Neural Network for Detecting Head Impacts from Kinematic Data

Michael Fanton, Nicholas Gaudio, Alissa Ling
CS 229 Project Report

1. Abstract

Mild Traumatic Brain Injury (mTBI) is a serious health concern, especially in contact sports such as football, and can cause acute and long term debilitating symptoms. The Camarillo Lab at Stanford has developed and deployed an instrumented mouthguard that records linear acceleration and angular velocity of head impacts in contact sports. However, to be useful, the device must be able to accurately classify between real impacts or false impacts (i.e. players spitting, chewing, etc.). In previous work, sequential feature selection was used to determine the most important classifier features to train a SVM classifier that classified between impacts and non-impacts in mouthguard data. We propose to use a neural network, which will automatically extract important features to distinguish between real and false impacts to a higher degree of accuracy. A low parameter neural network worked surprisingly well, achieving a 98.2% accuracy, 97.6% precision, 96.7% specificity, and 99.3% sensitivity, and out performed an existing SVM head impact classifier.

2. Introduction

Mild traumatic brain injury (mTBI), more commonly known as concussion, has become a serious health concern with recent increase in media coverage on the long term health issues of professional athletes and military personnel. Acute symptoms include dizziness, confusion, and personality changes which can remain for days or even years after injury [1]. Further, recent studies have shown that repetitive mTBI can lead to long-term neurodegeneration and increase the risk of diseases such as Alzheimer's and Chronic Traumatic Encephalopathy [2]. Although the mechanisms of this injury are not well understood, studies have shown that one of the biggest risk factors for mTBI is a history of prior mTBI [8]. Further, concussion symptoms are more severe with a longer recovery time if an individual does not rest after injury [9]. Therefore, it is imperative that individuals who have been suspected to have received a TBI be immediately removed from risky situations.

According to the CDC, contact sports such as football are one of the leading causes of mTBI. In these sports, mTBI is diagnosed by a sideline clinician through subjective evaluation of symptoms and neurological testing. Because of the large variance of symptoms within different individuals, and the pressure of athletes to return to play, mTBI can often be missed by these tests [11]. In efforts towards developing an objective diagnostic tool for concussion prevention, the Camarillo Lab at Stanford University created an instrumented mouthguard that rigidly connects to the upper dentition to record the linear acceleration and angular velocity of head impacts in six degrees of freedom. Whenever the linear accelerometer measures a signal of over 10g of acceleration, the device will trigger and record 200 ms of impact data. However, one of the primary challenges of this device is that it is prone to false positives, with non-impact events such as chewing, spitting, or dropping the mouthguard often triggering the device. In order for this device have promise to be used as a diagnostic tool in the future, it must be able to accurately classify between real impacts and false positives. Currently, this is done after the game; a research assistant will tediously watch hours of video footage time synced with the mouthguard, and each impact is manually labeled as a real impact or false positive. However, a machine learning classifier should be able to automatically differentiate between real and false impacts to a high degree of accuracy, as the kinematic data between these two impact types typically look distinct, as shown in Figure 1 in both the time and frequency domain.

In this project, our goal is to train a neural network, which will automatically extract relevant features, to classify between real impacts and false positives. The input to our algorithm is mouthguard time series data.

3. Related Work

Currently, there are a number of sensor systems used for measuring head impact kinematics in contact sports. Many of these systems use a simple linear acceleration threshold for differentiating impacts and non-impacts; however, this leaves the device prone to a large number of false positives. Many companies and research groups are developing proprietary algorithms for detecting impacts, but little has been published validating their accuracy [4]. The state-of-the-art for this problem is recent work from the Camarillo Lab, in which an impact classifier using a sequential feature selection was used to determine the most important classifier features (e.g. time domain features, power spectral density features, etc.), and these features were used to train a support vector machine [3,4]. While the results of this work were promising, achieving 87.2% sensitivity and 93.2% precision on a collegiate dataset, recent advances and increased adoption of neural nets for

