**<<Air Piano: Real-Time Finger Detection for Virtual Piano Playing>>**

*Sudhansu\_24177, DM, sudhansu2024@iisc.ac.in*

*Vaibhav\_24112, DM, gvaibhav@iisc.ac.in*

*Siddharth\_24857, DM, siddhartha1@iisc.ac.in*

*Proff. Pandarasamy Arjunan, RBCCPS, samy@iisc.ac.in*

**Overview**

**A drawing of a person playing a piano

AI-generated content may be incorrect.**Instead of playing a heavy, expensive instrument like a piano, you can wave your hands in the air to create sounds!

Leveraging the Nicla Vision board, we built a system to detect finger movements and turn them into music.

we developed a FOMO-based model to detect these gestures in real time, achieving a validation F1 score of 78.1%. To suit the Nicla Vision’s limited resources, the model was optimized and compressed into a lightweight 56 KB trained.tflite file.

Our pipeline translates detected finger positions into 8 virtual keys, sending serial commands to a connected computer to produce piano sounds.

This project showcases the successful integration of creativity, machine learning, and embedded systems, delivering an innovative and engaging user experience directly at the edge.

A hand holding a fist next to a sign

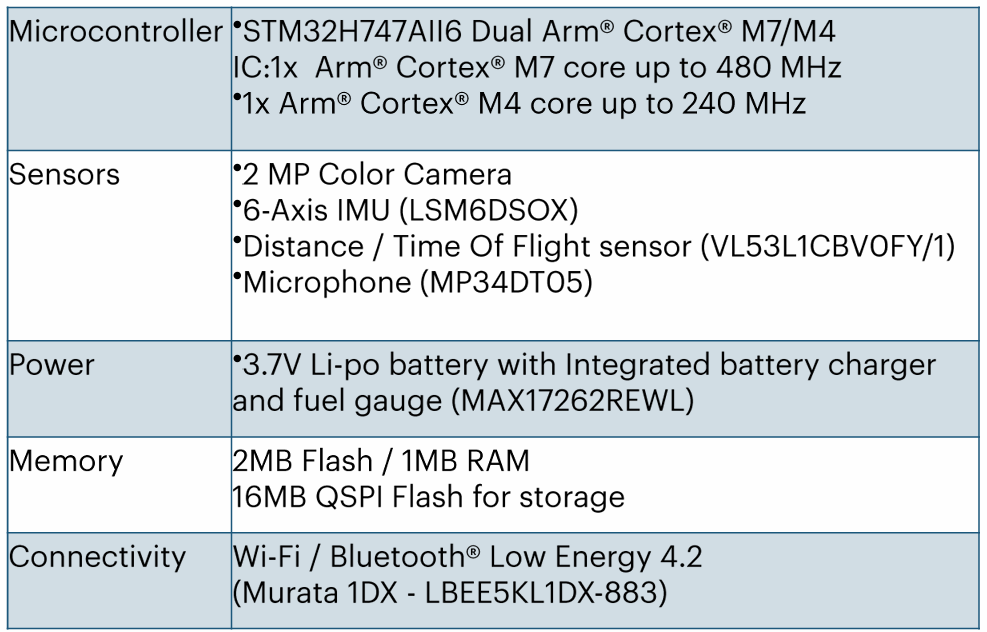
AI-generated content may be incorrect.**Background and Motivation**

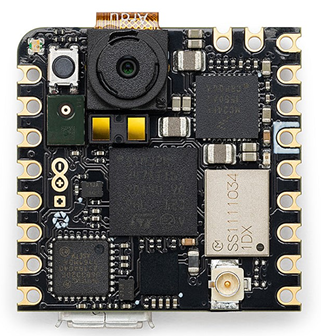
No heavy stuff to carry! Big instruments like pianos are hard to move around. We wanted   to make music easy by letting you play with just your hands in the air.​

Won’t cost a lot! Some instruments are super expensive. Our idea makes music fun and affordable for everyone.​

Super fun and friendly!Playing a piano just with our fingers dancing in the air! That’s the magic we wanted to bring to life.

**Methodology**

1. Hardware Requirements and Specifications



Nicla Vision

2. Software Used

**Edge Impulse Studio/Google Colab**: For training and deploying the FOMO(Object Detection) model used to detect finger position.

**OpenMV IDE**: For programming the Nicla Vision and integrating the model with custom logic.

**Python**: Used for preprocessing images and linking finger gestures to virtual keys, enabling real-time sound playback via serial communication.

