December 8 2025

Head-Motion Detection for Computer Interfacing

Christian Cherry

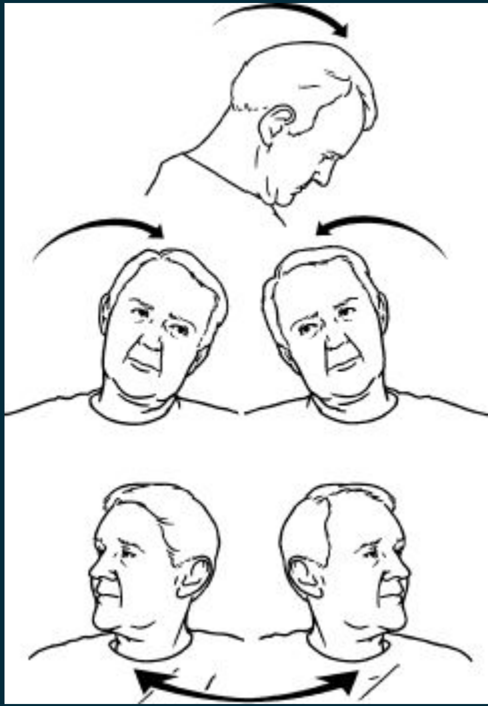
18-444C Embedded Machine Learning

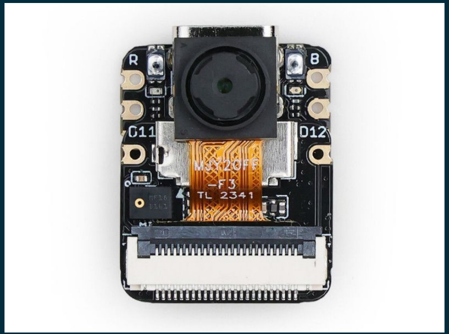
Project Overview

- General use computers and keyboards are commonly made to be used with hands

- Disabled individuals who can't extensively use their hands may have trouble using computers

- **GOAL:** Use Edge AI to detect different head motions of a user to interface with various devices, making for easier use of a computer or application for those who have trouble using their hands



