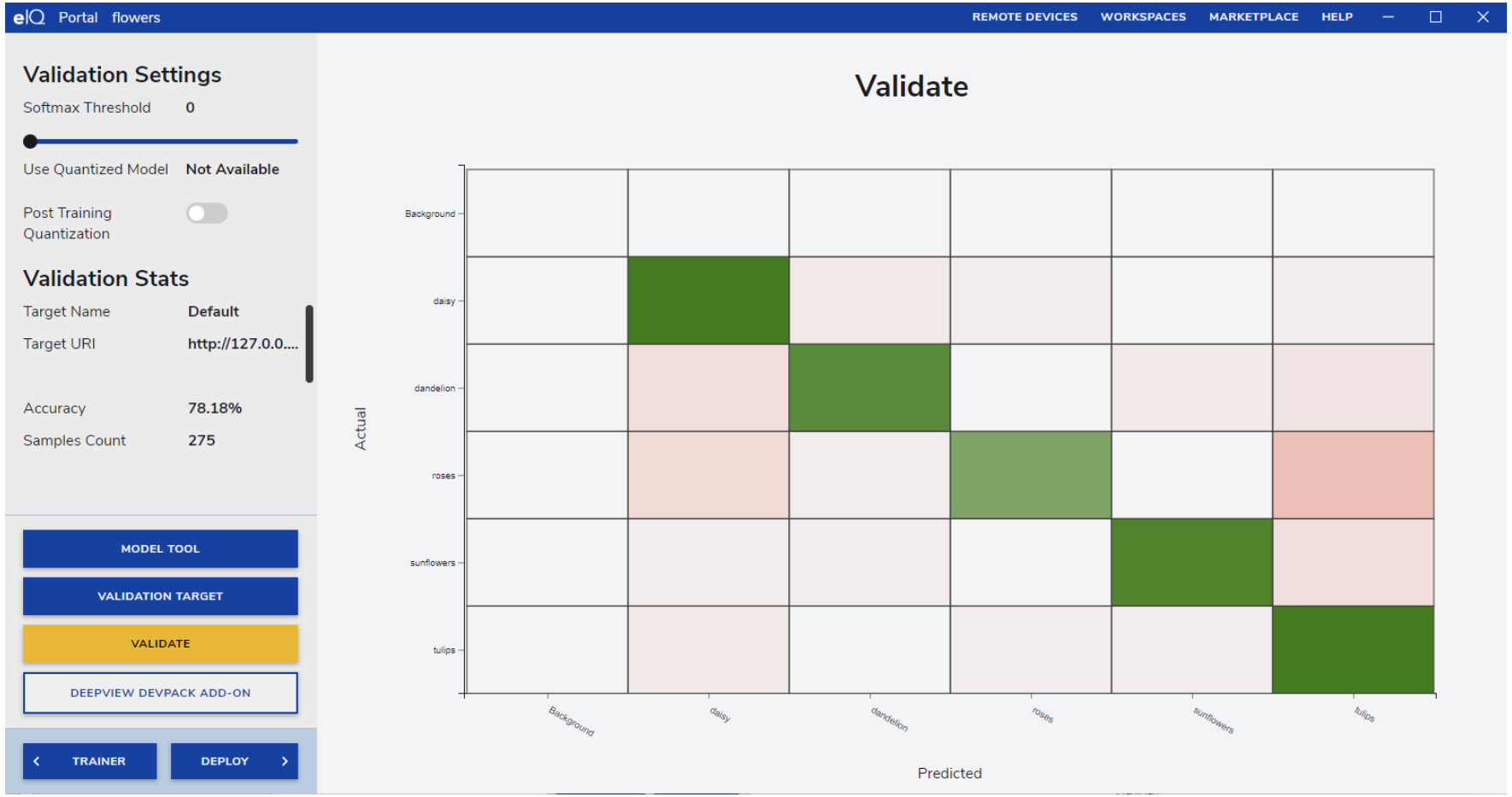
# eIQ PORTAL – TRAINER

- Adjust training options like learning rate, batch size, and epochs and see loss in real-time as the model trains