3. Working Principle

1. **Camera Initialization**  
   The Nicla Vision camera is set up to continuously capture image frames in real time.
2. **Image Preprocessing**  
   Each frame is cropped and resized to match the FOMO model's input requirements.
3. **A finger pointing at something

   AI-generated content may be incorrect.Running the Model**  
   The trained FOMO model identifies the "index" finger within the image, returning:
   * Label
   * Confidence Score
   * Bounding Box (x, y, width, height)

**Interpreting Results**  
The bounding box area is calculated (width × height) to estimate the finger's distance:

* + Larger area: Finger is closer to the camera (finger up – release)
  + Smaller area: Finger is farther away (finger down – press)

A diagram of a keyboard and hand

AI-generated content may be incorrect.

1. **Mapping to Piano Keys**
   * Image width (e.g. 240 px) is divided into 8 zones of 30 pixels each.
   * The center of the bounding box (center\_x = x + width / 2) determines which key zone the finger lies in:  
     key = center\_x // 30
2. **Detecting Press Events**  
   By comparing bounding box area changes across frames, the system infers when a press or release occurs.
3. **Continuous Loop**  
   This detection and mapping run in a continuous loop, enabling real-time interaction with the piano.

**Data collection**

Data was collected using an **Arduino Nicla Vision board** mounted 35 cm above a tabletop. The device captured video of an index finger interacting with a virtual piano interface. This 3-minute video was split into individual frames, resulting in approximately **683 images**.

These images were processed using **Edge Impulse**, where:

* **546 images** were used for training.
* **137 images** were reserved for validation.

The dataset was labeled into three classes:

* **Index finger** (with bounding box annotations).
* **Fist** (also with bounding box annotations).
* **Background** (unannotated images where no hand is visible), implicitly included as negative samples.

A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

Manual annotation was performed using Edge Impulse's interface to mark the position and dimensions of the finger and fist in each frame.

To enhance model generalization, **data augmentation** was performed using **Google Colab**, applying transformations such as rotation, scaling, and flipping. After augmentation, the dataset was expanded to a total of **1689 images**, ready for training computer vision models.

**Model development and compression**

The objective of the model development phase was to create an efficient object detection system capable of running on a highly resource-constrained embedded device—Nicla Vision. The task involved detecting where a finger is present in the camera frame.

**Data Preprocessing and Initial Model (Google Colab)**

The dataset consisted of images initially captured in RGB format at a resolution of 320×240 pixels. These images were down sampled and converted to grayscale with a 48×48 resolution to reduce computational load and memory usage during model training.

A simple neural network model was trained using this grayscale input. Before quantization, the model had a size of **351.41 KB** and achieved an accuracy of **74.34%**. However, for deployment on the Nicla Vision, the model was quantized to the int8 format to reduce size. After quantization, the model size was reduced to **96.40 KB**, with a trade-off in accuracy which dropped to **55.75%**.

**Optimized Model using Edge Impulse (FOMO - MobileNetV2)**

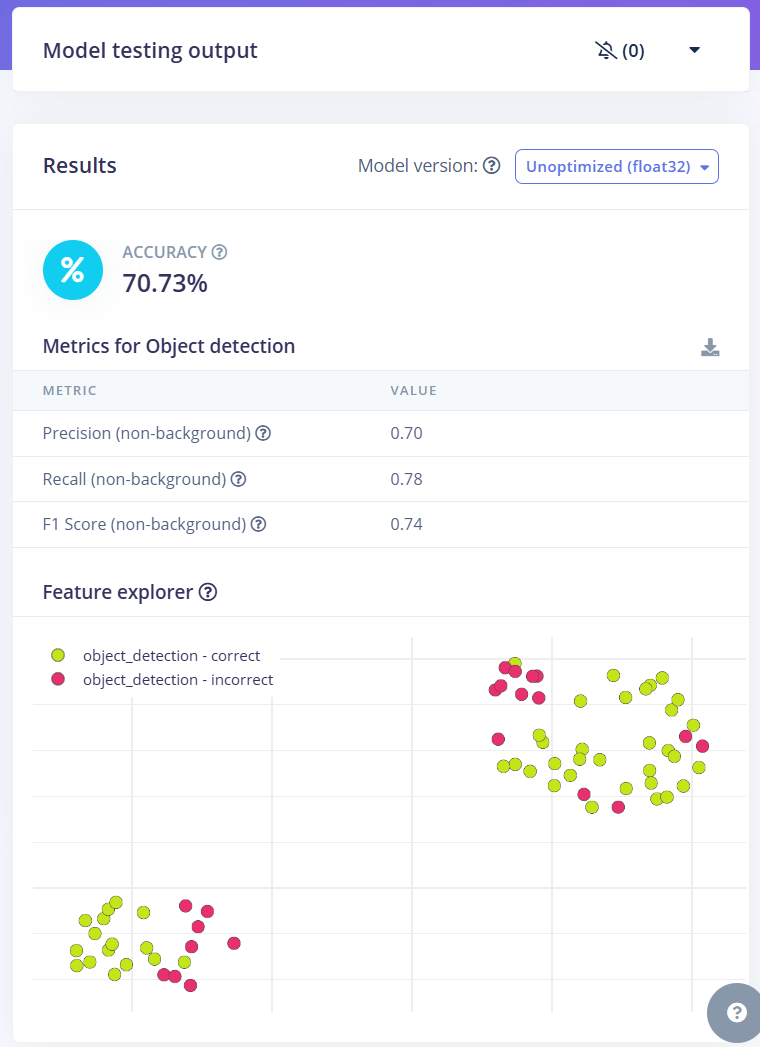
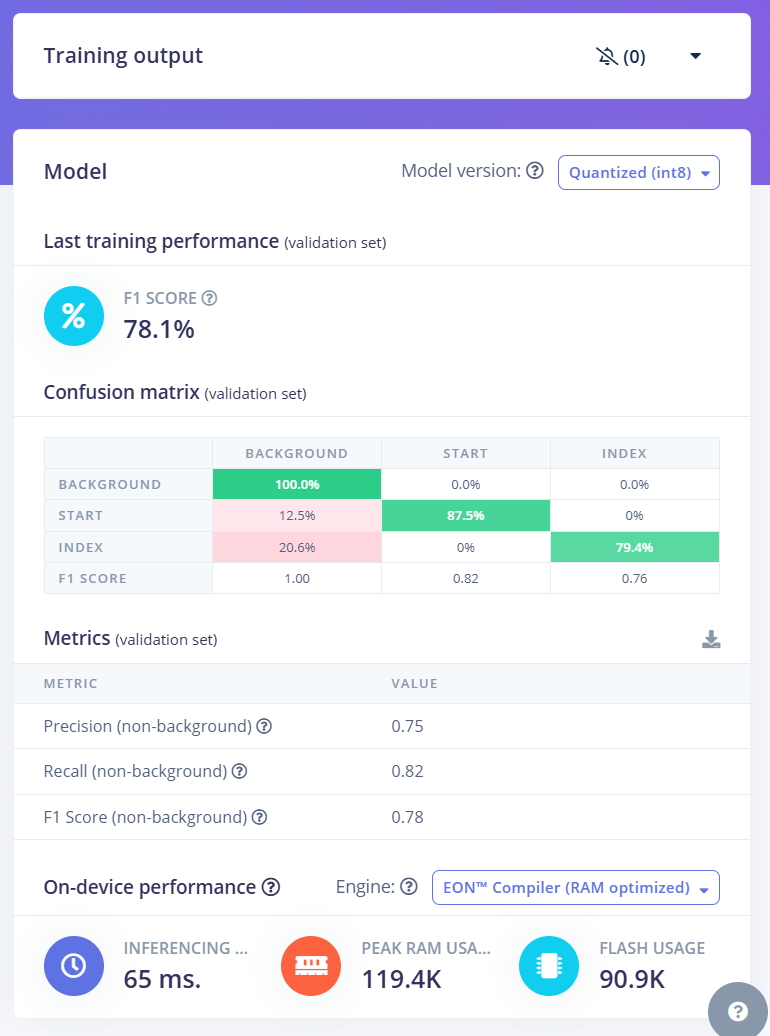
To further optimize performance, the model was retrained using Edge Impulse’s **FOMO (Faster Objects, More Objects)** architecture based on MobileNetV2 (0.35). This model architecture is designed for low-latency, memory-efficient object detection, suitable for microcontrollers.

