5-2020

A Capacitive Sensing Gym Mat for Exercise Classification & Tracking

Adam Goertz

A Capacitive Sensing Gym Mat for Exercise Classification & Tracking

Adam Goertz

April 24, 2020

An Undergraduate Honors Thesis

in the

Department of Computer Science and Engineering

College of Engineering

University of Arkansas

Fayetteville, AR

by

Adam Goertz

April 2020

University of Arkansas

0.1 Abstract

Effective monitoring of adherence to at-home exercise programs as prescribed by physiotherapy protocols is essential to promoting effective rehabilitation and therapeutic interventions. Currently physical therapists and other health professionals have no reliable means of tracking patients' progress in or adherence to a prescribed regimen. This project aims to develop a low-cost, privacy-conserving means of monitoring at-home exercise activity using a gym mat equipped with an array of capacitive sensors. The ability of the mat to classify different types of exercises was evaluated using several machine learning models trained on an existing dataset of physiotherapy exercises.

0.2 Acknowledgements

This research was supported by a University of Arkansas Honors College Research Grant

Contents

Chapter 1

Introduction

1.1 Problem

Stretches and non-aerobic exercises are often prescribed by physical therapists as a rehabilitation routine for treating acute or chronic injuries. These behavioral interventions are essential in producing optimal outcomes for patients. Outpatient programs are available for certain types of rehabilitation, but they often fall short of delivering a lengthy enough intervention requiring at-home continuation of physiotherapy. Additionally, at home physical therapy programs place a reduced burden on the patient, eliminating travel time, reducing expense, and freeing more time for medical professionals. The increased freedom of at-home rehabilitation is valuable, but it often comes at the cost of decreased adherence [1].

Patients who adhere to prescribed physiotherapy programs at the beginning of the training period are up to 20x more likely to continue to adhere throughout

the training period [2]. This indicates that providing monitoring, intervention, and encouragement early in the therapeutic process could have large benefits for patient outcomes.

Several strategies have been evaluated for their efficacy in improving adherence to at-home exercise and rehabilitation programs. Of these, two that show promise in improving adherence include 1) asking patients to keep records of both their symptoms and exercise activities and 2) engaging in tele-rehabilitation through email and telephone support [3]. Unfortunately, the benefits of self-reported adherence come with several limitations. Patients may fail to record, inaccurately recall, or deliberately mis-represent their adherence [3]. Therefore, there is a need for an approach that automatically and accurately collects information about exercise activity. Additionally, by providing physical therapists and other medical professionals with information about their patients' activity, they will be able to more efficiently direct their attention to patients who may be struggling to adhere to their prescribed programs.

Non-adherence to a physical therapy program significantly affects the speed and quality of recovery. In order to achieve the freedom of at-home rehabilitation and the accountability of inpatient or outpatient programs, an unobtrusive monitoring system is needed in order to provide medical professionals with information about their patients' activity and progress. Cameras are a dominant technology in telerehabilitation, but the introduction of a camera into private environments presents serious privacy concerns. There is a critical need for cognitive health systems that capture rehabilitative exercise without compromising users' privacy.

1.2 Thesis Statement

A low-cost means of tracking exercise and physiotherapy activities while maintaining patient privacy can be achieved using a gym mat equipped with capacitive sensors.

1.3 Approach

The solution proposed and developed in this project is to design and implement an array of capacitive sensors attached to an exercise mat. Capacitive sensors can be constructed out of inexpensive, flexible, and durable materials. While the mat is in use, the capacitive sensors are sampled by a capacitance-to-digital converter, and data is transmitted to a microcontroller. The microcontroller runs a machine learning model that is trained to differentiate among a several different exercises.

An Adafruit Feather M4 Express is used as the host microcontroller for this project; it runs the classification model which is implemented and trained in Tensorflow, and communicates with the FDC2214 capacitance-to-digital converter over I^2C.

Chapter 2

Background

2.1 Key Concepts

2.1.1 LC Oscillator

LC oscillators convert DC voltages to AC by alternately storing energy in a capacitor and an inductor. The simplest LC oscillator consists of a capacitor and inductor connected in parallel, along with a parallel DC voltage source connected through a switch.

Initially, the voltage source charges the capacitor. When the switch flips, disconnecting the voltage source and connecting the inductor, the energy stored in the capacitor as a voltage is released as current. This current flows through the inductor, which resists the change in current. As the current through the inductor rises, energy is stored in a magnetic field around the coils of the inductor. Eventually, the capacitor is discharged, and the current begins to drop. This decrease in

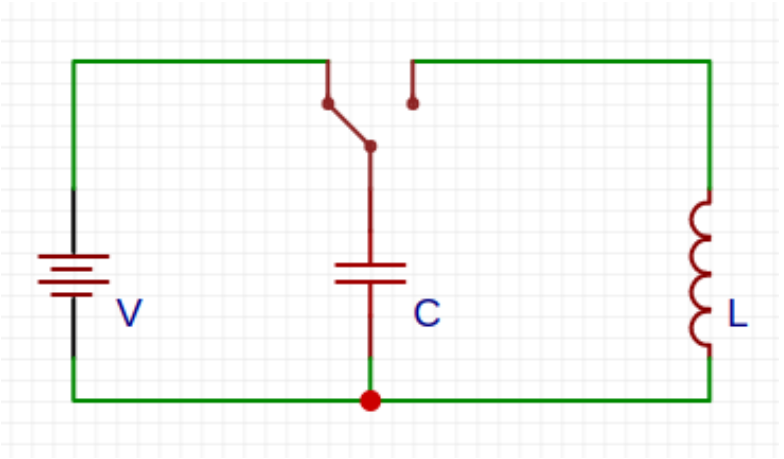

Figure 2.1: Simple LC oscillator

current is again resisted by the inductor, which produces a new voltage with the same polarity as that of the previously charged capacitor. This results in the flow of current in the same direction, charging the opposite plate of the capacitor. From here, the cycle repeats, but with opposite polarity and direction of current flow.