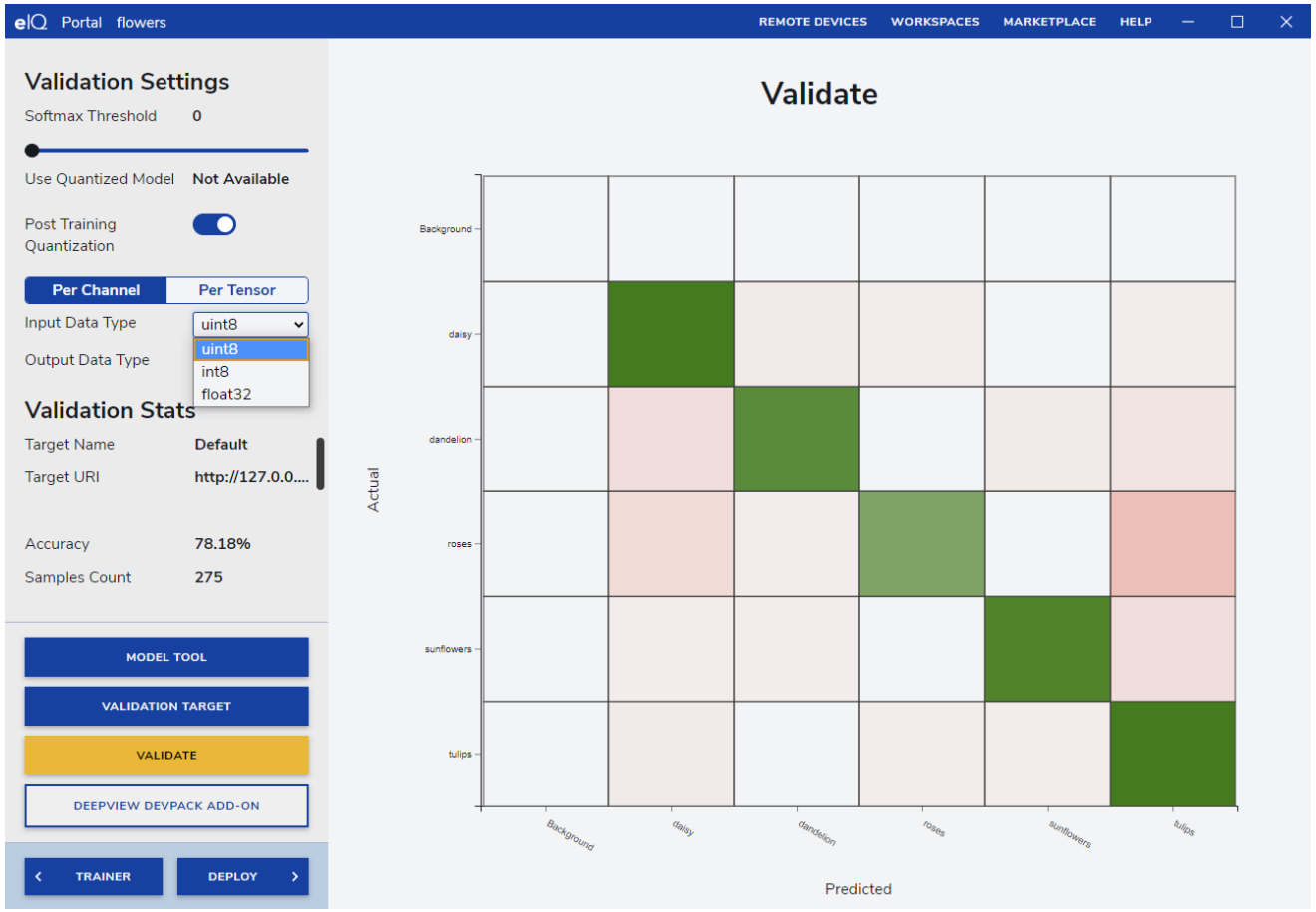
# eIQ PORTAL – VALIDATE

- See the results on a confusion matrix with the test images



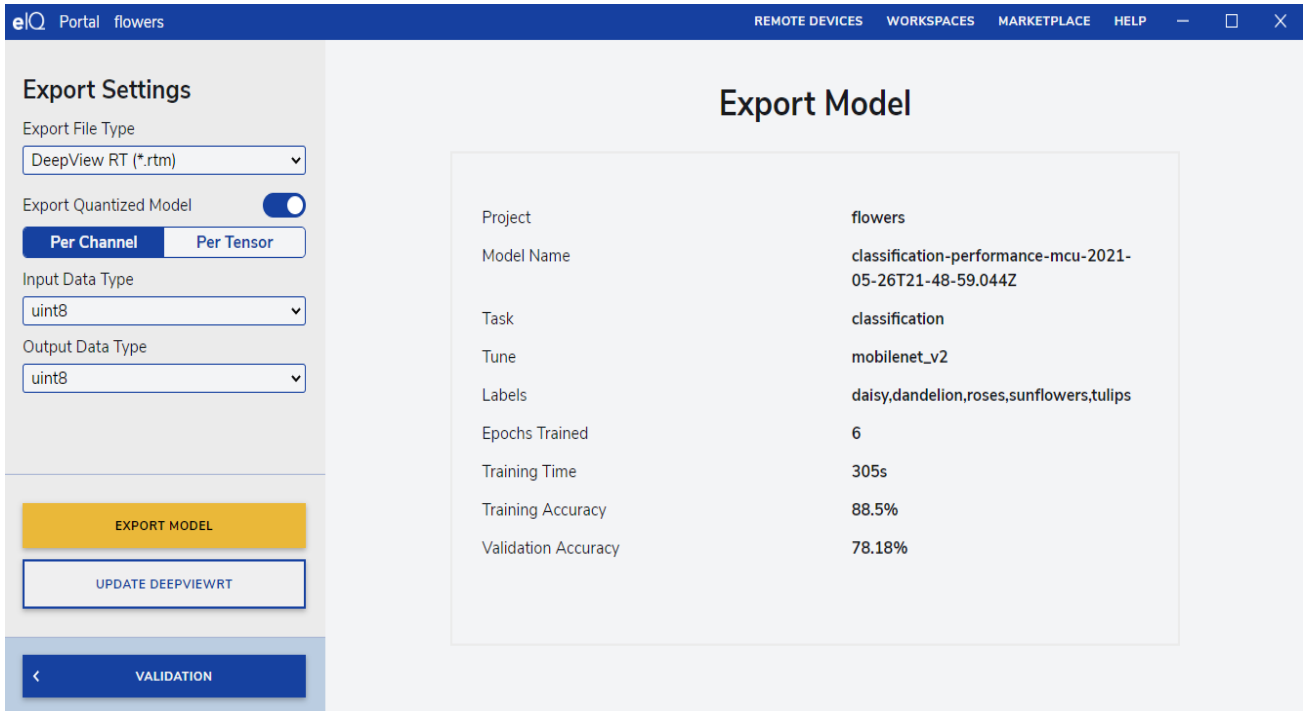
# VALIDATION QUANTIZATION

- Can also see easily effects of post-training quantization
- Select “Per Channel” or “Per Tensor” and Input data type



# eIQ PORTAL – DEPLOY

- Export the resulting model in DeepViewRT, TFLite, or ONNX formats
- Can quantize model before exporting
- Compatible with all the eIQ Inference Engines:
  - DeepViewRT Inference Engine
  - TensorFlow Lite for MPU
  - ONNX Runtime



# eIQ MODEL TOOL

- The **eIQ Model Tool** is used for the analysis of your already trained models including model and per-layer time profiling.

- It support BYOM path.
  - It Support Model conversion
  - It Support Model Quantization
  - It Support Per-Layer time profiling.

## eIQ COMMAND LINE TOOLS

- The eIQ Command line tools is command line interface for Advance users.
- eIQ Command-line Tools which also include a self-contained Python environment
- In order to use eIQ Command-line Tools, it is important either to launch them using the COMMAND LINE button from the Home screen or run <eIQ_Toolkit_install_dir>/bin/eiqenv.bat script which sets up the command-line environment.

# i.MX 8M Plus SoC

SECURE CONNECTIONS
FOR A SMARTER WORLD

# i.MX 8M PLUS FAMILY OF APPLICATIONS PROCESSORS

The i.MX 8M Plus family of processors is based on Arm® Cortex®-A53 and Cortex-M7 cores and delivers a new level of:

- **Machine Learning and Vision System**
- **Advanced Multimedia**
- **Industrial Networking and High Reliability**.

