
IE 417

Embedded AI

Lab: 03: Indoor Classification with Object Detection



Group Name: Friendly Group

Avinash Baraiya- 202201211

Malhar Vaghasiya - 202201183

Smeet Agrawal - 202101237

Priyesh Tandel - 202101222

Samarth Panchal- 202101456

Guided By Prof. Tapas Kumar Maiti

- **Introduction**

This project uses the OV7670 camera module and Arduino Nano 33 BLE Sense to classify indoor scenes, like kitchens and bathrooms, on a microcontroller. By capturing images and applying transfer learning with TensorFlow Lite for Microcontrollers, we'll deploy a compact model for recognizing indoor environments on limited hardware. This lab demonstrates how to bring basic computer vision to small devices.



- **Data Collection**

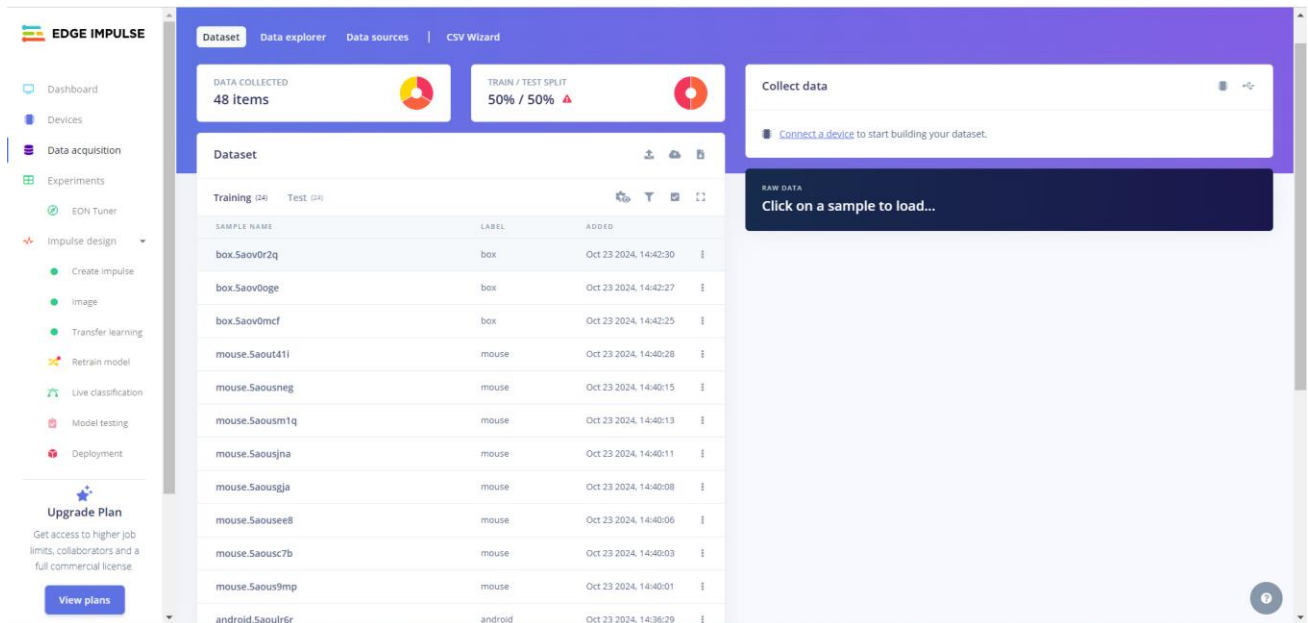
Indoor Classification with Object Detection

In this project, we use the Arduino Nano 33 BLE Sense and OV7670 camera module for indoor scene classification, focusing on detecting specific objects like a box, an Android device, and a mouse. Using TensorFlow Lite for Microcontrollers (TFLu), we apply transfer learning to classify these objects in indoor environments, enabling microcontrollers to perform basic object detection in real-world settings.

Steps for Data Collection:

- **Sensor Data:** Images were captured with the OV7670 camera module.
- **Objects Detected:** Three objects were identified for classification—box, Android device, and mouse.
 - **Box:** Identified to signal the presence of storage items.
 - **Android Device:** Detected to differentiate personal devices in a scene.
 - **Mouse:** Used to recognize workspace items, aiding in workspace classification.

This project highlights how to capture and classify indoor scenes using limited resources on microcontrollers.

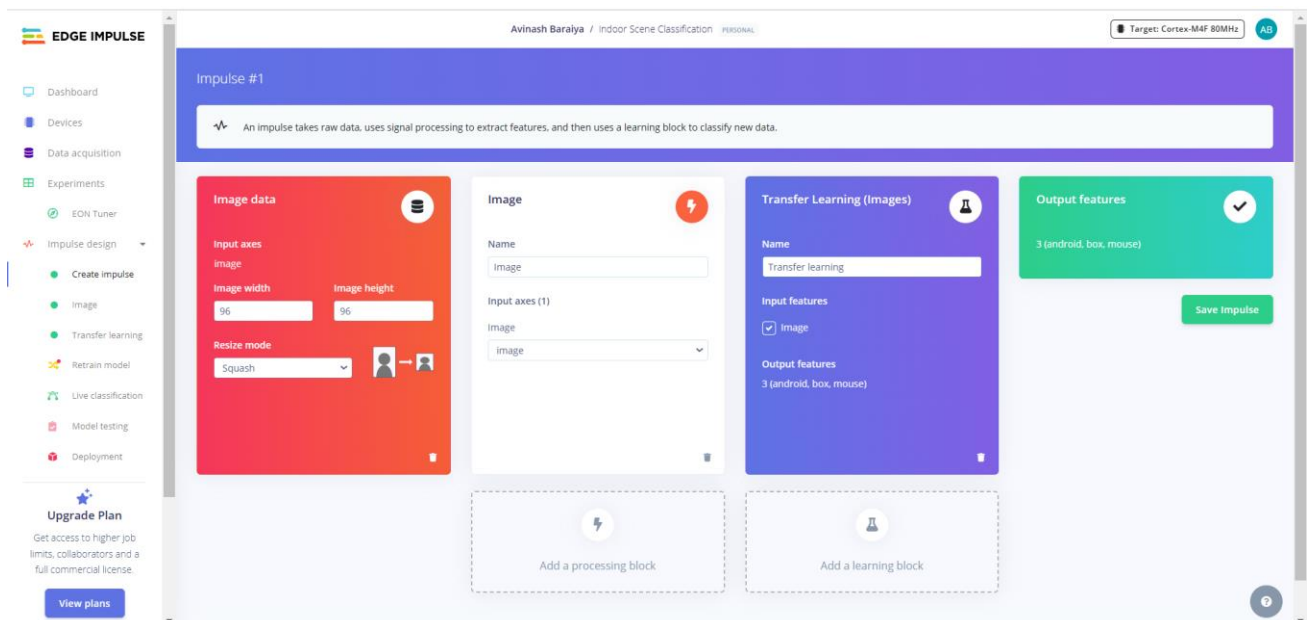


- **Impulse Creation**

Learning Block: Transfer Learning(images)

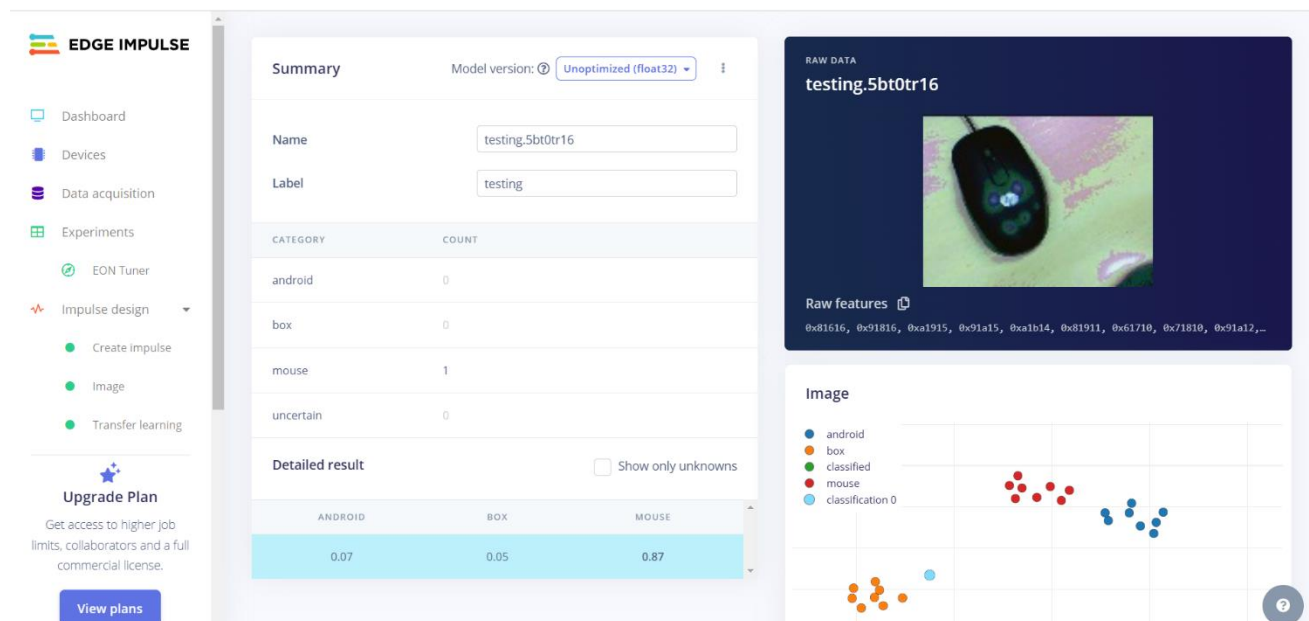
Input Features: Image

Output Features: Android | Box | Mouse

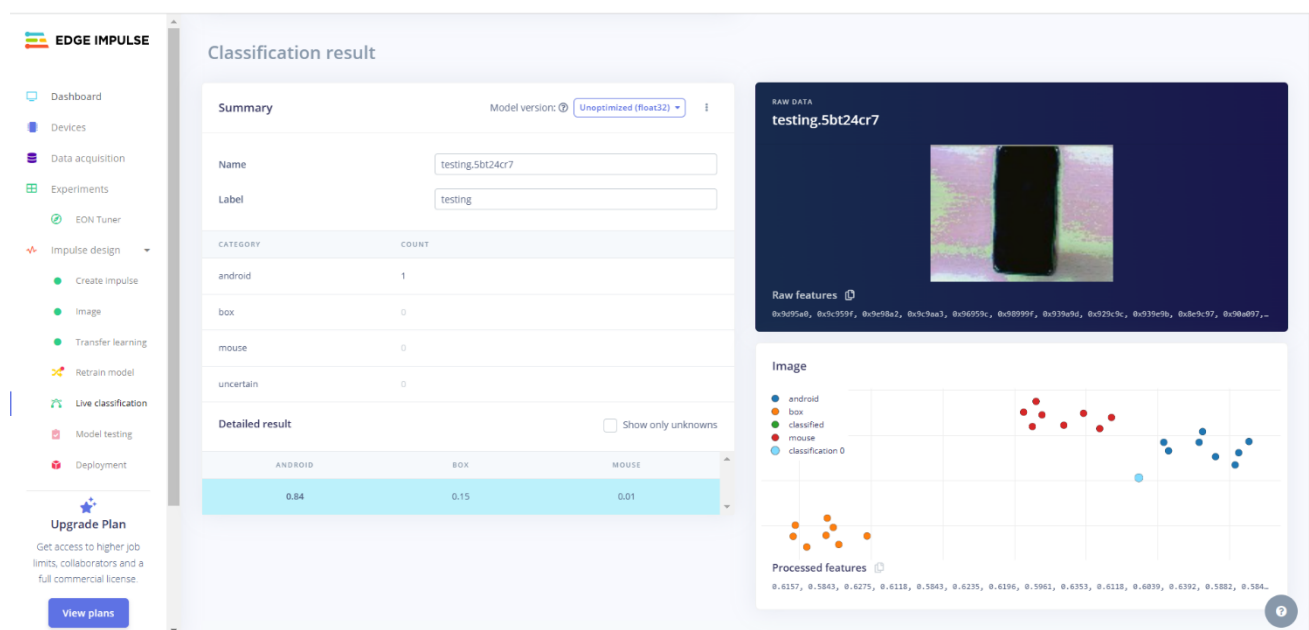


• Classification and Testing Results

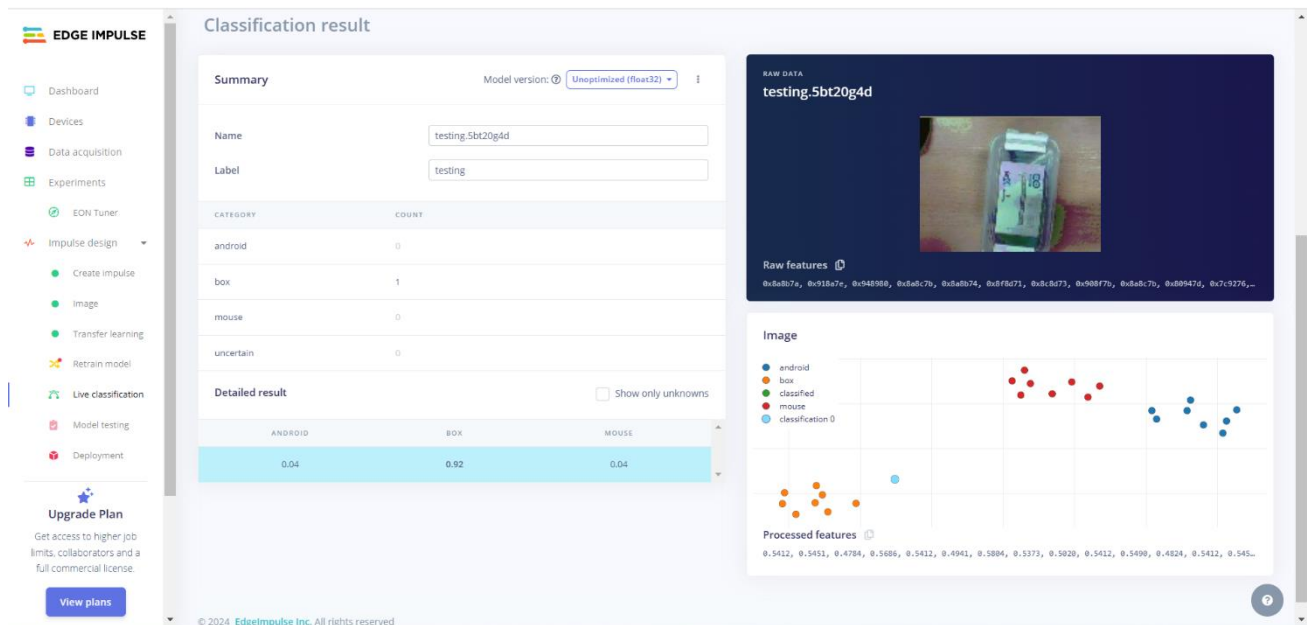
1. Mouse Sample Taken by Arduino and it correctly Predicted with value **0.87** that it is **Mouse**



2. Mobile Sample Taken by Arduino and it correctly Predicted with value **0.84** that is **Android**

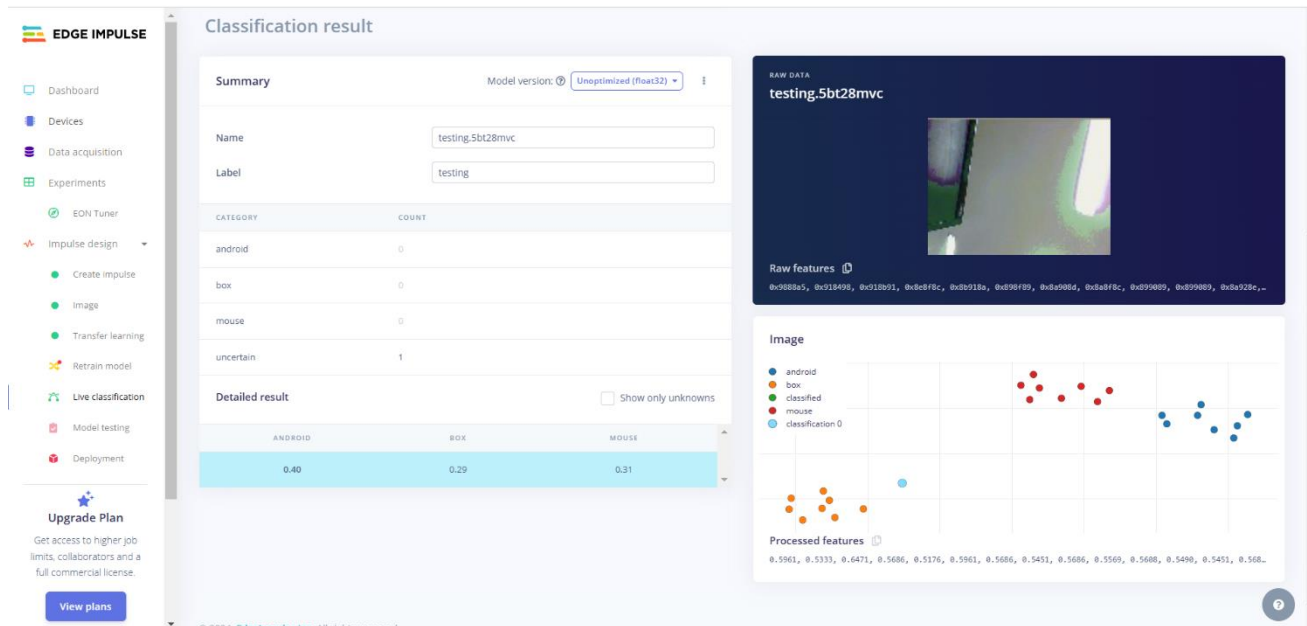


3. Box Sample Taken by Arduino and it correctly Predicted with value **0.92** that is **Box**



4. Random Sample Taken by Arduino and it correctly Predicted **Uncertain**.

Android - 0.40 | Box - 0.29 | Mouse - 0.31



Testing Data:

Test data

Classify all

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	ACCURACY	RESULT
box.5aouulg2	box	100%	1 box
box.5aouujnr	box	100%	1 box
box.5aouug5j	box	100%	1 box
box.5aouub7c	box	100%	1 box
box.5aouuacs	box	100%	1 box
mouse.5aoutkvh	mouse	100%	1 mouse
mouse.5aoutisg	mouse	100%	1 mouse
mouse.5aoutgdr	mouse	100%	1 mouse
mouse.5aoutel4	mouse	100%	1 mouse
mouse.5aoutdkv	mouse	100%	1 mouse

Model testing output

Model version: Unoptimized (float32)

Results

ACCURACY
91.27%

Metrics for Transfer learning

METRIC	VALUE
Area under ROC Curve	1.00
Weighted average Precision	1.00
Weighted average Recall	1.00
Weighted average F1 score	1.00

Confusion matrix

	ANDROID	BOX	MOUSE	UNCERTAIN
ANDROID	100%	0%	0%	0%
BOX	0%	93%	7%	0%
MOUSE	0%	12%	88%	0%
F1 SCORE	1.00	1.00	1.00	

Confusion matrix

	ANDROID	BOX	MOUSE	UNCERTAIN
ANDROID	100%	0%	0%	0%
BOX	0%	93%	7%	0%
MOUSE	0%	12%	88%	0%
F1 SCORE	1.00	1.00	1.00	

Feature explorer



- **Code Implementation**

<https://drive.google.com/drive/folders/1WiqxxtDulSYp1AO4rW0d3ulTBMM56kAC?usp=sharing>

- **Practical Video Link**

Part 1: https://youtu.be/fFQpoj-ub_g

Part 2: <https://www.youtube.com/shorts/BTtuHPeYFQE>