Maple Sung
CART451

Catch!

For this project, I was interested in trying to displace people's perception of machine recognition, however through their pet dogs. The objective was to use Edge Impulse to train a model that recognizes only dogs, deploy it into an Arduino and trigger a ping-pong launcher. The installation would be ideally placed with the Arduino camera facing inside a dog park, and the ping pong launcher facing out towards the general public.

Materials/Technology:

Data:

I needed to gather data to properly train an image recognition model. For dogs, I've borrowed from dog image datasets online as well as specific dog photos from friends or encounters of people with their dogs. Then I collected background images again from online datasets and of distinct places where I intend to test my installation. For the background images, I had a set with people in it and another without.

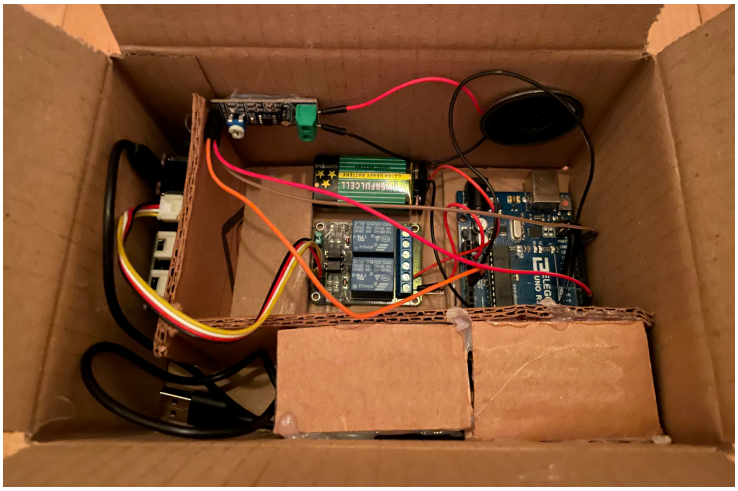
Training:

Edge Impulse developer platform was used to help me train various models and help me determine the most optimized model.

Technology / Materials:

Testing device (simple input - feedback gadget to test my trained models live)

- Edge Impulse Tiny ML Kit
- Relay
- Buzzer
- 6v battery
- External power bar
- Wires
- 5.5x7" cardboard box



Final device (ping pong launcher)

- Edge Impulse Tiny ML Kit
- 2 channel Relay
- 2 x DC motors
- Servo motor
- 6v battery
- Wires
- Cardboard tubes, plastic bottle
- glue gun
- 9.5x10.5 cardboard box

Implemented features:

Edge Impulse Model: (All models referenced below)

I chose to extract features from images resized at 96 x 96 and at the longest axis. Then configured the models using these settings: 20 training cycles, 0.0005 learning rate, with a validation size of 20%, auto-balance, and augmenting the data to make the data more robust and accurate. I selected the MobileNetV1 96x96 0.25 as my neural network architecture. I kept the same settings along all the models I trained to better analyze them.

Code:

Set trigger if ‘dog’ detection threshold is over set amount, experimented between 0.85 to 0.92.

Physical features:

My test machine was placed in a tiny box so it could be installed somewhat discreetly. If and when the camera catches an image of a dog, a loud beep would sound. The final ping pong launcher was also placed in a cardboard box to better contain the electronic and combine the launcher for ease of transport.

Analysis:

Testing:

