

WHICH HAND? TREMOR AVERAGE AREA AND PEAKS

ELEN4012A – EIE Investigation 2022 – Jesse van der Merwe (1829172)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This investigation focuses on the development and implementation of two methods used to produce a tremor severity rating that measures the tremor reduction after Focused Ultrasound Treatment on a patient's treated and untreated hand. This is to attempt to quantitatively determine whether the treatment is successful using image processing and computational analysis. This would remove the need for an expert physician and provide a more accurate and less biased result. This paper focuses on image pre-processing, as well as methods 2A (average area) and 2B(peak-trough distances). The results show that these methods reliably indicate the success of the treatment per patient and overall, by using the tremor severity ratings to determine whether the tremor has decreased in either hand.

1. INTRODUCTION

Focused Ultrasound Treatment is used to treat Parkinson's disease and essential tremor. This investigation aims at quantitatively investigating the efficacy of this treatment using computational and statistical analysis of spiral drawings. The aim is to provide insight about the extent that these results can be used to assess the severity of the tremor on the patient's treated vs untreated side after the treatment. This will help determine whether the treatment is successful in reducing the tremor or slowing the progression of the disease without the need for a medical practitioner with potential bias. This document details the various methods investigated and developed, including the pre-processing of the provided template data, as well as the analysis of the data and resulting quantitative values.

2. LITERATURE SURVEY

2.1. Focused Ultrasound Treatment

Focused Ultrasound Treatment (FUS) is a new and promising non-invasive treatment for movement disorders that cause involuntary rhythmic movements of body segments. Two such disorders, that produce similar symptoms, include Parkinson's disease (PD) and essential tremor (ET). Sound waves, which contain acoustic energy, are delivered through the physical barrier of the brain to create lesions on targeted brain tissue [1]. By creating these lesions on the part of the patient's brain responsible for the communication of sensory and motor signals, abnormal brain activity is interrupted, which reduces uncontrollable movements with immediate effect [1]. This reduces the unwanted tremor caused by PD or ET. This is a unilateral treatment since FUS is only performed on one side of the brain (usually on the dominant hand side [2]). This treatment shows immediate reduction in tremor on the treated side of the body.

2.2. Hand-drawn Shapes

The observational analysis of hand-drawn shapes by a neurologist is widely used as a severity test of movement disorders [3]. Analysis of handwriting is avoided due to stylistic differences skewing the test results [4]. An Archimedes spiral is often used as it captures the frequency, amplitude and direction of a tremor [4]. Long,

straight line drawings offer similar results. These drawings require one continuous pen motion and thus they are able to emphasise the abnormal movements specific to movement disorders [4]. The typical characteristics of tremor types can be found in [4], Tab. 1]. This table shows that ET and PD have similar characteristics, however, previous studies have shown that computational analysis of such drawings can reliably discriminate between ET and PD [3]. Further, the combination of traditional and computational analysis has provided significant progress in the classification of disease severity [5].

2.3. Existing Methods

Analysis of patterns in tremor diagnosis spiral drawings for automated classification [6]: This method focuses on finding the relative orientation of all pixels in a hand drawn Archimedes' Spiral. This approach is insensitive to drawing errors and produces accurate results. This approach is the basis for method 1 of this project.

Application of machine learning and numerical analysis to classify tremor in patients affected with essential tremor or Parkinson's disease [7]: Three methods try to match the qualitative clinical tremor ratings. 1) Digital pen spiral tracing analysis. 2) Analysis of a gyroscopic 'Shimmer' device data using RMS methods. 3) Analysis of the 'Shimmer' device data using machine learning decision tree algorithm. These methods matched the clinical tremor rating 78%, 42% and 82% respectively.

Quantification of tremor with a digitizing tablet [8]: Using a commercially available digitizing tablet, the tremor introduced by drawing an Archimedes spiral or writing words can be analysed. The quantification includes acceleration, peak-to-peak amplitude, velocity, power, and displacement. This approach is the basis for method 2 of the project, without the luxury of a digitising tablet.

Quantification of the drawing of an Archimedes spiral through the analysis of its digitized picture [9]: Spirals are analysed after reconstructing the temporal sequence of an Archimedes spiral drawing. Method 1) Cross-correlation coefficient with spiral template. Method 2) Mean and standard deviation of distance between each point of the spiral drawing and the corresponding point of the model. Method 3) Fourier Transform of reconstructed spiral.

3. PROJECT PLAN

At the commencement of this project, the submitted investigation Project Plan, found in Appendix 3, was discussed with Professor Aharonson, the supervisor of this project. It was quickly realised that the plan did not correctly cover the scope or complexity; no machine learning would be required, since image processing and computational analysis would be enough to answer the investigation question. Thus, a new project plan, scope and schedule was created. The corrected Gantt chart can be found in Appendix 7.

4. DATA

4.1. Database access and ethical clearance

Fully anonymised data of patients with either PD or ET was provided by the Rambam Medical Centre, Haifa Israel. Permission to use this data was subject to the obtainment of ethical clearance from the University of Witwatersrand. The Clearance Certificate for this project as well as the permission letter from Dr Schlesinger can be found in Appendices 5 and 6 respectively.

4.2. Patients

Out of the 122 patients, 34 are undergoing treatment for PD and the remaining 88 for ET. This investigation does not require any differentiation between diseases, but rather whether the tremor is reducing and treatment is working.

4.3. Data

The database consists of templates that are filled in with both the treated and untreated hands before and after receiving treatment. Each patient uses a pen to physically fill in a template. This is then scanned and saved as a PDF. Each template consists of two Archimedean spiral drawings (spiral A and spiral B), and multiple straight-line drawings (line-block C) as shown in figure 1A.

5. PRE-PROCESSING OF DATA

5.1. Original Spiral Isolation Code

Kelvin da Santos developed a pre-processing python script specifically for isolating the spirals for his master's research [10]. After meeting with him, permission was granted to use and further improve this code to better suit the needs of this project. His code is based on the 'OpenCV Text Detection (EAST text detector)' article by Adrian Rosebrock [11] which makes use of a text detector library to detect the labels "Drawing A", "Drawing B" and "Drawing C". This returns the four coordinates of each corner of these text areas, saved in a list. This process can be seen in figure 1B. From here, the width of each spiral was approximated using the start positions of "Drawing A" and "Drawing B". Finally, using crop tolerance values and further manipulation, each of the two spirals were cropped and saved as new images – as seen in figure 1C.

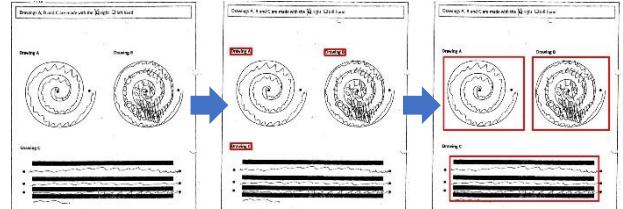


Figure 1: A) Original filled in template, B) text detection and C) image isolation and cropping procedure.

5.2. Improvement of Isolation Code

Too many incorrectly cropped spirals: Most of the scanned drawings are pixelated, slightly rotated, or contain erroneous pen or other markings. This causes the text detection code to sometimes incorrectly identify erroneous markings as text, which leads to inaccurately cropped spirals. This further causes the results of the image analysis to become skewed due to extreme data outliers. Since there are 122 patients, with each patient having anywhere between 2 and 20 scanned drawing, it was imperative to correct these inaccurately cropped spirals within the code. Thus, much time was spent improving the pre-processing code by adding more error detection and correction before the spirals were cropped. It is vital that as many spirals and line-drawings are isolated and accurately cropped to have as much useable data as possible.

Missing line-drawing C: This code did not extract the line-block C section of the templates. Thus, it was expanded to do so, since methods 2A and 2B require this line drawing.

Performing logic checks: To reduce the erroneous cropping, more logic checks were implemented. This included deciding whether the coordinates of the detected text matches the approximate expected positions of the text. Other detected text can be safely ignored since there should only be three sets of coordinates.

Not only relying on "Drawing A": The previous method implemented in the code relied on the position of "Drawing A" to determine both the width of the spirals and the cropping position of both spirals. However, in some instances, the "Drawing A" text was not detected, and thus the entire process would fail. To prevent this from happening, a combination of all the relative positions of any available detected text is used to ensure the best possible cropping positions. If no text is detected, average values are used instead.

Final cropped images: The cropped spiral A, spiral B, and line-block C for each scanned drawing of every patient is saved as a JPEG image. Each of these final images are converted to greyscale and resized to ensure consistent pixel distribution. The spirals are saved as 300×300 pixel images, and the line-blocks as 600×300 pixel images. This allows for all further comparison to be more accurate since every image is exactly the same size regardless of the image quality of the original scanned template.

5.3. Extraction of line-drawing C from line-block C

The line-block C for each scanned drawing had already been extracted. It was decided that only the top-most line-drawing of each line-block would be required as it is the largest line and provides the most space for patients to draw. Method 2, mentioned in section 7 below, analyses this line-drawing. Extracting only the line prevents the large black rectangles from influencing the image processing and analysis techniques. There is an overall rotation in many of the templates caused by human error whilst scanning. Thus, once the black rectangles are identified, compensation for this rotation needs to occur to prevent erroneously cropped lines.

Rectangle rotation correction: Using code snippets from the article “Cropping Rotated Rectangles from Image with OpenCV” by jdha [12], as well as the article “OpenCV shape detection” by Adrian Rosebrock [13], the coordinates of each of the detected black rectangles were saved in a list. This list is then sorted to ensure the coordinates are in a standard order: top-left, top-right, bottom-right, bottom-left. The only two rectangles needed are the top two, as these can then be used to approximate the coordinates of the line drawing that is positioned in between these rectangles. The first rectangle’s bottom coordinates become the line-drawing’s top coordinates. The second rectangle’s top coordinates become the line-drawing’s bottom coordinates. With these new coordinates, the `getPerspectiveTransform()` function from the OpenCV Python package is used in conjunction with the `warpPerspective()` function to produce a straightened line-image. A small number of pixels is then removed from each side of the new image to ensure that no remnants from the black rectangles persist. A brief logic check of whether the width is indeed larger than the height ensures the rotation of the rectangle is correct and makes an appropriate correction if it is not. This image is then saved as a JPEG. This process is summarised in figure 2.

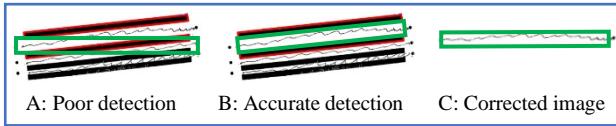


Figure 2: Rectangle detection and rotation correction process

Final extracted singular lines: The main goal of this pre-processing section is to successfully extract as many possible line-drawings as possible, with high precision and very little error. Out of the 1161 total scanned images, a total of 1018 line-drawing images were successfully extracted. This is a success rate of 87.8%. After further analysis, it was realised that most of the unsuccessful extractions were due to the fact that a handful of patients did not fill in the line-block C part of the template at all. Thus, the success rate of line-drawing extraction is even higher and sufficient for this project.

6. METHOD 1: EDGE ANGLE SPIRAL ANALYSIS

The spiral drawings on each template were analysed using a method researched and implemented by group member, Robyn Gebbie [14]. This chosen method and subsequent results will be briefly discussed to allow for a final comparison and conclusion to be drawn.

6.1. Method

Edge angles: Sobel edge detection filters are used to find the horizontal and vertical gradients of each pixel. The orientation (‘edge angle’) of each pixel is then found by taking the inverse tangent of the ratio of the gradients [15].

Pixel angles: The centre of each spiral is calculated. The ratio of the vertical and horizontal distances between the centre and each pixel is then found to take the inverse tangent and produce the ‘pixel angle’ [6].

Relative orientation: The relative orientation is found by subtracting the pixel angle from the edge angle for each pixel. This is plotted as a histogram. A high standard deviation of the data indicates a worse tremor. Further, a wide distribution of angles indicates a larger variety of angles and thus a worse tremor. A higher frequency of edge angles also indicates a worse tremor. The normalised standard deviation is found for every spiral of each patient to allow for comparison between patient’s own hands, as well as between patients in general.

6.2. Results

Two spirals, as seen in table 1, are used to demonstrate this method. Spiral A1 has a higher normalised standard deviation than spiral A2. The histogram in figure 3 also shows that spiral A1 has a higher frequency of the edge angles. This correctly indicates that spiral A1 has a worse tremor compared to spiral A2.