It is well suited for applications as:

- **Smart Home, Building, Retail and City**
- **Smart Factory and Industry IoT**


**Machine Learning & Vision, Advanced Multimedia, Industrial IoT**

# i.MX 8M PLUS KEY FEATURES

## High-Performance Power-Efficient

**High-Performance**

- Dual/Quad-core Cortex-A53 cores up to 1.8 GHz;
- Cortex-M7 up to 800MHz (task offload, power optimizations)
- 3D GPU and VPU enables efficient video and display
- DDR3L, DDR4, LPDDR4 (Inline ECC)

**Power-Efficiency**

- Dynamic Voltage Frequency Scaling (DVFS), power gating, clock gating.
- Built in 14nm FinFET LPC technology for low-power & high-performance

## Machine Learning, Vision and Voice

**Machine Learning**

- Neural Network Accelerator up to 2.3TOPS

**Vision System**

- Camera (up to 2 cameras):
- 2x MIPI-CSI (4 lanes each, 1080p)
- Camera ISP: 2x187MPix or 1x375MPix scale, de-warp

**Low-Power Voice**

- Low Power Voice Accelerator

## Advanced Multimedia

**Video:**

- 1080p60 video decoding (H.265, H.264, VP9, VP8)
- 1080p60 video encoding (H.265, H.264)
- 2D and 3D GPU

**Audio:**

- 18x I2S TDM (32-bit @ 768KHz),
- DSD512,
- SP/DIF Tx + Rx
- 8-ch PDM Mic input
- HDMI 2.0b Tx + eARC
- ASRC
- 8ch PDM DMIC input for voice capture

## Connectivity and Interfaces

**Display Interfaces**

- 1x MIPI-DSI
- 1x HDMI 2.0b Tx (+eARC)
- LVDS (4/8-lane) Tx
- Up to 3 display simultaneously

**High Speed Interfaces**

- 3x SDIO 3.0 for boot / storage / Wi-Fi (max flexibility)
- 1x PCIe 3.0 to connect to high-performing Wi-Fi solutions and other systems
- 2x Gigabit Ethernet with IEEE 1588, AVB (one with TSN, one with IEEE)
- 2x USB 3.0/2.0 OTG with PHY
- 2x CAN-FD

# i.MX 8M PLUS TARGET APPLICATIONS

## Machine Learning and Industrial Automation

- Machine Vision and Robot Controller
- Industrial Computer, Gateways, HMI
- Printers and Scanners
- Machine Visual Inspection
- Factory Automation

## Smart Home, Building and City

- Safety, Security and Surveillance
- Fleet Analytics and Driver Monitor
- Traffic Monitor and Flow Optimization
- Vision Payment Systems
- Targeted Advertisement
- Service Drones
- Alarm and AI Server Hubs
- Home Patient and Elderly Monitor

Home Gateway

## Consumer and Pro Audio/Voice Systems

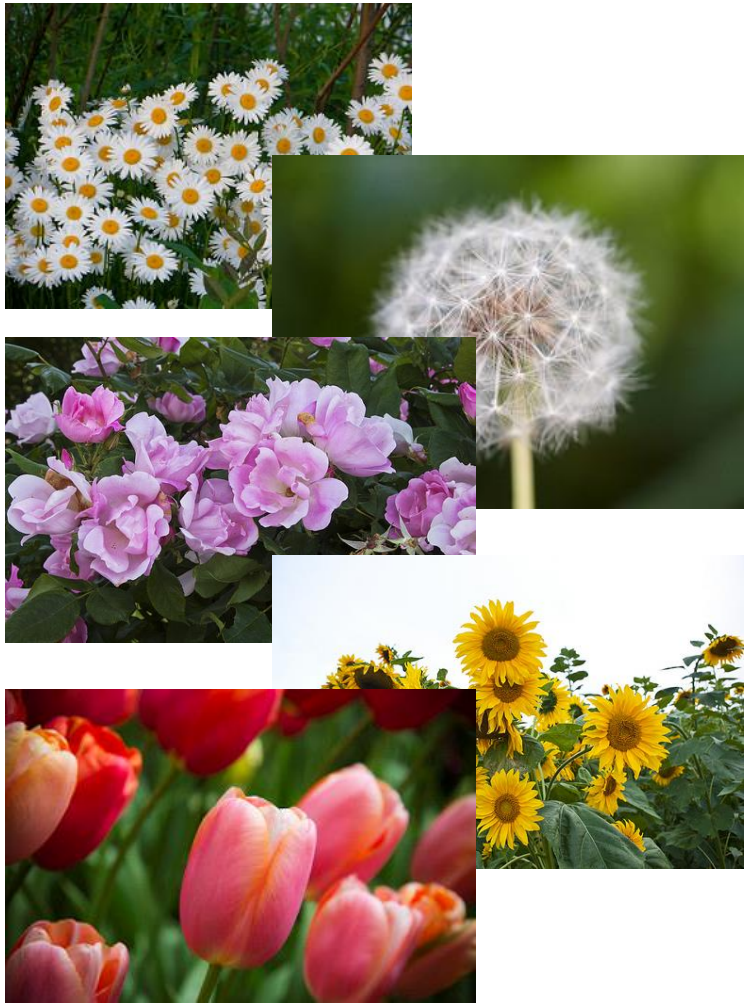
- Surround sound and sound bars
- Audio/video receiver
- Immersive Audio Products
- Wireless or networked smart speakers
- Personal Assistant
- Voice-assisted products