My steps for testing each model are to first make sure it is functioning using my cell phone because it was the quickest approach. Then it was important to learn what value each image I presented rated. I connected the Edge Impulse Tiny ML board to the Edge Impulse platform under devices and view the raw data using the terminal on my Mac.

```
m.sung — node /usr/local/bin/edge-impulse-run-impulse --debug — 80×24
Last login: Sun Dec  4 09:53:34 on ttys000
m.sung@mSungs-MacBook-Pro ~ % edge-impulse-run-impulse --debug
Edge Impulse impulse runner v1.16.1
[SER] Connecting to /dev/tty.usbmodem14301
[SER] Serial is connected, trying to read config...
[SER] Retrieved configuration
[SER] Device is running AT command version 1.7.0

Want to see a feed of the camera and live classification in your browser? Go to
http://192.168.1.3:4915

[SER] Started inferencing, press CTRL+C to stop...
Predictions (DSP: 17 ms., Classification: 628 ms., Anomaly: 0 ms.):
    dog: 0.67578
    not dog: 0.11328
    unknown: 0.21094
Predictions (DSP: 17 ms., Classification: 628 ms., Anomaly: 0 ms.):
    dog: 0.75781
    not dog: 0.14844
    unknown: 0.09375
Predictions (DSP: 17 ms., Classification: 628 ms., Anomaly: 0 ms.):
    dog: 0.80078
    not dog: 0.11719
    unknown: 0.08594
```

Subsequently, I deployed the model packaged for the Arduino library and then uploaded it onto my test device (Video also provided in repo). At this point, I only used my surroundings and photos of dogs as test subjects to conduct trials.

Actual testing (site: Lachine Canal, Montreal)

I placed the box with my test device right beside the canal that faces downwards and at a slight angle to raise the possibility of detecting dogs walking by.


person walking


person runnning


Multiple people walking


person + dog


people + stroller


people + dog

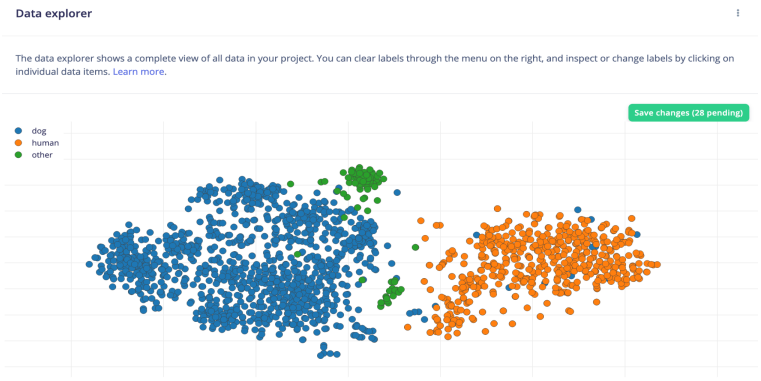

Cyclist

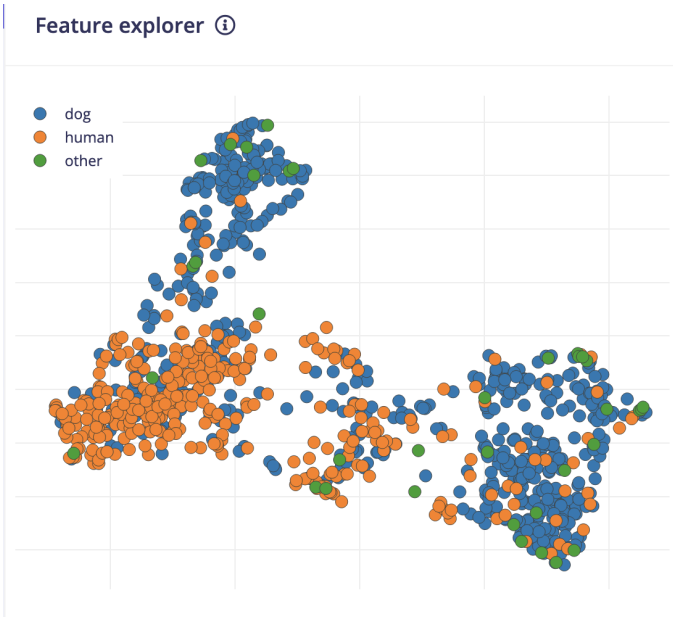
Analysis:

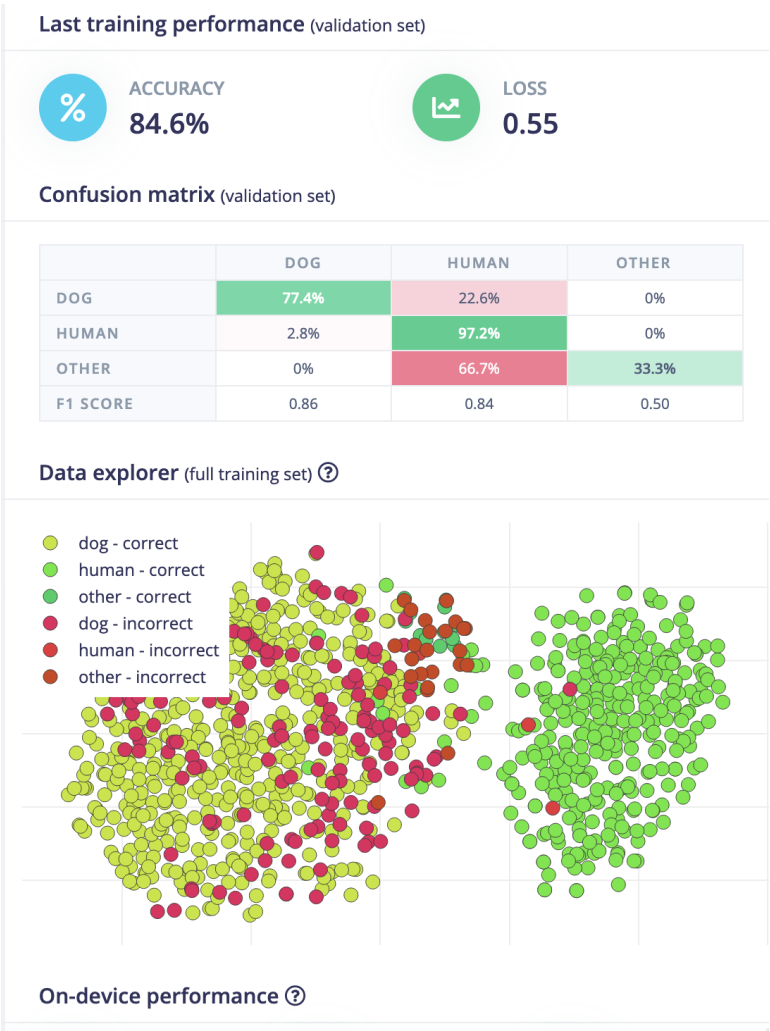
Models: (*numbers refer to amount of images for each object)

1. DOG, HUMANS (humans in the background), OTHER (birds + cats)

- Dog: 796, Humans: 450, Other: 46
  https://studio.edgeimpulse.com/public/154747/latest