The frequency with which this oscillation occurs is affected by the capacitance and inductance of the circuit. Specifically, an LC oscillator will oscillate at its resonant frequency, which is the frequency at which the inductive and capacitive reactance is equal [4]. At high frequency, a large capacitor does not have time to store a significant amount of charge, and therefore stores relatively little energy as an electric field. This can be seen from the equation for the reactance of a capacitor:

$$X_C = \frac{1}{\omega C} \tag{2.1}$$

where X_C is the reactance of a capacitor, ω is the angular frequency, and C is the capacitance. Conversely, an inductor acts like a short circuit at low frequency, as the current through the reactor changes slowly. An inductor's reactance is given by

$$X_L = \omega L \tag{2.2}$$

The capacitive and inductive reactance is equal at a frequency (in Hz) given by the following equation:

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{2.3}$$

2.1.2 Capacitive Sensing

There are several different, commonly-used modes of capacitive sensing. In mutual capacitive sensing, positively and negatively charged conductive plates are capacitively coupled, and proximity or touch is detected by a decrease in the measured capacitance between them, as nearby conductive objects capacitively couple to the positively charged plate.

In self capacitance, a single plate is charged, and the capacitance of the plate is measured. This capacitance is increased by the presence of conductive objects in the vicinity of the sensor, which form the second plate of the capacitor by coupling to the charged plate.

Capacitance can be measured via several means. One common method is

to charge the sensor plates with a fixed current and measure the time required to reach a threshold voltage. This time is proportional to the capacitance of the plate. Another method uses the sensor plate as the capacitor in an oscillator circuit, and measures the capacitance of the plate by measuring the oscillation frequency of the oscillator. The second method is employed by the capacitive sensing system in this project.

As seen above in section 2.1.1, the resonant frequency of the LC oscillator varies inversely with the capacitance of the circuit. The capacitance of the capacitive sensor is equal to the sum of any integrated capacitors in the LC tank and the capacitance of the sensor pad.

$$C_{total} = C_{sensor} + C_{integrated} \tag{2.4}$$

From this we can see that as the parasitic capacitance coupled to the sensor plate increases–for example, from the proximity of a conductive object like the human body–the oscillation frequency of the sensor decreases.

2.1.3 Machine Learning

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special category of neural network architecture that is especially common in image classification, object tracking, and text recognition applications [5]. The CNN structure mimics the function of biological visual processing, wherein different brain regions are activated by

different features in the visual field [6], such as lines, curves, or other simple shapes. CNNs utilize kernels (matrices of values) to perform convolutions over the input, learning to extract features that are useful in understanding images or other spatially-structured data. A convolution is performed by overlaying the input with each of potentially many kernels, performing a point-wise multiplication over each overlapping pair of values, summing the products together, then storing the result in the location underlying the center of the kernel. Each kernel is gradually shifted over the input. This process produces a set of new feature maps which highlight the areas where a particular kernel matched the input especially well. A simple kernel that detects vertical edges may look something like this:

-1	0	1
-1	0	1
-1	0	1

When the center of the kernel is aligned with a vertical edge, the pixels along the left side will have a significantly different value from those on the right, generating a large absolute value result of the convolution. When no edge is detected, the result of the convolution will be close to 0.

In a convolutional neural network, the kernel values are learned, enabling them to specialize for recognizing features that are desirable in a particular application. Each convolutional layer in a CNN consists of a number of different kernels, each of which produces a different feature map of the input image [5].

In addition to convolutional layers, max-pooling layers are used to produce shift-invariance in the feature maps. Max-pooling layers operate on each feature

map separately to preseve the maximum value in each region of pixels. Commonly, max-pooling layers utilize 2x2 pooling windows, but larger values are possible. By reducing the resolution of feature maps, small changes in location can eliminated [5].

At the end of a CNN is typically a flattening layer, which flattens the multi-dimensional data into a single dimension before feeding it into one or more fully-connected layers, which are used to do the final classification.

Long Short Term Memory Neural Networks

Long Short Term Memory (LSTM) networks are a special class of recurrent neural network (RNN) that utilizes special memory cells to understand long-term dependencies by preserving relevant information in excess of 1000 time steps [7].

As error propagates backward through a recurrent neural network during training, the error signal depends exponentially on the weights. This exponential behavior typically causes the error to either blow up (which can cause oscillations in the trained weights, as error varies dramatically over time) or vanish (which leads to extremely slow training speed) [8]. In order to remedy this, the LSTM architecture uses several gates to control access to the cell state, which ensure that the error signal changes linearly during backpropagation [8]. By moderating the speed of weight adjustment, LSTM networks are able to preserve important contextual information much more effectively than other RNNs.

2.2 Related Work

In [9], Sundholm et al. describe the use of a resistive, textile pressure sensor matrix for classifying and counting exercises. They develop a system capable of differentiating among 10 common exercises by analyzing the user's contact points with the pressure sensor matrix. The reported 82.5% recognition rate across all exercises shows that the pressure mat can be an effective means of classifying exercises. However, there are several limitations to the pressure mat approach. Firstly, pressure sensors provide no information about the user's movements above the surface of the mat except that which can be inferred from small changes in the pressure distribution of contact points. The 10 exercises selected for testing in [9] feature easily distinguishable contact patterns; however, this is not the case for all exercises. In order to accurately classify a broader range of exercises whose contact points may be similar, we need information about motion above the surface of the mat. This could be achieved using a depth camera as in [10], but the introduction of a separate piece of expensive equipment creates a significantly higher barrier to entry. The approach used in this project utilizes an inexpensive array of capacitive sensors which provide a third dimension of sensing capability without the expense or privacy concerns of a camera.

Chapter 3

Design & Implementation

3.1 Capacitive Sensor Array

3.1.1 Adafruit Feather M4 Express

The Adafruit Feather M4 Express microcontroller was chosen as the host microcontroller for this project due to its relatively large flash memory capacity, at 512 KB [11]. Since this microcontroller is used to perform machine learning classification, a large memory capacity is needed to store the model.

