CS395 Spring 2018—Final Project Report

Heart Rate Prediction with Deep Video Regression

Adam Barson abarson@uvm.edu

Daniel Berenberg djberenb@uvm.edu [*†]

May 8, 2018

1 Introduction

Millions of Americans are afflicted with panic disorder [1], a psychiatric disorder in which debilitating fear and anxiety arise with no apparent cause [2]. There are several clinically available methods to treat panic disorder, many of which involve either medication or intensive psychotherapy [3]. Past research [4] in the biomedical field has shown that by simply showing a panic disorder victim their heart rate on the onset of or during a panic attack, their episode was significantly mitigated or weakened in intensity.

Allowing people afflicted with panic disorder to access vitals such as heart rate and respiratory rate could not only benefit the longer term management of the disorder, but mitigate the risks and side effects of a live panic attack.

In order to expose this treatment method to as many victims of panic disorder as possible, we explore an element of the solution by making use of the ubiquitous smartphone. There is promising evidence that suggests pulse is detectable by processing videos in smartphone cameras. By pressing a finger against a smartphone camera with the flashlight activated, we can obtain a highly resolved clip of the blood pulsating within the finger. This project aims to leverage deep learning techniques in order to predict a patient?s heart rate from such a video.

[*] in collaboration with Dr. Ryan McGinnis, UVM

[†] under advisory of Dr. Safwan Wsah, UVM

2 Problem Definition

2.1 Task Definition

Our objective is to produce a fast, accurate within a small squared error, model that, given a sequence of the above described frames S of images $X_i \ldots X_n$, we output a value hr, the heart rate of the individual who submitted the sequence.

This problem is specifically intriguing not only for its real world application but because the task is vastly more dependent on the the temporal axis than other neural video processing tasks.

The dataset used by this project has been provided by Dr. Ryan McGinnis of the University of Vermont Biomedical Engineering department. Video data was collected by sampling 31 separate subjects; each patient recorded two videos of their finger– one at a resting heart rate, and one after an intense 60 second workout. All of the raw videos gathered during the study are around 30 seconds long. For each sample V, the heart rate h_r and respiratory rate rv for each were also recorded and are provided along with the video data.

2.2 Algorithm Definitions and Related Work

There are various algorithms that apply to deep neural video processing tasks; we explored three: 2D stacked convolutions with a subsequent LSTM layer or LRCN [5], Two stream networks that incorporate the use of optical flow [6], and 3D-Convolution [7]. Each of these models is designed to extract features from the temporal element of the sample in a unique way.

The **LRCN** uses stacked convolutional layers in order to extract hierarchical, semantically meaningful feature vectors which are then fed into an LSTM layer which uses these features to determine long term dependencies throughout the sample.

A **two stream network** uses two forms of input: raw frame sequences and the corresponding sequence of optical flow images that describes the change between some pair of frames. Formally, optical flow is defined as the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene [8]. Since the optical flow transformation embeds temporal data in a 2D image, the two stream CNN is entirely a convolutional neural network with some sequence of dense layers at the end. Our adaptation of the two stream CNN adopts the temporal stream of the data and leaves the spatial stream. As noted previously, each frame is inconsequential in determining the heart rate of the video; the temporal correlations and color variations over time are much more semantically meaningful.

Finally, our third perspective on video regression stems from the **3D convolutional network**, which considers a volume and convolves throughout it, learning hierarchical features as per 2D. We use a stripped down, version of the C3D [9], a commonly known and ?general purpose? 3D convolutional network, in which we several layers of filters in order to allow it to fit on our single Tesla K80.

Each of these models is a network which has a single, linearly activated output node for regression. These algorithms were each tested, with the 3D CNN ultimately performing the best.

3 Experimental Evaluation

4 Code and Datasets

The code base consists of a library called we_panic_utils, which contains several subpackages and modules for various forms of augmentation, preprocessing scripts, and architecture descriptions, video editing, dataset manipulation and creation, and other general utilities. Our library can be found here. After obtaining the data and organizing it correctly, the data must be preprocessed by running the `generate_dataset` in a bash or simialar unix terminal. After preprocessing the data, use the `run_it` bash script to interact with our codebase.

5 Conclusion