Technical Review Draft

Andrew Davis

CS 461

Fall 2018

Abstract

The Hierarchical Information Variant Exhanging Network (HIVENet) aims to us neural networks (NN) and their coefficients to train devices across a wireless network. The bulk of this project lies in researching a way to efficiently transfer these coefficients, as NNs can consist of large amounts of data. This process requires a system that accurately recognizes the facial structures of individuals, the ability to transfer data between edge- and core-devices, and a way to train NN on such devices given new input. The success of the project lies in the system as a whole to learn from disjoint entites that coordinate together to develop a cloud of information.

CONTENTS

1 Introduction

A vital portion of our project is ensuring a facial recognition framework is chosen, so as to avoid building this from the ground up. To ease this process, the framework chosen must have sufficient documentation, allowing for a quicker understanding and ability to manipulate data to our needs. Another portion of our project is the way in which images or videos are received into the system. Factors to include here are the quality of image, ease of setup, and the way in which the devices handle light. It is important to keep in mind the environment we will be in when presenting our project as well.

2 Facial Recognition Frameworks

2.1 Introduction

Our project requires software that utilizes a neural network to recognize images. In this case, those images are faces. Searching for, and understanding the facial recognition framework used for our project should not take up a lot of time. Therefore, our choice in this framework should be easy to use, read, and have the ability for us to extract data from the framework. Using this data will lead to the bulk of our project, ensuring the efficient transfer of NN coefficients to allow retraining of a centralized device's NN, and in turn edge devices' NNs. First, a brief overview of the framework, including levels of documentation, programming languages, etc., will be given. Then a more in depth look of the framework will be covered.

2.2 OpenFace

OpenFace is an open source software that uses deep NNs in facial recognition [1]. Its development was supported by the National Science Foundation, the Intel Corporation, Google, NVIDIA, and a number of other companies. There is much documentation on its use, as well as a Github repo. The software is implented using a combination of Python and Torch (both well documented languages). OpenFace has the ability to run on a CPU and CUDA, a product of NVIDIA.

The software is trained with a set of data, allowing for faces to be recognized. Then, new images to be detected are adjusted so as to produce the same image orientation everytime. Once this is done, OpenFace maps generic key features to the image, identifying features such as eyes, nose, and mouth. Clustering and classification techniques are then applied to complete the process.

There are four different facial recognition libraries presented. Their names, accuracy rates, and runtime are below:

Model	Accuracy	Runtime on CPU
nn4.small2	92.92%	75.67ms
nn4.small1	92.1%	82.74ms
nn4.v2	91.57%	69.58ms
nn4.v1	76.12%	58.9ms

2.3 Visual Geometry Group (VGG) Face Descriptor

VGG Face Descriptor [2] is a facial recognition framework developed at Oxford University that has the ability to train and recognize faces from video footage. This is done through matching of sets of faces from collected data, rather than individual faces. The framework uses Torch (same as OpenFace) as well as Caffe and MatConvNet as primary languages.

In documentation, there were two methods used by VGG. Their accuracy is noted below:

Method	Accuracy
SoftMax	92.27%
Embedding Loss	99.13%

The availability for the source code is currently unknown. An attempt to download is was made, but proved unsuccessful due to circumstances out of my control.

2.4 Deep Face Recognition

This framework, unlike the OpenFace, is not yet licensed. The code is available on Github, allowing for easy access. It uses Caffee, a deep learning framework that is also open source and has ample documentation.

The process of this framework starts off by downloading a dataset of faces. Each image is then converted to gray-scale and reduced or increased in size to a standard, based on facial landmarks such as eyes. The model itself is trained by caffee, as outlined in [3].

A pro of this option is the incorporation of an evaluation system, using a prominent standard, Labeled Faces in the Wild (LFW). Matlab code is also provided to further the evaluation process.

3 CAMERAS

3.1 Introduction

The impact of camera choice on our project will not be entirely known until implementation is enacted. Despite this, we will still choose the best option to ensure this easy-to-fix component does not hinder research and development. This decision comes from [4], which covers the benefits of industrial cameras over built-in ones. One such benefit is the ability for industrial cameras to handle poor lighting while cameras within computers cannot without the use of Software Development Kits (SDKs). The tech review here will cover the built-in camera on my personal device, a webcam extension, and a commercial camera.

3.2 Surface Pro 4 Camera

There are two cameras on this device. A front-facing, 5-megapixel, high-resolution camera, and a rear-facing, 8-megapixel camera, but the front facing one would be used. Both cameras record in 1080p. Windows also has an ability to perform facial recognition to unlock its device, which implies these cameras are capable of this task at close distances. Handling different lighting is the biggest issue with this option, but can be solved with the use of SDKs, as stated in the introduction section.

3.3 Logitech C920 Pro Webcam

This camera is a separate device that connects to a computer through USB. Resolution ranges from 720p at 30fps to 1080p at 20fps. Lighting issues with this device are not well documented. My research for this issue involved watching a tech review, with lighting adjusted. The results look promising, and could mitigate the errors that result from poor lighting. The downfall for this device compared to the previous one is monetary. Whereas the built-in cameras cost nothing, the Logitech camera adds up to around $80. Unfortunately, this device does not come with built-in facial recognition software.

3.4 Nikon D850

This camera is the highest quality of the three outlined. It is able to take videos in 4K resolution at 30fps, or 1280p at 24fps. These stats clearly out-perform the prior two devices, but are not necessary for our project. This, coupled with the extra work done to make this act as a live video recording device rule it out as a viable option, especially when there are two devices that can record already in play.

4 Conclusion

The tech products we choose here ought to provide the least amound of work, as our project is not to build a working facial recognition NN system, but to build ways to efficiently share data and train new NNs across many devices. As a result, the best option for a facial recognition framework is one that allows easy readbility, code manipulation, and much documentation. Out of the ones above, this is OpenFace. For a camera, we want one that will produce the least amount of problems with the least amount of setup. This, in my mind, produces a tie between a built in camera and the webcam. The decision will ultimately be based on group consensus.

References

[1] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, ”OpenFace,” [Online]. Available: *http://cmusatyalab.github.io/openface/#openface*. [Accessed November 1 2018].

[2] Omkar Parkhi, Andrea Vedaldi, Andrew Zisserman, ”VGG Face Descriptor,” *University of Oxford*, [Online]. Available: *http://www.robots.ox.ac.uk/ vgg/software/vgg_face/*. [Accessed November 1 2018].

[3] Alfred Xiang Wu, ”A Light CNN for Deep Face Representation with Noisy Labels,” [Online]. Available: *https://github.com/AlfredXiangWu/face_verification_experiment*. [Accessed November 1 2018].

[4] Baser, Cognitec, ”The Advantages of Industrial Cameras for Face Recognition Applications,” [Online]. Available: *http://www.cognitec.com/files/layout/downloads/White-Paper-Industrial-Cameras-Face-Recognition.pdf*. [Accessed November 1 2018].