3.1.2 FDC2214

The capacitive sensing is handled by an FDC2214 from Texas Instruments. The FDC2214 is a high-resolution EMI-resistant capacitance to digital converter, which uses an LC oscillator to detect changes in the self capacitance of a sensor plate.

The FDC2214 can sample from up to 4 channels sequentially and produces up to 28 bits of precision in its measurements. To interface with the Adafruit Feather M4 Express, the FDC2214 uses a standard I^2C communication protocol. This protocol allows any I^2C compatible microcontroller to program the FDC's configuration registers and retrieve the sensor data. The FDC2214 measures capacitance by charging the connected sensor plate, then measuring the frequency of the LC oscillator.

The FDC2214 measures the LC oscillator frequency and transmits digitized frequency values to the host processor. The digitized frequency values are calculated using the formula

$$DATA = \frac{f_{sensor} \times 2^{28}}{f_{ref}} \tag{3.1}$$

where f_{sensor} is the measured oscillation frequency of the LC tank, f_{ref} is the frequency of the reference oscillator, and $DATA$ is the digitized frequency. For this application, the internal reference oscillator of the FDC2214 was used, which has a frequency of 40MHz. Using an oscilloscope, the oscillation frequency of the LC tank and attached sensor pad was measured to be approximately 3.2MHz. This result was confirmed by calculating the sensor frequency using the above formula and the received data values.

Using this information and the known configuration of the FDC2214, the baseline capacitance of the sensor pad can be calculated. The prototype design utilized a 33pF capacitor and 18μH inductor for the LC tank. The capacitance of the sen-

sor pad itself can be calculated using the formula

$$C_{sensor} = \frac{1}{L \times (2\pi \times f_{sensor})^2} - C_{tank} \tag{3.2}$$

which yields a baseline sensor pad capacitance of 104pF, though this measurement necessarily includes a small amount of stray parasitic capacitance from the environment.

3.1.3 FDC2214 Library

In order to interface with the FDC2214 and make it easier to adjust the configuration parameters, we wrote a small C++ library which utilizes the standard Arduino Wire library to communicate with the FDC2214 over I^2C. This library allowed us to adjust many of the configuration parameters, such as the sampling mode (differential or single-ended), number of channels sampled, sample reference count, settle count, drive current, and deglitch frequency. This library was initially adapted from the FDC2214 library by github user zharijs [12] , but the interface has been adapted significantly to improve readability and allow for the configuration of a larger number of parameters that were not available in zharijs' original implementation.

3.1.4 Capacitive Sensing Plates

The basic design for the capacitive sensing mat consists of three identical arrays of 4 capacitive sensing plates, for a total of 12 sensor plates. The plates are rect-

angular in shape and arranged around each of the four sides of a mat section. The size of the plates was selected to maximize the sensing range. Data from the Texas Instruments data sheet for the FDC2214 capacitive sensing board indicates that a sensor size of approximately 150 cm^2 is gives a maximum effective sensing range of 60cm. The larger the sensor's range above the surface of the mat, the more data can be collected about the user's movements; therefore, the capacitive sensing plates were designed to be rectangular sheets of adhesive copper tape, measuring 5x28cm, for a total sensor area per pad of 140 cm^2. The sensor pads are connected to the FDC2214 capacitive sensing board through length-equated, shielded copper wires. In order to ensure that each wire has approximately equal length, the wires are connected to the long, inner edge of each sensor pad and routed first to the center of the sensor array, before being routed to one corner of the sensor array.

In order to avoid interference from other wires, the sensor plates themselves, or outside noise, the wires are covered in a conductive sheath which acts as a shield from electro-magnetic interference. This shield can either be grounded or charged through a buffer to match the voltage of the sensor pad it is connected to. During testing, both the grounded and actively shielded wires showed a slight reduction in noise at the expense of a decrease in sensitivity.

During initial development, a single array of 4 sensor pads was used to fine-tune the design and experiment with different techniques for improving sensitivity and reducing noise. In this preliminary configuration, each trace was connected to a different input of a 16:1 multiplexer. This design worked reasonably well, and would have had the capacity to scale to the full 12-pad design. However, it had a

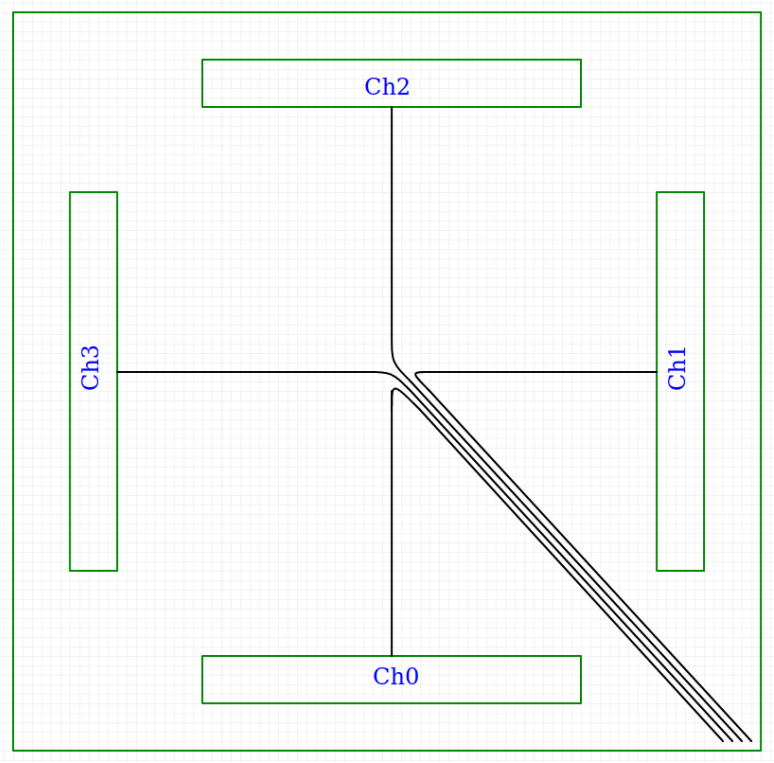

Figure 3.1: Capacitive sensor array layout

few key drawbacks.