Table 1: Spiral A1 &A2 image and normalised standard deviation

	Spiral A1: Before treatment	Spiral A2: 1 month after treatment
Extracted spiral image		
Normalised standard deviation	0.64	0.15

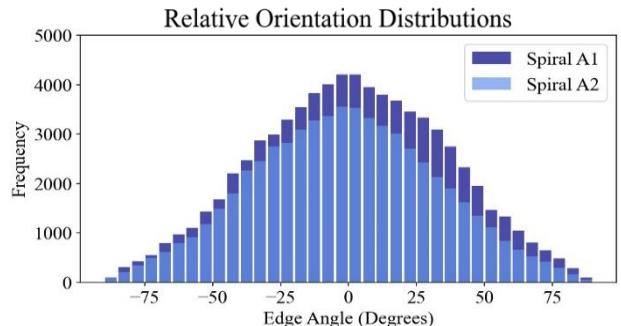


Figure 3: Relative orientation distributions of spirals A1 and A2

7. METHOD 2: AVERAGE AREA AND PEAKS

After the pre-processing mentioned in section 5.3 above, each of the top-most line drawings has been successfully extracted for each patient's scanned drawings. These will be referred to as 'the line(s)'. There are a total of 1018 useable lines that were extracted and saved as JPEG images. This method relies on converting the line into a python list, on which analysis can be made.

7.1. What can be extracted from a line?

If a patient has a severe tremor, it is extremely difficult for them to draw a straight line. Through observation, it was noted that the number of peaks is an indication of the frequency, and thus the severity, of the patient's tremor. In general, the more peaks indicate a worse tremor. Furthermore, the distance between each adjacent peak and trough provide another indication of the severity of the tremor. However, a larger distance generally only indicates a worse tremor when occurring with a high number of peaks. With this in mind it was decided to calculate the average area under the line, as well as the number of peaks and peak-trough distance for each line. This can easily be achieved in Python using the various mathematical packages. However, first a line function (in the form of a list) must be extracted from the line itself.

7.2. Determining the line function

Table 2: Patient #5's treated hand: Extracted line images

	BEFORE	
	AFTER 1 YEAR	

Line noise reduction: Many of the lines still contained erroneous pixel markings that needed to be removed. Pixelation and blurriness due to the poor quality of each scan also needed to be reduced. This will allow for higher-quality analysis. First, each line is converted to grayscale. Then, using the *point()* function from the PIL.Image Python package [16], the colour of each pixel above a particular greyscale value threshold is mapped to black. All other pixels are mapped to white. This allows any grey erroneous pixels to be removed. Further, this leaves only the required pixels, which are now guaranteed to be black. The result of this process can be seen in table 2.

Converting to function: Using the NumPy Python package's *argwhere()* function [17], the indices of pixels that are black in colour are read into a multidimensional array. The x- and y-coordinates of each index is then extracted, which successfully creates two arrays of corresponding coordinates for the x- and y-values of each pixel found in the line. By sorting these according to the x-values, these arrays can easily be plotted on a graph. These arrays will be referred to as the 'line function'. For easier analysis, each graph is shifted down by the average y-value so that it is centred around the x-axis. It can quickly be seen that these points are extremely scattered and noisy, with hundreds of peaks and troughs, as seen in figure 4A.

Graph noise reduction: To reduce the noise of the resulting graphs, the real one-dimensional Fourier Transform *rfft()* is computed using the SciPy.fft Python package [18]. All entries in the y-value array are real which means the faster and more optimized *rfft()* can be used. As seen in the example in figure 4B, there is only a small range of useful frequencies present. Only the lowest 5% of the frequencies are kept, and the higher, unwanted frequencies caused by the pixelated/blurry input are discarded. The inverse real discrete Fourier Transform is then applied, resulting in noise-free line-graphs, as seen in figure 4C.

Extreme frequency bug fix: It should be noted that some of the line-graphs produced an extreme range of frequencies. An example of this can be seen in figure 4A; the right graph has a much smaller range of y-values and yet considerably more erroneous noise-induced peaks compared to the left graph. Keeping 5% of the frequencies in this example did not reduce the noise sufficiently. To fix this error the graph noise reduction process was repeated until sufficient noise reduction had occurred. It was known when this point was reached when the number of peaks is accurate to the line itself and not due to noisy plotting.

7.3. Method 2A: Average Area Under Line Function

Once the line function had been successfully determined, the NumPy Python package's *trapz()* function [17] is used to calculate the total area under the absolute value of the line function, as seen in figure 4D. However, it was realised that if the line was drawn slanted then this total area is exaggerated, regardless of tremor. Thus, it was decided to find the average area of small segments along the line. This provided a more reliable tremor severity rating and helped to reduce the impact of slanted lines.

7.4. Method 2B: Average Peaks × Peak-Trough Distance

The SciPy.signal Python package's *find_peaks()* formula [18] was used to calculate the total number of peaks and troughs within the line function, as well as the coordinate positions of each. These can be seen in figure 4E. From here these values are iterated through to calculate the distance between each peak and adjacent trough(s). Finally, the average peak-trough distance value is found. Since a larger average peak-trough distance generally only indicates a worse tremor when occurring with a high number of peaks, the two values are proportionally linked to the severity of the patient's tremor. Thus, the product of these two values is used as the indication of tremor severity for method 2B.

7.5. Additional Values

Some additional values were calculated for each line, including the standard deviation of both the total and average areas, the maximum value of the line function and the maximum frequency of each line function's Fourier Transform. This can be seen in table 3. These values were not analysed due to time constraints of the project.

7.6. Results

The results from two chosen lines will be used to demonstrate the validity of this method. Patient 5 was chosen since there is significant improvement between the lines drawn by their treated hand before receiving treatment and 1 year after treatment. These two lines will be referred to as the ‘before-line’ and ‘1-year-line’. These line images can be seen in table 2 above, with the processes behind methods 2A and 2B shown in figure 4A-E. Table 3 shows the final calculated values. The values for the before-line are higher than the 1-year-line in all instances. However, as mentioned above, only the values required for methods 2A and 2B are focused on in this report.

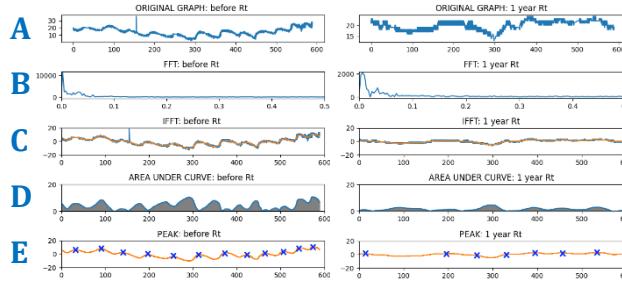


Figure 4: Noise reduction and analysis stages for method 2A & 2B

Table 3: Calculated values from the line function of patient #5’s treated hand before and 1 year after treatment

Calculated Value from Line Function	Before	1 year
Total area (pixels^2)	15442.10	4316.75
Standard deviation of total area	5.52	2.10
Maximum value (pixels)	21.42	4.17
Maximum frequency (Hz)	11087.03	2093.07
METHOD 2A: Average area (pixels^2)	13.44	0.86
Standard deviation of average area	122.74	51.09
Number of peaks	12	7
Average peak-trough distance (pixels)	5.81	2.43
Num. Peaks \times Avg. Peak-Trough Distance (METHOD 2B)	69.67	17.02

7.7. Future Improvements

Slanted lines: With method 2B, a result of a large peak-trough distance coupled with a very low number of peaks was often an indication of a slanted line image. Likewise, with method 2A, an incorrectly large average area was often an indication of a slanted line. More advanced pre-processing could be developed to straighten the lines as much as possible. However, first, research should be performed and/or medical professionals consulted, to determine whether a slanted line is an indication of a worse tremor. This would mean that such slanted lines are valuable and should not be corrected.

Additional values: As mentioned above, the four additional values calculated for each line should be analysed to determine whether they provide further insight. Further research in the frequency domain should be conducted as it is a powerful tool that could provide interesting and unique insights.

8. OVERALL DISCUSSION OF RESULTS

Various tremor severity ratings were determined for each patient’s treated and untreated hand over various time periods. Method 1 uses the normalised standard deviations of the relative orientation distributions of spiral A. Method 2A uses the average area under the line C. Method 2B uses the product of the number of peaks and average peak-trough distance of line C. To determine whether these methods are effective indications of the treatment’s success, the average tremor severity rating, as well as percentage of patients whose tremor improved, was determined. These graphs can be seen in figures 5-7. Full sized versions can be found in Appendix 9.

Tremor severity: There is an immediate decrease in the average tremor severity of the treated hand in all methods. This indicates that the treatment is successful. Further research should be conducted to determine whether these values reflect the medial diagnosis of each patient. Interestingly, both methods 2A and 2B demonstrate a decrease in the tremor severity of the untreated hand as well. This should be further analysed to determine whether it is accurate or a flaw of these particular methods.

Percentage of improved patients: Table 4 shows the average percentage of how many of patients improved after treatment over time. Methods 1 and 2B have very similar averages which indicates that the FUS treatment successfully reduces the tremor in the treated hand. Method 2A has a lower average. Further investigation should be conducted to determine which method most accurately reflects the medical diagnosis of each patient.

Amount of data of later years: As indicated by the red lines in figures 5-7, the number of patients with data for later years drastically decreases. This affects the reliability of these years’ results due to insufficient data.

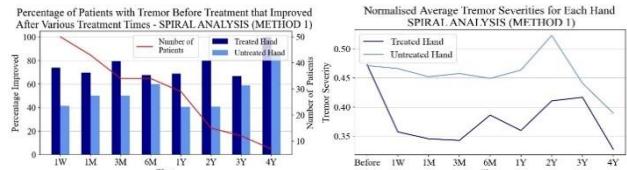


Figure 5: Results of method 1 – spiral analysis

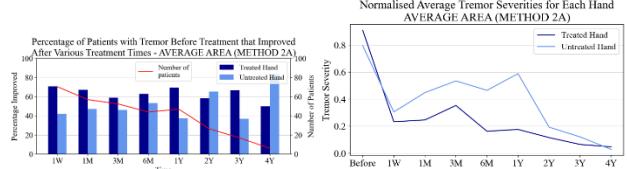


Figure 6: Results of method 2A – average area

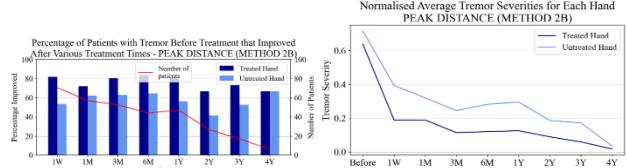


Figure 7: Results of method 2B – peak-trough distance

Table 4: Average percentage of improved patients according to each method implemented

% Improvement of Patients	Method 1	Method 2A	Method 2B
TREATED HAND	75.8%	63.5%	75.7%
NON-TREATED HAND	53.4	51.4%	57.4%

9. FUTURE IMPROVEMENTS

Bringing in an expert physician: The accuracy of these implemented methods should be evaluated by an expert physician or medical professional in order to truly determine whether the results are accurate and reliable enough to be used in the medical field.

Combining methods: Research should be conducted to determine whether combining the results of all methods would produce a more reliable result and/or whether methods 2A and 2B can be applied to the spiral images. This would allow for additional comparative opportunities to further validate the results of methods 2A and 2B.

AI and machine learning: The above computed data can be compiled with existing patient information – such as age, medical history, dominant hand etc. This would allow for a machine learning model to be developed to potentially predict whether a patient would have successful treatment, or even how much improvement would be achieved in each hand. This would be invaluable to the medical physicians and specialists and the patients themselves who are opting into this relatively new FUS treatment.

Pathological vs physiological tremor: Further research should be conducted into whether these methods can accurately detect physiological tremor (peak-trough distance of less than 0.5mm which occurs in normal persons [8]) or whether it only can detect mild pathological tremor (peak-trough distance of greater than 1mm which impairs a patient's function [8]). This would determine whether control groups can be analysed to better provide training data for potential machine learning applications.

10. CONCLUSION

The implemented methods reliably produce three different tremor severity ratings. These can be used to determine which of a patient's hand had been treated, whether the treatment was successful and the amount by which the tremor was reduced in either hand. This points to the potential of using such methods as non-biased evaluation tools of the treatment's efficacy, without needing to involve an expert physician's observation. However, there are plenty of improvements to be made, and further validation of the results is required.

