**Convolutional Neural Networks: Predicting Human Activity and Measuring Susceptibility to Faulty Data**

CS4991 Capstone Report
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Bryant Chow

May 9, 2024

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS

Briana Morrison, Department of Computer Science

**ABSTRACT**
During my internship at MITRE, I applied convolutional neural networks on human activity recognition data to act as a proof of concept for using machine learning on electromagnetic data of a similar format. Working with my project manager and a co-intern, we used PyTorch to complete our project in three main stages. First, we used PyTorch's custom data loaders to transform human activity recognition time series data into spectrographs with labels, combined the spectrographs into a single image, and then prepared the images to be fed into different ResNets. Next, we loaded ResNets to be applied to the data. The models were tuned using Optuna which pruned and searched for optimal hyperparameters. Last, we simulated error from poor measurements by adding guassian noise to certain spectrographs and then trained the ResNets using different combinations of faulty data to determine when it is most susceptible. We found that the models performed well on data with this format and which situations the model showed weaknesses. Given more time, I would continue this work by using live data or by trying to detect more specific movements.

## 1. INTRODUCTION
The way somebody is moving can give a lot of information about somebody that cannot be seen on the surface. Do they have an injury? Are they stressed or anxious? Are they fatigued? Human activity recognition can be applied to areas such as health, where it can help detect unseen problems or track the effects of injury. It can be used to study someone's psyche during a conversation based on body language or jitters. It can also be applied to sports, where it can be used to study an athlete's form when moving.

Convolutional neural networks are useful at categorizing images by detecting certain defining features within them. If a spectrograph image of time series human activity data was created, could a convolutional neural network detect actions by finding common patterns? How much data would be needed? How complex of a neural network would be needed? A tool able to automatically detect these things would be immensely useful, so my work at MITRE looks to investigate the feasibility of using machine learning as a solution by first using models to detect basic actions.

## 2. RELATED WORKS
In a similar work done by Vrigkas et al. (2015), his team also experimented with using deep learning on human activity recognition. They discuss the potential benefits of being able to detect human activity recognition, just as I do in my work. They used deep learning on video data with a subject performing an action in frame, and tracked the subject to determine motions and actions they were performing. The drawback of their approach compared to mine is that a video feed of a subject is needed for the model to make predictions. This is not only harder to obtain, but it is limited to the angle of capture, which can be obstructed easily. Since my work used direct subject data from devices such as an Apple Watch, it is more direct and has less room for noise; though more nuanced movements may not be detected.

Another similar work was done by Bayat et al. (2014). They also used human activity data from an accelerometer in a smartphone either held or on a person. Their results are comparable, as they used other machine learning models such as random forests or SVMs to do their categorizations. Bayat et al.'s work does not do the same investigation into their model's susceptibility to noise and faulty data collection as my experiments, however.

## 3. PROJECT DESIGN

At MITRE, the purpose of my internship project was to experiment with convolutional neural networks on varying data qualities.

### 3.1 Data Preparation

To start, the human activity recognition data was prepared by using a standardized format to make it useful for training convolutional neural networks. The raw data came as a collection of csv files, each csv file containing a particular subject's results for a specific action trial. Each individual participated in 15 different trials, all categorized by action performed: walking, jogging, standing, sitting, walking up stairs, or walking down stairs. The subject data within each csv is time series data of four measurement types: acceleration, rotation rate, gravitation acceleration, and attitude in the xyz axis. A separate csv file containing the subject's age, sex, weight, height, and more was also provided.

For this project, a custom dataloader was made using PyTorch to turn the csv time series data into a collection of images to be used in convolutional neural networks. First, the dataloader created four RGB images for each csv file by creating a spectrograph for each measurement type. The dataloader then combined the four images by stacking the RGB channels to create a single 12 channel PyTorch tensor representing the original csv file. This tensor was then reduced by averaging the channels by color, converting it to a three channel tensor or a single RGB image. The image was then labeled based on the action performed. Last, the tensor normalized the images based on the requirement of the model it was being fed into. To summarize, the dataloader created a dataset of labeled, normalized, RGB images.

### 3.2 Control Model

Using PyTorch, a Resnet18 model was loaded to be used for control results. The dataset was split into 80 percent training data and 20 percent testing data. The hyperparameters for the Resnet18 model were tuned using Optuna.

### 3.3 Creating Noise

The next phase of the project was to create error within data to simulate measurement failure or noise. This data would be used to measure the effectiveness of a model during the potential failure of gathering accurate data. For example, if one of the devices within an apple watch were to malfunction.