First, the resistance of the 16:1 MUX was significant, at around 90Ω as measured through a multimeter. Secondly, using a single multiplexer allowed the use of only one input channel of the FDC2214. Consequently, the effective sampling rate was reduced by half. This reduction in the sampling rate is due to the lack of synchronization between the microcontroller that requests a reading from the FDC2214, and the actual sensing performed by the FDC2214 capacitive sensing board. When the microcontroller hot-swaps the FDC2214 to sample from a new sensor pad by swapping which wire is connected through the multiplexer, it is

not guaranteed that this swap will occur at the beginning of a sample cycle. If a sample is already in progress (which is very likely), that sample will be corrupted, and the system must wait for the next sample to be valid. Due to the complexity of detecting this error, it was far simpler to wait for 2 full sample cycles after hot swapping before reading a new sensor value.

In order to correct both of these deficiencies, the revised design utilizes 4 separate, smaller multiplexers. Each of the 4 wires (one for each sensor pad) from a given sensor array runs to a different TS5A3359 SP3T Bi-Directional Analog 3-1 MUX/DEMUX. These 3-1 multiplexers were chosen for their very low input resistance, 900 mΩ [13], and are used to select which of the 3 sensor arrays is currently being sampled. Each of the 4 multiplexers is connected to a different sensor channel on the FDC2214. This configuration is used in order to minimize the sensing frequency penalty from hot swapping the sensor arrays. After each channels current sensor reading is read, it can immediately be hot swapped to read from the next sensor array. Meanwhile, the other 3 channels are sampled.

3.1.5 Active Shielding

One of the key design considerations in the creation of the sensor arrays is their effective range and sensitivity. However, during initial testing there was a significant range decrease when placed on the floor, as compared to when elevated over the floor (such as when on a table). The likely cause of this decrease in sensitivity is due to electro-magnetic interference from conductive or grounded objects (such as wiring, air ducts, etc) in the floor. In an attempt to mitigate the interference from

the floor, we utilized an active shield. The active shield is a second conductive plate located directly under each sensor pad. These shielding plates are charged to the same voltage as the sensor plate by connecting the appropriate sensor channel pin on the FDC2214 to an Op-Amp which acts as a unity-gain buffer. Because the voltage oscillates at a frequency in the MHz range, the LTC6228 Op-Amp, which has a gain-bandwidth product of 890MHz [14], was chosen for this application. Since it is being utilized as a unity-gain buffer, the effective bandwidth is 890MHz, which is more than sufficient for this application.

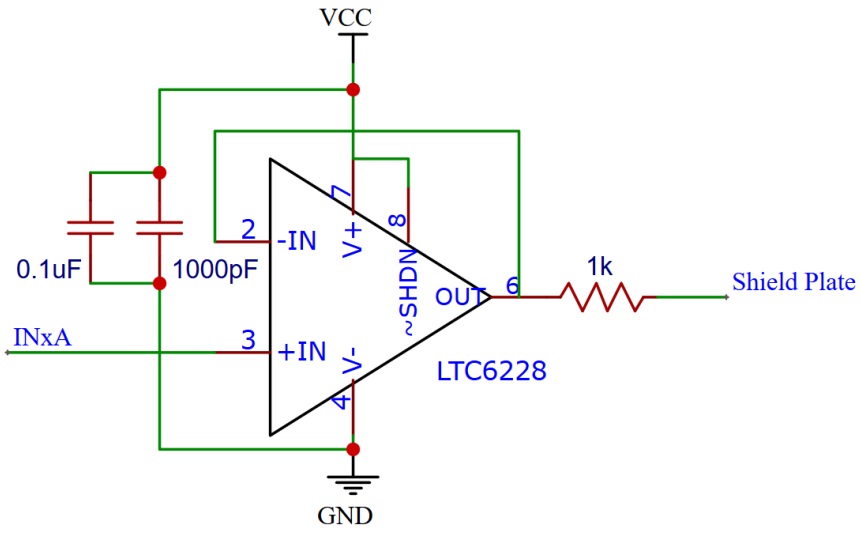

Figure 3.2: Unity-gain buffer

One of the primary challenges in designing this active shielding system was eliminating noise and high-frequency oscillations in the buffer output. 0.1μF and 1000pF bypass capacitors are placed in parallel between the positive and negative supply voltages to filter out oscillations and noise in the supply voltage. Because

the LTC6228 is not designed to drive capacitive loads, a 1kΩ resistor is placed between the output pin and shield plate to act as a low-pass filter, which further reduces ringing and unwanted oscillation on the output.

The shield plate was tested with two configurations. In the first, the shield was located on a separate piece of acrylic, which created a separation between the sensor and shield of 5mm. The second configuration utilized insulating tape to separate the sensor pads, rather than acrylic. This is mimics the approach used in the Texas Instruments guide for active shielding [15]. However, this approach did not show any significant benefit over using a separate piece of acrylic. The primary advantage of the tape-separator was the reduced material cost and thickness of the sensor array.

Tests of the active shield in both configurations showed little improvement in the range and sensitivity of the sensor pads when placed on the floor. In fact, the sensor output showed periodic, random spikes in capacitance that were not present without the active shield. There was no noticeable improvement in SNR. The failure of the active shield is likely due to the presence of noise voltage on the shield plate [16]. A more sophisticated shield driver may be necessary to eliminate this noise.

3.2 Exercise Classification

In order to classify the data produced by the sensor array and retrieved from the FDC2214, the machine learning model must be capable of classifying a multi-

variate, time-ordered data stream. Several machine learning architectures were considered for their ability to satisfy this requirement. The first approach uses a convolutional neural network (CNN). Using this architecture, the stream of sensor data from the capacitive sensor pads is separated into time slices. Each time slice consists of a fixed number, *N*, of input tensors, where each tensor has 12 entries–one for each sensor pad in the mat. This Nx12 tensor of sensor data is the input to the convolutional neural network. The model performs convolutions on the input matrix, extracting features that are useful in classifying the exercises. One limitation of this approach is the requirement of a fixed time slice size, as N is fixed by the model. Since different exercises take different amounts of time to perform, the ideal time-slice size is unlikely to be the same for all exercises.

