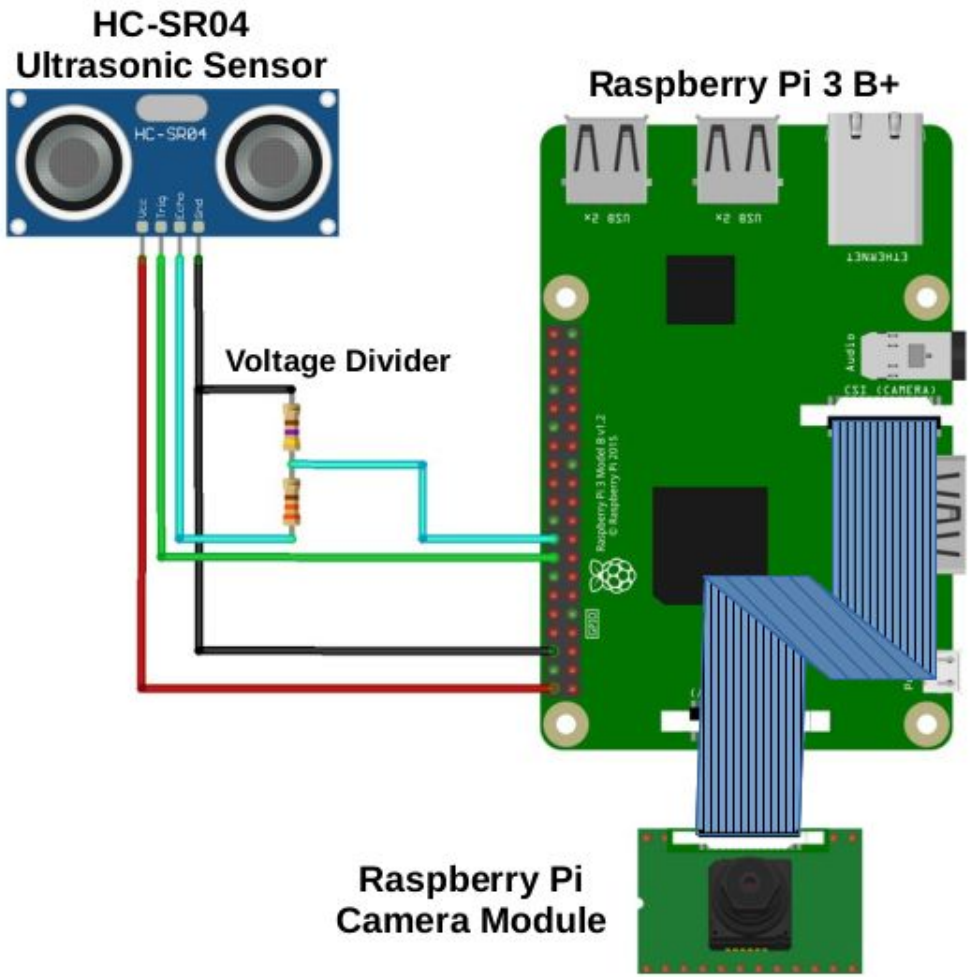
Embedded Facial Detection and Identification

Deployment of Tensorflow Lite and OpenCV on the Raspberry Pi 3B+

Andrew O'Shei
Intégration et développement sur des systèmes nomades
Master Informatique - Universite Paris 8