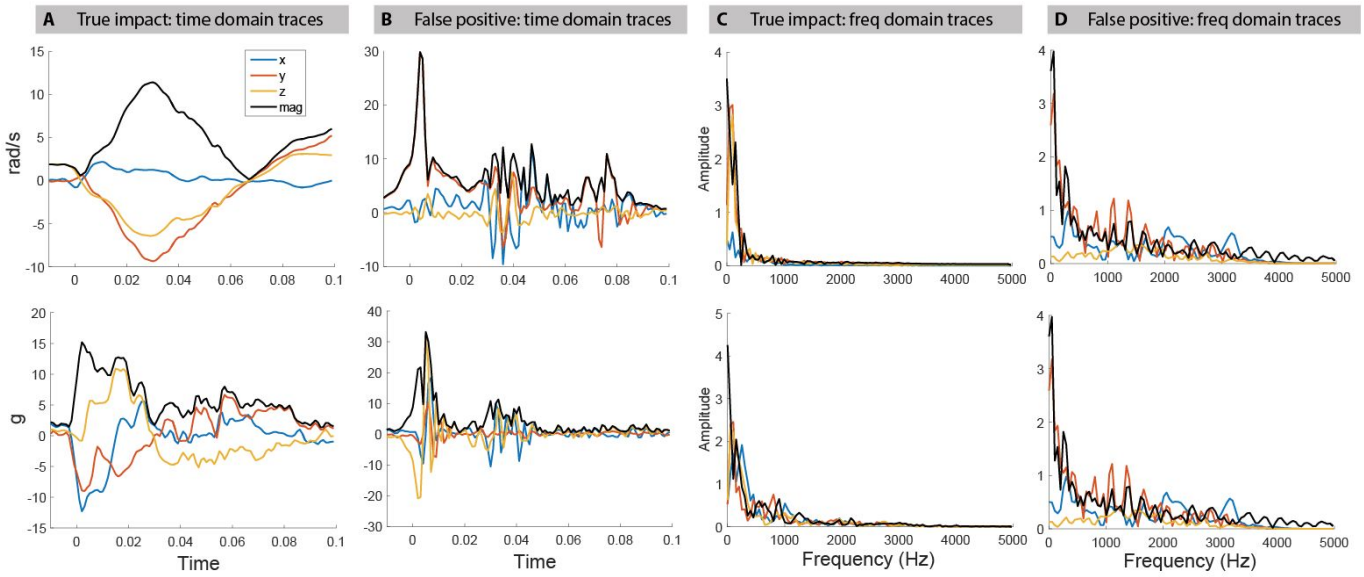

Figure 1: ***Example time and frequency domain traces from true and false impacts.*** *A single representative true and false impact time trace is plotted. Real impacts tend to be comprised of lower frequency signals, while false positive impacts (due to biting, spitting, chewing, etc.) tend to have higher frequency content.*

detecting human activity have shown this could be a promising approach for this application. Further, as the Camarillo Lab begins to disseminate their device around the country to different colleges and high schools, it is expected that there will soon exist a significantly larger dataset of real and false positive impacts to work with, and a neural network should perform better than SVM in predicting a non-labeled impact or nonimpact on a large dataset.

To the best of our knowledge, only one study has attempted to use a neural network algorithm for detecting head impacts and non-impacts from kinematic sensor data; this study used a simple net with a single fully-connected layer, and only achieved 47% specificity and 88% sensitivity on their dataset of soccer athletes [5]. Convolutional neural nets (CNN) have been used to great success for Human Activity Recognition from accelerometer and gyroscope time series data. Ronao et al. [6] developed a deep CNN to detect human activity from smartphone sensor data, and was able to achieve a classification accuracy of 95.75% on differentiating between six different human activities. PerceptionNet [7] improved upon this performance, achieving an accuracy of 97.25% on the same dataset. These results suggest that deeper CNN's could have merit for detecting head impacts, which could be considered a simpler, binary classification of human activity from kinematic data. CNN's have also been broadly useful for classification, localization, and recognition tasks, such as biomedical image segmentation, with a limited dataset [15].

4. Dataset and Features

Our dataset is 527 examples of which half are labeled real (or true) impact and the other half are labeled as false impacts. The dataset was obtained by instrumenting Stanford football athletes over the Fall 2017 season with the Camarillo Lab instrumented mouthguard. To obtain the ground truth labels for the dataset, videos of each game and practice, time synced with the mouthguards, were analyzed according to the methodology outlined by Kuo et al [13]. Through manual video analysis, the time of every helmet contact event was noted. These helmet contact events were then matched with the instrumented mouthguard sensor impacts. Mouthguard events with helmet contact clearly identifiable in video were labeled as real impacts. Mouthguard events in which there was clearly no helmet contact were labeled as false positives. Mouthguard events in which the view of the player in video footage was obscured or unidentifiable were discarded and not used in the dataset.