11. REFERENCES

- [1] H. Baek, D. Lockwood, E. J. Mason, E. Obusez, M. Poturalski, R. Rammo, S. J. Nagel and S. E. Jones, "Clinical Intervention Using Focused Ultrasound (FUS) Stimulation of the Brain in Diverse Neurological Disorders," *Frontiers in Neurology*, vol. 13, 2022.
- [2] M. Rohani and A. Fasano, "Focused Ultrasound for Essential Tremor: Review of the Evidence and Discussion of Current Hurdles," *Tremor and Other Hyperkinetic Movements (NY)*, vol. 462, no. 7, 2017.
- [3] N. Seedat, V. Aharonson and I. Schlesinger, "Automated machine vision enabled detection of movement disorders from hand drawn spirals," *2020 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1-5, 2020.
- [4] J. Alty, J. Cosgrove, D. Thorpe and P. Kempster, "How to use pen and paper tasks to aid tremor diagnosis in the clinic," *Pract Neurol*, vol. 17, pp. 456-463, 2017.
- [5] V. Aharonson, N. Seedat, S. Israeli-Korn, S. Hassin-Baer, M. Postema and G. Yahalom, "Automated stage discrimination of Parkinson's disease," *BIO Integration*, vol. 1, no. 2, pp. 55-63, 2020.
- [6] A. Wille, M. Sangaré and S. Winter, "Analysis of Patterns in Tremor Diagnosis Spiral Drawings for Automated Classification," Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, Bochum, 2013.
- [7] S. Aktar, H.-Y. Tseng, P. Barthelmess, P. R. Cohen, N. D. Darnall, K. C. Donovan and D. C. Lin, "Application of machine learning and numerical analysis to classify tremor in patients affected with essential tremor or Parkinson's disease," *Gerontechnology*, vol. 10, no. 4, pp. 208-219, 2012.
- [8] R. J. Elble, R. Sinha and C. Higgins, "Quantification of tremor with a digitizing tablet," *Journal of Neuroscience Methods*, vol. 32, pp. 193-198, 1990.
- [9] F. Miralles, S. Tarongí and A. Espino, "Quantification of the drawing of an Archimedes spiral through the analysis of its digitized picture," *Journal of Neuroscience Methods*, vol. 152, pp. 18-31, 2006.
- [10] K. da Silva, "Tremor Classification," 7 September 2020. [Online]. Available: https://github.com/kdasilva835842/tremor_classification. [Accessed 29 September 2022].
- [11] A. Rosebrock, "OpenCV Text Detection (EAST text detector) - PyImageSearch," PyImageSearch, 20 August 2018. [Online]. Available: <https://pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/>. [Accessed 16 July 2022].
- [12] jdiao, "Cropping Rotated Rectangles from Image with OpenCV," jdiao's digital space, 23 February 2019. [Online]. Available: https://jdiao.github.io/2019/02/23/crop_rotated_rectangle_opencv/. [Accessed 26 July 2022].
- [13] A. Rosebrock, "OpenCV shape detection - PyImageSearch," PyImageSearch, 8 February 2016. [Online]. Available: <https://pyimagesearch.com/2016/02/08/opencv-shape-detection/>. [Accessed 20 July 2022].
- [14] R. Gebbie, "Analysis of the Effect of Focused Ultrasound Treatment on Hand Tremor: Quantification of Tremor in Spiral Drawings," 4th Year Investigation Report 22G05, School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, 2022.
- [15] P. Delmas, "Edge Detection," in *COMPSCI 373 Semester 1*, Auckland, The University of Auckland, 2010, pp. 40-51.
- [16] Alex Clark and Contributors, "Image Module - Pillow (PIL Fork) 9.3.0 documentation," pillow, 2022. [Online]. Available: <https://pillow.readthedocs.io/en/stable/reference/Image.html>. [Accessed 6 September 2022].
- [17] NumPy Developers, "NumPy documentation - NumPy v1.23 Manual," NumPy, 2022. [Online]. Available: <https://numpy.org/doc/stable/index.html>. [Accessed 5 September 2022].
- [18] The SciPy community, "SciPy documentation - SciPy v1.9.3 Manual," SciPy, 2022. [Online]. Available: <https://docs.scipy.org/doc/scipy/index.html>. [Accessed 9 September 2022].

Appendix 1: Reflection on Working as a Group

REFLECTION ON WORKING AS A GROUP

ELEN4012A – EIE Investigation 2022 – Jesse van der Merwe (1829172) – Group 22G05

This project required learners to create groups of 2 to not only work on the final investigation, but also create tender bids, project plans, etc. After forming a group with Robyn Gebbie (2127777) at the beginning of the year, we quickly got started on researching all the available project outlines from various lecturers (supervisors). After a few weeks, we met to decide on our top three. After some debate we narrowed it down to five options, and thus our first group hurdle was reached; which three topics to bid on?

We decided to contact the respective supervisors via email to ask questions clarifying anything we were unsure of from the project descriptions. Fortunately, this allowed us to narrow down our list of five to the final list of three! From here we got together and wrote the tender bid outlines for the three top projects. We then split the workload of finishing and finalising these bids and did the rest at home.

After submitting our three tender bids, we waited anxiously for the final allocations. We were lucky to get our second choice: “Which Hand?”, supervised by Dr Aharonson. We then met to discuss the project brief and scope. Soon we met with our supervisor to continue this discussion. We then divided up the work of creating the project plan and completed our sections at our own individual pace, within a determined time frame. Finally, we met again to combine, finalise, and submit the Project Plan document.

This would quickly become the norm for us, as we both preferred to work on our own and then later meet up (often with our supervisor) to discuss, compare and finalise. These “meetings” were almost exclusively online, using either Discord or MS Teams as the VOIP and screen-sharing software. Our supervisor encouraged us to turn on our cameras to provide more engagement and interactivity. This was wonderful until my internet became too poor to keep up with the live video feeds. We were able to effectively use both Discord and MS Teams as group work platforms by having multiple channels in which we sent and shared various important resources, data, documents, and even just reminders and to do lists.

I utilised GitHub as the basis for all my code and data storage. It was very convenient for me to have a detailed history and way in which to revert to previous versions of code. In future group work environments, I will encourage all group members to join the same repository on GitHub to share, edit, collaborate, and monitor each other’s code. GitHub has incredibly useful groupwork tools: code reviews, source control, repository branching, detailed history logs, to name a few.

During the actual project, Robyn and I worked well together. We split the work as equally as possible:

- Robyn focused on researching and implementing a more technical and mathematically complicated data analysis method.
- I focused on improving the existing data pre-processing code as well as implementing a simpler data analysis method.
- We both did background research to ensure a basic overall understanding of the topic at hand. This was especially important since this is a biomedical topic, and neither of us did biomed.

I believe that we utilised many of the strengths of group work. Not only did we split the workload, but we did it in such a way that played to our individual strengths. For example, since I graduated from game design engineering, I already had a better understanding of the Python programming language. I thus volunteered to tackle the image pre-processing section as it required me to navigate another student’s code, and this was easier to do with a basic understanding of the language.

Our time management was well thought out, implemented, and relatively strict, while still giving each other flexibility and support if needed. However, our initial Project Plan did not fully grasp the scope or complexity as intended by our supervisor. Specifically, we had included machine learning as a requirement, but after discussing further with our supervisor we came to understand that only image processing and computational analysis of the results was actually required to answer the investigation question. This meant that our original Project Plan was incorrect and needed to be updated. Fortunately, once we came up with a new Project Plan the timeline and schedule were followed leaving plenty opportunity for us to meet with our supervisor and get feedback before each of the milestones of this project. Such milestones included the project presentation and Open Day.

While Robin and I were happy to work individually on our parts of the project and then get together to discuss, combine and conclude, I do think there was room for more collaboration to better streamline the process of working in a group. Thus, in the future I will ensure to include more frequent group check-ins as well as more time to review each other's work before the project is finalised.

It was a pleasure to work with Robin in this final investigation project. We were thrown into the depths of a challenging biomedical investigation question, but we truly learned an incredible amount during this process while trying to produce high quality engineering work.

Vered Aharonson was a fantastic supervisor who never failed to answer any question. She was available pretty much all the time via e-mail and always keen for a video chat on MS Teams, even when she was travelling the world attending research conferences. Both Robin and I are very grateful for her input, support, and guidance. We hope we did her proud in this project.

Appendix 2: Tender Documents



Tender Bid for Project Number: 22P59

Project Title: Which Hand?

Group Number: 22G05

Project Overview:

In order to begin the design and implementation of the hand drawing analysis, sufficient research must be done to understand the effects of Parkinson's disease on a person's motor functions, and how this would affect their drawing ability. Research on image processing would also be important to understand the data provided for the project. The project will require image processing in order to convert the hand drawing data into a comprehensive database that can then be utilized for machine learning. The actual implementation of the project will involve the analysis of the data provided and the creation of an artificial intelligence program that is capable of interpreting the information and determining what stage of Parkinson's a specific subject is at. The final stage of the project will be testing the application against known values in order to ensure a product of high enough accuracy. Any necessary changes and improvements to the image processing and machine learning will need to be implemented. The resulting data will be used to efficiently analyze the effectiveness of Focused Ultrasound as a treatment for Parkinson's disease. The machine learning program can be used to compare the patient data before the start of the treatment to the data 2 years into the treatment. The program should effectively display the difference between the stages of Parkinson's displayed in each case.

Preliminary Budget & Resources:

In order to successfully implement this project, access to extensive data on the effects of Parkinson's disease on hand drawings, as well as images drawn by patients that can be used to produce a machine learning database will need to be provided by the university. The effectiveness of the program will need to be verified by comparing the results of patients in known stages of Parkinson's. It will therefore be necessary for the known stage of Parkinson's disease to be provided for the image data.

An image processing library will be required to analyze and manipulate the image data provided for the project. This library will need to convert the images into a data format so as the machine learning database can be developed. OpenCV is a commonly used image processing tool for machine learning, it is open source and therefore free to use.

A database will need to be developed with the hand drawing image data; this will require databasing software. Open-source software such as MySQL and Cassandra are available and free to use.

The development of the machine learning program will be done in Python as many machine learning related resources support this language and are readily available.

Weekly Milestones:

Week 1: Perform thorough research into the Focused Ultrasound (FUS) treatment, Parkinson's disease and Essential Tremor to better understand the subject matter. Further research into the software requirements and solutions, as well as the computational and statistical analyses that will be used to provide quantitative analytics and results. Ensure all plans and deliverables are completely in-line with the standards, ethics and requirements of the professionals and data providers.

Week 2: Start designing image processing techniques (if necessary) and finalize what software will be used.

Week 3: Start applying the data preparation and image pre-processing to generate a comprehensive dataset so as to develop the dataset that will be further analyzed as well as data analysis techniques.

Week 4-6: Start developing the software: further image processing, data analysis, signal processing, output, etc. Ensure to develop tests and other checks that always ensure the utmost accuracy and quality output.

Week 7-8: Finalize the software and start performing final quality and accuracy tests.

Risks and Mitigation:

This project is attempting to replace subjective, non-standard, labour-intensive task of an expert physician evaluating the effectiveness of the specific Parkinson's treatment. However, this means that the results of this project need to be reliable enough to be relied upon in a medical environment. Thus, the biggest and most important risk to consider regarding this project includes having an error margin that is too large, as this would result in unreliable results that could mislead physicians and patients. This risk can only be mitigated if the error margin of the final product is small enough, according to the professionals within this field, so that the results can be confidently and consistently relied upon. If this error margin cannot be achieved, then disclaimers and other ethical considerations must be discussed with all parties involved before using any of the results of the final product.

As mentioned, the use, analysis and result-distribution of the data provided offers ethical considerations that must be adhered to at all times. Ethics waivers and agreements must be drafted up and agreed to by all parties before the commencement of this investigation.