System Specs

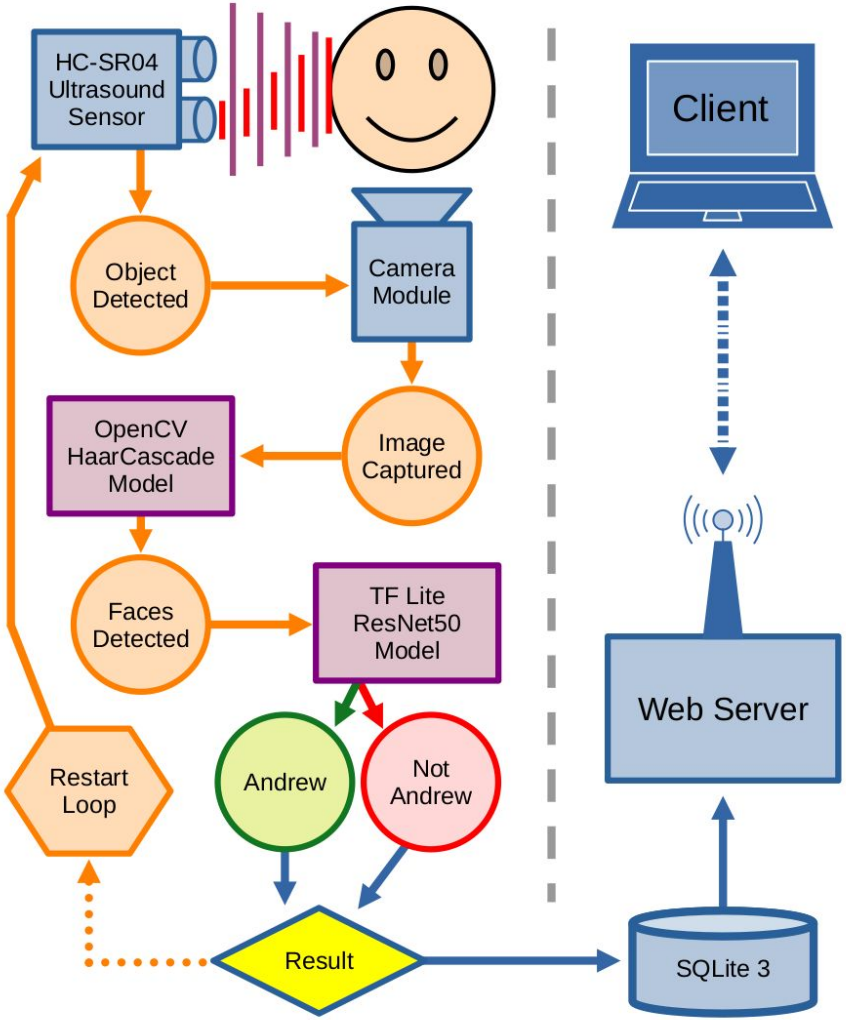
- Raspberry Pi 3 Model B+
  - Single Board Computer
  - ARM-Cortex A53 1.4GHz
  - 4 Cores
  - 1Gb DDR2 RAM
  - Wifi and Bluetooth
- Raspberry Pi Camera Module
  - 5 Megapixels
  - 1080p
- HC-SR04 Ultrasonic Sensor
  - Range 2cm - 400cm
  - Angle 15 degrees



Application flow

The application has two main components each operates on an independant process

- Face Detection / ID System
  - Hardware components
  - Image processing
  - Machine Learning
  - Database writes
- Web Server
  - Handles client connections
  - Database reads