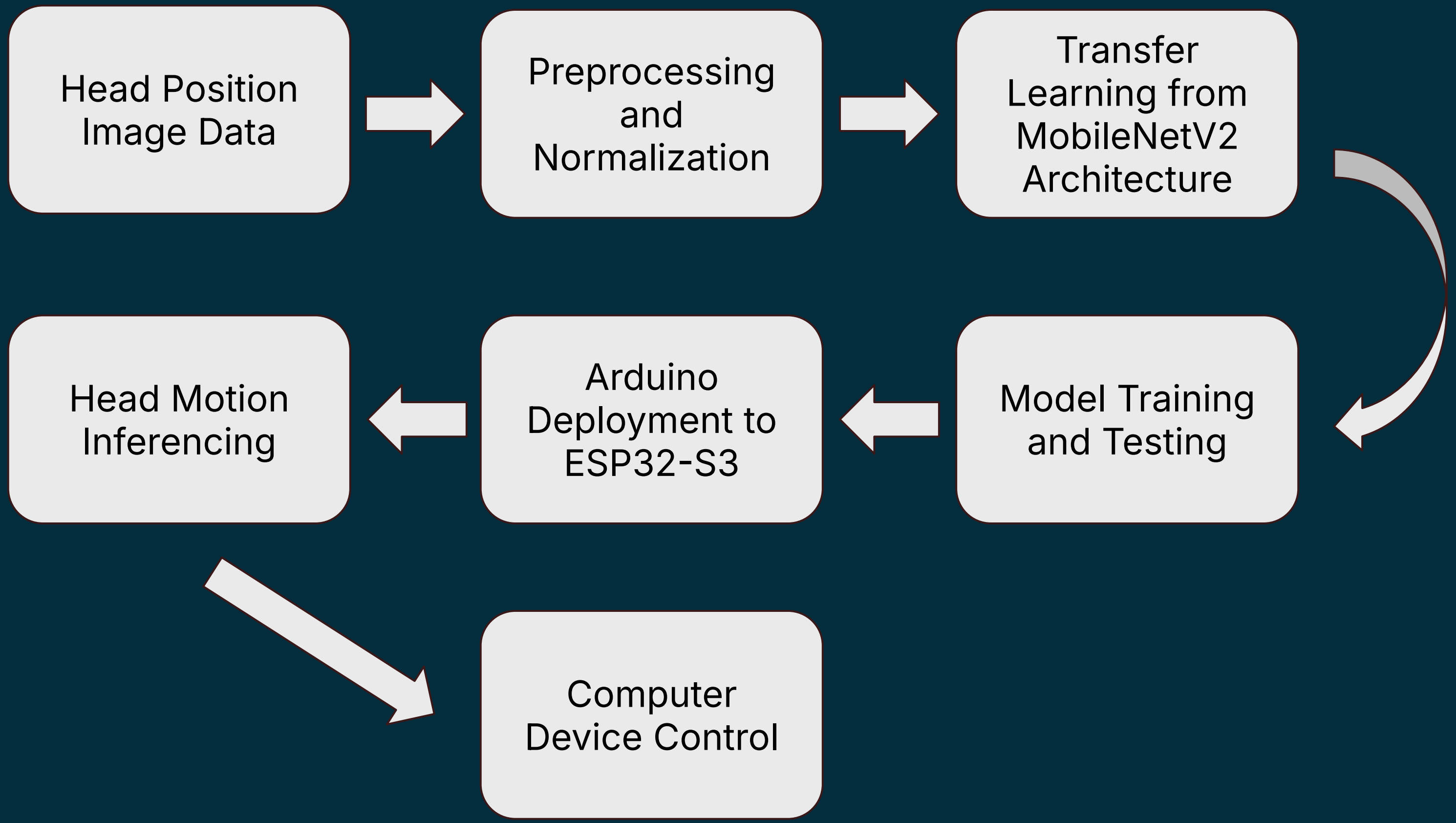


https://iris.hattiesburgclinic.com/patadv/exkit/Geriatric%20Resource%20Library/English/0300000583arth3m_English.html

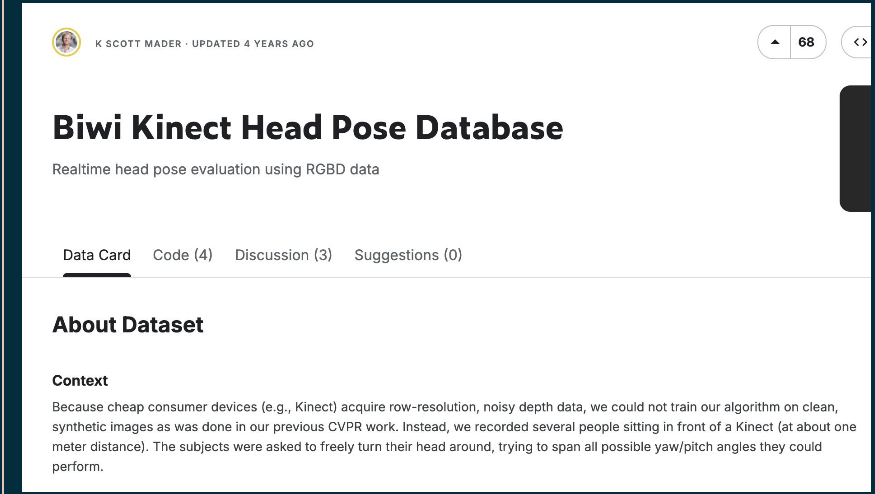
Using Embedded ML for head motion detection allows for an accessible and modular way to control a computer using head gestures.