We split our dataset up into 70% for training, and 15% for both evaluation and testing for when leveraging K-fold cross validation and 70% for training and 30% for evaluation for when training our selected architecture. Each example has dimension 199x6 comprised of 6 time traces of length 199 (200 ms). The six time traces are the linear acceleration at the

head center of gravity in the x, y, and z axes, and angular velocity of the head in the x, y, and z anatomical planes. The data was sampled with a time step of 1000 Hz, with 50 ms recorded pre-trigger and 150 ms post-trigger for 299 data points. Figure 1 shows the time and frequency characteristics of one representative impact and non-impact. True impacts generally are comprised of lower frequency content, while false impacts have much higher frequency content, which is supported by Figure 1. Intuitively, this makes sense, as biting or dropping the mouthguard would likely result in a high frequency noisy signal, while football head impacts typically have frequency content in the 20-30 Hz range [10].

Data was pre-processed using standardization by subtracting out the mean of each sensor's values and dividing by the standard deviation.

5. Methods

A convolutional neural network is a class of deep neural networks comprised of convolutional layers. In convolutional layers, each sensor measurement is convolved with a weighting function w. In the case of a 1D input to a 1D convolutional layer, the i^{th} product element can be found as follows:

$$c_i = b + \sum_{d=1}^{D} w_d x_{i+d-1}$$

Where b is the bias term, D is the filter width, and w_d are each of the filters. In the case where the input to the 1D convolution is a multi-channeled, such as our application where we are stacking six input signals, the output of the convolution each channel is added to give the following:

$$c_i = b + \sum_{s=1}^{S} \sum_{d=1}^{D} w_d x_{i+d-1}$$

Where s is the number of input channels (in our case, six). The output of a 1D convolutional layer is a single vector. In 2D convolutional layers, this process is repeated in two dimensions, providing a two dimensional output. Convolutional neural networks commonly have pooling operations, which combine outputs of neuron clusters at one layer into a single neuron in the next layer. Max-pooling layers use the maximum value from a specified cluster of neurons, while average pooling uses the average of a specified cluster. Further, dropout layers can be added to help prevent overfitting; a dropout layer will randomly ignore a certain percent of the layer interconnections during training.

We investigated multiple different convolutional neural network architectures using Keras and Tensorflow written in Python; specifically, we developed both sequential models and recursive network models. In investigating proper model architecture, we utilized the K-fold cross validation technique (k=10) as we knew that 527 examples is not a very large amount and gathering more data was not feasible within the scope of this project. In training all of our networks, the number of epochs was increased indefinitely, until five consecutive epochs did not result in an improved evaluation binary cross entropy loss. Following completion of training, the model at the end of the epoch with the lowest evaluation loss was saved and used for analysis.

We developed and compared two primary architectures. The "RecursiveNet" model has the most convolutional layers as seen in Figure 2a. In a recursive architecture, inputs to later hidden layers are concatenated with outputs of earlier hidden layers. The idea