# Lab Overview

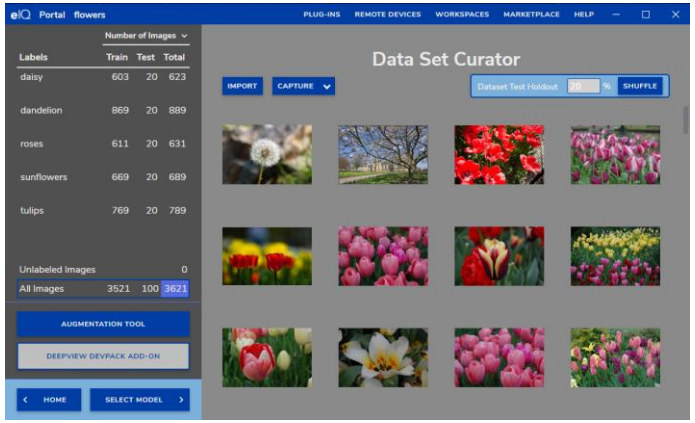
SECURE CONNECTIONS
FOR A SMARTER WORLD

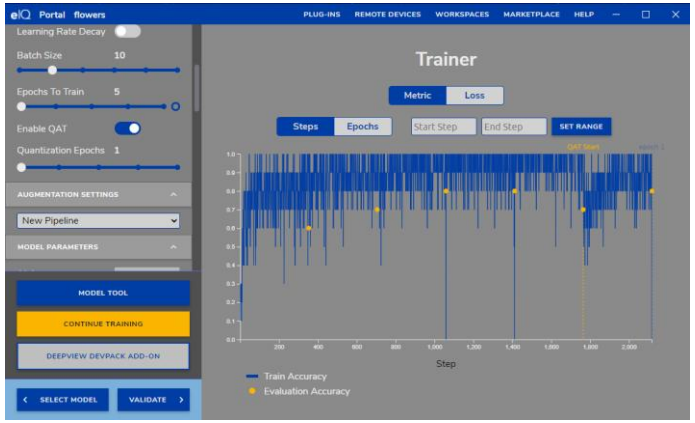
## THE GOAL

We can use a group of labeled photos….



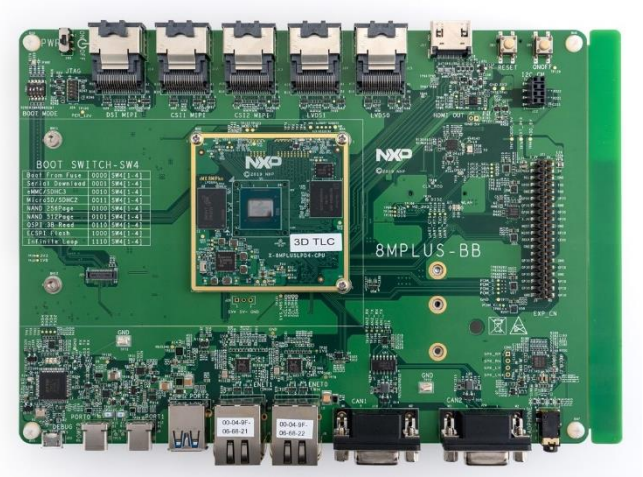
…in the eIQ Toolkit's model creation tool…



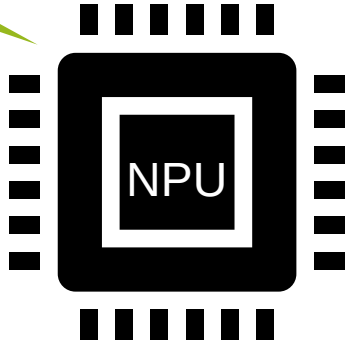

…to get a file that can tell a computer how to identify flowers from other images.