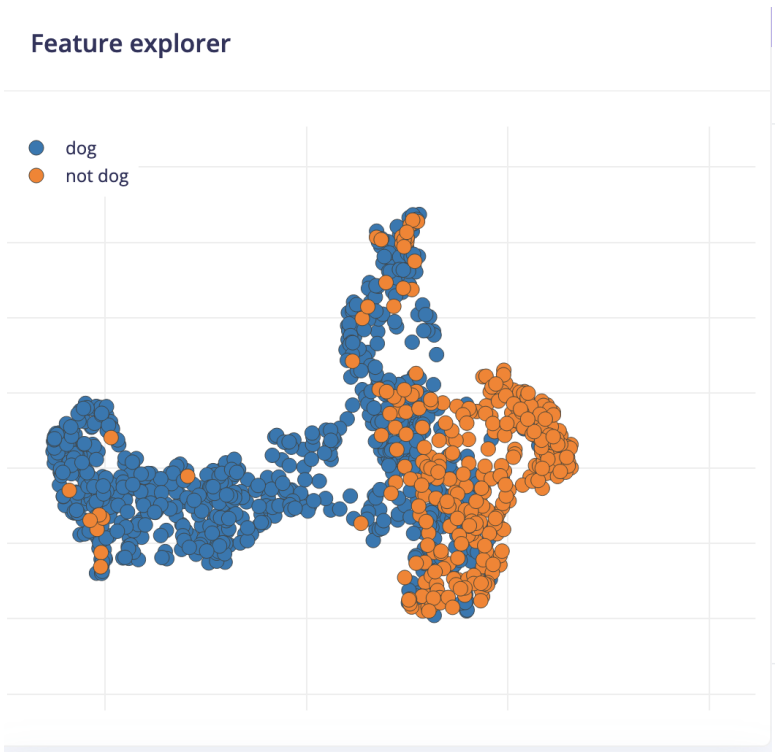


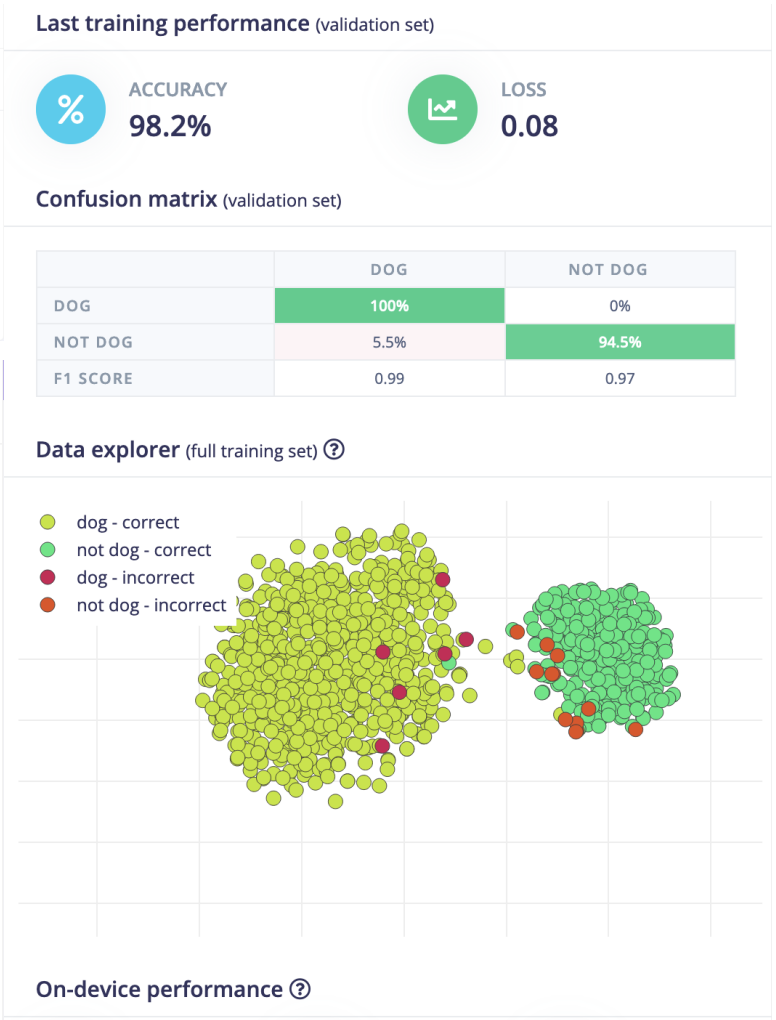


NOTE: Even though the 'other' in this model had low accuracy, the final model showed this to be a good training set where the 'dog' and 'human' showed enough of a difference.

