# Capstone Status Report for Week 3

Noah Buchanan (Team 2)

nbucha00@g.uafs.edu

January 24, 2022

## 1 Team Members

My team includes the following members:

1. Sasha Lawson
2. Alana Matheny
3. Sam Donaldson
4. Noah Buchanan

## 2 Goals from Previous Week

The following goals were set in the previous weekly report:

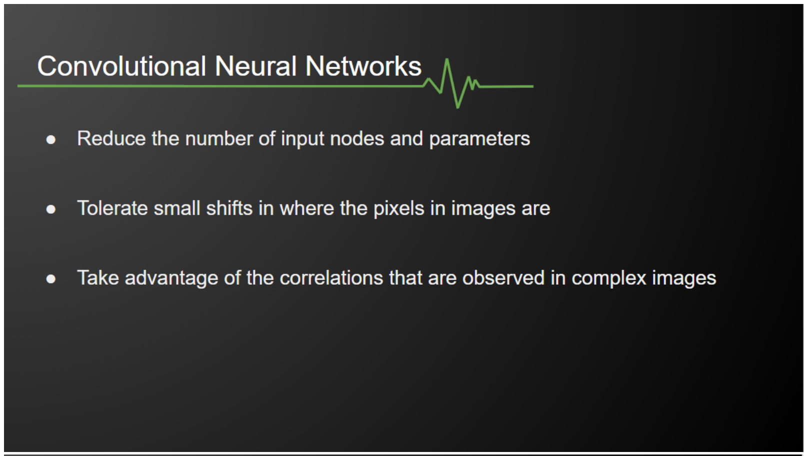
1. **Task 1**: Review and make changes as needed to any of my sections before presentation.
2. **Task 2**: Run through practice presentation with team members to prepare for presentation.
3. **Task 3**: Start looking into how to construct our different models we will be using.

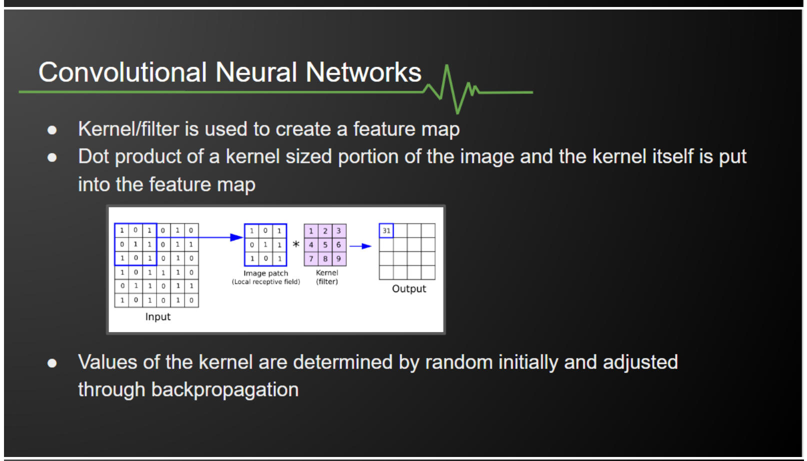
## 3 Completion of Tasks Planned

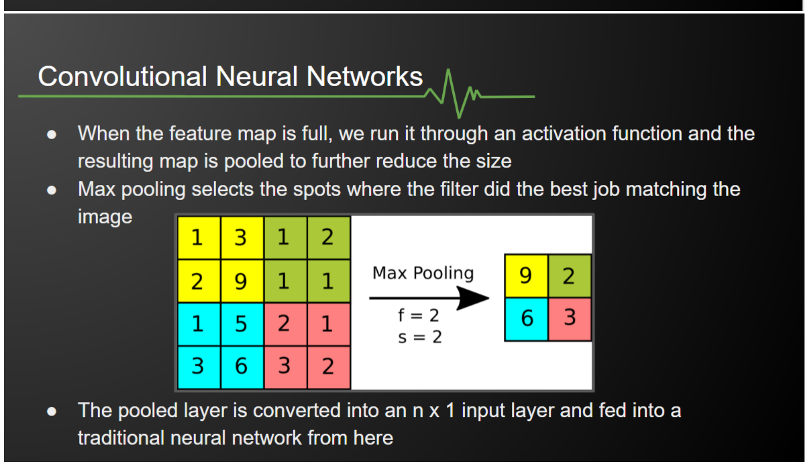
The following work has been accomplished with respect to my previously set goals:

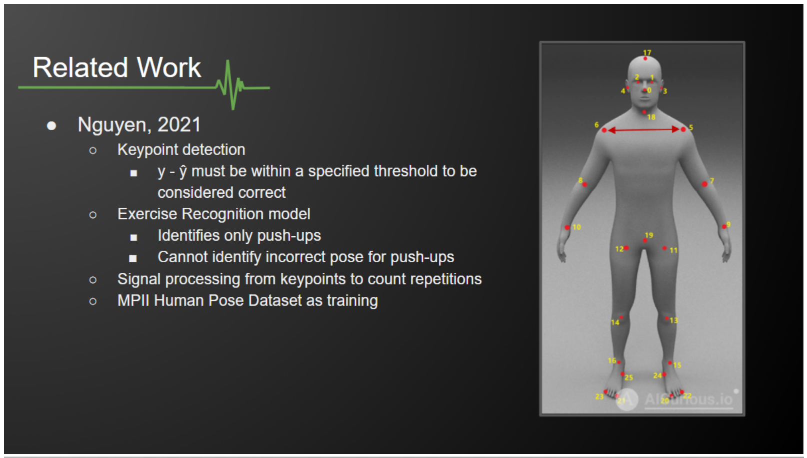
### Task 1: Review and Adjust before Presentation

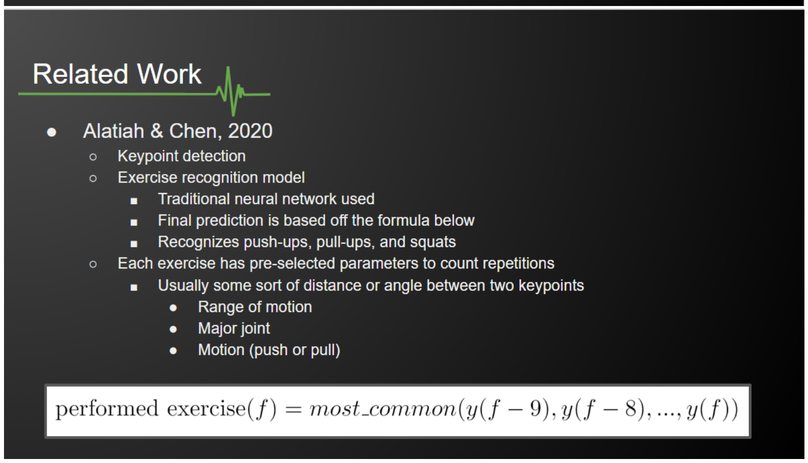
All work was completed as expected, and approved by the team. I made these slides for our presentation:


Convolutional Neural Networks
Reduce the number of input nodes and parameters
Tolerate small shifts in where the pixels in images are
Take advantage of the correlations that are observed in complex images


Convolutional Neural Networks
Kernel/filter is used to create a feature map
Dot product of a kernel sized portion of the image and the kernel itself is put into the feature map
Input
Image patch (Local receptive field)
Kernel (filter)
Output
Values of the kernel are determined by random initially and adjusted through backpropagation


Convolutional Neural Networks
When the feature map is full, we run it through an activation function and the resulting map is pooled to further reduce the size
Max pooling selects the spots where the filter did the best job matching the image
1 3 1 2
2 9 1 1
1 5 2 1
3 6 3 2
Max Pooling
f = 2
s = 2
9 2
6 3
The pooled layer is converted into an n x 1 input layer and fed into a traditional neural network from here


Related Work
Nguyen, 2021
Keypoint detection
y - ŷ must be within a specified threshold to be considered correct
Exercise Recognition model
Identifies only push-ups
Cannot identify incorrect pose for push-ups
Signal processing from keypoints to count repetitions
MPII Human Pose Dataset as training

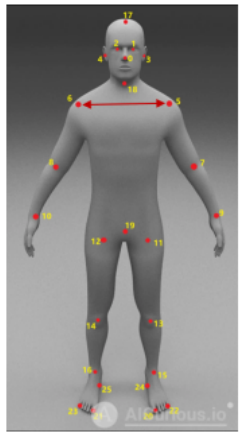

Related Work
Alatiah & Chen, 2020
Keypoint detection
Exercise recognition model
Traditional neural network used
Final prediction is based off the formula below
Recognizes push-ups, pull-ups, and squats
Each exercise has pre-selected parameters to count repetitions
Usually some sort of distance or angle between two keypoints
Range of motion
Major joint
Motion (push or pull)
$\text{performed exercise}(f) = most\_common(y(f-9), y(f-8), ..., y(f))$