The second architecture tested was chosen in an attempt to address this problem. That architecture was the LSTM architecture. The LSTM architecture was chosen for its ability to understand long-term dependencies. This feature is desirable when attempting to classify an exercise based on a long stream of time-series data, as the data stream may have a length in excess of 50 samples. Typical RNNs are not capable of preserving useful information when the input sequence length exceeds 5-10 samples.

This project implements both the CNN and LSTM architectures using Keras, then exports the models using Tensorflow Lite for microcontrollers, a lightweight implementation of the popular machine learning library Tensorflow. One limitation of doing classification on a microcontroller is that the model size must be carefully managed to ensure it is sufficiently accurate without requiring too much

of the microcontroller's memory. The Adafruit Feather M4 Express has 512 KB of flash memory [11], so the combined size of the model and the code for running the capacitive sensing must be smaller than 512KB. Also, because the model is initially trained using double-precision floating point parameters and then exported to 8-bit integers (in order to reduce the model size and increase the speed of classification), there is the potential for a significant loss of accuracy during the export process.

Chapter 4

Evaluation Methodology & Results

4.1 Data Acquisition and Processing

Due to the outbreak of the COVID-19 virus and the subsequent closure of research labs on campus, there was no opportunity to collect data from subjects using the previously discussed capacitive sensor array. In order to train and evaluate the machine learning models developed to classify the sensor data, a pre-existing dataset provided by UC Irvine was used. This dataset consists of 30 individuals performing 7 different physiotherapy exercises while being recorded using 4 different sensing modalities [10]. The 4 sensing modalities include a depth camera, a pressure sensing mat, and 2 accelerometers placed on the wrist and thigh.

This dataset was chosen for several reasons. First, the exercises recorded by

Exercise #	1	2	3	4	5	6	7
Exercise Name	Knee-Roll	Bridge	Pelvic Tilt	Clam	Back Extension	Prone Punches	Superman

the dataset include common physiotherapy movements, which is the general subcategory of exercises that this project is targeted towards. Secondly, the sensing modalities used enable the close replication of the kind of data that would be acquired from the capacitive sensing array. The depth camera and pressure mat data can be combined to mimic the proximity detection of a capacitive sensing array.

We chose to use both the depth camera and pressure mat data because they each offer specific information that would be captured by a capacitive sensor array. The depth information provided by the depth camera is analogous to the change in capacitance that is associated with proximity in a self-capacitive sensor. However, because the depth camera is positioned above the participant, some information about the position of the participants' limbs is lost when they are positioned directly under the torso. Additionally, a gym mat combined with a capacitive sensor acts as a rudimentary pressure sensor. The pressure on the mat causes compression in the mat material, which brings the user into closer proximity to the sensor. For these reasons, we also utilized the pressure sensor data. The accelerometer data was not used, because the information it provides does not have a close analogue in the capacitive sensor array.

The first step in processing the dataset was to match depth camera and pressure mat readings by time. Although the depth camera and pressure mat were both reported as sampling at a frequency of 15Hz, inspection of the dataset showed that the depth camera's sample rate was actually approximately 9 Hz. Therefore, it was not possible to simply match depth camera and pressure mat readings 1-1.

Instead, we parsed and stored the timestamps for each depth camera and pressure mat sample, then matched a depth camera sample to each pressure mat sample by finding the depth camera sample with the minimum absolute-value time difference. This approach gives an effective sample rate of 15 Hz for both sensors.

In order to more closely resemble the data that would be acquired from the capacitive sensors developed in this project, the dataset was processed to compute an approximate value for each of the 12 sensor pads of the capacitive sensor array. In the provided dataset, the depth camera recorded video of the participants from the neck down. This video was downscaled to a 12x16 grid of depth value pixels on the range [0,1], with 0 representing the closest pixel to the camera and 1 the furthest away. This is the opposite of the capacitive sensor, which returns larger capacitance values when the participant is close. To correct this, each depth camera value was subtracted from 1.

The pressure mat data was reported as taking values between 0 and 1; however, the provided dataset included floating point numbers with values in excess of 1000. In order to normalize the data, each entry was divided by the maximum recorded value in its data file.

To approximate the discrete sensor pads on the sensor array, the 16x12 array of depth camera values and the 32x16 array of pressure mat values were each grouped into 12 regions. Each region consists of a set of pixel values that would be in approximately the same location as one of the sensor pads in the capacitive sensor array. Each sensor pad region is described by two ordered pairs which represent the (row, column) of the top left corner, and (height, width) of the pad.