# THE PROCESS




These are **tulips!**



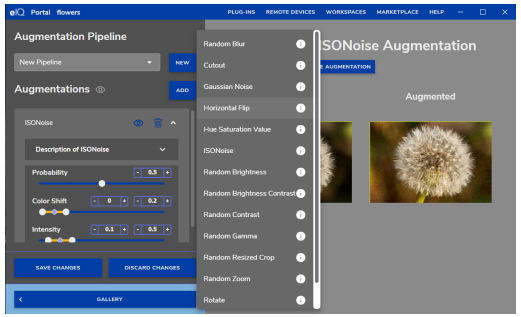
## THE OBJECTIVES

### Section 1
Importing Data into eIQ Portal

- How to import large datasets into eIQ Portal
- Importing data using a script

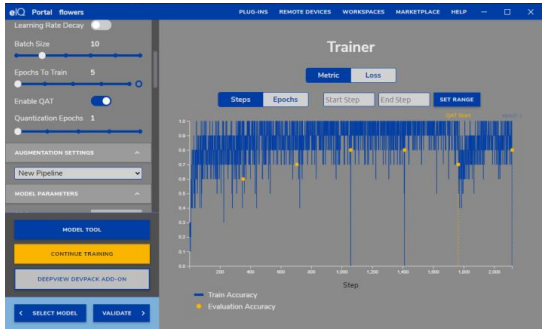
### Section 2
Viewing and Augmenting

- How to change the dataset inside the application
- How to use augmentation pipelines



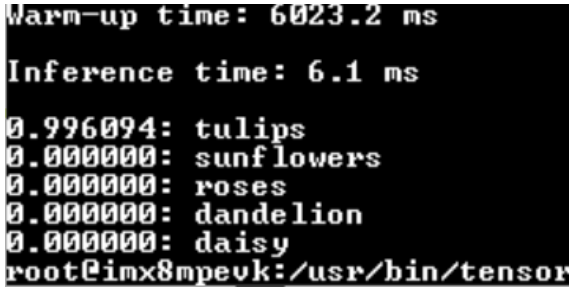

### Section 3
Train the Model

- How to train a model in eIQ Portal
- Creating models for NPU devices




### Section 4
Deploy the Model

- How to test a TFLite model on the i.MX 8M Plus
- How to use the onboard NPU



## ABOUT THIS LAB

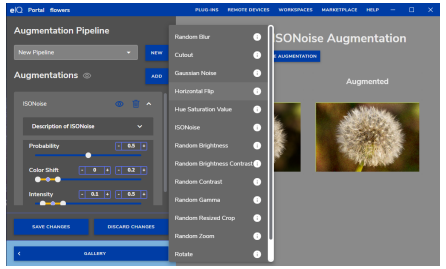
- Follow along using the “Training Image Classification Models for i.MX Applications Processors.pdf” found on the desktop.
- Use the laptop and the i.MX 8M Plus provided for you.
- Each lab section builds on itself. Checkpoint files have been provided in case some participates cannot complete certain sections.
- Feel free to ask questions!

**Section 1**

Importing Data into eIQ Portal

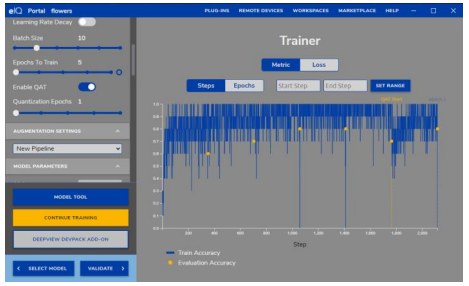
**Section 2**

Viewing and Augmenting



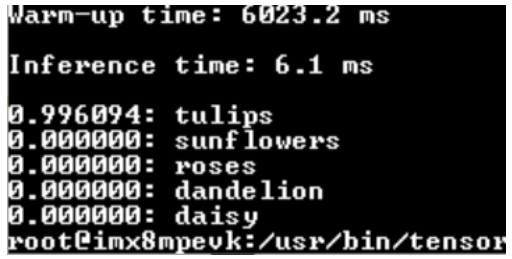

**Section 3**

Train the Model




**Section 4**

Deploy the Model



SECURE CONNECTIONS
FOR A SMARTER WORLD