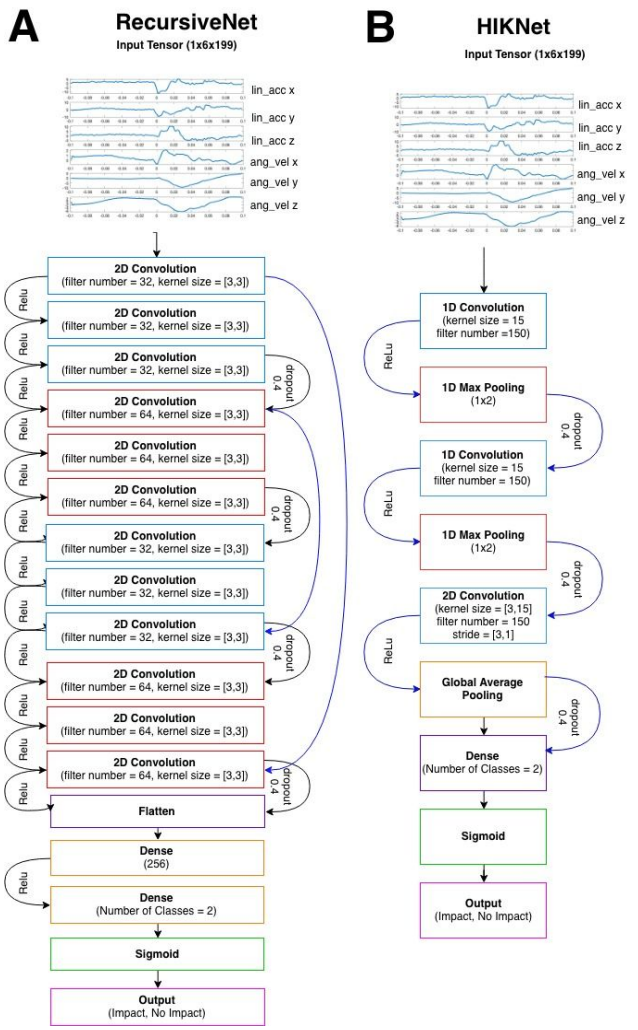

*Figure 2: **A) RecursiveNet architecture schematic. B) HIKNet architecture schematic.***

behind leveraging a recursive model was to further prevent overfitting of our small dataset in a very deep architecture. The dimensional representations of the input data were held constant across the hidden layers of the network (using padding). Our last model, "HIKNet" is a sequential convolutional neural network with architecture shown in Figure 2b. In this architecture, 1D convolutional layers feed into a late 2D convolution. This model is based off of PerceptionNet [7]. The intuition behind this structure is that the 1D convolution acts to extract high-level features of the motion signal, feeding into a 2D convolution which fuses the sensor signals together. The late 2D convolution helps to prevent overfitting of the data. No layers in the HIKNet were padded as the network is not as deep and thus lower dimensional representations proved not only permissible but also beneficial for classification applications.

6. Experiments/Results/Discussion

In all testing, the metric we optimized for was accuracy, but we also performed tests on precision, specificity, and sensitivity. The equations for the metrics are described below, where *TP* is "true positive," *TN* is "true negative," *FP* is "false positive", and *FN* is "false negative."

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad precision = \frac{TP}{TP + FP} \qquad specificity = \frac{TN}{TN + FP} \qquad sensitivity = \frac{TP}{TP + FN}$$

Using our baseline hyperparameters in preliminary testing, we found that the HIKNet had comparable accuracy to RecursiveNet at a much lower computational cost. Thus, we focused our hyperparameter tuning on the HIKnet architecture.

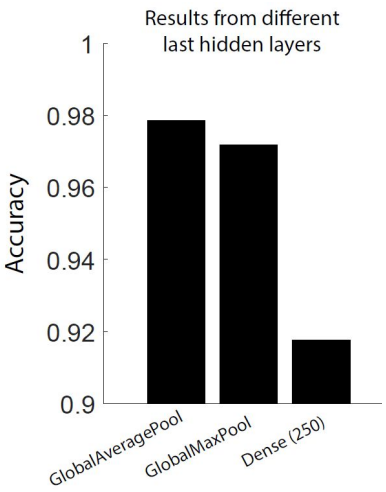

Figure 3: ***Last layer selection effect on accuracy performance.*** *The last layers tested were global average pool, global max pool, and a dense layer.*

We tuned the final HIKNet using a "greedy" optimization scheme for number of 1D conv layers, 2D conv layers, and type of final layer. Because our parameters were initialized to random values, and convergence is highly dependent on weight initialization [7,14], each experiment we did was repeated 10 times and metrics were averaged over those trials. We tested 1, 2, 3, and 4 1D conv layers and found that difference in performance was minimal. The number of 2D convolutional layers also did not make a significant difference in the performance metrics. Thus we chose two 1D conv layers and one 2D conv layer because having less parameters increased the speed of the net. For our last hidden layer, we tested a Global Average Pool, Global Max Pool, and Dense layer with 250 nodes. As shown in Figure 3, Global Average Pooling resulted in the best accuracy because it emphasizes the connection between features maps and categories, and it has no parameter to optimize, thus prevents overfitting. The Dense layer has fully connected layers which can be prone to overfitting and thus not generalizable [12].