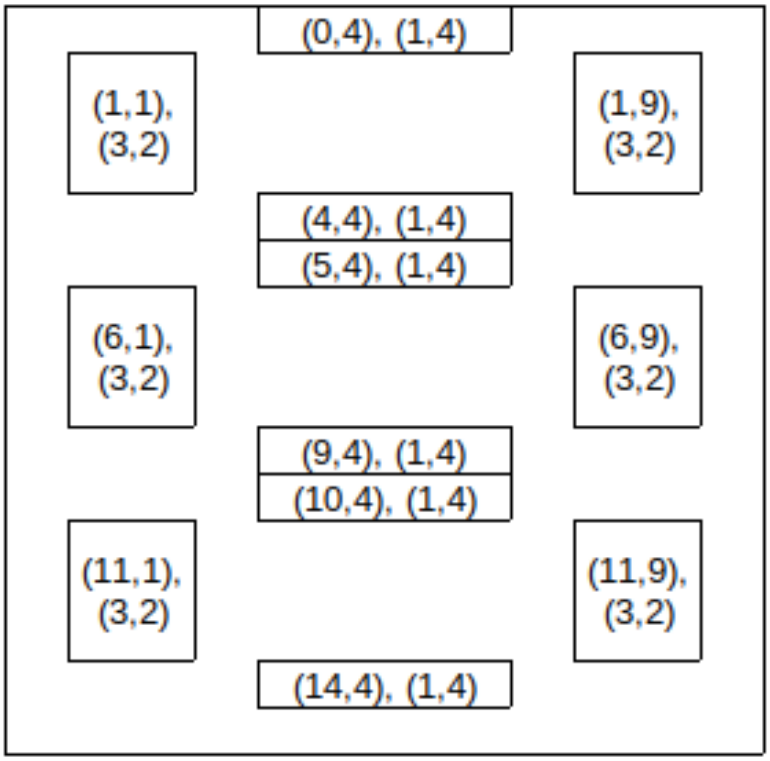

Figure 4.1: Map of capacitive sensor locations onto 12x16 depth camera data

For each sensor pad, data entries that were directly over one of the sensor pads were assigned a weight of 1, while entries that were nearby, but not directly over the sensor pad, were assigned a lower weight. For each entry with a Manhattan distance, d, from the pad, the data value, v, is weighted according to the formula

$$v = \frac{1}{2^d} \tag{4.1}$$

These weighted values were then summed and normalized by dividing by the sum of the weights to produce average depth-camera and pressure mat values

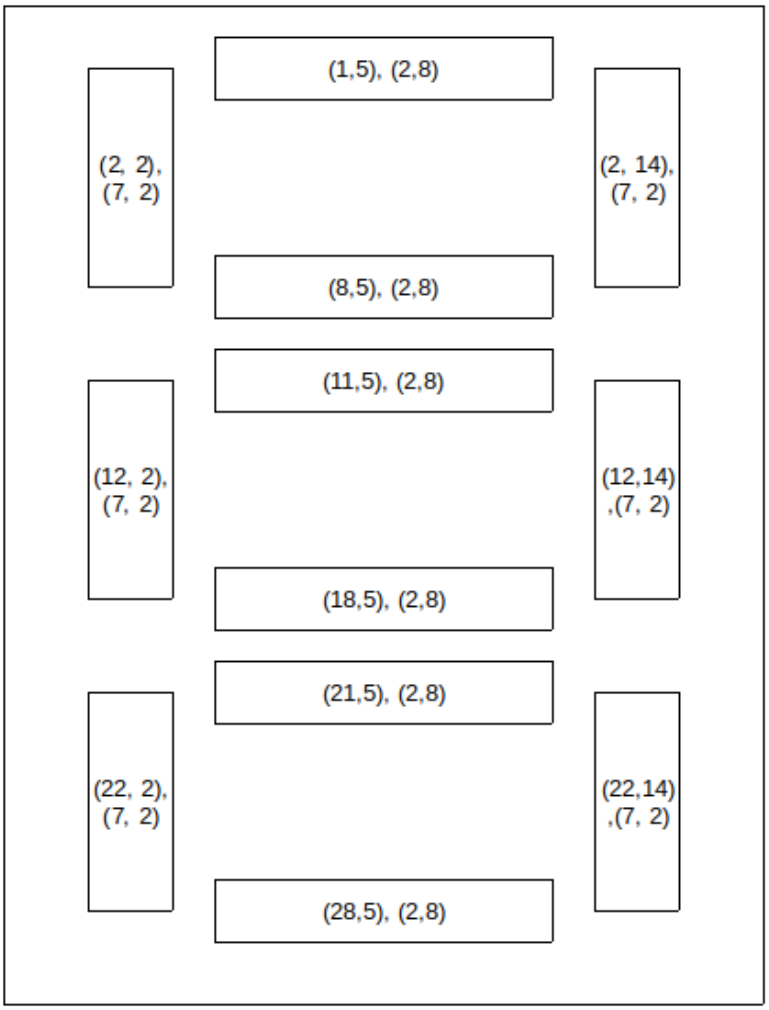

Figure 4.2: Map of capacitive sensor locations onto 32x16 pressure mat data

between 0 and 1 for each sensor pad. The arithmetic mean of the pressure mat and depth camera pad values was calculated for each sensor pad, and these 12 pad values were the output of the data processing stage for each sample in the dataset.

After combining the depth camera and pressure mat samples, the mock capacitive sensor samples were grouped into time slices of 75 samples each. With a 15

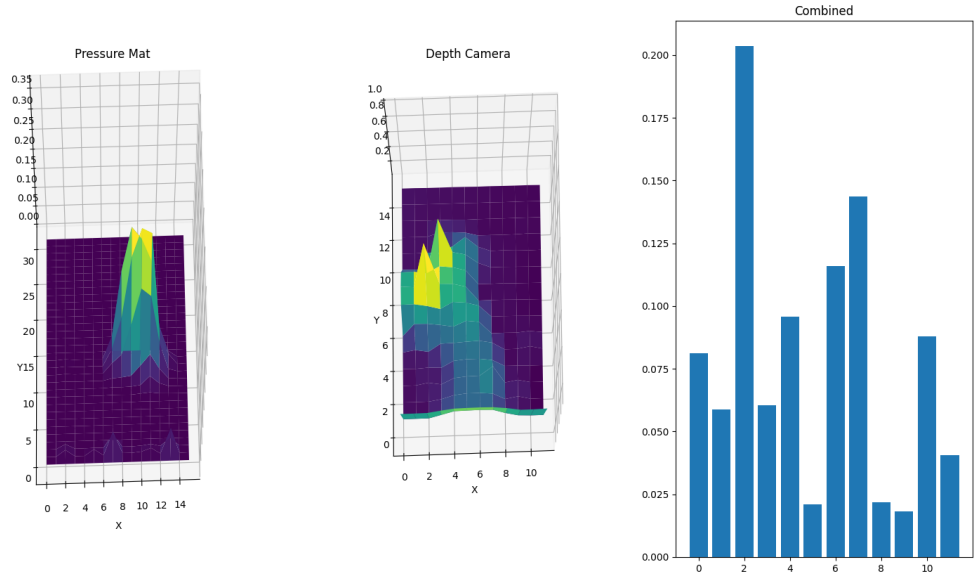

Figure 4.3: Pressure mat, depth camera, and converted mock capacitive sensor data.

