**Daycare Face Recognition**

* Create a face recognition system that can accurately identify N individuals via a video camera mounted at the daycare entrance and notify the kids and employees via some sort of speech agent as to who is here.
* Ex: “Henry, your mom is here!” or “Hey everyone, Fatema is here!”

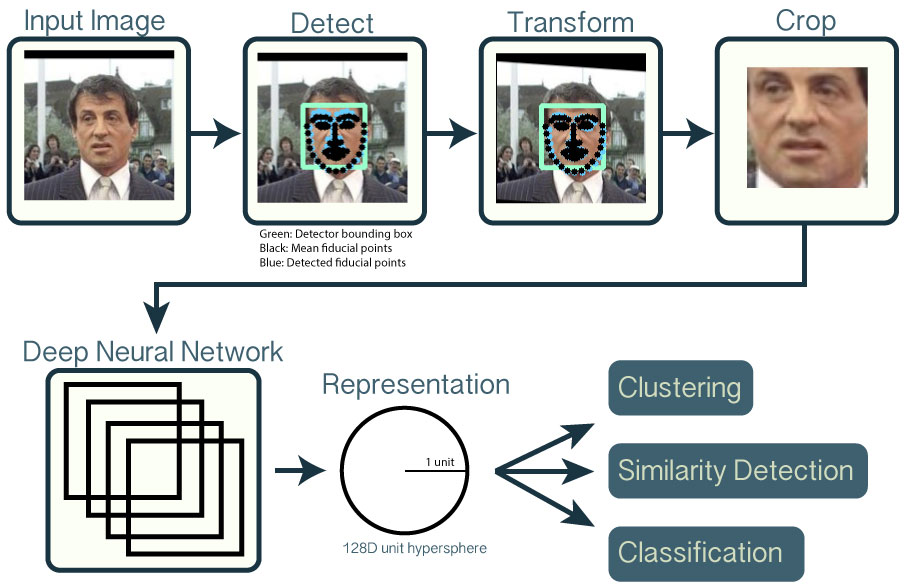
**Motivation**

* What makes a good CS project is either a large number of users or a cool story. As it’s quite hard to do the former, especially with any certainty in the given time frame, this at least shows community engagement, creativity, and the use of pretty new technologies.

**Outline**

* Door-mounted camera will detect when a face is present and attempt to match the face with one of N individuals on whom the system has been trained.
* If face is detected:
  + If does not match with 1 of N individuals:
    - Speech agent communicates “Someone is here!”
  + If matches with 1 of N individuals:
    - If individual is tied to a specific child (mother, father, grandmother, etc.):
      * Speech agent communicates “[child\_name], your [relationship\_to\_child] is here!”
    - Else:
      * Speech agent communicates “Hey everyone, [indentified\_person] is here!”

**Things we (probably) need to use**

* Raspberry Pi and camera module
  + Goal of this is to take the video to identify when there is a person present, and who that person is
  + Not sure if we will use this or some other IOT video camera
* FaceNet
  + <https://arxiv.org/pdf/1503.03832.pdf>
  + Paper that provides a pretrained model for face recognition
  + Learns a neural network that encodes a face image into a vector of 128 numbers. By comparing two such vectors, you can then determine if two pictures are of the same person
* TensorFlow
  + Framework to help do triplet loss, which will make it clearer when we have different images of the same person vs. different images of different people
  + Encoding distances
* Keras
  + Upper level framework on top of TensorFlow which will enable quick changes and iterations to the model
  + Basically all the work is done with this
* OpenFace
  + Carnegie Mellon open source project which uses FaceNet to create a system that can grab image of someone’s face and recognize it
  + <https://cmusatyalab.github.io/openface/>
  + 
* Home Surveillance – Brandon Joffe
  + Open source project on GitHub which uses OpenFace to create a home security system which uses IOT devices such as security cameras to detect people and who they are
  + Can take a look at his architecture to see how we will set up the entire system (what hardware, how to connect things, etc.)
  + Benefit is that this is a system that is always watching, more in line with what we will be implementing – not sure about what issues we could run into using the Raspberry Pi so have to do more research on this
  + <https://github.com/BrandonJoffe/home_surveillance#license>
* Speech agent
  + Something like Alexa to speak to the kids / employees and communicate the curated messages
    - Not sure if it’s possible to trigger Alexa without voice commands, regardless can look into developing a custom Skill to do this
    - <https://developer.amazon.com/docs/ask-overviews/understanding-the-different-types-of-skills.html>
    - <https://developer.amazon.com/docs/custom-skills/understanding-custom-skills.html>
  + If using Alexa proves to be too complicated, can implement some simple text-to-voice software to just read our commands, push the audio to a Bluetooth speaker

**Steps (before 8/5)**

* Get needed hardware
  + Camera unit (Raspberry Pi video camera module is preferable for affordability, If not we can look at the ones used by Brandon Joffe)
  + Wires and stuff, power source, computing power (Pi or cloud or laptop), etc.
  + Voice agent
    - Computer directly outputting audio to Bluetooth speaker with text-to-audio software
    - Or, send messages to be read by Alexa
  + I’ll confirm this stuff as much as possible and order to your house so we can hit the ground running and aren’t waiting for hardware to be delivered
* Figure out CNN (convolutional neural network) architecture
  + Whether we will use OpenFace (and possibly Joffe’s stuff) or use the pretrained FaceNet network to train our implement our own model from scratch
    - This really depends on what’s easier to do for our application
* Plan out architecture
  + Base off of people that have built similar systems before, determine what libraries, packages we will need and also if there is open source code to help us with this
    - Online research and scour GitHub

**Next steps (8/5 – 8/17)**

* Set up minimal viable project
  + System that can detect when a person is present
  + System that can recognize, with > 50% accuracy, 1 individual. This can be my mom, as it will be easy to get training pictures of her and to test on her as opposed to some random parent
  + Should have architecture set up to output string “Hey everyone, Mohua is here!”
* Iterate and add features (ideally)
  + Train on multiple people for detection, add relevant outputs
  + Connect with speaker, whether it be Alexa or not, to speak to the kids
* Final version (really ideally)
  + Train on all 12 parents, > 50% accuracy on each
  + Cloud computing, instead of local Pi or laptop
* Create a simple 3-5 minute video explaining how we built the system, problems we encountered along the way, and a demo of it working at the daycare
* Document progress w/ pictures and notes
* I’ll create a Git repo for the project, we can upload our code there

**Things to do**

* Check out the FaceNet paper
  + <https://arxiv.org/pdf/1503.03832.pdf>
* Look through OpenFace and Joffe’s stuff
  + <https://cmusatyalab.github.io/openface/>
  + <https://github.com/BrandonJoffe/home_surveillance#license>
* Read up on face recognition systems in general to get more insight on how they work, also check out convolutional neural networks and read up on how AI works in general
* Python 3
  + I still have to look more into the exact architecture we’re going to use, but 99% chance we will just be using Python (for the recognition model and maybe fitting different components together) and Linux/GIT commands (to download the stuff we need)
  + The coding itself shouldn’t be super complicated, most of the challenge will be in getting the different parts of the system to work together and tuning it to actually work decently well
  + My recommendation would be to get some working knowledge of Python (how functions work, different data structures, syntax things, etc.) and to read some stuff on how TensorFlow and Keras works with Python
  + Also definitely download Python 3 on your computer and play around with it, there’s good tutorials online from Quantopian and Codecademy, lots of practice problems on HackerRank