We also did a parameter sweep to find the optimal filter size, kernel width, and dropout threshold. The filter size was changed between the values of 15 and 200, the kernel width was between 0 and 50, and the dropout threshold was swept between 0 and 0.6. The optimal dropout threshold was found to be 0.4, kernel width was 15, and filter number was 150. We found that the optimal kernel width and dropout threshold for HIKNet was the same for the PerceptionNet [7]. We believe we did not overfit our data because we utilized dropout layers and optimized for the epoch number with the lowest evaluation loss.

The final performance metrics are summarized in Table 1. For our final version of HIKNet, we also computed the area under the receiver operator characteristic curve (AUC_{ROC}) and area under the precision-recall curve (AUC_{PR}). The ROC curve plots true positive rate against false positive rate at different thresholds; an ideal classifier would have an area under the curve of 1.0. The PR curve plots precision against recall (sensitivity) at various thresholds; likewise, a perfect classifier would have an area under the curve of 1.0. Compared to the SVM classifier [3,4] trained on our dataset, HIKNet had higher performance metrics on the same time series data set. This is probably because the SVM needed manual feature extraction which inherently has error associated with it, whereas the neural net automatically detected the best features. However, there are still some advantages of using the SVM because there is physical intuition of what the

features are in the SVM, whereas in the neural net, the features are a mix of unknown parameters and the algorithm is a black box.

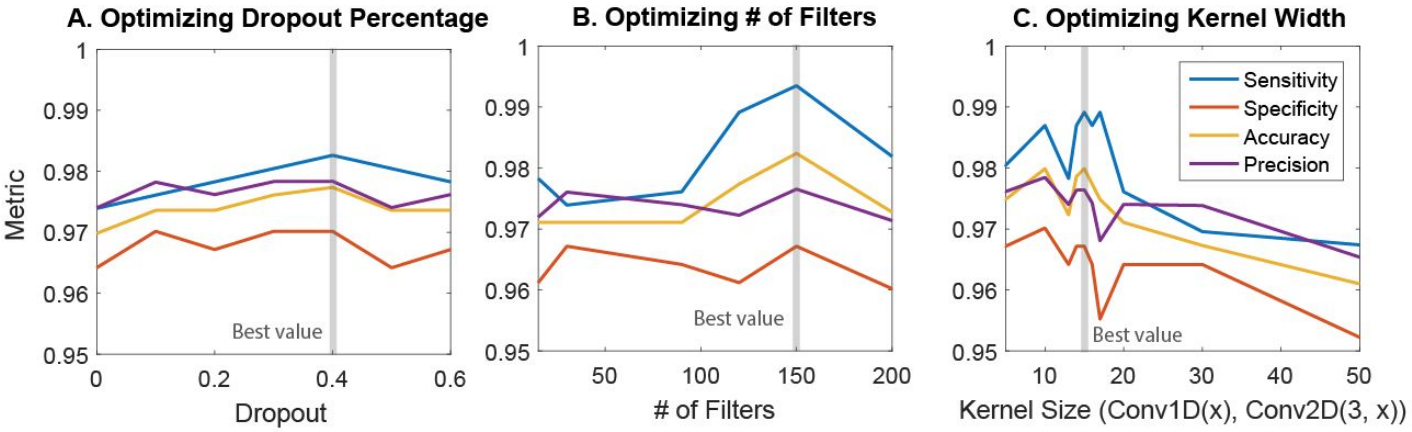

Figure 4: ***Optimization metrics.*** *Dropout percentage, filter number, and filter width were optimized in a parameter sweep.*

7. Conclusion/Future Work

In conclusion, a low parameter neural network performed very well and achieved better performance metrics than the existing SVM classifier trained on the same mouthguard time series data set. We created two deep convolutional neural networks, one that was recursive and one that was sequential, and although both performed similarly, we chose the HIKNet because it had fewer parameters and a higher confidence in its ability to generalize to other kinematic time series datasets than the RecursiveNet based on our literature search. We used a greedy optimization scheme to build the architecture of HIKNet, and did a parameter sweep to find the optimal filter size, kernel width, and dropout percent.