Hz sample frequency, each time slice covers 5 seconds of time. We chose a 5 second time slice because some exercises, such as the superman, include instructions to hold a position for up to 5 seconds [10].

4.2 Results

The 75x12 time slices of converted sensor values described in section 4.1 were used as the inputs to both the LSTM and CNN models. The data was separated 80/20 into training and testing sets, with the testing data drawn uniformly from each exercise type. With a time slice size of 75, or 5 seconds, there were 2070 inputs in the training data set, and 518 inputs in the test data set.

4.2.1 LSTM

The LSTM model was trained over 5 epochs using a batch size of 24. The overall model accuracy for the LSTM architecture was quite low, at only 42.5%. Classifying among 7 exercises, this is roughly three times as good as random guessing, but nevertheless unsatisfactory. It is possible that with further tuning and optimizations, this network could achieve a passable accuracy; however, we were not able to find a successful configuration.

4.2.2 CNN

The CNN architecture proved to be much more effective at classifying the mock capacitive sensor data. This model was trained over 10 epochs with a batch size of 24. The model itself consisted of 2 convolutional layers, followed by a max-pooling layer, another convolutional layer, and 2 dense layers. After fitting, the model accuracy was measured to be 89.2%.

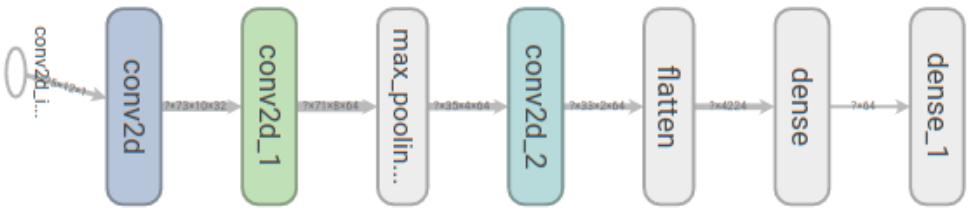

Figure 4.4: Convolutional neural network architecture.

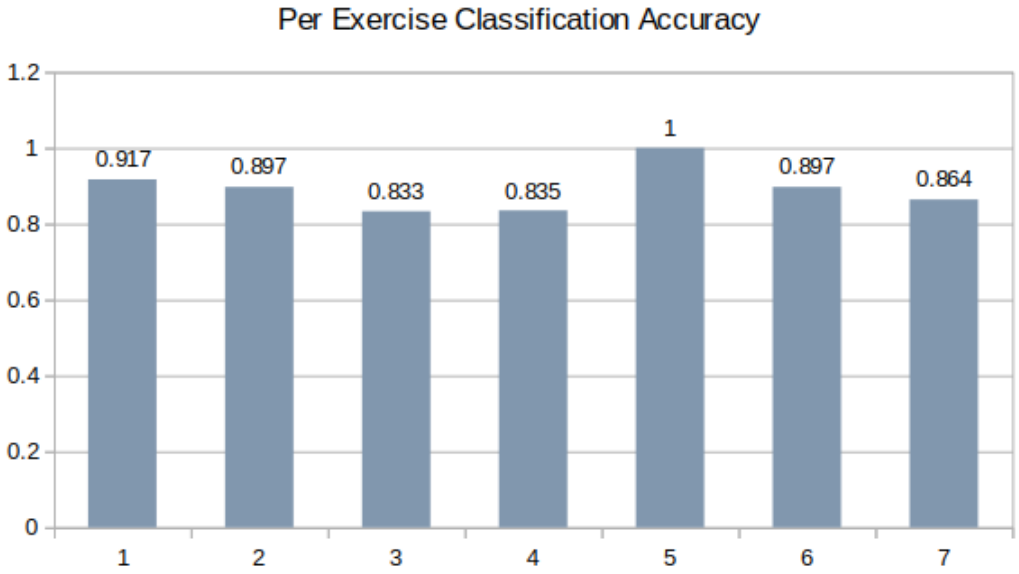

Figure 4.5: Per exercise classification accuracy.

Classification accuracy varied from 83.3% to 100%. The worst performing exercises were the pelvic tilt (83.3%) and the clam (83.5%). The best performing exercise was the back extension (100%). The poor performance of the pelvic tilt is likely due to the extremely subtle movement pattern. The only movement during the exercise is a slight posterior tilt of the pelvis, which would show as a slight decrease in pressure in the area of the glutes and a slight increase in pressure around the lumbar spine. The clam has a more distinctive movement pattern, but was the only exercise in the data set to be performed on 2 sides [10]. The mirroring of some samples from others likely led to its lower classification accuracy.

To further verify the model's effectiveness, we performed leave-one-out cross-validation, using each of the 30 participants as a testing data set. Unsurprisingly, the performance of the model decreased; however, its overall accuracy remained reasonable, at 73.8%. One consequential finding of this analysis is that the standard deviation of the classification accuracy for each exercise is quite high: between 21-44%. Unsurprisingly, the exercises with the most variable accuracy were those with the least distinctive motion pattern (e.g. the pelvic tilt).