2. DOG , NOTDOG (backgrounds with and without humans + birds + cats)

- Dog: 792, notDog: 330
  https://studio.edgeimpulse.com/public/155457/latest

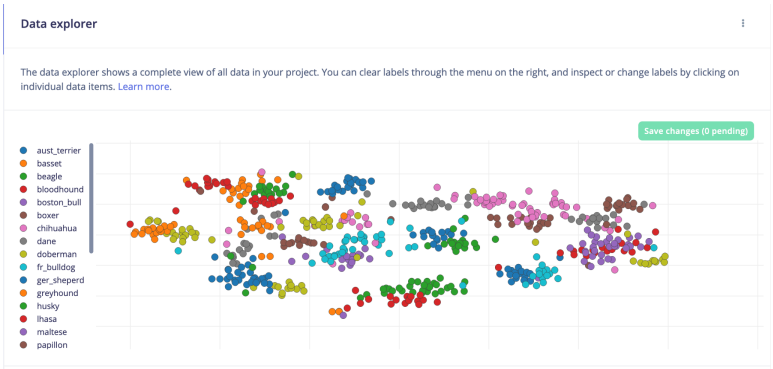


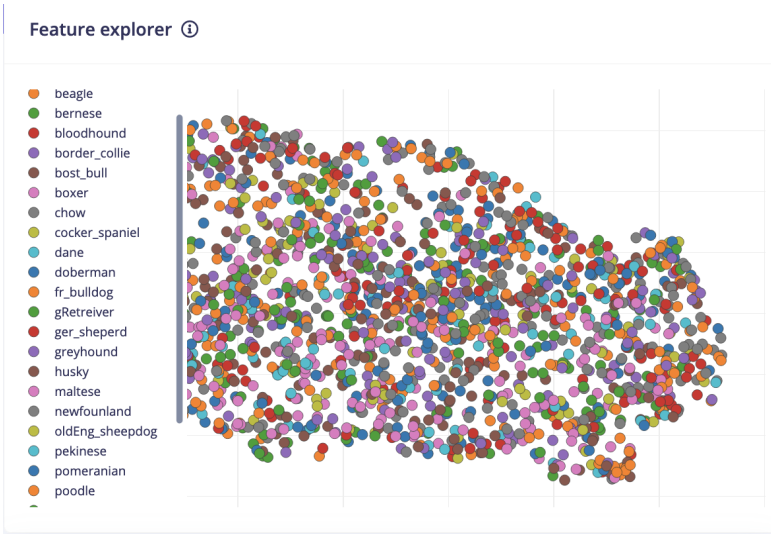


NOTE: The data explorer in the model here shows great accuracy in determining ‘dog’ and ‘not dog’, however, ‘dog’ did have 100% accuracy. Needed to explore more.