- *Bandwidth:* A lot less data to process with camera feature extraction versus processing large video data
- *Latency:* Collecting data from a computer camera increases latency versus running an separate on-device model
- *Economics:* The cost of buying an Embedded ML device is more accessible than having to buy an entire computer to use the same feature
- *Reliability:* No reliance on network connection, can be portable and modular to other devices without relying on a specific computer's camera system
- *Privacy:* Having on-device compute without transferring facial data gives users a guarantee that their information is secure

Block Diagram

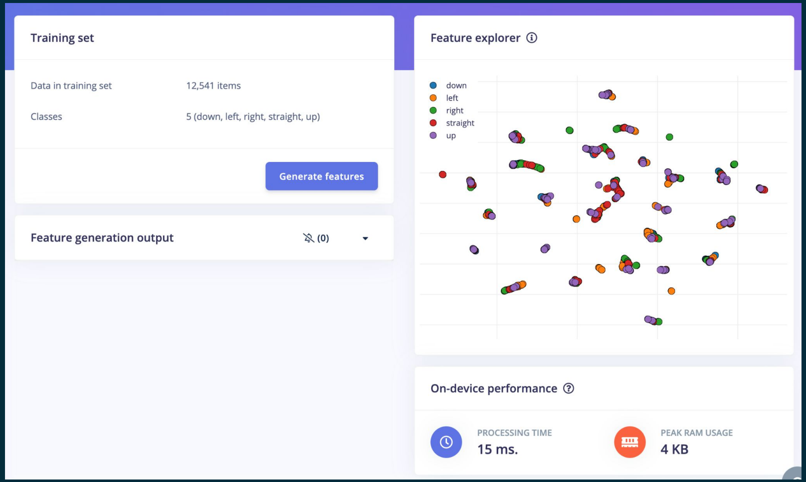


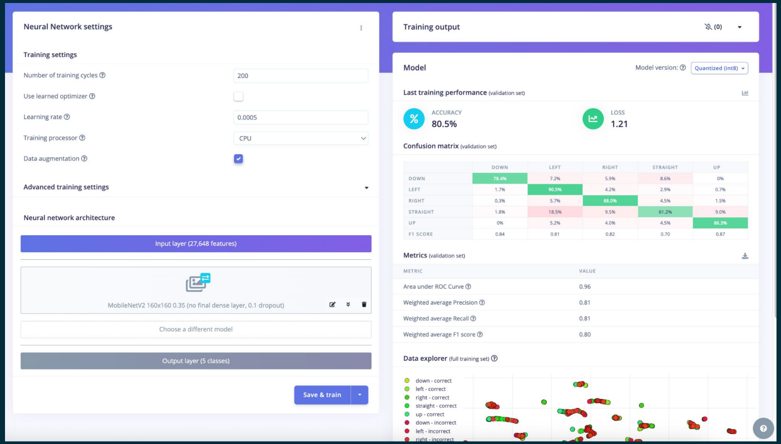
Data Collection



- Head pose estimation dataset from Kaggle.
- Used AI-labeling in Edge Impulse for “up”, “down”, “left”, “right”, and “straight” labels.
- *Challenge:* Manually cleaning up dataset for more accurate labels took quite a while.

Preprocessing an NN Architecture





- Utilized Edge Impulse's built in Image pre-processing
- *Challenge:* I initially used a pre-trained pose estimation block, which had way better accuracy, but this turned out to be too big to fit on my embedded camera
- Utilized the MobileNetV2 160×160 0.35 NN architecture to train model
- *Challenge:* Since only int8 quantization fits on my embedded camera, finding the right balance of efficiency and accuracy with the MobileNet architectures took much trial and error.

Training and Testing Results

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
frame_00082_rgb	left	100%	1 left
frame_00198_rgb	down	100%	1 down
frame_00412_rgb	up	0%	1 uncertain
frame_00257_rgb	up	0%	1 left
frame_00131_rgb	up	100%	1 up
frame_00372_rgb	up	0%	1 uncertain
frame_00615_rgb	up	100%	1 up
frame_00032_rgb	right	0%	1 uncertain
frame_00597_rgb	up	100%	1 up
frame_00203_rgb	down	100%	1 down
frame_00604_rgb	up	100%	1 up
frame_00164_rgb	up	100%	1 up
frame_00256_rgb	up	0%	1 left

Model testing output

(0)

Results

Model version: Quantized (int8)

ACCURACY
76.89%

Metrics for Transfer learning

METRIC	VALUE
Area under ROC Curve	0.96
Weighted average Precision	0.82
Weighted average Recall	0.81
Weighted average F1 score	0.80

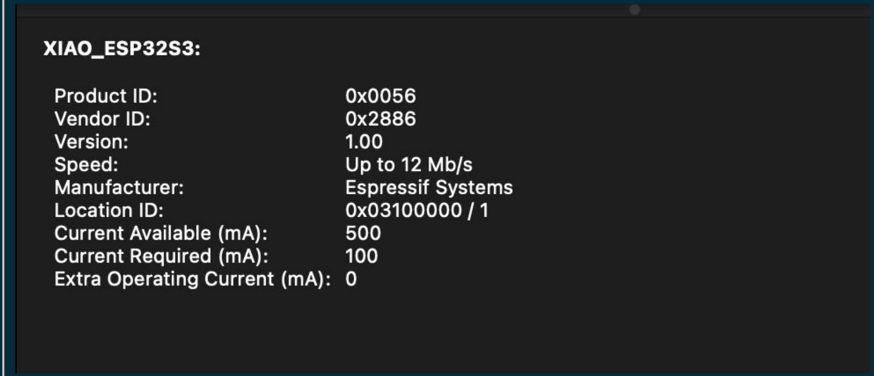
Confusion matrix

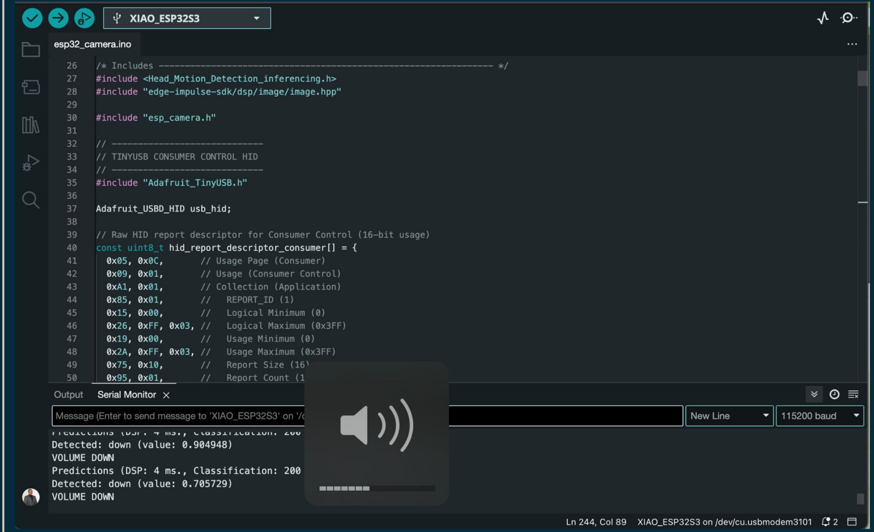
	DOWN	LEFT	RIGHT	STRAIGHT	UP	UNCERTAIN
DOWN	74.1%	5.5%	4.3%	4.6%	0.6%	11.0%
LEFT	0.8%	89.2%	2.6%	3.0%	0.6%	3.8%
RIGHT	0.8%	4.3%	85.3%	3.5%	0.8%	5.3%
STRAIGHT	1.5%	13.8%	5.9%	56.7%	6.5%	15.6%
UP	0%	6.5%	2.9%	3.6%	81.5%	5.5%
F1 SCORE	0.83	0.82	0.83	0.67	0.86	

Feature explorer

The model has the most trouble classifying when your head is “straight”.

Model Deployment





- Once the model is deployed, we connect the XAO ESP32-S3 to our computer via a TinyUSB Adafruit library.
- Real-time deployment yields a classification time of 200ms.
- We can then develop an embedded ML application to detect head motions to control the computer interface.
- *Challenge:* The accuracy of the real-time model is heavily biased towards “up” and “down”, which is comes from combination of physical position sensitivity and model accuracy <80%. In addition, accuracy decreases when there are multiple people in the camera frame.