Exercise	Accuracy	Std. Dev.
Knee-roll	0.77	0.34
Bridge	0.68	0.37
Pelvic Tilt	0.56	0.44
Clam	0.77	0.26
Back Extension	0.94	0.21
Prone Punches	0.69	0.36
Superman	0.80	0.26
Overall	**0.74**	**0.13**

Table 4.1: Leave-one-out cross-validation results

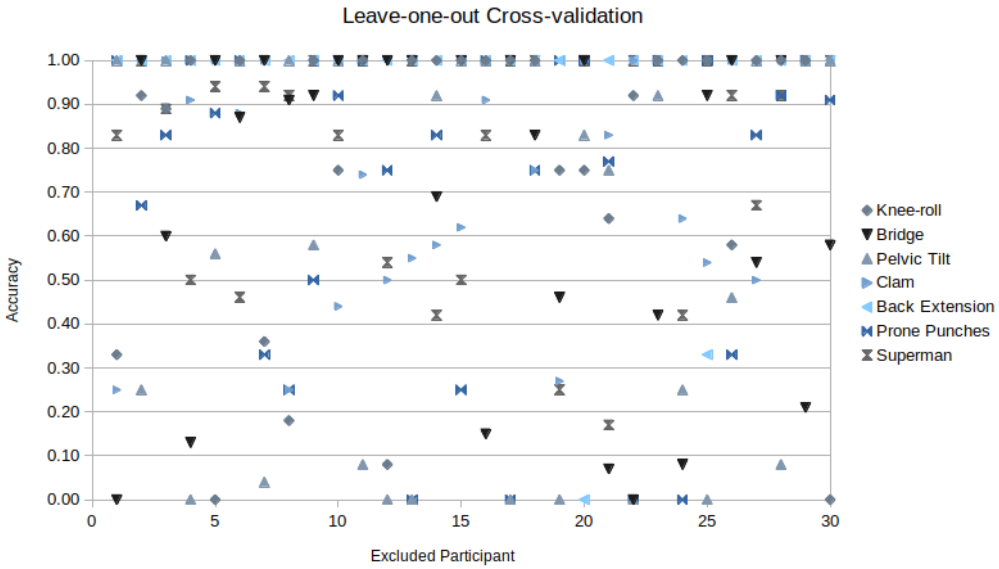

Figure 4.6: Leave-one-out cross-validation results.

4.2.3 Exported Model

Due to the convolutional neural network's significantly better performance than the LSTM network, only the CNN model was exported using Tensorflow Lite (TFLite). The unoptimized, exported model size was over 1.3MB, which is far too large to fit in the 512KB flash memory of the Adafruit Feather M4 Express. However, after applying the default TFLite converter optimizations, the model size shrunk to only 333KB.

Chapter 5

Conclusion

5.1 Summary

Capacitive sensors have the potential to provide a cost-effective, privacy-preserving alternative to other sensing modalities such as cameras and pressure mats in the detection and classification of exercises. Although this project was not able to sufficiently evaluate the efficacy of the capacitive sensor array, it was able to show that the sensor configuration is capable of providing data that is useful in differentiating among different exercises when processed using a convolutional neural network.

5.2 Future Work

Further research is needed to improve the range and sensitivity of the capacitive sensors in the gym mat. Although the sensor configuration used in this project has been reported to have an effective detection range up to 60cm [17], we were unable to replicate these results. This indicates that other techniques may be needed to eliminate noise and other stray parasitic capacitance from the sensor readings. The hardware and software for multiplexing the sensor signals together is also in need of further testing. Although it was fully designed for this project, the lab shutdown occurred before it could be implemented and tested. Additionally, one important component of this application that was not fully addressed in this paper is the counting of exercise repetitions. In order to count exercises, some additional techniques, such as dynamic time warping and exercise templates (as used in [9]) may be necessary.

References

[1] S. Bassett, "The assessment of patient adherence to physiotherapy rehabilitation," *New Zealand Journal of Physiotherapy*, vol. 31, pp. 60–66, 01 2003.

[2] K. Jack, S. M. McLean, J. K. Moffett, and E. Gardiner, "Barriers to treatment adherence in physiotherapy outpatient clinics: A systematic review," *Manual Therapy*, vol. 15, no. 3, pp. 220 – 228, 2010.

[3] S. B. Gaikwad, T. Mukherjee, P. V. Shah, O. I. Ambode, E. G. Johnson, and N. S. Daher, "Home exercise program adherence strategies in vestibular rehabilitation: a systematic review," *Physical Therapy Rehabilitation Science*, vol. 5, no. 2, pp. 53–62, 2016.

[4] "LC Oscillator Basics," Aug. 2015. Accessed on Apr. 6, 2020. Available: https://www.electronicshub.org/lc-oscillator-basics/.

[5] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, "Recent advances in convolutional neural networks," *CoRR*, vol. abs/1512.07108, 2015.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[7] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, Oct. 2000.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, "Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp 14, (New York, NY, USA), p. 373382, Association for Computing Machinery, 2014.

[10] A. Wijekoon, N. Wiratunga, and K. Cooper, "MEx: Multi-modal Exercises Dataset for Human Activity Recognition," *arXiv preprint [Web Link]*, 2019.

[11] "Adafruit Feather M4 Express - Featuring ATSAMD51." Accessed on Jan. 22, 2020. Available: https://www.adafruit.com/product/3857.

[12] zharijs, "FDC2214," Mar. 2018. Accessed on Jan. 29, 2020. Available: https://github.com/zharijs/FDC2214.

[13] "TS5A33591-SP3T Bidirectional Analog Switch 5-V/3.3-V Single-Channel 3:1 Multiplexer and Demultiplexer," Jan. 2016. Accessed on Mar. 2, 2020. Available: http://www.ti.com/lit/ds/symlink/ts5a3359.pdf.

[14] "LTC6228/LTC6229," Sept. 2019. Accessed on Feb. 25, 2020. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/LTC6228-6229.pdf.

[15] D. Wang, "Capacitive Sensing: Ins and Outs of Active Shielding," Feb. 2015. Accessed on Feb. 21, 2020. Available: http://www.ti.com/lit/an/snoa926a/snoa926a.pdf.

[16] A. Rich, "Shielding and Guarding How to Exclude Interference-Type Noise What to Do and Why to Do It – A Rational Approach," *Analogue Dialogue*, 1983. Accessed on Apr. 14, 2020. Available: https://www.analog.com/media/en/technical-documentation/application-notes/41727248AN_347.pdf.

[17] Y. Yu, "Capacitive Proximity Sensing Using FDC2x1y," Oct. 2017. Accessed on Apr. 9, 2020. Available: http://www.ti.com/lit/an/snoa940a/snoa940a.pdf.