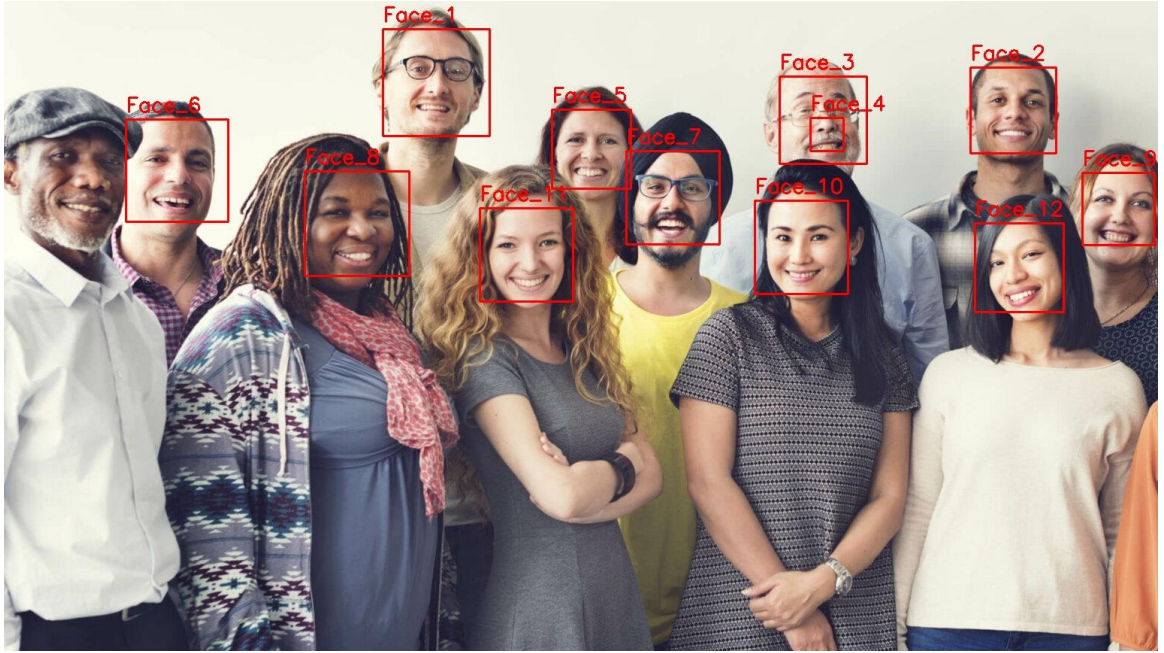
OpenCV HaarCascade Algorithm

Machine learning based approach for object detection

Trained using positive and negative image classes

- **Positive images:** Images containing the object
- **Negative images:** Images not containing the object

The pre-trained FrontalFace_Alt2 model was selected for this application as it provided a decent detection rate with low occurrences of false positives in testing



Faces detected using the HaarCascade_FrontalFace_Alt2 Model
Original Image Source: martech.org

The Dataset

The dataset consisted of 368 images classified in two classes

- **The positive class :** *Andrew*
  - 100 unique images
- **The negative class :** *Not Andrew*
  - 268 unique images
- Each image contains a single face
- Images size 250 x 250 pixels

The images in the *Not Andrew* class were derived from Google's *Facial expression comparison dataset***

https://research.google/tools/datasets/google-facial-expression/

**R Vemulapalli, A Agarwala, *A Compact Embedding for Facial Expression Similarity*, CoRR, abs/1811.11283, 2018.

Dataset Augmentation

Augmentation of the dataset was used to obtain a final dataset with 1668 images

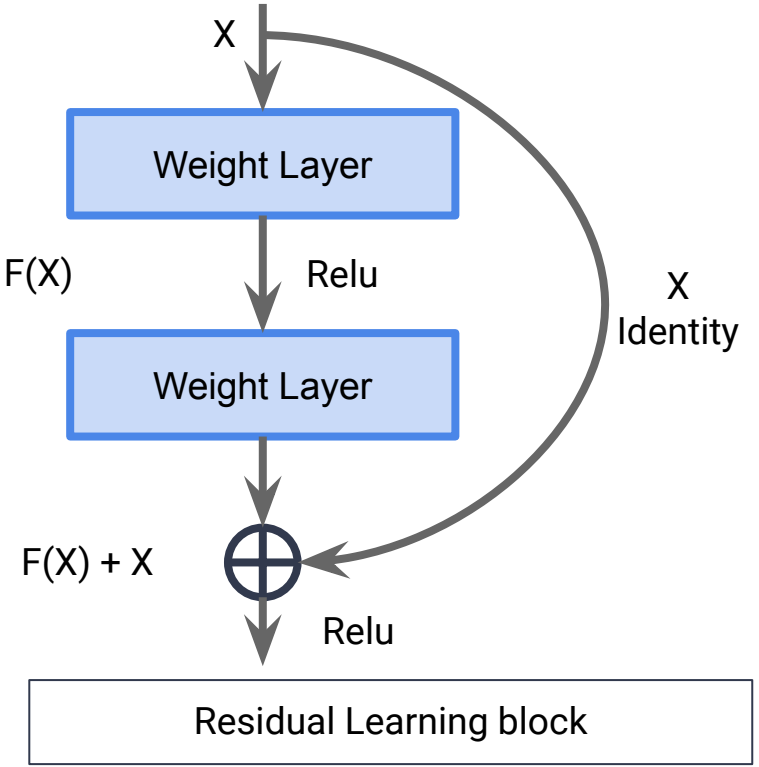
- Horizontal Flip
  - Vertical flips were not used as I do not anticipate the application taking pictures of upside down faces
- Scale / Zoom
  - Nearest neighbor was used to fill empty space
- Rotations
- Distortions