As an immediate next step, we can apply our neural networks to a larger mouthguard dataset as more data is collected in the Camarillo Lab to further tune parameters. In future work, we can use the same mouthguard and video impact footage to create a dataset with more specific labels, i.e. where the impact was located on the head, body impact, or no impact. Using this data, we could create a softmax classifier to predict whether an impact occurred and where it occurred on the head and body. Lastly, once more concussion data is obtained, we could create a neural network that could detect whether an impact occurred and predicts if the impact resulted in a concussion or not. This would require additional data beyond just the mouthguard data such as clinical diagnoses and medical records. The ultimate goal would be to have a device that could instantly tell if an impact resulted in concussion; although it may take years to obtain the dataset needed to train this classifier, the performance of our network architecture gives promise that this could be possible using a similar methodology as put forth in this work.

Table 1: ***Performance comparison of the HIKNet and SVM [3,4] classifiers.*** *The metrics compared were accuracy, precision, specificity, sensitivity, and the areas under the receiver operating and precision-recall curves. HIKNet outperformed the SVM classifier in all metrics on the current dataset.*

	Accuracy	Precision	Specificity	Sensitivity	AUC_{ROC}	AUC_{PR}
HIKNet	98.2%	97.6%	96.7%	99.3%	0.993	0.994
SVM [4,5]	93.7%	92.3%	92.8%	94.6%	0.963	0.926

8. Contributions

Michael Fanton - Developed HIKNet neural network architecture in Keras, set up architecture optimization, helped with statistical analyses, provided background information
Nicholas Gaudio - Lead the insight into Keras and the model architecture setup, created the RecursiveNet, setup auto epoch stopping and saved the best epoch model, conducted experiments to find the optimal filter width and filter number.
Alissa Ling - Preprocessed data, wrote the K-fold function, optimized the dropout threshold, lead the final poster, wrote first draft of sections.

9. References

1. Langlois, Jean A., Wesley Rutland-Brown, and Marlena M. Wald. "The epidemiology and impact of traumatic brain injury: a brief overview." *The Journal of head trauma rehabilitation* 21.5 (2006): 375-378.
2. Ramos-Cejudo, Jaime, et al. "Traumatic Brain Injury and Alzheimer's Disease: The Cerebrovascular Link." *EBioMedicine* (2018).
3. Wu, Lyndia C., et al. "A head impact detection system using SVM classification and proximity sensing in an instrumented mouthguard." *IEEE Transactions on Biomedical Engineering* 61.11 (2014): 2659-2668.
4. Wu, Lyndia C., et al. "Detection of American football head impacts using biomechanical features and support vector machine classification." *Scientific reports* 8.1 (2017): 855.
5. Motiwale, Shruti, et al. "Application of neural networks for filtering non-impact transients recorded from biomechanical sensors." *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on*. IEEE, 2016.
6. Ronao, Charissa Ann, and Sung-Bae Cho. "Human activity recognition with smartphone sensors using deep learning neural networks." Expert Systems with Applications 59 (2016): 235-244.
7. Kasnesis, Panagiotis, Charalampos Z. Patrikakis, and Iakovos S. Venieris. "PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion." *Proceedings of SAI Intelligent Systems Conference*. Springer, Cham, 2018.
8. Lasry, Oliver, et al. "Epidemiology of recurrent traumatic brain injury in the general population: A systematic review." *Neurology* 89.21 (2017): 2198-2209.
9. Williams, Richelle M., et al. "Concussion recovery time among high school and collegiate athletes: a systematic review and meta-analysis." *Sports medicine* 45.6 (2015): 893-903.
10. Laksari, Kaveh, et al. "Mechanistic Insights into Human Brain Impact Dynamics through Modal Analysis." *Physical review letters* 120.13 (2018): 138101.
11. Albicini, Michelle, and Audrey McKinlay. "A review of sideline assessment measures for identifying sports-related concussion." Journal of concussion 2 (2018): 2059700218784826.
12. Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).
13. Kuo, Calvin, et al. "Comparison of video-based and sensor-based head impact exposure." *PloS one* 13.6 (2018): e0199238.
14. Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010.
15. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Code for final project:

https://drive.google.com/file/d/1rj8O4d13DrKCyMT8792yOdQQPY6ChnGn/view?usp=sharing