

ECE 568 – Final Project Report

Facial Recognition System on Aerial Vehicle

Team:

- Himal Agrawal (agrawalh@purdue.edu)
- Maria Nieves Brunet (mbruneta@purdue.edu)
- Joseph Yanos (yanosj@purdue.edu)

Brief Description of Facial Recognition Algorithm

To perform facial recognition, we use OpenCV, Python, and deep learning. The technique is called deep metric learning.

Deep metric learning is different because instead of trying to output a single label, we are outputting a real-valued feature vector.

For the *dlib* facial recognition network, the output feature vector is 128-d (i.e., a list of 128 real-valued numbers) that is used to quantify the face.

The network architecture for face recognition is based on ResNet-34 from the *Deep Residual Learning for Image Recognition* paper by He et al., but with fewer layers and the number of filters reduced by half.

The network itself was trained by Davis King on a dataset of ~3 million images. On the Labeled Faces in the Wild (LFW) dataset the network compares to other state-of-the-art methods, reaching 99.38% accuracy.

Both Davis King (the creator of *dlib*) and Adam Geitgey (the author of the *face_recognition*) have written detailed articles on how deep learning-based facial recognition works. We refer to the articles below for more details on how deep learning facial embeddings work.

- *High Quality Face Recognition with Deep Metric Learning* (Davis)
- *Modern Face Recognition with Deep Learning* (Adam)

List of code sources and tutorials

- Real-Time Face Recognition

<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

<https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826>

- Raspbian Stretch: Install OpenCV 3 + Python on your Raspberry Pi

<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/>

- Raspberry Pi Face Recognition

<https://www.pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/>

Guide – All the steps are equivalent for either Raspberry Pi or Laptop running Linux

1- Install OpenCV

- Install dependencies
- Download the OpenCV source code
- Compile and Install OpenCV,
- Test

2- Test the Camera

- Check all drivers are correctly installed

3- Face Detection

- Configure libraries for face recognition
- Make directories for each person you would like to recognize
- Gather the faces dataset (photos)
- Compute face recognition embeddings
- Test recognition of faces in live video streams.