3. DOG BREEDS

- Dog breeds: 4042
  https://studio.edgeimpulse.com/public/165076/latest

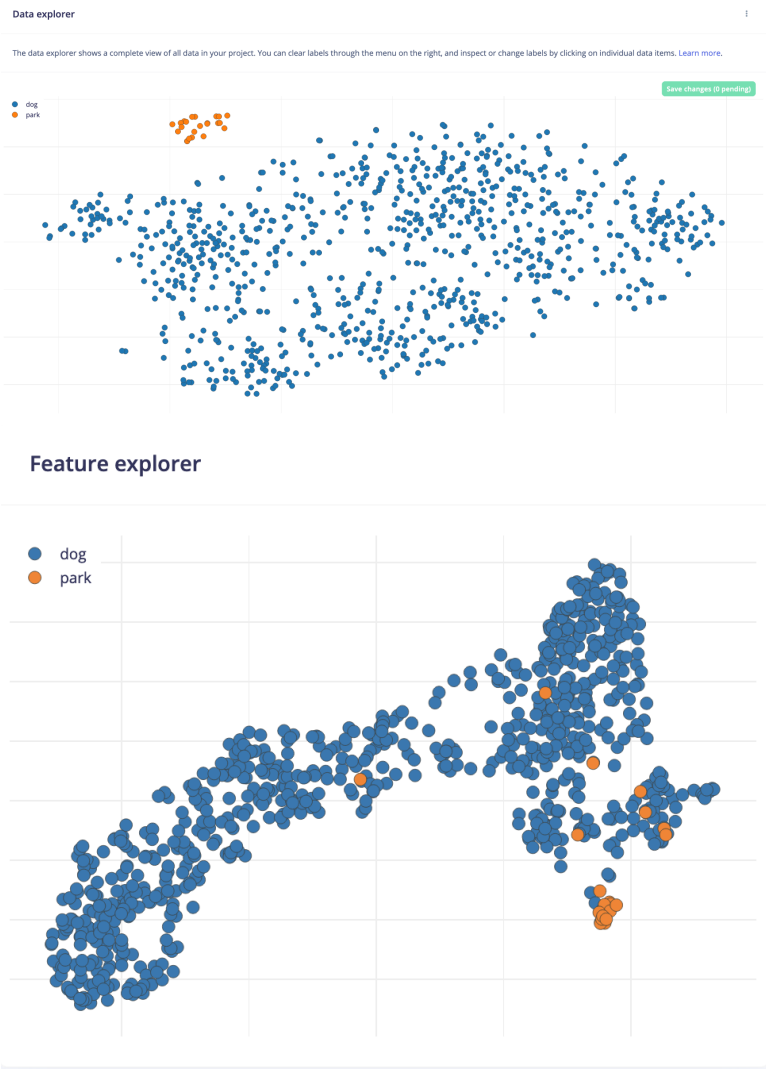




NOTE: Edge Impulse has lots of trouble with too many objects.

4. DOGS, PARK (exact location background)

- Dog: 660, park: 92
  https://studio.edgeimpulse.com/public/166674/latest



NOTE: This model was made to see if I could cheat the inferencing by using images of exact dogs and location.

Conclusion:

Out of all the models I had trained, I only implemented and tested three of the four models: dog - notdog, dog - humans - other and dog - park. I observed the first two models return non-conclusive results so I experimented with the trigger threshold in the Arduino code. It seemed either the model was too sensitive or the camera module from the Arduino kit was not powerful enough. I did notice that when I viewed the camera output through Edge Impulse's platform, the feed was not the best quality and lagged a lot. Also, since I wanted to deploy into an Arduino Nano, I could not train my model in any architecture more powerful than the 'MobileNetV1 96x96 0.25'. With these limitations in mind, I then decided perhaps I could cheat the system and train my final model with images only gathered from the same location I've been conducting my tests. Unfortunately, it was still unsuccessful. I believe the model would function if I used a higher-quality camera because I had made a test with the same person walking by the device at different speeds. The sound, meaning the detection of a dog was triggered if he walked by faster (refer to videos: person_walking vs person_walking2). Finally, I did feel disingenuous if I did use the final device where ping pong is launched instead of a sound. Because most of the time dogs do not stand still, so all they have to do is pass by the camera, also, owners usually stay in one spot inside a dog enclosure. Effectively, it would seem like any other motion detection instead of a device that recognizes a dog.

** video references in 'testResults' folder of repo

References:

Brown, Avi. “Using Edge Impulse AI Inferences to Trigger Events in Arduino C++.” *Medium* (blog), June 4, 2021. https://avi-brown.medium.com/using-edge-impulse-ai-inferences-to-trigger-events-in-arduino-c-7f7edbb11143.

https://kaggle.com/competitions/dog-breed-identification.

“ESP32-CAM Face Recognition for Home Automation – Robot Zero One,” April 13, 2019. https://robotzero.one/esp32-face-door-entry/.

Latest Open Tech From Seeed. “Everything About TinyML - Basics, Courses, Projects & More!,” June 14, 2021. https://www.seeedstudio.com/blog/2021/06/14/everything-about-tinyml-basics-courses-projects-more/.

How to Make Ping Pong Ball Launcher at Home - Full Auto Electric Machine Gun, 2017. https://www.youtube.com/watch?v=87dd0elBw_E.

“How to Program / Upload Code to ESP32-CAM AI-Thinker (Arduino IDE) | Random Nerd Tutorials,” February 4, 2020. https://randomnerdtutorials.com/program-upload-code-esp32-cam/.

https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-cpu-gpu-targets/edge-impulse-linux-macos.

https://www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset.