Emotion Classification on the Edge Using Two Stage Learning Algorithm

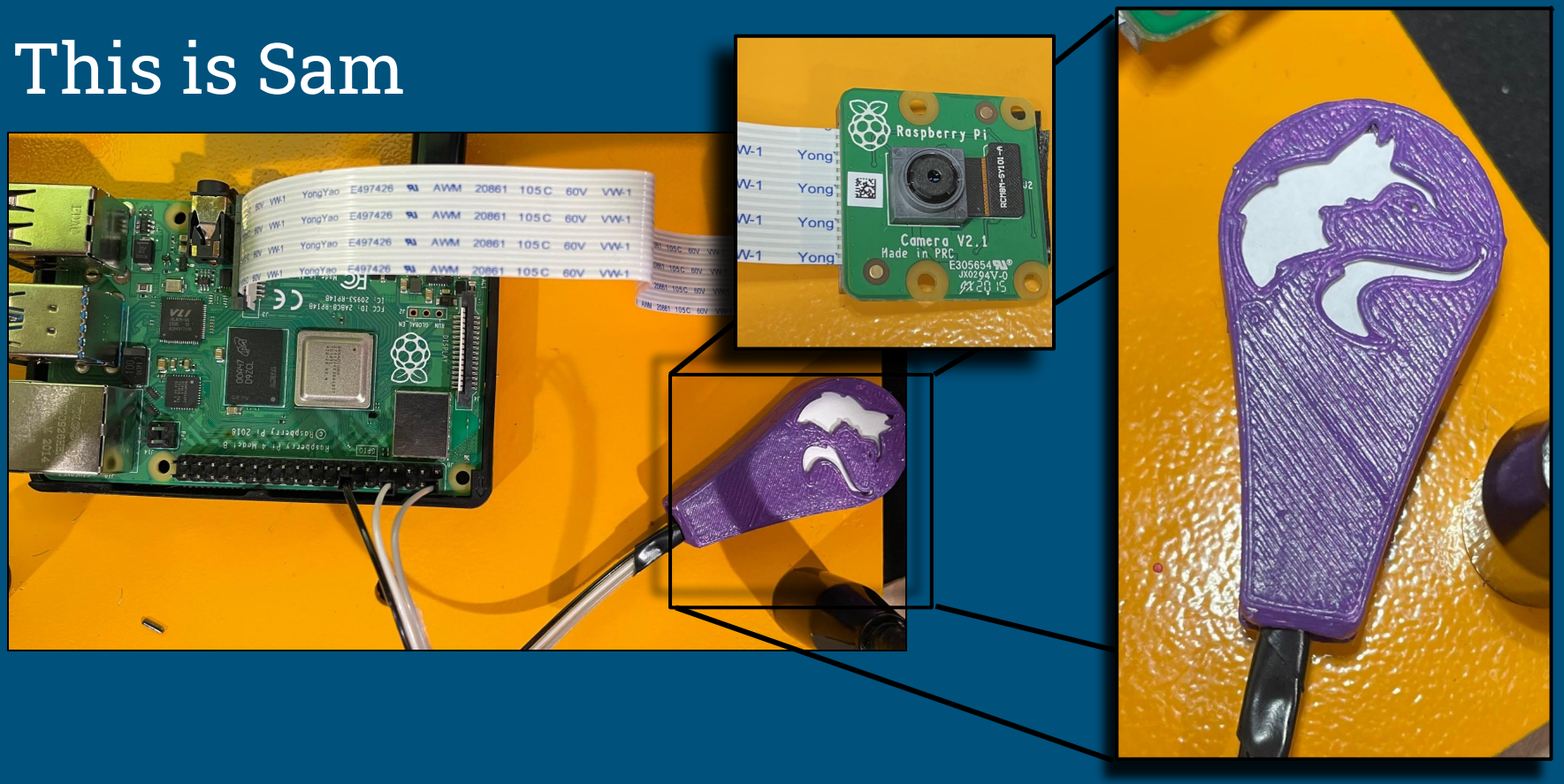
Elliot Blanford
EEP 552
3/2/2021

Emotional intelligence is one of the biggest predictors of success in life and work.

Noticing and reacting appropriately to someone’s emotions is a critical part of emotional intelligence

What if there was a machine that anyone could use to supercharge their emotional intelligence?

This is Sam



Device Requirements

- Capture video stream with camera
- Analyze faces in the frame and classify their emotions
- Do it quickly. You will see a frames per second (FPS) tracker
- Provide feedback to a user via buzzer when a change in emotional state is detected

DEMO

kaggle

- FER-2013 dataset 28,709 examples, 48x48 pixel grayscale, label 0-6
- 6 layer convolutional neural network to classify one of seven emotions
- F1 score: 0.70

0: Anger

1: Disgust

2: Fear

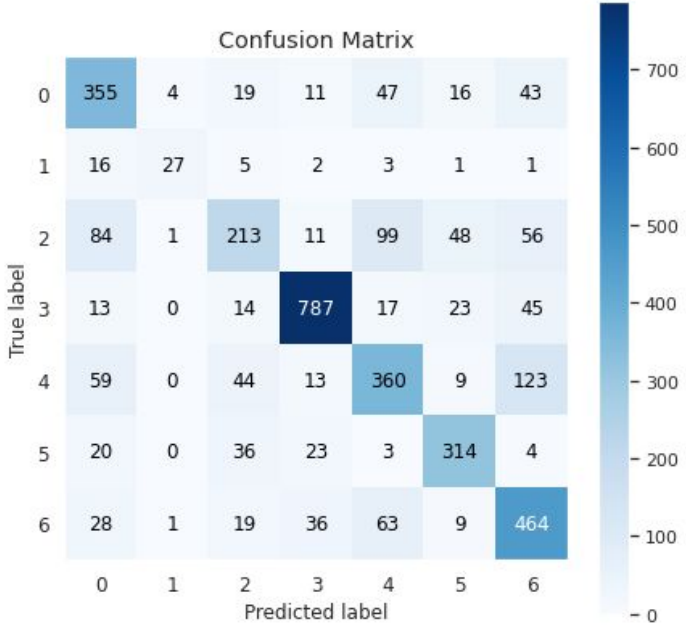
3: Happiness

4: Sadness

5: Surprise

6: Neutral

	precision	recall	f1-score	support
0	0.62	0.72	0.66	495
1	0.82	0.49	0.61	55
2	0.61	0.42	0.49	512
3	0.89	0.88	0.88	899
4	0.61	0.59	0.60	608
5	0.75	0.79	0.77	400
6	0.63	0.75	0.68	620
accuracy			0.70	3589
macro avg	0.70	0.66	0.67	3589
weighted avg	0.70	0.70	0.70	3589



The Metrics

- Classification accuracy by F1 score:

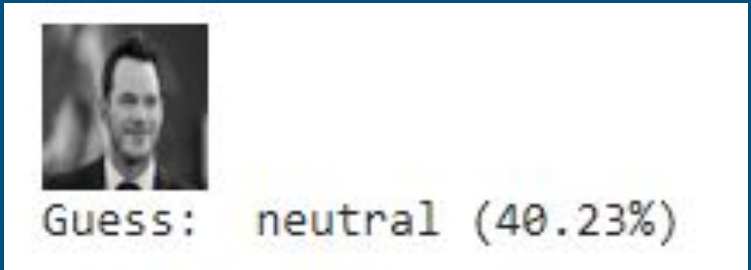
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}$$

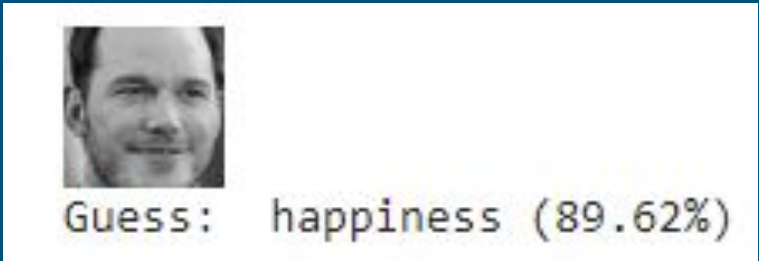
- Inference speed by frame rate

- The tradeoff

Challenges

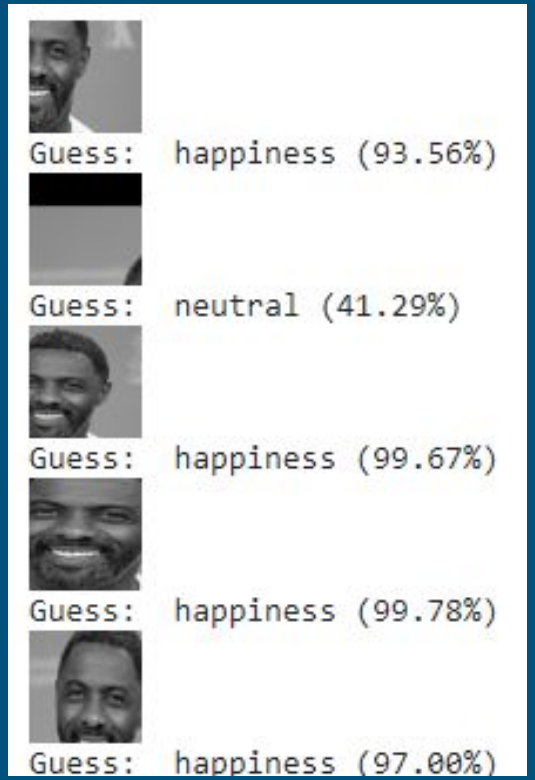
1. Transfer learning: the accuracy is much worse when the image comes from the Raspberry Pi Camera.

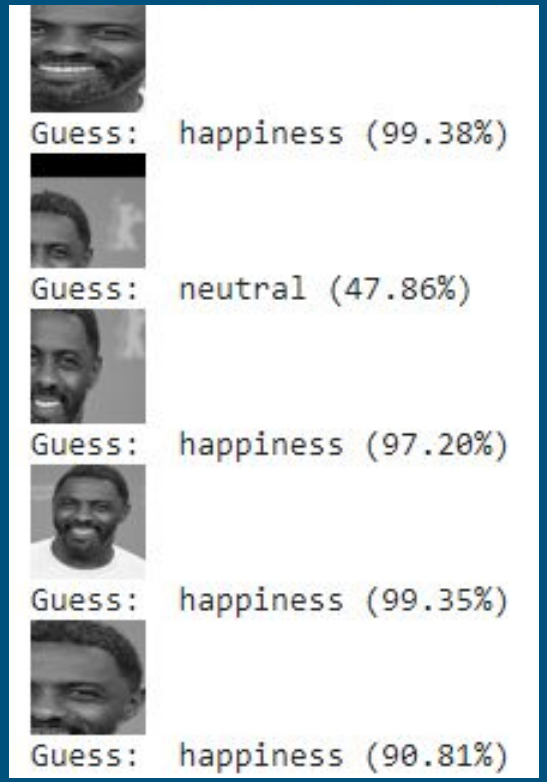




The key difference between training and inference on the device is the background. This is why it is necessary to use the two stage algorithm.

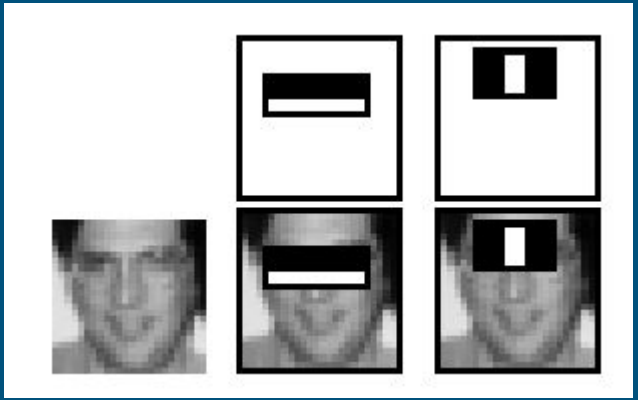
The classifier is robust to errors in the ROI bounding box

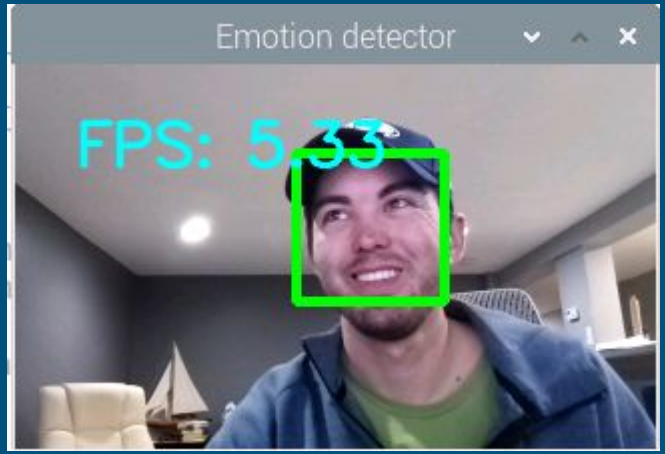




Answer

Viola-Jones cascade classifier for face detection





https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

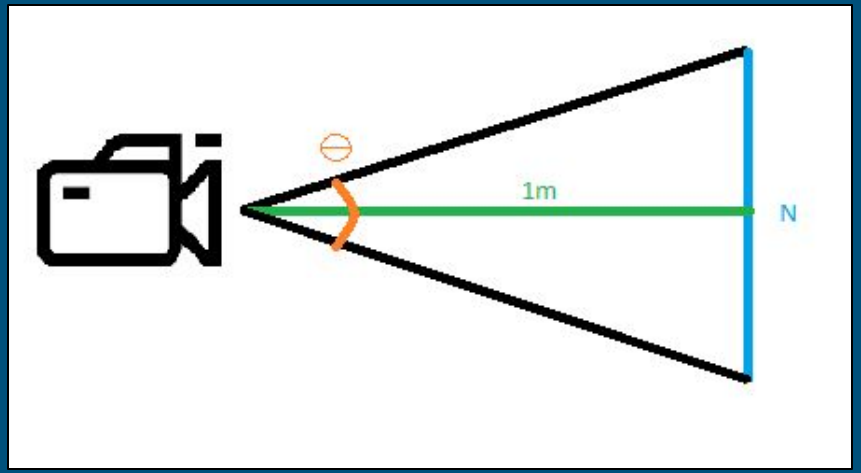
Challenges

1. Transfer learning: the accuracy is much worse when the image comes from the Raspberry Pi Camera.
2. How long is an emotion displayed on the face?

Strategies to improve the frame rate?

- TFLite model converts model weights from 8 byte floats to 4 byte ints
- The minimum face size parameter in the cascade classifier
- Using ‘shallower’ neural network helped a lot for training time but not for inference: <0.1 fps
- TFLite optimizers: +1.0 fps
- Reduce resolution from 1280x720 to 320x192: +0.9 fps
- Hide camera preview: +0.2 fps
- Running headless mode: <0.1 fps

How much can I scale down the image?



$$N = \tan(\Theta/2)*2*1m$$

FOV (degrees)	FOV in radians (Θ)	View area at 1m (N)	# pixels for a face	Compression
62.2	1.0856	1.2065	212.2	4.4
48.8	0.8517	0.9072	198.4	4.1

Lessons Learned

- ML on the edge is intimidating but not impossible
- Running a neural net on Raspberry Pi is hard
- Be ready to adapt!
- When you reboot the RP4, GPIO pins go high and it starts buzzing like crazy

Things I tried for the first time

- Raspberry Pi
- Linux
- Github
- SSH
- Tensor Flow
- TFLite
- OpenCV
- RPi.GPIO
- Fusion 360
- 3D printing

Thank You

Questions:
elliot.blanford@gmail.com

BOM

1x Raspberry Pi 4B
1x 3.5A USB-C Power Cable
1x Raspberry Pi Camera V2
3x 15cm jumper wire, 28 gauge, 0.5mm female connector
1x vibrating motor, 3V DC, 85mA, 12000 rpm
1x BC547 transistor

Consumable:
4in Electrical tape
1g Solder, SAC305
7g PLA filament

Appendix

My Github: https://github.com/Elliot-Bl/Emotion_Detector

Guide to set up ML on RP4. This is a fantastic guide and probably saved me a million billion hours in getting my first model running on RP4.
https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi

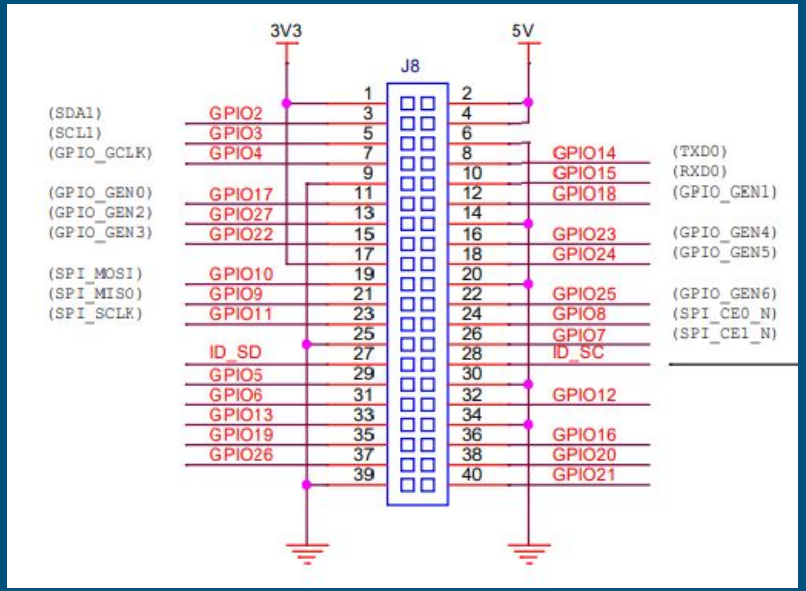
An education start up using emotion detection to gauge learning
https://www.cnn.com/2021/02/16/tech/emotion-recognition-ai-education-spc-intl-hnk/index.html

The Buzzer

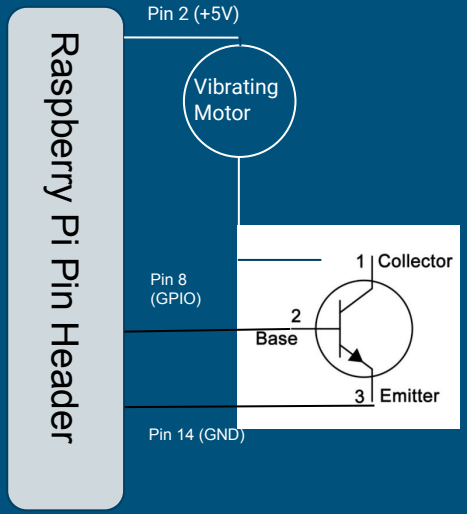
I included the buzzer output mostly as an excuse to learn about GPIO pins.

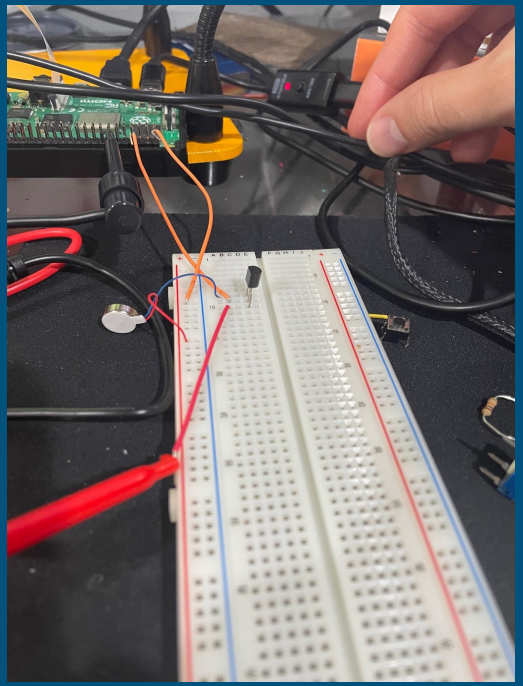
I had to use the 5V rail because it was the only thing that had enough current to run the motor. The interesting (dangerous?) thing is that the 5V potential is always on, so I had to use a transistor controlled by a GPIO pin

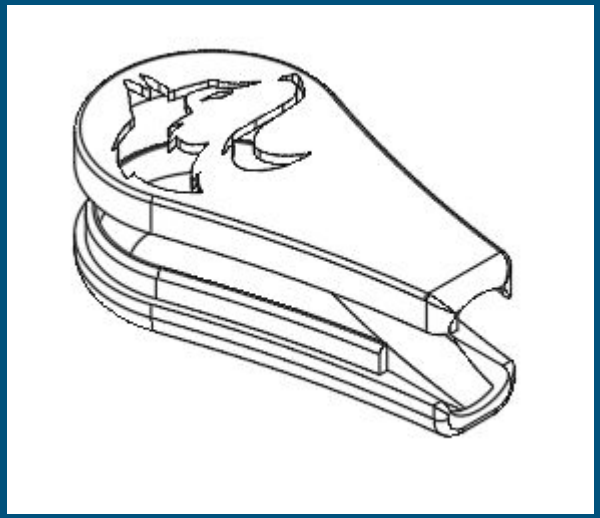
Fun fact: the logic to decide when to buzz took more lines of code to implement than the Viola-Jones face detector!



Buzzer Module







Known Bugs

When you boot up the RP4, GPIO 8 goes high and it starts buzzing like crazy

Challenges

- Transfer learning: the accuracy is much worse when the image comes from the Raspberry Pi Camera.
- How long is an emotion displayed on the face?
- What about individual baselines? In lie detection, the objective is to establish a baseline then look for deviations from that. So the first step is to do some calibration for the individual. This would require some re-training of the neural net. Using the neural compute stick you can do some edge training with k-nearest neighbors, last layer retraining, or weight imprinting.

What is the confidence?

The output of the neural net is a tensor (aka array) of 7 floats.

```
mapper = {0:'anger', 1:'disgust', 2:'fear', 3:'happiness', 4: 'sadness', 5: 'surprise', 6: 'neutral'}
```

I print out the highest confidence answer and if it is above a certain threshold, I will also print the confidence. It's not really a % but that's a little easier for me to wrap my head around.

```
Guess:  neutral
Guess:  neutral
Guess:  neutral (53.68%)
Guess:  neutral
Guess:  neutral (50.37%)
Guess:  neutral
```

Outline

- The project
- The metrics
- The challenges
- Lessons learned
- Tips

Motivation