

EDGE INTELLIGENCE LABORATORY 5

MACSE604

**DANISH RYWEN T
25MML0048**

EDGE IMPULSE

1.Click Data Acquisition:

Navigate to the Data Acquisition section, where all sensor and image data collection is managed.

2.Select Connect to device:

Choose the option to connect a device for data collection, which allows real-time image capture.

3. Use mobile phone and scan the QR code:

Select the mobile phone option and scan the displayed QR code using your phone's browser to establish a connection.

4.Connect the mobile phone as a camera:

Once connected, the mobile phone camera acts as an input device for capturing images directly into Edge Impulse.

5.Capture sample images:

Use the phone camera to capture images under different lighting and angles to improve model robustness.

6.Collect images for training and testing:

Capture a sufficient number of images and split them into training and testing datasets for proper model evaluation.

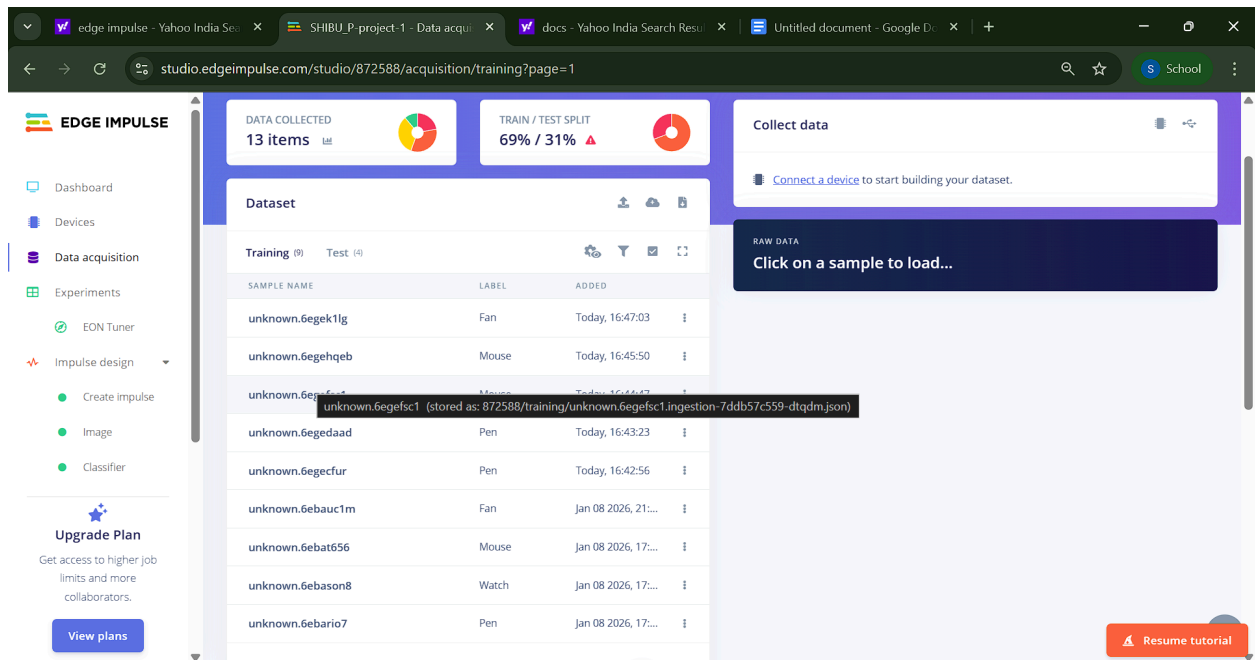
7.Rename images with meaningful names:

Assign clear and descriptive filenames to images to help identify classes and improve dataset organization.

8.Label all images correctly:

Apply accurate labels to each image so the model can correctly learn and distinguish between different classes.

Training Data with size if Nine (9):