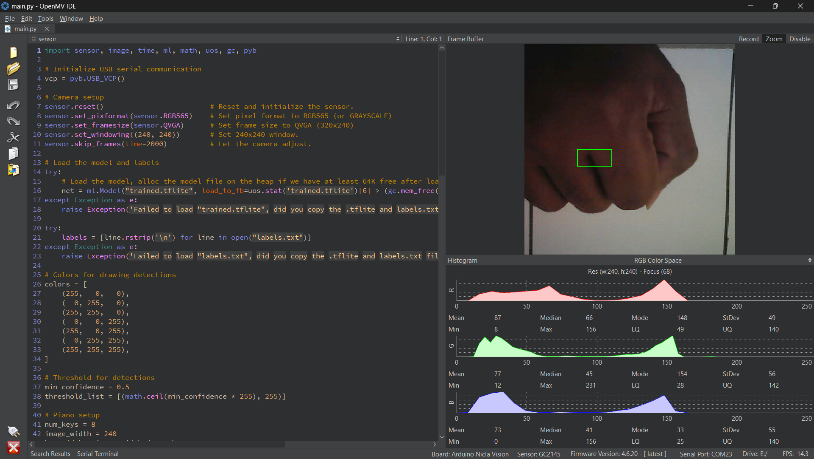
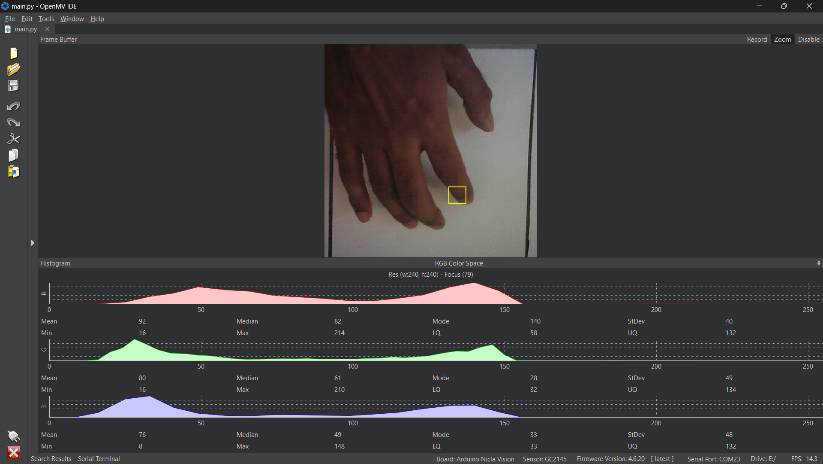
* **Input layer**: 96×96 grayscale image (9,216 features)
* **Output layer**: 2 classes (Finger, Fist)
* **F1 Score (Validation)**: 78.1%
* **Precision (non-background)**: 75%
* **Recall (non-background)**: 82%
* **Inference time**: 65 ms

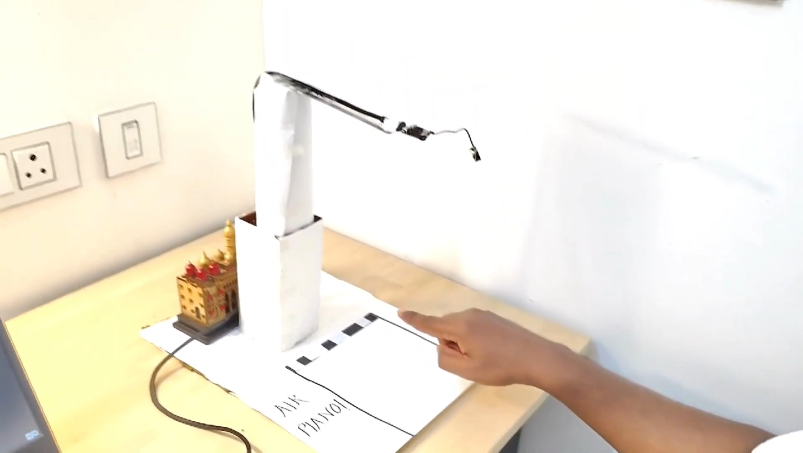
**Final Deployed Model**

The final model was exported as a **TFLite file** (trained.tflite) with a compact size of just **56.0 KB**. During testing, it achieved:

* **Accuracy**: 70.73%
* **Precision**: 70%
* **Recall**: 78%
* **F1 Score**: 74%



**Prototype and Demo**

****

**Project Resources:**

Github link: <https://github.com/btech10423/Air-Piano-Real-Time-Finger-Detection-for-Virtual-Piano-Playing.git>

youtube video link: <https://youtu.be/-nB3dS6YtFM>

**Challenges and Workarounds**

**Gesture Detection Struggles**: The model initially confused fists and fingers, especially in different lighting. We fixed this by adding more varied images and using data augmentation, boosting accuracy to a 78.1% F1 score.

**Slow Response Time**: The piano lagged on the Nicla Vision, so we scaled images to 96x96, used grayscale, and optimized the model to 56 KB, making it fast enough for real-time play.

**Finding the Right Threshold**: It took iterative trial and error to set the perfect threshold index finger area for playing and releasing piano keys, ensuring the system accurately detected finger presses.

**IMPORTANT LEARNING**

**Believe on your idea! We changed our idea 3 times, to arrive at a final one.  
Don’t be rigid on your idea be flexible. Initially our idea was very wage and unrealistic then with brainstorming with our team we figure out ways with time and ideas were clicking. And finally, we arrived at our final concept.**

References:

understanding object detection with FOMO:

<https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>

Video explanation: <https://www.youtube.com/watch?v=VzJZM5p24Tc&pp=ygUjZm9tbyBtb2RlbCBvYmplY3QgZGV0ZWN0aW9uIGVkZ2UgYWk%3D>