I also heavily adjusted our related work:

### 2.3 Related Work

**Push-up Counter**

In [21] a utilization of OpenCV and Deep Learning was demonstrated to complete the task of counting push-ups. Their specific approach involved human keypoint detection (or human pose estimation). They used the MPII Human Pose Dataset and various push-up videos from Facebook as their training dataset. They used a unique metric to measure the quality of these keypoints, Percentage of Correct Keypoints (PCKs): "A detected joint is considered correct if the distance between the predicted and the true joint is within a certain threshold. The threshold here is chosen as 0.25 times of the distance between 2 wrists (or the distance between point 5 and point 6 in the image)."



This model for key point extraction is separate from the exercise recognition model. Signal processing from the keypoints is what is used to count repetitions. In their conclusion they stated they felt they could have achieved better results with more keypoints and leveraging the keypoints to correct wrong push-up poses. We will attempt to achieve both of these. The "sweet spot" of keypoints will likely be found through experimentation so we cannot report on that as of now. As for bad exercise poses, we intend to measure and fix these through the angles between specified keypoints (the best keypoints to measure will also likely be apparent through experimentation).

**Recognizing Exercises and Counting Repetitions in Real Time**

In [22] pose tracking and exercise recognition is used to count repetitions on pull-ups, push-ups, and squats. Their method involves three phases; pose tracker to identify and track users, exercise recognition to detect the name of the appeared exercises, and a counter to count and indicate the correct and incorrect repetitions. For pose tracking they use OpenPose. For exercise recognition however, the authors of this paper did not use a convolutional neural network rather a normal neural network; it is our teams' belief that we can achieve better results through a convolutional neural network. For repetition counting they first use a formula and the predictions of their neural network to predict the exercise as follows:

$$\text{performed\_exercise}(f) = \mathit{most\_common}(y(f-9), y(f-8), ..., y(f))$$

Where, f: frame, and y: predicted label from the neural network.

Once the exercise is detected, there are pre-selected parameters such as exercise range of motion, the major joint, and type of motion (push or pull) that will be used to count repetitions. We will likely use the same formula for detecting exercise as them. However, as stated before, they did not use a convolutional neural network; we will attempt to implement this and achieve better results.

As well as slightly adjusting our convolutional neural network background per our departmental advisor's recommendation (only the part that was adjusted is shown):

### 2.1 Overview

**Convolutional Neural Network**

The traditional neural network is a powerful tool that can be used in a variety of ways within limitations; one such limitation is the extremely costly computation of large amounts of data i.e images. Classifying multiple images in real time with millions, potentially billions, of parameters (depending on the resolution of the images and the amount of layers and neurons used) is not practical. This is where convolutional neural networks (CNNs) come in handy. CNNs do three specific things to make image classification practical:

1. Reduce the number of input nodes and parameters.
2. Tolerate small shifts in where the pixels are in the image. Essentially meaning that through each filter that is applied, a band of width $=\ \text{floor}(\frac{\text{length}(f)}{2})$ in pixels, where $f$ is the filter, will be removed from the edges of the image.
3. Take advantage of the correlations that are observed in complex images.

The first thing a CNN does is apply a filter to the input image. Filters are sometimes also called kernels. For this explanation the former will be used. A filter is a smaller matrix of pixels typically around the size of $3 \times 3$. The intensity of each pixel in the filter is determined by back propagation. To begin with the pixel values in the filter are randomized.

### Task 2: Practice Presentation

All work was completed as expected. We ran through a practice presentation for timings sake as well as preparing everyone and shifting our responsibilities where we thought necessary.

### Task 3: Beginning Model Preliminaries

This task was delayed due to the increased time commitment to the first task. This will be deferred to this week, specifically how to go about creating wire-frames and key-points.

## 4 Additional Work Completed

In addition to the goals set for the previous project, I also completed the following work:

1. Proofread our objective statement and abstract that had been revised by Sasha.

## 5 Overall Group Dynamics

We used both class sessions this week to meet and work as a team. In addition, we also scheduled an additional meeting on Sunday at 2:00 pm to practice our presentation. All members were in attendance. We are all making excellent progress.

## 6 Goals for This Week

The following goals and tasks are planned to be completed by our next status report:

1. **Task 1**: Help the team to create our interfaces (actual code, not mock-ups).
2. **Task 2**: Research which metrics will be best for analyzing our key-points performance and what loss function to utilize for this task.