Problems:

- Not a replacement for a truly robust dataset
- Prone to overfitting
- Undesirable artifacts in some images

Residual Neural Networks - ResNet50

- Proposed in 2015 (He et al.), Microsoft Research*
- The ResNet model Won the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) in 2015
- ResNet50 - 50 Layer Convolutional Neural Network
- Includes Residual Learning Blocks to maintain features across layers
- The network is a bit complex for embedded system inference
- Takes a long time to train, it would have been impractical if we did not have access to a GPU



*K He, X Zhang, S Ren and J Sun, *Deep Residual Learning for Image Recognition*, Microsoft Research 2015.

Training Results

Validation Dataset:

- Tensorflow (TF) Validation Set
  - 50% Split of 1668 images
- Post Validation Dataset
  - 20 images of Andrew
  - 20 images of other people

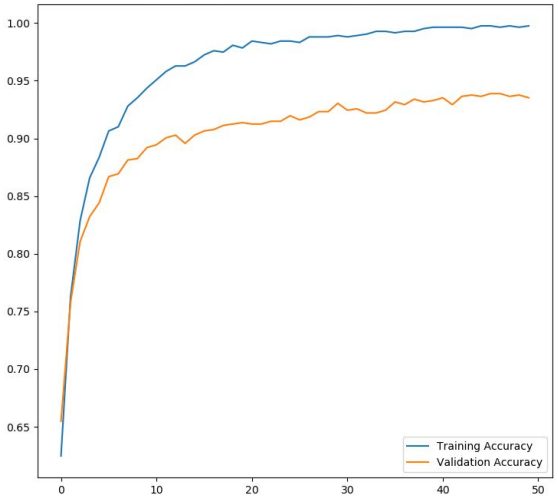
Results:

- TF Validation Accuracy = 92%
- Post Validation Accuracy = 90%
- Post Validation Recall = 1.0
- Post Validation Precision = 83%
- Post Validation F1 Score = 0.9

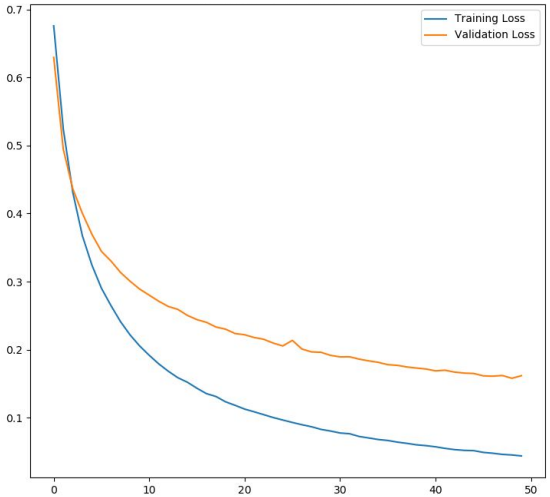
Qualifications:

- The TF Validation set included augmented versions of images that appeared in the Training set, this can cause overfitting
- Validation set is too small to make strong claims
- Not the same validation set used during training

Training and Validation Accuracy*



Training and Validation Loss*



**Accuracy and Loss measured over 50 training Epochs.*

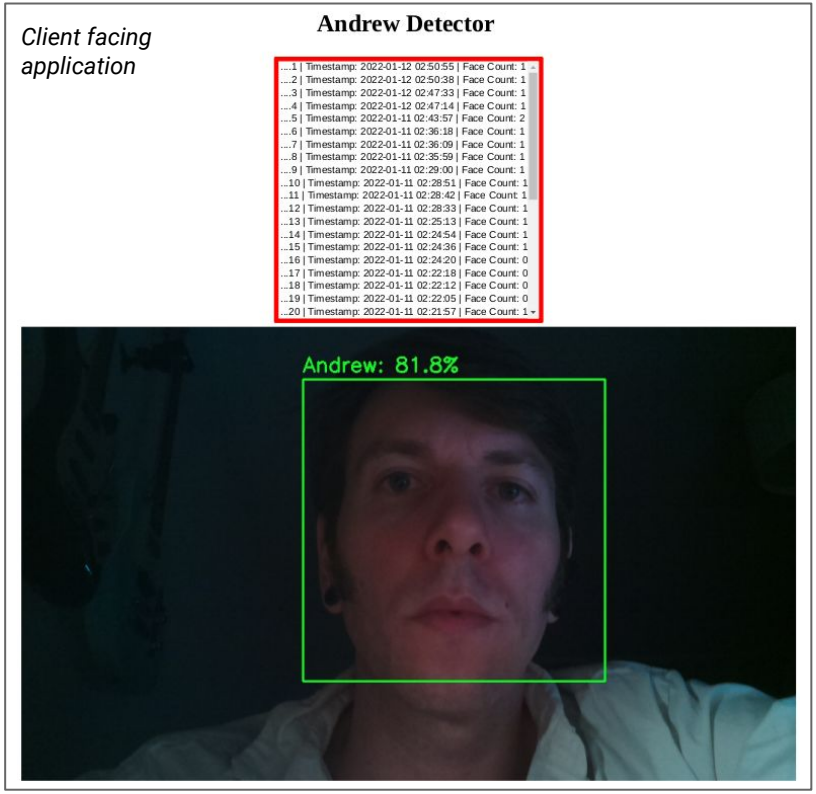
Training Set

Validation Set

Performance Analysis

- Considering the size of the initial dataset the real world performance is not bad
- The performance could be improved by expanding on the HaarCascade FrontalFace model
  - Pictures taken at angles are not detected
  - FrontalFace could be used in tandem with other models in OpenCV, assuming the RasPi has enough resources
  - ResNet50 was trained on images of varying angles
- A more robust dataset will likely outperform the augmented dataset
- Most of my face images in the dataset were taken from the same camera in indoor lighting conditions
  - More variety in lighting should improve dynamic range of the detection algorithm
  - Using different cameras should improve generalization

Client facing application



Primary Sources:

- Haar Cascade Model
  - OpenCV Docs, https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
  - Padilla R, Costa Filho CFF and Costa M, *Evaluation of Haar Cascade Classifiers for Face Detection*, ICDIP: International Conference on Digital Image Processing Vol. 6, April 2012.
- ResNet50
  - K He, X Zhang, S Ren and J Sun, *Deep Residual Learning for Image Recognition*, Microsoft Research 2015. https://arxiv.org/pdf/1512.03385.pdf
- Data Augmentation
  - Tensorflow Docs: https://www.tensorflow.org/tutorials/images/classification
  - D Nikolaiev, *Real-time 'me-not_me' Face Detector*, September 2021. https://www.linkedin.com/pulse/real-time-me-notme-face-detector-dmytro-nikolaiev/
- Face Dataset
  - R Vemulapalli, A Agarwala, *A Compact Embedding for Facial Expression Similarity*, CoRR, abs/1811.11283, 2018. https://research.google/tools/datasets/google-facial-expression/

Thank you for your attention!