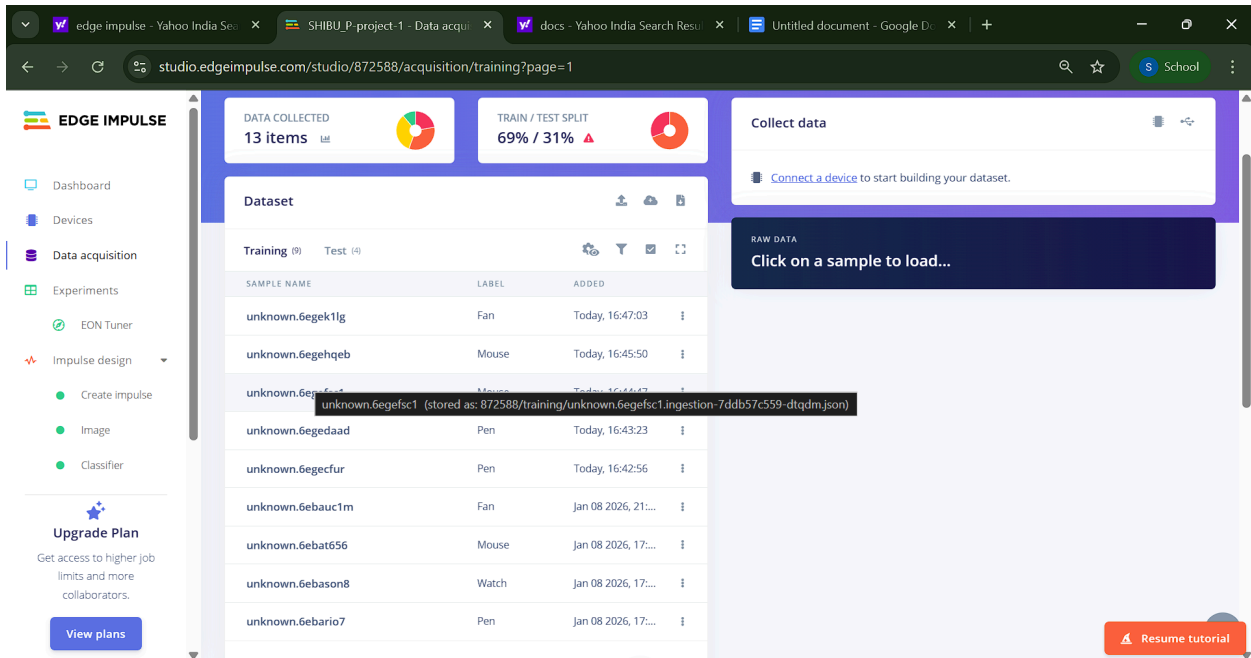
The screenshot displays the Edge Impulse Studio interface for a data acquisition project. The top navigation bar shows the project name 'SHIBU_P-project-1 - Data acquisition' and the current page 'studio.edgeimpulse.com/studio/872588/acquisition/training?page=1'. The main content area is divided into several sections:

- DATA COLLECTED:** 13 items.
- TRAIN / TEST SPLIT:** 69% / 31%.
- Collect data:** A section with a 'Connect a device' button to start building the dataset.
- Dataset:** A table showing the training and test sets. The training set contains 9 items, and the test set contains 4 items.
- RAW DATA:** A section with a 'Click on a sample to load...' button.

The dataset table is as follows:

SAMPLE NAME	LABEL	ADDED
unknown.6egek1lg	Fan	Today, 16:47:03
unknown.6egehqeb	Mouse	Today, 16:45:50
unknown.6egefcsc1	Pen	Today, 16:43:23
unknown.6egedaad	Pen	Today, 16:42:56
unknown.6ebauc1m	Fan	Jan 08 2026, 21:...
unknown.6ebaf656	Mouse	Jan 08 2026, 17:...
unknown.6ebason8	Watch	Jan 08 2026, 17:...
unknown.6ebario7	Pen	Jan 08 2026, 17:...

Testing Data with size of Four:



The screenshot displays the Edge Impulse Studio web interface. On the left is a sidebar with navigation options: Dashboard, Devices, Data acquisition, Experiments, EON Tuner, and Impulse design. The 'Impulse design' section is expanded, showing options to 'Create impulse', 'Image', and 'Classifier'. An 'Upgrade Plan' button is also visible. The main area shows a 'Dataset' section with a table of 13 items. The table has columns for 'SAMPLE NAME', 'LABEL', and 'ADDED'. The items are categorized into 'Training (9)' and 'Test (4)'. A tooltip is visible over the item 'unknown.6egefc1', showing its full path: 'unknown.6egefc1 (stored as: 872588/training/unknown.6egefc1.ingestion-7ddb57c559-dtqdm.json)'. On the right, there's a 'Collect data' section with a 'Connect a device' button and a 'RAW DATA' section with a 'Click on a sample to load...' button. A 'Resume tutorial' button is at the bottom right.

SAMPLE NAME	LABEL	ADDED
unknown.6egek1lg	Fan	Today, 16:47:03
unknown.6egehqeb	Mouse	Today, 16:45:50
unknown.6egefc1	Pen	Today, 16:43:23
unknown.6egefc1	Pen	Today, 16:42:56
unknown.6ebauc1m	Fan	Jan 08 2026, 21:...
unknown.6ebata56	Mouse	Jan 08 2026, 17:...
unknown.6ebason8	Watch	Jan 08 2026, 17:...
unknown.6ebario7	Pen	Jan 08 2026, 17:...

1.Open Impulse Design and click Create Impulse.

Go to the Impulse Design section and click “Create Impulse” to define the data flow for model training.

2.Add a Processing Block as Image.

Select the Image processing block to convert raw image data into features suitable for learning.

3.Add a Learning Block as Image Classification.

Choose the Image Classification learning block to enable the system to classify images into predefined categories.

4.Click Save Impulse.

Save the impulse configuration so the processing and learning blocks

are applied to the dataset.

5.In Impulse Design, click on Image.

Open the Image processing page to configure parameters such as image size and color depth.

6.Click Generate Features to extract image features.

Generate features from the uploaded images, transforming visual information into numerical representations.

7.After feature generation is complete, go to Image Classification (Classifier).

Navigate to the classifier section to prepare the model for training using the extracted features.

8.Click Start Training.

Start the training process, where the model learns patterns from the labeled image data.

9.Once training is finished, the model accuracy will be displayed for the images you uploaded.

The platform shows performance metrics such as accuracy, indicating how well the trained model classifies the images.

10.Test with live classification:

Try using sample test data

edge impulse - Yahoo India Se...Profile - Projects - Edge Impulse - docs - Yahoo India Search Resu...+studio.edgeimpulse.com/studio/profile/projects

EDGE IMPULSE

S

ProjectsCustom ML blocks

S

SHIBU_P

DEVELOPER

Enable MFA

Multi-factor authentication is now available for all users. [Set up now.](#)

Organizations

Allow anyone on your team to collaborate on multiple datasets, automation, and models in a shared workspace.

Request Enterprise Trial

Projects

Sort + Create new project

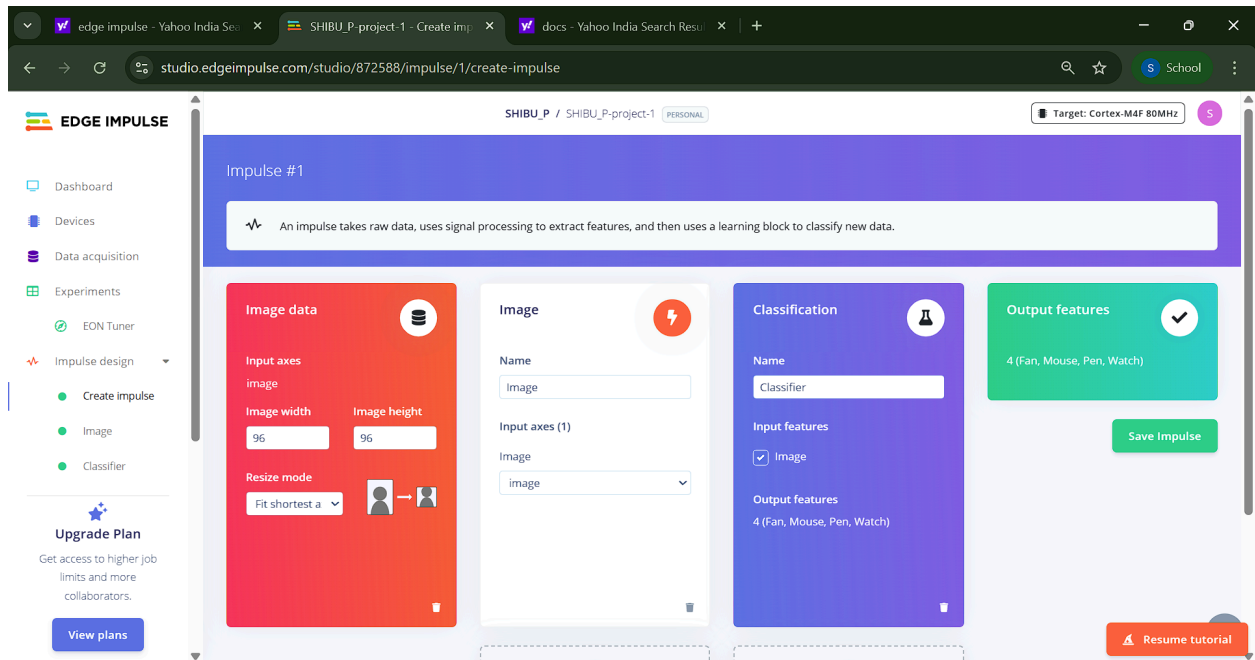
2 of 3 private projects remaining. Want access to more? [Request Enterprise Trial.](#)

S

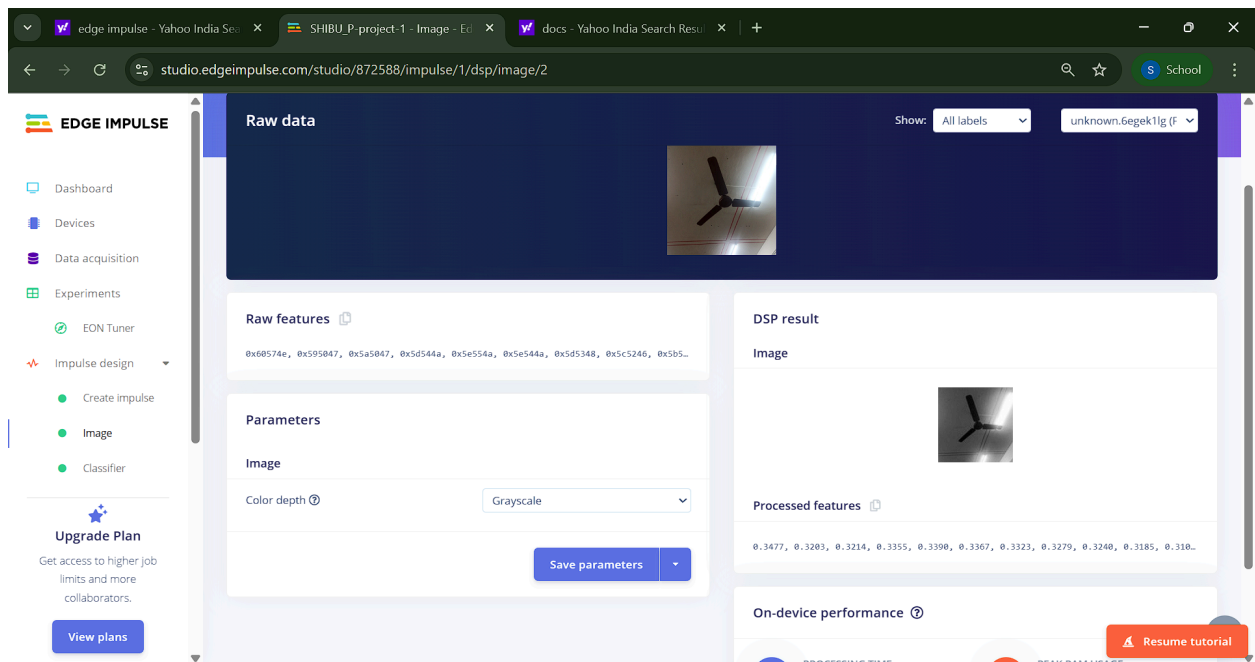
SHIBU_P / SHIBU_P-project-1

?

Creating Impulse:



Changing into grayscale image:



Generate features:

The screenshot shows the Edge Impulse Studio web interface at the URL `studio.edgeimpulse.com/studio/872588/impulse/1/dsp/image/2/generate-features`. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design, and an Upgrade Plan section.
- Top Bar:** Displays the 'Parameters' tab and the 'Generate features' button.
- Training set:** A table showing the data used for training:

Parameter	Value
Data in training set	9 items
Classes	4 (Fan, Mouse, Pen, Watch)
- Feature generation output:** A section showing the results of the feature generation process, currently displaying 0 items.
- Feature explorer:** A scatter plot visualizing the features for the four classes: Fan (blue), Mouse (orange), Pen (green), and Watch (red). The plot shows distinct clusters for each class.
- On-device performance:** A section for monitoring the performance of the model on a device, with a 'Resume tutorial' button.

Classifier with two Convolution and pooling layer with dropout and flatten layer and achieved an accuracy of 50 %:

The screenshot shows the Edge Impulse Studio interface for a project named "SHIBU_P / SHIBU_P-project-1". The "Neural Network settings" panel is active, displaying training parameters: 10 training cycles, learning rate of 0.0005, and CPU as the training processor. The "Neural network architecture" section shows a sequence of layers: an input layer with 9,216 features, followed by two "2D conv / pool" layers (16 and 32 filters respectively), a "Flatten layer", a "Dropout (rate 0.25)" layer, and an "Output layer (4 classes)". The "Training output" panel on the right shows the last training performance on the validation set: Accuracy of 50.0% and Loss of 1.40. A confusion matrix is also displayed, showing a 50% accuracy for the "MOUSE" class. Metrics for the validation set include Weighted average Precision of 1.00 and Weighted average Recall of 0.50.

Neural Network settings

Training settings

- Number of training cycles: 10
- Use learned optimizer: ☐
- Learning rate: 0.0005
- Training processor: CPU

Advanced training settings

Neural network architecture

- Neural network
- Transfer learning

Input layer (9,216 features)

2D conv / pool layer (16 filters, 3 kernel size, 1 layer)

2D conv / pool layer (32 filters, 3 kernel size, 1 layer)

Flatten layer

Dropout (rate 0.25)

Add an extra layer

Output layer (4 classes)

Training output

Model version: Quantized (int8)

Last training performance (validation set)

- ACCURACY: 50.0%
- LOSS: 1.40

Confusion matrix (validation set)

	FAN	MOUSE	PEN	WATCH
FAN	-	-	-	-
MOUSE	0%	50%	50%	0%
PEN	-	-	-	-
WATCH	-	-	-	-
F1 SCORE		0.67	0.00	

Metrics (validation set)

METRIC	VALUE
Weighted average Precision	1.00
Weighted average Recall	0.50

[Resume tutorial](#)

The screenshot shows the Edge Impulse Studio interface for the same project. The "Neural network architecture" panel is active, displaying the sequence of layers: an input layer with 9,216 features, followed by two "2D conv / pool" layers (16 and 32 filters respectively), a "Flatten layer", a "Dropout (rate 0.25)" layer, and an "Output layer (4 classes)". The "Metrics (validation set)" panel on the right shows the last training performance: Weighted average Precision of 1.00, Weighted average Recall of 0.50, and Weighted average F1 score of 0.67. A "Data explorer" panel shows a scatter plot of the full training set, with points colored by class and correctness. The "On-device performance" panel shows the engine used (EON Compiler) and the performance metrics: Inference time of 587 ms, Peak RAM usage of 182.7K, and Flash usage of 106.1K.

Input layer (9,216 features)

2D conv / pool layer (16 filters, 3 kernel size, 1 layer)

2D conv / pool layer (32 filters, 3 kernel size, 1 layer)

Flatten layer

Dropout (rate 0.25)

Add an extra layer

Output layer (4 classes)

Metrics (validation set)

METRIC	VALUE
Weighted average Precision	1.00
Weighted average Recall	0.50
Weighted average F1 score	0.67

Data explorer (full training set)

- Fan - correct
- Mouse - correct
- Pen - correct
- Watch - correct
- Mouse - incorrect

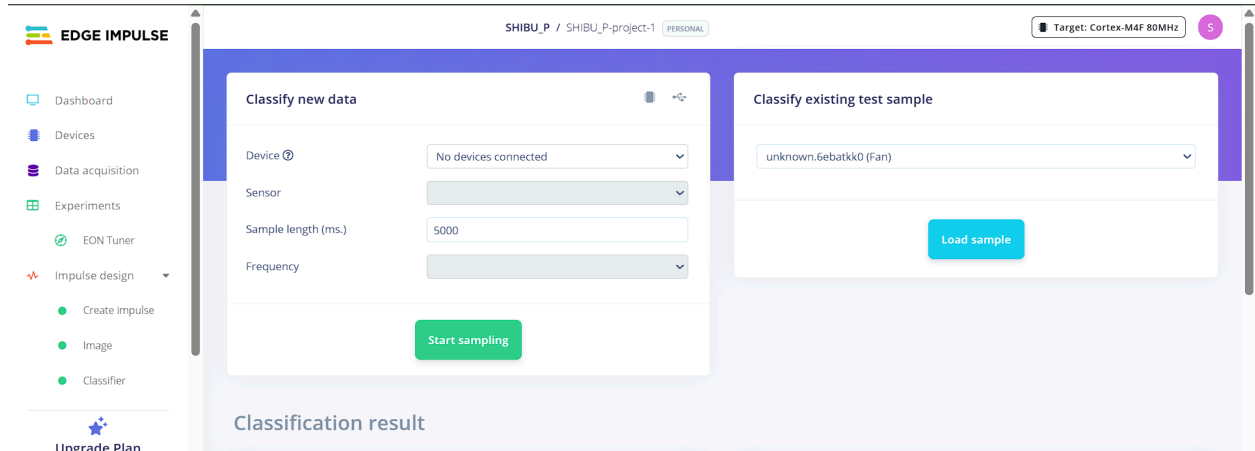
On-device performance

Engine: EON Compiler

- INFERENCE TIME: 587 ms.
- PEAK RAM USAGE: 182.7K
- FLASH USAGE: 106.1K

[Resume tutorial](#)

Testing with (Fan) image:



The model predicted an highest probability as Fan:

