University of Applied Sciences
HOCHSCHULE
EMDEN·LEER

Manual

Machine Vision Using Portenta H7

Author: Vatsal Mahajan (7025694)
Author: Manoj Selvaraju (7025649)
Author: Vijay Singh (7025700)
Course of Studies: Department of Electrical and Computer Science

First examiner: Prof. Dr. Elmar Wings
Submission date: January 31, 2025

University of Applied Sciences Emden/Leer · Faculty of Technology · Electrical and Computer Science Department
Constantiaplatz 4 · 26723 Emden · http://www.hs-emden-leer.de

Contents

List of Figures

List of Tables

1 Introduction

Welcome to the Face Recognition for Access Control System Manual. This system uses the Arduino Portenta H7 and Vision Shield, along with Edge Impulse Studio, to deliver an efficient and accurate face recognition solution for access control. Our system designed to ensure security and reliability using MobileNet V2 algorithms and it is ideal for applications in offices, restricted zones and other areas requiring controlled access.

This manual provides a step-by-step guide to understanding, setting up and using the face recognition system. Whether you are setting up for personal use or professional applications, this manual will help you utilize the system's functionalities efficiently.

1.0.1 Key Features

- **Portable and Energy-Efficient:** Powered by the Arduino Portenta H7, the system is compact, energy-efficient and easy to set up, making it perfect for a variety of applications.
- **Customizable User Profiles:** Add or remove face profiles effortlessly to keep the access control system up-to-date with changing needs.
- **Easy Deployment:** Deploy trained models seamlessly on the Portenta H7, ensuring quick and hassle-free integration with the hardware.
- **Scalable for Any Setup:** Whether it's a small office or a large facility, the system can easily scale to handle varying requirements. Updating datasets and retraining models is simple and efficient.
- **Clear and Insightful Visualizations:** Get a better understanding of data and model performance through detailed, easy-to-interpret visualizations.

1.0.2 How to Use This Manual

This manual is structured to provide a comprehensive guide to using the Face Recognition for Access Control System. Below is an overview of the key sections and how they can assist you:

- **Introduction:** Learn about the system's purpose, key features and overall workflow.

- **Main Functionality:** Get a clear understanding of how the system processes data, trains models and recognizes faces in real time.
- **Installation and Configuration:** Follow detailed instructions to set up the hardware and software, install necessary tools and configure the system for use.
- **Getting Started:** Learn how to prepare datasets, connect the hardware, power the system, and run the pre-installed model.
- **Maintenance:** Follow guidelines to maintain hardware and update software for optimal performance.
- **Safety Guidelines:** Steps to ensure safe operation with precautions for hardware, environment and data security.
- **Troubleshooting:** Get tips to resolve some common hardware and software issues with clear solutions.

1.0.3 Basic Info on the System

The system uses the MobileNet V2 algorithm, a convolutional neural network optimized for embedded vision applications. MobileNet V2 is known for its ability to deliver high accuracy with reduced model size, making it ideal for deployment on resource-constrained devices like the Portenta H7 hardware. The CAD model of the system is listed as shown in the figure 1.1. ALso, the GUI of the system will be available as shown in the figure 1.2.

We are confident that our system will provide you with the tools and insights needed to effectively monitor and provide access control. If you have any questions or need further assistance, our support team is always ready to help. Let's take a step towards smarter and more secure access control with our Face Recognition System, ensuring a seamless blend of technology and security for your environment!

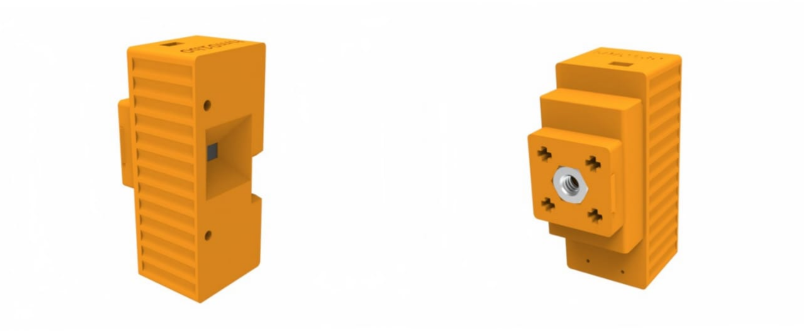
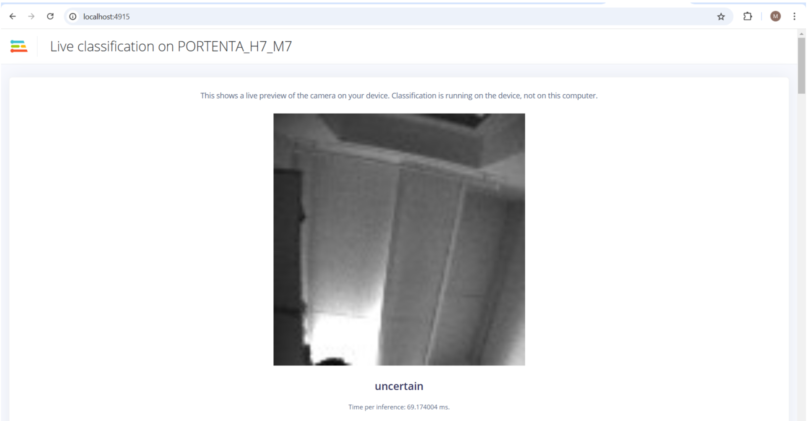


Figure 1.1: CAD Model of the system



Live classification on PORTENTA_H7_M7

This shows a live preview of the camera on your device. Classification is running on the device, not on this computer.

uncertain

Time per inference: 69.174004 ms.

Figure 1.2: GUI of the application

2 Main Function

2.1 User Interaction and Workflow

The Face Recognition for Access Control System is designed to provide a seamless and user-friendly experience. Below are the key functionalities and how users can interact with the system:

2.1.1 User Registration

Adding a New User:

To add a new user, the system administrator or authorized user can use the system's interface to capture the user's face image. The user simply needs to capture images from the Vision Shield camera and do the model retraining along with redeployment as mentioned in the report.

Profile Management:

Each user profile can be labeled with a unique identification name for easy management. The system allows administrators to add, update or delete user profiles as needed.

2.1.2 Access Control

Real-Time Recognition:

When a user approaches the access point, the system automatically captures their face image using the Vision Shield camera. The captured image is compared against the registered profiles in real-time.

Access Granting/Denial:

If the system recognizes the user and matches their face with a registered profile, access is granted. If no match is found, access is denied and an alert can be sent to the administrator for further action.

2.1.3 System Feedback

GUI Interface:

The system provides immediate feedback through a graphical user interface (GUI) hosted on the local host. The GUI displays whether the person is identified or not, along with relevant details such as the Person name and the inference time. This interface is accessible to administrators for real-time monitoring and decision-making.

2.2 Model details:

The system leverages the MobileNet V2 algorithm optimized for edge devices like the Portenta H7. This ensures fast and accurate face recognition while maintaining low power consumption. The model is trained using Edge Impulse Studio, which allows for easy customization and retraining based on specific user requirements.

3 List Of Parts

3.1 Hardware Parts

Please refer to Table 3.1 for the system deliverable part details as shown in figure 3.1.

Title	**Version**	**Description**	**Source**
Laptop	Windows 11	Lenovo 15.6 Inch Full HD Notebook	Amazon
Portenta H7 Board	Latest	Development Board	Arduino
Vision Shield	OV5640 Camera	Camera module for Portenta H7	Arduino
USB Cable	USB C	Cable for connecting Portenta H7 to laptop	Amazon
Micro SD Card	8 GB	Storage for Vision Shield	Amazon

Table 3.1: Hardware List

3.2 Software Parts

Please refer to Table 3.2 for software requirement for the system.

Title	**Version**	**Description**	**Source**
Arduino IDE	2.3.2	IDE for Arduino boards	arduino.cc
Edge Impulse CLI	1.15.4	CLI for Edge Impulse models	edgeimpulse.com
Python	3.10	Python programming language	python.org

Table 3.2: Software Details

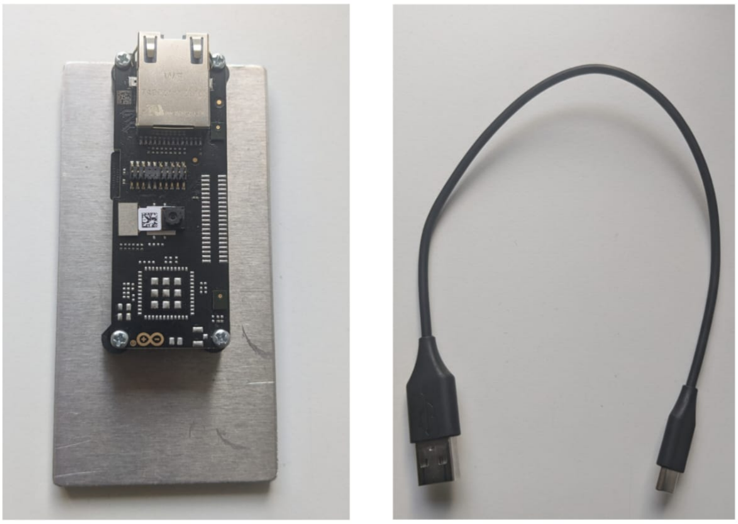


Figure 3.1: Hardware parts - Portenta H7, Vision Shield and USB-C cable

4 System Setup and Configuration

This chapter provides detailed steps to set up the hardware and software required for the Face Recognition Access Control System. Follow the instructions carefully to ensure a smooth installation, configuration and debugging process.

4.1 Software Requirements

- **Python 3.10 or latest version:** Required for running scripts and managing data preprocessing or additional configurations.
- **Arduino CLI:** Essential for managing Arduino board configurations, libraries, and code deployment via the command line.
- **Edge Impulse CLI:** Required for running the model in debug mode and streaming real-time results.

4.2 Installation Steps

1. **Python Installation:**
   - Download the latest version of Python (3.10 or above) from https://www.python.org.
   - Run the execution file and follow the installation instructions for your operating system.
   - During installation, check the box to add Python to your system's PATH.
2. **Arduino CLI Installation:**
   - Visit https://arduino.github.io/arduino-cli to download the Arduino CLI for your operating system.
   - Extract the downloaded files to a preferred location.
   - Add the CLI path to your system's PATH variable.
   - Verify installation by running `arduino-cli version` in your terminal or command prompt.
3. **Edge Impulse CLI Installation:**
   - Install Node.js from https://nodejs.org.
   - Open a terminal and run: `npm install -g edge-impulse-cli`.

- Verify installation by running: `edge-impulse-daemon -version`.

4. **Clone the Project Repository:**
   - Clone the software repository containing the project code by running: `git clone https://github.com/project-repository.git`
   - Review the provided files and ensure all dependencies are properly configured.

4.3 Initial Configuration

1. **Python Configuration:**
   - Install the required Python packages by running: `pip install -r requirements.txt`.
   - Ensure all packages are installed successfully.
2. **Arduino CLI Configuration:**
   - Add the Arduino board manager URL by running:
     `arduino-cli config add board_manager.additional_urls https://downloads.ar`
   - Update the board index:
     `arduino-cli core update-index`.
   - Install the Portenta H7 board core:
     `arduino-cli core install arduino:mbed_portenta`.
3. **Edge Impulse Configuration:**
   - If you want to retrain or customize the model via Edge Impulse Studio:
   - Connect the Portenta H7 to your computer via USB.
   - Run the following command to start the daemon: `edge-impulse-daemon`.
   - Log in to Edge Impulse Studio when prompted and link your device.
   - Use Edge Impulse Studio to update or retrain the model with your input dataset, then export it as a file `.zip` for deployment in Arduino CLI.

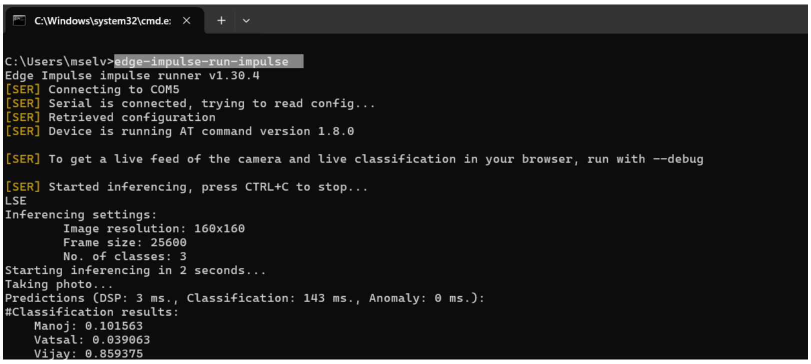
5 Getting Started

5.1 Connecting the Hardware

1. Mount the Vision Shield to the Portenta H7 carefully.
2. Insert the Micro SD card into the Vision Shield.
3. Connect the Portenta H7 to a USB-C power adapter (5V, 2A).
4. Check the LED indicator on the Portenta H7, that indicates the device is powered ON.

5.2 Execute the Model

1. Once the Portenta H7 is powered ON, the pre-installed model will automatically initialize and be ready to run.
2. Open a terminal on your computer and enter the following command to start the inference process:
   ```
   edge-impulse-run-impulse
   ```
   This will run the model on the Portenta H7 and provide real-time recognition results in the terminal as shown in figure 5.1.
3. If you need detailed logs or debugging information, use the debug mode by running the following command:
   ```
   edge-impulse-run-impulse -debug
   ```
   This mode will display additional information, such as inference timing, data acquisition details for visualization as shown in the figure 5.2.

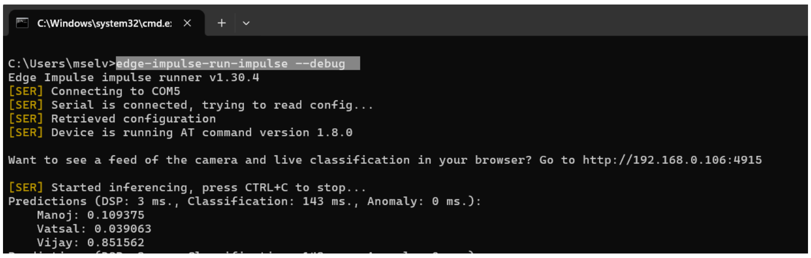


```
C:\Users\mselv>edge-impulse-run-impulse
Edge Impulse impulse runner v1.30.4
[SER] Connecting to COM5
[SER] Serial is connected, trying to read config...
[SER] Retrieved configuration
[SER] Device is running AT command version 1.8.0

[SER] To get a live feed of the camera and live classification in your browser, run with --debug

[SER] Started inferencing, press CTRL+C to stop...
LSE
Inferencing settings:
        Image resolution: 160x160
        Frame size: 25600
        No. of classes: 3
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 3 ms., Classification: 143 ms., Anomaly: 0 ms.):
#Classification results:
    Manoj: 0.101563
    Vatsal: 0.039063
    Vijay: 0.859375
```

Figure 5.1: Running the inference



```
C:\Users\mselv>edge-impulse-run-impulse --debug
Edge Impulse impulse runner v1.30.4
[SER] Connecting to COM5
[SER] Serial is connected, trying to read config...
[SER] Retrieved configuration
[SER] Device is running AT command version 1.8.0

Want to see a feed of the camera and live classification in your browser? Go to http://192.168.0.106:4915

[SER] Started inferencing, press CTRL+C to stop...
Predictions (DSP: 3 ms., Classification: 143 ms., Anomaly: 0 ms.):
    Manoj: 0.109375
    Vatsal: 0.039063
    Vijay: 0.851562
```

Figure 5.2: Debug Mode

6 Maintenance

6.1 Hardware Maintenance

- Regularly check the Vision Shield camera lens to ensure image clarity.
- Inspect all connectors (USB-C, Vision Shield) to ensure they are secure and free from dust or debris.
- Check the Portenta H7 board for any signs of physical damage or overheating during operation.

6.2 Software Updates

- Periodically check for updates to the Arduino CLI and Edge Impulse CLI.
- Re-train the model with new datasets if significant changes in user profiles occur.
- Update firmware for the Portenta H7 if a newer version is available from Arduino.

7 Safety Guidelines

- Use only certified power sources and cables to prevent damage to the Portenta H7 board.
- Keep the system in a clean, dry environment to prevent dust or moisture damage.
- Avoid pointing the camera at reflective surfaces to prevent glare and ensure image clarity.
- Position the system in a secure location to prevent tampering or theft of hardware components.
- Periodically monitor the system for unusual behavior such as unauthorized access attempts.

8 Troubleshooting

In this section, possible errors are listed along with their troubleshooting solutions.

8.1 Hardware Troubleshooting

1. The system does not power ON.
    - Check the connectivity with USB-C cable and verify the power source. Ensure the USB cable is functioning properly.
2. The system camera is not functioning, unable to capture images.
    - Verify that the Vision Shield is properly mounted to the Portenta H7 system. Test the camera with a simple sketch in the Arduino IDE.

8.2 Software Troubleshooting

1. The model fails to deploy using the Arduino CLI.
    - Ensure the correct board core is installed. Recheck the compile and upload commands.
2. The system outputs recognition results with low accuracy rate.
    - Improve lighting conditions and ensure faces are properly aligned with the camera. Re-train the model with more diverse datasets.
3. The Edge Impulse CLI cannot detect the Portenta H7.
    - Restart the device and reconnect it to your computer. Ensure the correct port is selected in the Arduino CLI.
    - Verify that the Edge Impulse CLI is correctly installed.