We created two scenarios of missing data, one where the data is known to be incorrect and one where the data is unknown to be incorrect. In both scenarios, the dataloader replaced the measurement types specified to be incorrect with gaussian noise. When the model is unaware that the data is incorrect, all measurement types are taken into account during the averaging of the RGB layers; therefore, the incorrect data still has an effect on the resulting image. When the model is aware of the incorrect data, the incorrect measurement is omitted from the averaging; therefore, the model creates its predictions on the safe data.

### 3.4 Running Tests

The complete set of tests were completed again using Resnet18 tuned by Optuna. There was a test case for each scenario when a measurement type was missing for both situations where it was unknown and where known data missing. Finally, we conducted a set of test cases for a combination of the previous ones, though these tests were not complete by the end of my internship. The goal of running these tests was to find the extent of Resnet18s effectiveness for this data and to explore which measurement types were relatively critical.

## 4. RESULTS

Our control results with the initial data produced a testing accuracy of 98.6%. The loss and accuracy over epochs for our control test is seen in Figure 1.


Figure 1: Control Test Loss and Accuracy over Epochs

| Unknown Missing Measurement | Accuracy |
| --- | --- |
| Attitude | ~100% |
| Gravity | 92.59% |
| Acceleration | 88.89% |
| Rotation Rate | 92.59% |

Table 1: Unknown Missing Measurement Test Accuracies

| Known Missing Measurement | Accuracy |
| --- | --- |
| Attitude | 96.3% |
| Gravity | ~100% |
| Acceleration | 98.15% |
| Rotation Rate | 98.15% |

Table 2: Known Missing Measurement Test Accuracies

The results of the unknown missing data tests are seen in Table 1 and the results of the known missing data tests are seen in Table 2. From these initial results, it is seen that when the data is known to be missing, the model performs better as the noise does not incorrectly affect the images. These results show that three of the four measurement types being available is sufficient to yield passable results.

The results were inconsistent and no assumptions were made about which measurement types were relatively critical. Furthermore, the resulting accuracies of the tests were suspiciously high, meaning that our model is possibly overfitting or that there is an insufficient amount of data. Not only do the accuracies of the models used with corrupted data increase when compared to the control, which is highly unexpected, but also some tests, such as that achieved when the gravity measurement was unknown to be missing, have seem to have impossibly high accuracies. Due to these concerns, further testing is required to check for overfitting; this can be done by first increasing the number of epochs to see if loss stops to drop and starts to increase which would indicate overfitting. Also, we could rerun the tests using cross validation on the dataset to further make the model overfitting resistant. If these tests show no overfitting, it would then be beneficial to rerun the experiments after more data is gathered. Once this extra experimentation is carried out, we could be more comfortable with the associations made from the data. The current loss and accuracy over epochs for the unknown missing gravity test can be seen in Figure 2.

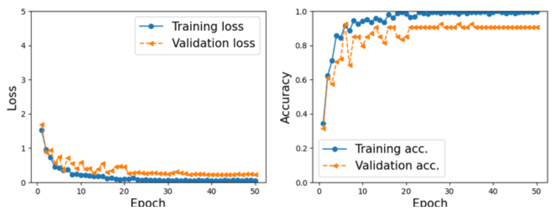


Figure 2: Unknown missing gravity test loss and accuracy over epochs

## 5. CONCLUSION

My work at MITRE has produced basic results which can be used as a foundation for more complex human activity recognition projects. The results show the capabilities of CNNs and how reliable they are against erroneous data. With this, others can decide whether or not they believe pursuing human activity recognition in such a way is currently feasible. Furthermore, building upon my work can lead to useful projects in the fields of sports, medicine, security and more.

## 6. FUTURE WORK

This project can be built upon and used as a framework for more complex uses. The convolutional neural networks can be used on more complex or specific datasets catered to certain purposes. The data loader that replicates error by implementing noise can also be reused on other projects.

The convolutional neural networks could also be subjected to live testing. For further exploration, my models could be fed live data from a device such as an Apple Watch to give real results instead of simulated results.

## REFERENCES

Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A review of Human Activity Recognition Methods. *Frontiers in Robotics and AI*, *2*. https://doi.org/10.3389/frobt.2015.00028

Bayat, A., Pomplun, M., & Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. Procedia Computer Science, 34, 450–457. https://doi.org/10.1016/j.procs.2014.07.009