Tender Bid for Project Number: 22P30

Project Title: Sign Language Translator for the Hearing and Speech Impaired

Group Number: 22G05

Project Overview:

In order to fully appreciate the extent of this project, research will have to be undertaken regarding South African sign language (SASL). While it is not (yet) one of South Africa's official languages, it is recognized as a teaching-subject in schools and universities across South Africa. With all this in mind, it is obvious that developing systems that will enable the translation of SASL into audible speech is invaluable to not only the deaf community, but all communities of this country. Further research will be conducted to discover if there are any current solutions to developing such a system (that can, for example, translate hand signals and movements). The design of the image capturing techniques, separation and analysis of the inputted data will need to be planned and complex algorithms developed. A dataset and database will need to be captured, designed and developed that will best contain and allow for ease of analysis of the data. From here, software will need to be investigated and designed that will convert the processed signals into audio speech signals (and text output if possible). This entire system will need to be one self-contained and portable program. As with any video recording and use of personal data, ethics applications and waivers will need to be developed and agreed to by all parties. The final stage of the project will be testing the system in order to ensure a product of high enough accuracy, as incorrect translations would be detrimental to such an important project.

Weekly Milestones:

Week 1: Perform thorough research into SASL, as well as the software requirements and solutions. Ensure all plans and deliverables are completely in-line with the standards, ethics and requirements of the deaf-community.

Week 2: Start designing image processing techniques (if necessary) and finalize what software will be used.

Week 3: Begin collecting data (if necessary) and storing it into the database – develop the dataset that will be analyzed as well as data analysis techniques.

Week 4-6: Start developing the software: image processing, data analysis, signal processing, audio processing and output, text generation and output, etc. Ensure to develop tests and other checks that always ensure the utmost accuracy and quality output.

Week 7-8: Finalize the software and start performing final quality and accuracy tests.

Preliminary Budget & Resources:

Fortunately, the University of Witwatersrand (Wits) has many of the required resources for this project readily available. The incredible ESD laboratory and other computer and software components available to students allow for the cost of this project to remain low.

If anything, the major cost will include any costs of software not provided by Wits that is required to perform the image capturing, data processing and data conversion required. However, since Python is the preferred programming language for such applications (especially in the machine learning field), most libraries are open source and free to use and develop further.

An image processing library will be required to analyze and manipulate the input (video data for example) and convert it into a data format so that a database can be developed. This will require database software; fortunately, open-source software such as MySQL and Cassandra are available for students for free.

Depending on how the data is captured, and whether it must be done within this project (or if it is provided beforehand) hardware resources might be required, such as a camera.

Risks and Mitigation:

The biggest and most important risk to consider with this project includes the miss-translation of the inputted sign language. A high rate of miss-translation is unacceptable and will not be considered a successful solution. Mitigating this will require accurate data input, well-programmed software solution, and continuous input from the deaf-community and professional SASL contact that checks the accuracy of the output of the system. Fortunately, one of the students has a SASL contact that would be willing to provide this quality assurance if Wits is unable to provide such a person.

Further, the emotions conveyed when communicating with SASL are often entirely based on facial expressions and body gestures, which is incredibly difficult to convey using a generated audio and text output. This is another risk that is related to the miss-communication of such a system and must be closely monitored to ensure that the inaccuracies are minimal.

As mentioned, the use of video recording and use of personal data provides ethical considerations that must be adhered to at all times.



Tender Bid for Project Number: 22P21

Project Title: App to scan pool water test strips
Group Number: 22G05

Project Overview:

In order to begin the design of the app to scan pool water strips, sufficient research will need to be done to understand the possible outcomes displayed by the pool water strips and their corresponding courses of action. Research on image processing and effective image capturing in different light conditions will also be important for this stage. The design of the image capturing techniques and separation and analysis of each testing strip segment will need to be planned and complex algorithms developed. After the design stage, a database will need to be developed with the use of the processed image data. This data will be used to train the artificial intelligence program to recognize the colors present on the testing strips and will be used to determine the appropriate course of action for each color. The database will need to consider different light conditions in the images. The actual implementation of the project will involve image capturing, processing and utilizing machine learning to analyze the pool water testing strips and advise on a plan of action. This project will be developed into an application. The final stage of the project will be testing the application against known values and making any necessary changes to the coding and image processing.

Preliminary Budget & Resources:

Since this is a machine learning project, the majority of the resources that will be required are software based.

The image capturing stage will require the use of a cellphone camera and will not require any resources to be provided by the university.

An image processing library will be required to analyze and manipulate images. This library will need to convert the images into a data format so as the machine learning database can be developed. OpenCV is a commonly used image processing tool for machine learning, it is open source and therefore free to use.

A database will need to be developed with the test strip image data; this will require databasing software. Open-source software such as MySQL and Cassandra are available and free to use.

Application development can be done with the use of Unity which is free for students. The application development will be done in Python as this is the most common language used for machine learning and comes with many supporting resources.

Weekly Milestones:

Week 1: Research on pool water test strip outcomes and possible solutions. A plan on image processing techniques and how the app development will take place.

Week 2: This week will focus on a general project design, highlighting the image processing section and determining the software required for each stage of development.

Week 3: Image processing design; this week should focus on capturing the image data and designing an image processing approach.

Week 4: This week will focus on implementing algorithms to capture and analyze the pool water test strip images, developing a dataset.

Week 5: This week will focus on the implantation of machine learning and application development.

Week 6: This week will focus on finalizing features and ensuring application covers requirements.

Week 7: Testing of application with the use of test strips in various lighting conditions.

Week 8: Making any necessary adjustments to project and finalizing project report.

Risks and Mitigation:

Several risks exist in the implementation of this application. Although these risks are not necessarily harmful, they could pose a risk to the success and usefulness of the application.

It is essential that the final application meets the project requirements and provides a correct course of action for the chemical ranges on the testing strip. If the image processing is performed incorrectly, the result determined by the machine learning program could suggest an incorrect course of action for the pool water correction. This could be avoided by thoroughly testing results against known values, ensuring the image processing and image analysis with the use of machine learning has been performed correctly and accuracy.

An important project requirement is that the application is able to work successfully to analyze images in all light conditions. If this requirement is not carefully considered, the program could mistake a specific color for another. In order to ensure that this does not take place, it will be necessary to thoroughly test the application with test strip images in different light conditions.

Software advancements could mean that the app becomes unusable in coming years. The solution to this is to ensure the software is adaptive.

Appendix 3: Project Plan Report

Investigation Project Plan

Robyn Gebbie (2127777), Jesse Van Der Merwe (1829172)

School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2050, South Africa

Abstract — The EIE Investigation Capstone Project “Which Hand”, supervised by Dr V. Aharonson, will be undertaken during the second block of semester 2 of 2022. This report describes the project plan, work breakdown structure and Gantt charts, which have been generated using project management software Mind Genius and MS Project. Further, the required resources are described, and risks and ethical considerations included.

I. INTRODUCTION

The Electrical/Information Engineering Investigation Project requires students to undertake a significant investigation that involves research, design, implementation, and investigation of a system typical of a complex engineering problem. The project undertaken for this investigation is that of “Which Hand”, supervised by Dr V. Aharonson, which requires the implementation of computational and statistical analysis to provide quantitative insight to assess the severity of a subject’s tremor in order to determine the outcome of Parkinson’s disease and Essential Tremor treatment. This report will cover a comprehensive plan as to how this project will be implemented, taking into consideration any risks and ethical concerns that may arise during the execution of the project.

II. PROJECT BACKGROUND

Focused Ultrasounds (FUS) treatment has the potential to reduce the tremor in patients with Parkinson’s Disease (PD). The effectiveness of this treatment can be determined by analyzing traced spiral drawings done by PD patients in order to determine the tremor present in the drawings. In order to determine whether this treatment is decreasing the severity of PD in patients with the illness in a way that decreases the influence of human error and bias, a machine learning approach can be developed to determine whether a spiral drawing has been done by a patient with PD or simply by a patient that has Essential Tremor (ET). This model can be created with the use of Artificial Intelligence (AI) and will require use of image classification and preprocessing software.

III. PROJECT DESCRIPTION

This project will involve research and investigation into FUS and its effect on reducing the tremor in subjects who have PD or ET. The treatment is applied to one side of the brain and the result of the treatment will therefore only be visible in the treated side of the brain. The project involves the implementation of a model that minimizes the effect of human error and bias in the diagnoses process of PD.

The implementation of this project will require data in the form of traced spiral drawings drawn by subjects that have been diagnosed with PD and have the illness to varying severity, as well as subjects that do not have PD, but simply show signs of

ET. Each subject will need to trace the spiral with both their right and left hands in order for the effectiveness of the FUS treatment to be determined on the treated side of the brain. In order to get permission to use this data, a waiver must be submitted to Rambam Medical Center. A close consideration of data ethics must be carried out throughout this process.

Research will need to be done on the effectiveness of the FUS treatment and how it affects the tremor present in drawings drawn by patients. This research will be used in coordination with an analysis of the spiral drawings drawn by patients in order to determine whether the treatment is lessening the severity of PD in the side of the brain receiving treatment.

Research into effective methods of classifying the drawings as either being drawn by a subject with PD or ET will need to be done. This research will be largely focused on machine learning models that will be capable of accurately processing and classifying the spiral image data. This research will include investigation into appropriate methods to perform image processing and preprocessing in order for a dataset to be created for the implementation of the model.

The implementation of the AI model will require data preparation and preprocessing in order for the spiral image data to be used effectively. The implementation of this project will involve the development of a machine learning program that is able to accurately classify the PD and ET drawings and determine whether a drawing has been done by a subject that has PD. The model will help to determine the effectiveness of the FUS treatment on the PD patients.

IV. PROJECT MANAGEMENT

For the project to be delivered successfully and on time, a comprehensive plan needs to be developed. This plan needs to include the project scope, as well as any project deadlines with a comprehensive description of the requirements for each deadline. A work breakdown structure (WBS) has been generated for this project using [Mind Genius 20](#). This allows for the entire project to be broken down into manageable sections, whose overall lengths could then be estimated. Exporting this into [Microsoft Project](#) allowed for the further addition of relevant project data, including lengths of individual tasks as well as the order and reliance of tasks. Both these aforementioned project management software tools will help generate and control the scope of the project, as well as the success criteria. This will be expanded upon in the Methodology section below.

V. METHODOLOGY

A detailed and complete WBS of the entire project consisting of 5 main components was created. This complete WBS can be found in appendix A. This WBS was then imported into MS Project, and the approximates hours per task allocated. This allowed MS Project to produce a Gantt chart and overall timeline. This chart can be found in Appendix A. According to the approximate hours per task currently allocated, the entire project will take 300 hours, as required by the CBO. The project should also be completed by mid-November. However, this current project plan allows for almost no error or variability. It is known that errors, mistakes, and delays are rampant in projects, and thus more efficient resource leveling, and task allocation must occur before the project can go forward. This includes better allocation of the hours between the two group members, as well as more streamlined order in which tasks are to be completed. Each of the five main components are detailed below.

A. Investigation

The first component of this project includes investigation (figure 1). To better understand the field in which this project operates in, a few hours should be spent researching the basics of the associated medical theory. This includes Parkinson's Disease (PD), Essential Tremors (ET), Static Spiral Tests (SST), the bias and inaccuracies of healthcare professionals, as well as the various methods of diagnosing PD/ET. Even though the project is simply to evaluate the effectiveness of the treatment, both PD and ET (neurological disorder) need to be researched in order to understand the similarities and differences. This will be vital if the functionality of differentiating between the two is to be implemented.

Next, research must be conducted in order to determine the best possible data preparation method in order to process the provided images, as well as to pre-process the resulting data. These provided images are pictures/scans of spiral drawings of patients that have PD or ET, as further expanded in the Required Resources section of this report. The way in which the data is prepared and split into training data, as well as data augmentation and hyper-parameter optimization must also be researched and carefully considered, as these aspects will greatly vary and affect the final results.

Finally, research must be conducted in order to determine the best possible machine learning/deep learning method to be implemented in order to

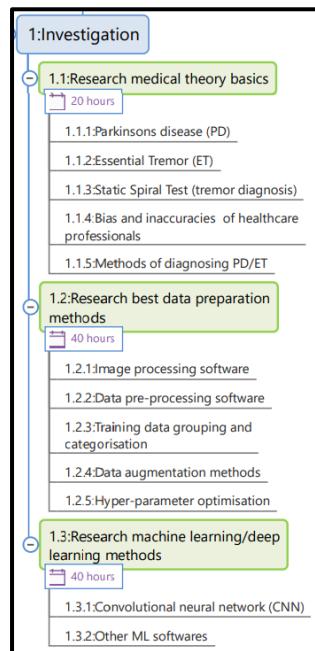


Figure 3: WBS - Investigation

generate the most accurate and useful results.

B. Design and Implementation

After the above research has been conducted, the design and implementation of each section can commence (figure 2). Once the most applicable image processing method is determined, this can be designed, coded, and implemented. Following, once the image processing has been complete, and the most applicable data pre-processing method is determined, the data-preprocessing software can be designed, coded and implemented. Following, once the data has been pre-processed and optimized, and the best machine learning method chosen, this software can be designed, coded and implemented in order to generate final results.

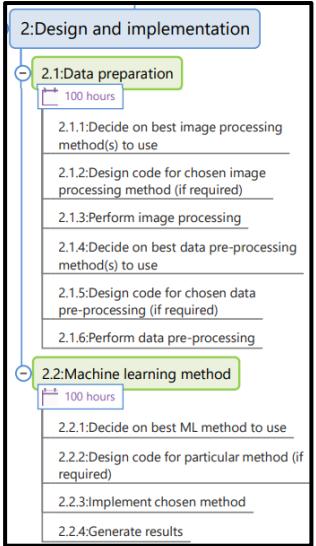


Figure 1: WBS - Design and Implementation

This section makes up the bulk of the project and thus 100 hours were initially allocated for each subsection. However, after loading the WBS into MS project and allocating more specific hours to each subtask, it was determined that perhaps 75 hours for data preparation implementation, and 215 hours for machine learning method implementation would be more appropriate. This includes factors such as both members of the project team being able to work on the first subsection concurrently, therefore halving the overall time required. These intricate details will be further evaluated and worked out during the first few days of the project in order to better determine a suitable schedule for coding and implementing of the software.

C. Budget

After investigation and research has occurred, it will be clear as to whether any software required for this project will need to be bought, or whether such software resources are freely available already – either through free packages on the internet, or through the university. If such a purchase is required, it must be discussed with the project supervisor in order to generate and submit a purchase order to the School.

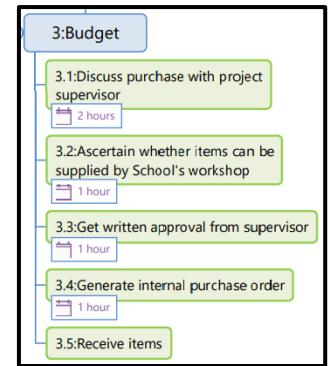


Figure 2: WBS - Budget

D. Validation (Success Criteria)

Once the final solution has been designed and implemented, it will need to be tested and validated in order to determine whether the solution acquired is accurate and/or useful. Both

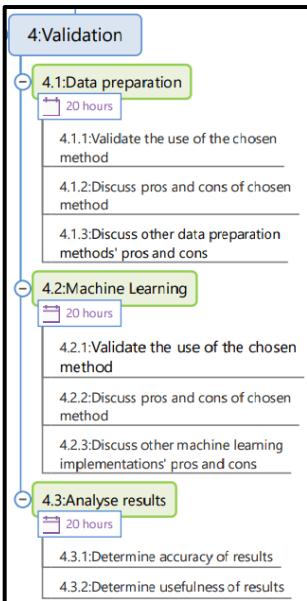


Figure 4: WBS - Validation

based on the particular test data used. This will help determine whether the solution is useful, as only a very high percentage accuracy will be considered acceptable in this particular application. The mis-classification between PD and ET might be considered acceptable since both PD and ET are movement disorders that result in tremor [10], however, effectively zero mis-classification between PD/ET and the control subjects is required.

E. Documentation

Finally, documentation must be planned for since this project involves not just the research, design and implementation of an applicable solution, but also the final submission of a project report. Further, each group member must keep a detailed Engineering Notebook throughout the project. Weekly meetings with the supervisor must be held, with minutes and agenda recorded at each. An ethics waiver must be filled in and approved (discussed further in the Ethics section below). A final group presentation must be prepared, and an individual interview conducted. All of these are important aspects of the project that must be considered and planned for accordingly.

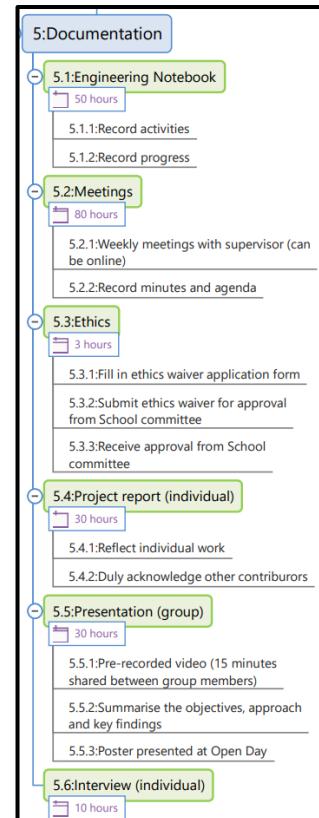


Figure 5: WBS - Documentation

VI. REQUIRED RESOURCES

This project will require the use of Artificial Intelligence (AI) to perform a statistical analysis of spiral drawings done by patients that have Parkinson's Disease. The drawings will be used to train a machine learning model to differentiate between drawings that have been drawn by patients that have Parkinson's and those who have an Essential Tremor (ET).

A. Training Data

In order to train the AI model, sufficient data containing the spiral drawings done by individuals that have been diagnosed with Parkinson's Disease and those who do not have the illness will be required. This data will be provided by Rambam Medical Center and will be in the form of scanned images containing traced spiral drawings for each subject. Each subject will draw a clockwise and an anticlockwise spiral using each hand, totaling in 4 spirals for each patient. The tremor present in each spiral will be analyzed in order to determine the likelihood of the patient having Parkinson's Disease and the severity of the Parkinson's Disease for each patient that has been diagnosed and is undergoing treatment.

Data containing information about each subject's age, gender and date of PD diagnoses will also be provided. This data can be used to train a model to recognize the effect of these input characteristics on the severity of PD in each subject.

B. Image Processing Software

The project involves researching image preprocessing techniques that can be used to analyze scanned images of the spiral drawings. As a starting point, some image processing techniques will be considered, and further research will be done on these techniques during the implementation of this project in order to find the optimal method.

One of the most widely used image processing and visual AI techniques is the use of a Convolutional Neural Network (CNN). CNN is a deep learning model that processes data in a grid format, making it an optimal method for image processing. This method has many supporting architectures, with some of the common ones being AlexNet, GoogleLeNet and VGG. These image processing architectures are supported by Python and can be implemented with the use of environments such as VSCode or Jupyter Notebook.

An alternative to CNN is a Graph Neural Network, this model can be implemented in Python with the use of packages such as Deep Graph Library, PyTorch Geometric and Spektral.

C. Image Preprocessing Software

It is required to research image preprocessing techniques in order to prepare the image data for the processing stage. The preprocessing resources are dependent on the machine learning model that is being implemented.

For most models, image preprocessing will require the resizing of all images to a 256x256 pixel dimension, image augmentation and test-time augmentation. These processes can be achieved with the use of a Python package and can be implemented using environments such as VSCode or Jupyter Notebook. Python has

several packages that support the image preprocessing implementation, some of which are Scikit-image and Keras.

VII. RISKS

Before beginning this project, it is important to consider any possible risks that implementing this project may pose. The risks are related to the type of data that is required to train the AI models and general risks that come with implementing an AI.

A. Data Privacy

The data required for this project contains personal medical information for various subjects. These people have given permission for their data to be used in research, however it is important to not to violate the conditions under which the data is allowed to be used. Violation of these conditions could pose harm or distress to the people whose information is contained in the data.

B. Bias

It is important that a fair AI model is developed that is inclusive of all social groups. If the AI model is fed training data that excludes a particular class of people, it will produce results that do not consider the full population demographic. The AI must therefore receive data that does not allow any form of unfairness to develop.

C. Unreliable Results

One of the greatest risks is that the resulting AI model will not accurately predict the severity of PD or the presence of PD in a test subject. This risk is related to the accuracy of the data preparation process and the implementation of the machine learning model. In order to ensure that this does not occur, it is important to perform extensive tests with data with known results in order to ensure that the model is producing the correct results.

D. Security

Due to the complexity of training an AI model, it is initially difficult to tell whether the data provided is accurate and has undergone the proper preprocessing. Bad data can be introduced into the model, resulting in incorrect results that do not meet the final requirements. The extensive testing of all stages of process can allow for cases like this to be caught and dealt with.

VIII. ETHICS

This project involves the use of data obtained through human research. This data contains information about a test subject's age, gender and date of Parkinson's Disease diagnoses, as well as spiral drawings done by the subject that can be used to track the severity of PD in the patient. The subjects whose data is available for this study have given permission for their information to be used for research.

In order to use the data, an ethics waiver must be submitted to the Rambam Medical Center. The facility will then provide permission for the data to be used for this project.

A risk analysis can be done to categorize the risk at which the subjects are put in when participating in the research. The risk can be categorized as either having no risk, minimal risk, low risk, medium risk or high risk. In order for the research to have no risk, no human participants must be involved. High risk research involves research performed with human participants in which the participants are in risk of harm, this includes investigation into sensitive topics that may cause the participants emotional distress. This research can be considered to be medium risk research. The data used for this research has been obtained with the use of human participants, these participants have been diagnosed with PD and are required to trace a spiral with their left and right hands. The research has the potential to cause the subject emotional distress under the condition that the research results in the discovery that the subject has a severe or worsened stage of PD.

IX. CONCLUSION

This project is to be completed within the second block of the second semester of 2022. This plan serves as the basis from which to orientate the workflow and task allocation between group members in order to ensure a well planned and implemented project that is not rushed or left to the last minute. An ethics waiver must be completed, submitted and approved before use and manipulation of the data can occur. After research and investigation with regards to the required resources and methods to be used within this project, the design and implementation of a solution will begin. The methodology has been briefly mentioned and is to be further fleshed out as each section is tackled during the project. The solution is to be analyzed and validated, with the results and accuracy of such results being determined and thoroughly documented.

REFERENCES

- [1] N. Seedat, V. Aharonson, and I. Schlesinger, "Automated machine vision enabled detection of movement disorders from hand drawn spirals," in 2022 IEEE International Conference on Healthcare Informatics (ICHI), 2020: IEEE, pp. 1-5.
- [2] M. Tripathi, "Image Processing using CNN | Beginner's Guide to Image Processing", *Analytics Vidhya*, 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/>. [Accessed: 24-Jul- 2022].
- [3] P. Walpita, "Convolutional Neural Networks For Artificial Vision", *Medium*, 2020. [Online]. Available: <https://priyalwalpita.medium.com/convolutional-neural-networks-for-artificial-vision-455be7c85d15>. [Accessed: 24-Jul- 2022].
- [4] J. Brownlee, "Best Practices for Preparing and Augmenting Image Data for CNNs", *Machine Learning Mastery*, 2022. [Online]. Available: <https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/>. [Accessed: 24- Jul- 2022].

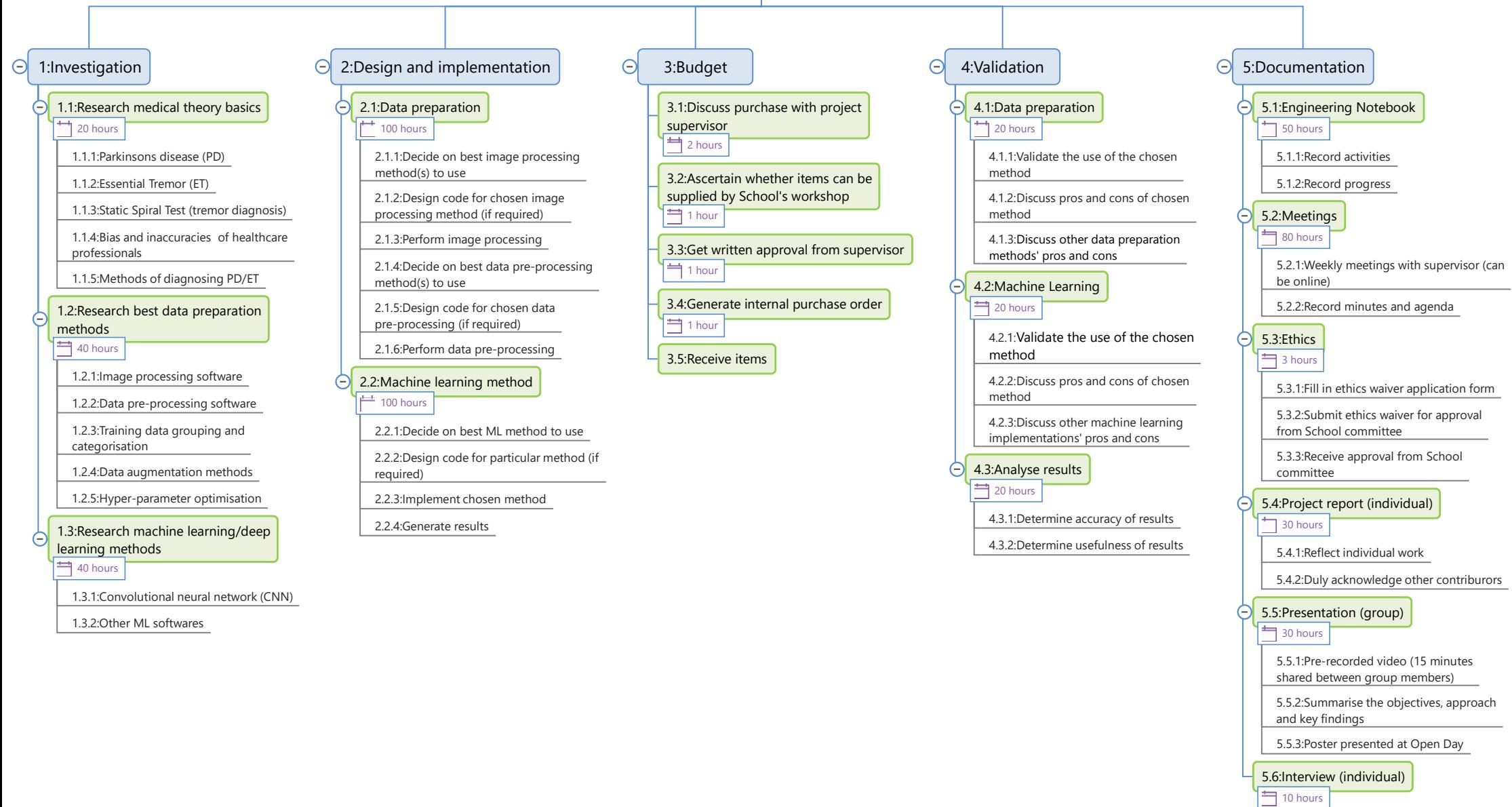
- [5] J. Leban, "Image recognition with Machine Learning on Python, Image processing", *Medium*, 2020. [Online]. Available: <https://towardsdatascience.com/image-recognition-with-machine-learning-on-python-image-processing-3abe6b158e9a>. [Accessed: 24- Jul- 2022].
- [6] R. Yadav, "Top 7 Image Processing Libraries In Python", *Analytics India Magazine*, 2022. [Online]. Available: <https://analyticsindiamag.com/top-8-image-processing-libraries-in-python/>. [Accessed: 24- Jul- 2022].
- [7] A. Fredrick, "Getting Started with Image Preprocessing in Python", *Engineering Education (EngEd) Program / Section*, 2021. [Online]. Available: <https://www.section.io/engineering-education/image-preprocessing-in-python/>. [Accessed: 24- Jul- 2022].
- [8] "Overview of Graph Neural Networks", *OpenGenus IQ: Computing Expertise & Legacy*, 2022. [Online]. Available: <https://iq.opengenus.org/graph-neural-networks/>. [Accessed: 24- Jul- 2022].
- [9] R. Teja, "Let's Talk About Graph Neural Network Python Libraries!", *Medium*, 2021. [Online]. Available: <https://towardsdatascience.com/lets-talk-about-graph-neural-network-python-libraries-a0b23ec983b0>. [Accessed: 24- Jul- 2022].
- [10] S. Singh, "Risk Assessment for AI projects", *LinkedIn.com*, 2021. [Online]. Available: <https://www.linkedin.com/pulse/risk-assessment-ai-projects-sonu-singh>. [Accessed: 24- Jul- 2022].
- [11] Microsoft, "Microsoft Project", 2021. [Online]. Available: <https://www.microsoft.com/en-za/microsoft-365/project/project-management-software>. [Accessed: 24- Jul- 2022].
- [12] MindGenius, "MindGenius 20", 2021. [Online]. Available: <https://www.mindgenius.com/mindgenius-20/>. [Accessed: 24- Jul- 2022].

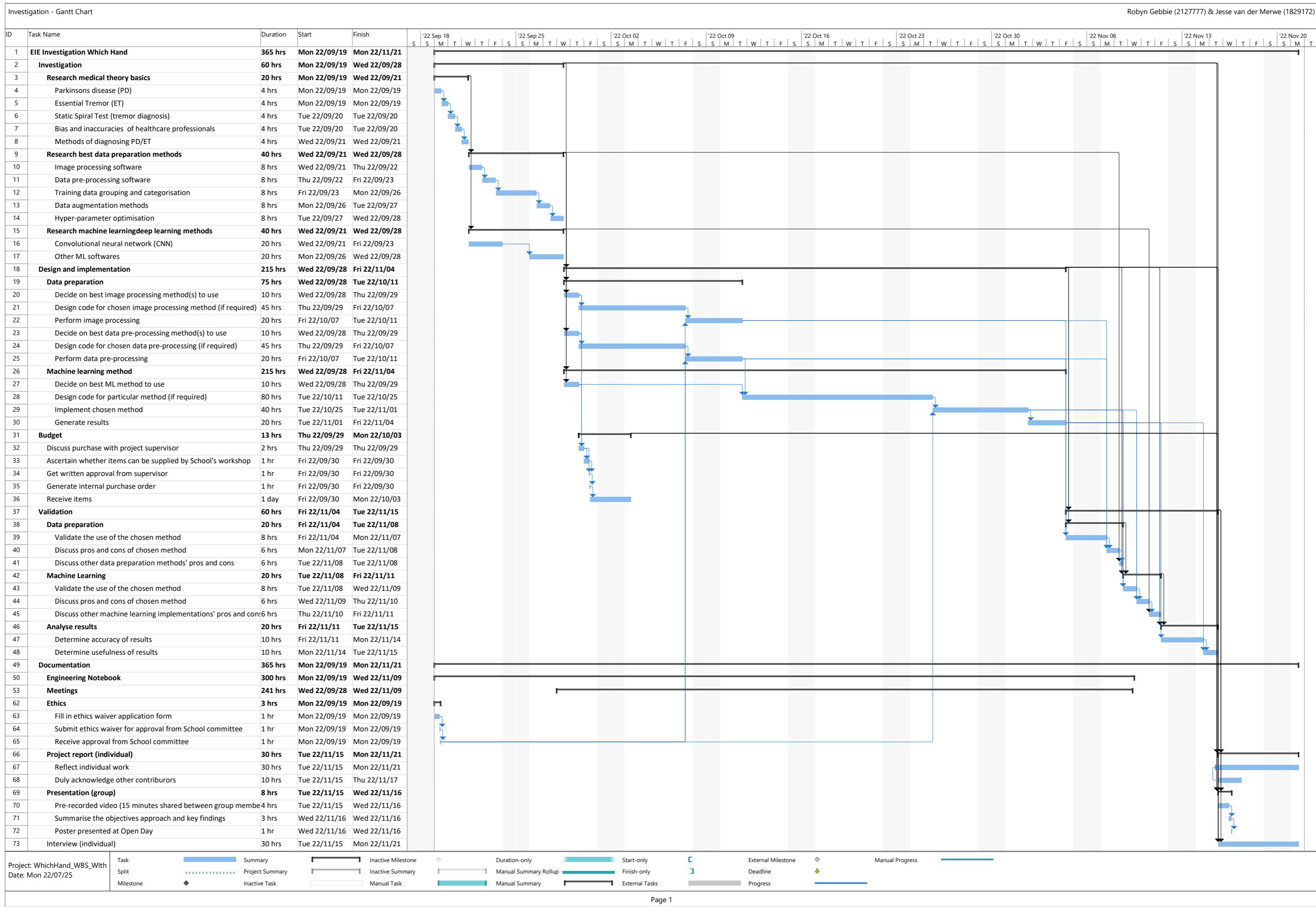
Appendix A

1. Work Breakdown Structure
2. Gantt Chart

EIE Investigation: "Which Hand"

19 Sep





Appendix 4: Agendas & Minutes of Cohort 8's Meetings

APPENDIX 4A: Meeting Details and Agendas of Each Weekly Cohort 8 Meeting

WEEK 1

1.1. Meeting Details

Date: 21/09/2022	Time: 10:00		
Group 1: 22G05	Chairperson 1: Jesse van der Merwe	Secretary 1: Robyn Gebbie	
Group 2: 22G50	Chairperson 2: Keowa Pretorius	Secretary 2: Casey Steyl	

1.2. Agenda Items

- Opening and welcome
- Apologies (excuse members who can't make it)
- Introductions from each group
- General questions
- Closure

WEEK 2

2.1. Meeting Details

Date: 26/09/2022	Time: 10:30		
Group 1: 22G02	Chairperson 1: Ilan Malkin	Secretary 1: Adam Blumenthal	
Group 2: 22G03	Chairperson 2: Gabriella Ndhlovu	Secretary 2: Mpho Moloi	

2.2. Agenda Items

- Opening and welcome
- Apologies (excuse members who can't make it)
- Reading and adoption of minutes of previous meeting
- Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Standup Session)
- Answers for all previous raised issues: Check if unreachable supervisor contacted groups, do we submit individual or cohort, whether or not we're splitting into two groups, answer to if each person has to chair or just one group.
- General questions
- Closure

WEEK 3

3.1. Meeting Details

Date: 03/10/2022	Time: 10:30		
Group 1: 22G09	Chairperson 1: Kavisha Reddy	Secretary 1: Zakhna Dhiraj	
Group 2: 22G85	Chairperson 2: Reza Seedat	Secretary 2: Isaiah Chiraira	

3.2. Agenda Items

- Opening and welcome
- Apologies (excuse members who can't make it)
- Answers for all previous raised issues and raising of any new concerns.
- Reading and adoption of minutes of previous meeting
- Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Standup Session)
- General questions
- Closure

WEEK 4

4.1. Meeting Details

Date: 10/10/2022	Time: 10:30		
Group 1: 22G46	Chairperson 1: Neo Mthimkhulu	Secretary 1: Tshepo Chokoe	
Group 2: 22G58	Chairperson 2: Cynthia Kijjambu	Secretary 2: Mampuru Molebogeng	

4.2. Agenda Items

- Opening and welcome
- Apologies (excuse members who can't make it)
- Answers for all previous raised issues and raising of any new concerns.
- Reading and adoption of minutes of previous meeting
- Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Stand Up Session)
- General Outstanding questions
- Closure

WEEK 5

5.1. Meeting Details

Date: 17/10/2022	Time: 10:30		
Group 1: 22G20	Chairperson 1: Sanet Witbooi	Secretary 1: Nomtha Njamela	
Group 2: 22G34	Chairperson 2: Maria (Louise) Botha	Secretary 2: Lutfiya Charfaray	

5.2. Agenda Items

- Opening and welcome
- Apologies
- Answers for all previously raised issues and raising of any new concerns
- Allow each group to present their progress for the previous week and their plans for the following week, as well as allow for feedback or information sharing from other groups
- General questions / any other business
- Closure

WEEK 6

6.1. Meeting Details

Date: 24/10/2022	Time: 10:30		
Group 1: 22G01	Chairperson 1: Mukundi Mushiana	Secretary 2:	Sinazo Thungo
Group 2: 22G73	Chairperson 2: Lungelo Chala		

6.2. Agenda Items

- Opening and welcome
- Apologies
- Answers for all previously raised issues and raising of any new concerns: Open day issue was addressed by Dr. Genga.
- Allow each group to present their progress for the previous week and their plans for the following week, as well as allow for feedback or information sharing from other groups
- General questions / any other business
- Closure

WEEK 7

7.1. Meeting Details

Date: 31/10/2022	Time: 10:30		
Group 1: 22G19	Chairperson 1: Faatimah Mahomed	Secretary 1:	Milan Jeevan

7.2. Agenda Items

- Opening and welcome
- Apologies
- Allow each group to present their progress for the previous week and their plans for the following week, as well as allow for feedback or information sharing from other groups
- General questions / any other business
- Closure

APPENDIX 4B: Minutes of Each Weekly Cohort 8 Meeting Taken by the Secretary(s)

WEEK 1 – MEETING 1

Date and Time

21/09/22 at 10:00

Venue

Microsoft Teams

Names of Participants

Students of groups 22G09, 22G46, 22G19, 22G01, 22G73, 22G05, 22G34, 22G02, 22G20, 22G50, 22G85, 22G58, 22G03

Meeting Hosts

22G05

Chair: Jesse van der Merwe

Secretary: Robyn Gebbie

22G50

Chair: Keowa Pretorius

Secretary: Casey Steyl

Purpose of Meeting

This meeting was an introductory meeting to get to know the members of cohort 8. The meeting also provided a forum for discussion about future meetings and how they would be conducted.

Agenda Items

- Group members introduce themselves
- Description of each group's project
- Decide on date and time of meetings going forward
- Decide on how future meeting agenda items will be decided upon
- Allocate meeting hosts for future meetings

Discussion Notes

Group Introductions

- 22G09: Using control knowledge to improve the grip of a bionic hand.
- 22G46: Building an improved sensing armband for a bionic hand.
- 22G19: Using AI to convert sign language to text.
- 22G01: Assisting visually impaired by building cane with a camera that notifies user of obstacles using audio.
- 22G73: Designing a heart rate monitor that will be used to identify health problems in users.
- 22G05: Analysing spiral drawings to determine if FUS is effective treatment for patients with Essential Tremor and Parkinson's Disease.
- 22G34: Building a computerised tremor assessment for patients with Parkinson's Disease.
- 22G02: Determining difference between persuasive and non-persuasive speech using data and signal analysis.
- 22G20: Using neural networks to monitor heart conditions and determine what the patient is suffering from.
- 22G50: Using machine learning and signal processing to diagnose cardiac conditions using cardiac sounds.
- 22G85: Determining the correlation between Radiohead albums using digital signal processing.
- 22G53: Evaluating changes occurring in virus trajectories by using entropy to measure these changes.
- 22G03: Investigating the effect of lightning on nerve strands. Generating lightning current using impulse generator. Investigating the stability of using coaxial cables instead of actual nerves.

Group Organisation

The chairperson asked the groups if they had managed to get in contact with their supervisors. It was found that all groups were able to contact their supervisors and had made arrangements to meet with their supervisors privately with the exception of one group. This group would continue to try make contact. If they were still unable to make contact by the next meeting the other groups would contact their supervisors and ask for advice.

Future Meeting Arrangement

- A roster was set up to determine who was going to chair the meetings each week. This was done by making a spreadsheet available to the cohort and allowing for each group member to sign up.
- It was decided to create an agenda system in order to decide on the topics that will be discussed in the future meetings. This was done by making a Google Document available to the whole cohort, each member could add their topics of concern to the document. It was decided that if the agenda items were not discussed in meetings due to time constraints they would flow over to the next meeting.
- It was decided that future meetings would be held on Mondays at 10:30. This day was decided upon by allowing the members of the cohort to vote between Mondays, Wednesdays, and Fridays.
- It was decided that the meetings did not need to include the group supervisors.

Questions

Several questions were raised that would need to be answered by the head of this course.

- Should each person take individual minutes or is one set of minutes for everyone fine?
- Can the cohort split into 2 groups for meetings and if so, do we need to combine the meeting minutes from these 2 groups.

The chairperson of this meeting will email the head of the course to find the answers to these questions.

Next Meeting Date

26/09/2022

Adjournment

The meeting was adjourned by Jesse van der Merve at 10:54.

Cohort 8 Minutes by 22G50

21 September 2022 10:am

Jessies group and my group are chairing - 22G05 & 22G50
Just some general conversation in the group about what the project will entail

A quick vote for which day to have the meeting. Monday wins.

Group 9 talks about their project to hold a pen and a marble with a pen and a bionic hand.

22G46 - Designing armband to take emg signal from amputees to a bionic hand to control its movement.

22G19 - Sign language translation device - will implement it using gloves with sensors connected to arduinos. Gestures transmitted to bt speaker to hear and understand the sign language.

22G01 - Making a device for assisting visually impaired individuals. Cane with ultrasonic sensors and camera connected to reaberry pi and sensor connected to arduino. If person approaches object their cane gives feedback as vibration. Bt earphones will also make noise to move away from object.

22G73 - Building device to monitor heartrate and will be using LED's to measure blood pulsations in finger and compute heartrate.

22G05 - analysing handrawn circles and analysing to see if persons Parkinson's treatment is working and helping fix the disease

22G34 - app to help physicians detect tremors in people with Parkinson's

22G02 - understand difference between persuasive and non-persuasive speaking. Speech processing

22G20 - Building and coding a system which will take in capillary readings to identify neurological data from patients and coalesce a database.

22G50 - use neural networks and AI to monitor heart conditions. Automatically determine cardiac murmurs using ai and machine learning and signal processing

22G85 - Radiohead has Easter eggs in music. Correlation and signal processing and machine learning to find correlation between two albums

22G58 - Developing quantification model for inferring changes on viral sequences. Develop parametrics to evaluate changes that have occurred between viral sequences. Specifically for covid 19. Check degree of changes and evolutionary changes. Once they have model they hope to test with covid 19, influenza and hiv to check how feasable and accurate the model is.

They met up with people from bioinformatics to assist them with finding artificial changes. Not prevalent in literature and so needs to be figured out this week.

22G03 - investigating effect of lightning on nerve strands in nervous system. Generate lightning and then use coaxial as nerve proxy to see effects induced - currents and voltages. Collect this data and create model on how expect nerves to behave in these lightning electromagnetic fields.

Generate current impulses similar to lightning. Use impulse generator. Objective is how to use the generator and understand circuit. Generated lightning currents already. Expected rise and fall times. Research coaxial instead of nerve use for the project and how this will affect measurements. Effect of an open nerve on the human body with standing waves and electrical impulses.

Added a channel to teams thing where we can talk about small questions we have

Anyone struggling to talk to supervisors? - Adam was emailed on Monday and they still have no response. They will try call him and see what the problem is. Will check up with this next week.

Any questions for other groups? Not necessarily.

Need to set up a roster for who will be chairing each week. A google sheet will be made and shared on the whatsapp group

Okay with 10 on Monday? If people have loadshedding just notify everyone beforehand. Group 2 wants to chair next week and group 3 next week - Both groups will chair

Can we make it 10:30? Doesn't seem to be a problem. Did a vote and everyone is happy.
All weeksmeetings to be changed to 10:30

Minutes to be uploaded to files section of MS teams group on the day of the meeting and no later.

Check we can submit minutes as a cohort or individually - check this for all cohorts

Any other general questions?

Jessie will email and clarify about how chairpersons will work. Since we have an exceptionally large cohort,

Check if we can split the cohorts up as we get more detailed to save time

Make a google doc for the agenda for each week. Chairpersons will carry over things from previous weeks and anyone can access it and add their own things to the agenda. Only one google doc that continuously grows

Can we split into two groups? Will we have two separate minutes? Jessie will email and find answers to this

22G05 and 22G50 closed the meeting. Next meeting on Monday September 26 2022
@10:30

Cohort 8 Meeting Minutes

Location : Microsoft Teams
Date : 26 September 2022
Time : 10:30
Attendees : All groups from cohort 8 are present
Chair : Gabriella Ndhlovu, Ilan Malkin
Secretary : Mpho Moloi, Adam Blumenthal

10:33 Meeting Starts

- Opening and Welcome by Gabriella Ndhlovu
- Issue about meeting recordings
- Suggestion to not record meetings so as to create a safe space for everyone to voice out their concerns about anything and everything
- Instead of recording, it was suggested that people go through minutes to find out what was said in the meetings

10:37 Apologies: Gabriella

- One member from 22G46 (Tshepo Chokoe) couldn't make it

10:38 Reading and adoption of minutes of previous meeting: Gabriella

- Last week we discussed the scopes and objectives of the projects
- How will chairing work
- Check if we can split the cohort into two different groups
- Have google form to put minutes and agenda
- Groups who couldn't reach their supervisors

10:40 Standup Session: Ilan

- Jesse sent an email to the course coordinators and got no response
- Allowing each group to present

22G09

Precision grip. Got their parts. Tested parts. Problem and sorted it out. After meeting they focusing one sensor this week. Redid mechanics of hand. Make one sensor and test it out.

22G46

- Managed to order some parts. Waiting for delivery
- Finished paper design. Put order today and waiting for parts
- For this week, design the 3D CAD design for electrodes. 2D print of substrate electrodes

22G19

- Making flat sensors.
- Been through a few designs
- Problem with resistance values
- Got components, waiting for others. Hopefully will get others before the week ends.

22G01

- Working on project alone so scope had to change. Met with supervisor and presented new reduced scope.
- Ordered components. Haven't received anything.
- This week, waiting for components and doing software part of the project

22G73

- Worked with filter and amplifier
- This week: look at software. Can't order components as ethics application hasn't been approved.

22G05

-Last week: did research and background and got in touch with MSc students through help of supervisor which will give them code and software

-Will start implementing code that does image processing this week

22G34

- Got ethics approval
- Chose python for mobile development
- Doing tremor analysis
- Chose what to do for database
- Want to create user interface this week

22G02

- Extracted features for data analysis
- Want to recode to make it work with big data

-Find out what testing to use. Generalizing for data analysis

22G20

-Solved ethics, got clearance

-Compiled database and collection

-This week: build software to analyse data

22G50

-Did new ethics letter last week

-Continuing feature extraction for heart monitoring system

-Want to train data for Machine Learning program

22G85

-Did background about digital signal processing

-Need to purchase the albums first

22G58

- Working on info entropy system. Got additional info from bio informatics department. Checked if scope needed to be adjusted.

-This week: present adjusted scope for review. Information entropy model

22G03

- Generated impulse. Weekly meetings with supervisor.

-Got idea from supervisor to first simulate the model.

-Need to know if results getting are correct.

-Goal to model and simulate system on Simulink first. Model should be using nerve, using coax cable to get the experiment results.

-Final model should be more nerve based than coax based.

-Software tools to use models nerve behavior.

-Looked at Hodgkin model for the nerve model.

-This week planning to finish simulation. Reading to figure out how to get data.

10:58 Answers for all previous raised issues: Gabriella

-No response from email suggesting the cohort split into two groups

- Look at the matter of submitting own minutes or group minutes next week
- Dr. Pantanowitz finally responded and the meeting was set with students

10:59 General Question: Gabby

- Follow up on the email sent last week
- It is suggested that groups ask for help from other groups

11:07 Closure

- Gabriella Ndhlovu adjourned the meeting

Meeting Minutes 26 September 2022

Cohort 8

1. Opening and welcome

Gabby:

Isaiah: Recordings so we can look back on stuff?

Ilan: Meetings don't need to be recorded.

2. Apologies (excuse members who can't make it)

22G46 one member couldn't make it.

3. Reading and adoption of minutes of previous meeting

Gabby: How chairing works. We will address stuff raised last week. Split into 2 groups of 6 groups. Look back at some groups they couldn't reach supervisors.

4. Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Standup Session)

Ilan: Quick summary and plans for next week. Any blockers and advice from other groups.

5. Answers for all previous raised issues: Check if unreachable supervisor contacted groups, do we submit individual or cohort, whether or not we're splitting into two groups, answer to if each person has to chair or just one group.

No response from lecturers about splitting into smaller groups and waiting to find out about minutes for it.

Adam contacting him and got back to them last week.

6. General questions

Problems and/or questions:

7. Closure

Concluding meeting and message other groups doing similar thing if need help.

Group	Response
22G09	Precision grip. Got their parts. Tested parts. Problem and sorted it out. After meeting they focused on one sensor this week. Redid mechanics of hand. Make one sensor and test it out.
22G46	Ordered some parts. Waiting on them to be collected/delivered. Focused on design and did it on paper. Ordered components for this week and waiting to get them to build their circuits. 3d cad design to be started for electrodes. Get substrates for electrode.
22G19	Making flat sensors. Been through a few designs. Having a few problems with resistance values. Got a few today and end of this week should have them all.
22G01	Working alone. Redo the scope and present the scope to supervisor about project and what he intends to do. Meeting supervisor and discussed new reduced scope. Ordered some components. Waiting for them to arrive. Haven't received them as of yet. Working on software part of project in the meantime.

22G73	Working on filter and amplifier last week. Looking at software side of project. Haven't ordered components and waiting on ethics approval. If order now may need to change components while ethics approval.
22G05	Background and research for project. Vered put her in contact with previous masters students. Masters will give some code and software for them.
22G34	Ethics approval granted. Did some research. Python and ? for mobile development. Digital spiral assessment for database. Focusing on reg page for patients. Allow user to draw on spiral tem
22G02	Extracted all features. Research on stats tests. This week generalising for data analysis.
22G20	Ethics clearance. Data collected. Build software environment for analysing data once samples are taken.
22G50	Whole new ethics letter last week. Feature extraction for heart monitoring system. Segments split to train ML program.
22G85	Did background for project. About digital signal processing. Need to purchase the albums first.
22G58	Working on info entropy system. Got additional info from bio informatics department. Checked if scope needed to be adjusted. Presenting adjusted scope with scope. And start implementation of group.
22G03	Generated impulse. Weekly meetings with supervisor. Got idea from him to first simulate the model. Need to know if results getting are correct. Goal to model and simulate system on simulink first Model should be using nerve using coax cable. Final model should be more nerve based than coax based. Characteristics of nerves. Fluids that surround nerve. Software tools to use models nerve behaviour. Hodgkin model. This week planning to finish simulation. Reading to figure out how getting data.

Meeting Minutes 3 October 2022 by Zakhna Dhiraj

Chaired by Groups 9 and 85

Cohort 8

1. Opening and welcome
2. Checked attendance of all groups.
3. Apologies
 - Adam sent apologies as he will join the meeting a bit late.
4. Revised last week's minutes.
5. Follow up on last week's queries.
 - Follow up regarding the query on whether to split the cohort into 2 groups – Yuval Genga said yes, it is allowed.
 - Groups that have Professor Pantanowitz and Professor Rubin as supervisors have a similar meeting on Thursday thus, they we can split accordingly.
 - A discussion occurred as to whether splitting the cohort was necessary.
 - A vote was done to determine whether the group should split from next week.
6. Progress from each group:

Group	Progress
22G09	Constructed 3 force sensors, however measurements are inaccurate, therefore a measurement system for the force sensors needs to be created. This system will be tested for a few days to see the consistency of the measurements. The design of the packaging for the circuit will also be worked on this week. This week will also be used to work on the control system and feedback from the sensors.
22G46	The power supply was completed. Testing of the arm band will be done, currently working with galvanized steel and brass electrodes. Currently completing the building of the signal conditioning and the design of the signal processing circuit. If extra time, the previous arm band will be tested.
22G19	Tested different sensors, materials, and configurations and established a design for a sensor. This week will be used to test the sensor and establish more consistent readings over different periods of usage.
22G01	Received components on Friday, the software component of the design was worked on. This week will be used to build an ultrasonic sensor as well as a vibrating feedback mechanism and a piezoelectric buzzer, as well as an accelerometer with a text-to-speech speaker.
22G73	Ordered components. The ethics community said the submission was too late thus a heartbeat will have to be modelled which will be worked on this week. There were issues with the input thus an oscillator had to be designed. Waiting for components to start the signal conditioning aspects.
22G05	Met with a master's student who completed similar research. A small portion of the student's code will be implemented as part of the preprocessing of the data. Further implementation of the software will be done this week.
22G34	Created a login page, a menu page and a form for the doctor to fill out the patient's information. A database was created to store the patient's information. This week's focus is to query patients' data from the database and to set out a template spiral drawing for the patients to draw on.
22G02	Analysis of debating speeches verses ted talks. Focused on extracting features from debating speeches using clusters and MPI. This week will be used to conduct feature extraction and preprocessing on the ted talk speeches.

22G20	Last week was used for image processing while this week will be used for preprocessing and analyzing code. This will allow interpolated samples to be collected so that the control model can start being designed. There was a problem with components as they could not be ordered through Wits, therefore they have been ordered via the supervisor which will cause a delay of a few days.
22G50	Last week was used to test different algorithms in diagnosing different murmurs. Some feature extraction was completed to train different machine learning models which resulted in a high accuracy. This week will be used to merge the different algorithms and feature extraction together to diagnose a heartbeat.
22G85	Obtained the albums and tested the musical similarity between the two albums and added a third one as a control measure. This week will be used to alter the crossfade of the albums as it may help detect more similarities.
22G58	Started to implement the entropic model and this week will be used to simulate it and plot all the entropic parametric curves. The nucleotide motifs from Covid-19 virus will also be looked at.
22G03	Got induction measurements of voltage and current that was induced in the coaxial cable and discussed the measurements with the supervisor. A question was asked as to how the students know these measurements are not noise and discussed how the protection of the measurement system can act as a frequency filter. The frequencies that will be allowed and not allowed through the faraday cage need to be determined and the simulation needs to be completed.

7. From the votes the majority voted that the group should split. From next week, the groups will be split and will attend different meetings at the same time. The splitting of the groups will be discussed on Whatsapp.
8. The meeting was closed off by Kavisha Reddy.

Meeting Minutes 3 October 2022

Chaired by Groups 9 and 85

Cohort 8

1. Opening and welcome
2. Checked attendance of all groups.
3. Apologies
 - Adam sent apologies as he will join the meeting a bit late.
4. Revised last week's minutes.
5. Follow up on last week's queries.
 - Follow up regarding the query on whether to split the cohort into 2 groups – Yuval Genga said yes, it is allowed.
 - Groups that have Professor Pantanowitz and Professor Rubin as supervisors have a similar meeting on Thursday thus, they we can split accordingly.
 - A discussion occurred as to whether splitting the cohort was necessary.
 - A vote was done to determine whether the group should split from next week.
6. Progress from each group:

Group	Progress
22G09	Constructed 3 force sensors, however measurements are inaccurate, therefore a measurement system for the force sensors needs to be created. This system will be tested for a few days to see the consistency of the measurements. The design of the packaging for the circuit will also be worked on this week. This week will also be used to work on the control system and feedback from the sensors.
22G46	The power supply was completed. Testing of the arm band will be done, currently working with galvanized steel and brass electrodes. Currently completing the building of the signal conditioning and the design of the signal processing circuit. If extra time, the previous arm band will be tested.
22G19	Tested different sensors, materials, and configurations and established a design for a sensor. This week will be used to test the sensor and establish more consistent readings over different periods of usage.
22G01	Received components on Friday, the software component of the design was worked on. This week will be used to build an ultrasonic sensor as well as a vibrating feedback mechanism and a piezoelectric buzzer, as well as an accelerometer with a text-to-speech speaker.
22G73	Ordered components. The ethics community said the submission was too late thus a heartbeat will have to be modelled which will be worked on this week. There were issues with the input thus an oscillator had to be designed. Waiting for components to start the signal conditioning aspects.
22G05	Late last week met with a master's student who completed similar research. A small portion of the student's code will be implemented as part of the preprocessing of the data. Further implementation of the software will be done this week.
22G34	Created a login page, menu page and a form for the doctor to fill in to add new patients. Also set up a database to store the patients details. Planned to focus on searching for existing patients on the database and creating a spiral drawing templates.
22G02	Did analysis of debating speech vs Ted talks specifically extracting features from debate speeches. Planned to do more preprocessing.

22G20	Worked on image processing code. Planned to continue working on the processing code as well as analysis. Reported on a setback as far as components ordering is concerned.
22G50	Worked on different algorithms to diagnose S1S2 for different types of murmurs. Planned to do code integration.
22G85	Tested for music similarity between OK computer and in rainbows using machine learning algorithms with Kid A as a control album. Planned to test 01 10 theory.
22G58	Implemented the entropic model. Planned to start simulation and putting all entropy parametric curves.
22G03	Gave a feedback on their meeting with the supervisor on protecting the measurement equipments.

Weekly Meeting Minutes

Date | Time: 10 October 2022 | 10h30 – 11h00

Chairpersons: Neo Mthimkhulu and Cynthia Kijjambu

Secretaries: Tshepo Chokoe and Mampuru Molebogeng

Week 4

Agenda:

1. Opening and welcome
2. Apologies (excuse members who cannot make it)
3. Answers for all previous raised issues and raising of any new concerns.
4. Reading and adoption of minutes of previous meeting
5. Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Stand Up Session)
6. General Outstanding questions
7. Closure

Proceedings:

1. Opening and welcome by Neo Mthimkhulu and Cynthia Kijjambu
2. Check attendance for all groups to report any students in absentia
 - All members were present for the meeting
3. Follow up on last week's queries
 - The cohort decided to chair the meeting like the previous ones and implement the splitting in Week 5, to avoid time delays
 - The cohort is to be split into two sub-cohorts that will run concurrently, how the groups would be split will be decided before the end of the week according to project focus.
 - It was raised that the splitting at this point could be redundant since the project timeline is reaching conclusion. A new voting poll was made and majority voted for not splitting
4. Revised last week's meeting minutes to check for any outstanding queries

5. Progress of each group is summarised in a table below

Group	Progress report
22G09	Finish up sensor design and fit it onto the bionic hand. This week focus is to control the finger movement to improve precision. Integrate sensor into the bionic hand.
22G46	Done signal conditioning subsystem and integrating the subsystems. Plan is to conclude integration of subsystems this week
22G19	Have three designs on sensors. Plan is to integrate other sensors by the end of this week
22G01	Finished doing obstacle and fault detection subsystem. Halfway through voice feedback subsystems where the blind person will be listening through Bluetooth. Integrate subsystems altogether.
22G73	Trying to model and finished the circuitry. This week will be starting with the mobile app.
22G05	Extracted all spirals for images. Devised methodology to detect tremor severity
22G34	Created the assessment template to access the patient information from the database. Signal and image processing to be done.
22G02	Started the pre-processing and statistical testing, which hasn't run according to plan. Plan is to try finish testing this week.
22G20	Completing imaging and pre-processing code and received components. Planning to finish code and make 3D chin rest.
22G50	Pretty much done and everything works accordingly, left with testing the auscultation prototype.
22G85	Continuing with similarity testing, found that there is correlation between the

	algorithms and currently looking for ways to improve
22G58	Completed the entropy computation model, the next focus would involve devising the time evolution of entropy parametric curves
22G03	Supervisor advised that there is always interference. Therefore, it must be minimised and quantified. Induction measurements to be done.

6. General Outstanding questions

- Adam asked if we will be allowed to make changes on our projects after the 3rd of November 2022. It was advised that the question be taken up with course coordinators and/or supervisors.

7. Closure by Cynthia Kijambu and Neo Mthimkhulu

Week 4 Cohort 8 Meeting Minutes

Date: October 10th, 2022

Chaired by: Cynthia Kijjambu (22G58) and Neo Martin Mthimkhulu (22G)

Secretary: Mampuru Molebogeng

1. Opening and welcome: Cynthia Kijjambu and Neo Mthimkhulu
2. Apologies (excuse members who can't make it)
 - All members were present for the meeting
3. Answers for all previous raised issues and raising of any new concerns.
 - The cohort decided to chair the meeting like the previous ones and implement the splitting in Week 5, to avoid time delays
 - The cohort is to be split into two sub-cohorts that will run concurrently, how the groups would be split will be decided before the end of the week according to project focus.
 - It was raised that the splitting at this point could be redundant since the project timeline is reaching conclusion. A new voting poll was made and majority voted for not splitting
4. Reading and adoption of minutes of previous meeting
5. Allow each group to present their progress for the previous week and their plans for the following week, and allow for feedback or information sharing from other groups (Stand Up Session)

Group	Project
22G09	Finish up sensor design and fit it onto the bionic hand. This week focus is to control the finger movement to improve precision.
22G46	Done signal conditioning subsystem and integrating the subsystems. Plan is to conclude integration of subsystems this week
22G19	Have three designs on sensors. Plan is to integrate other sensors by the end of this week
22G01	Finished doing obstacle and fault detection subsystem. Halfway through voice feedback subsystems where the blind person will be listening through Bluetooth
22G73	Trying to model and finished the circuitry. This week will be starting with the mobile app.
22G05	Extracted all spirals for images. Devised methodology to detect tremor severity
22G34	Created the assessment template to access the patient information from the database.
22G02	Started the pre-processing and statistical testing, which hasn't run according to plan
22G20	Completing imaging and pre-processing code and received components. Planning to finish code and make 3D chin rest
22G50	Pretty much done and everything works accordingly, left with testing the auscultation prototype
22G85	Continuing with similarity testing, found that there is correlation between the algorithms and currently looking for ways to improve
22G58	Completed the entropy computation model, the next focus would involve devising the time evolution of entropy parametric curves

22G03	Worked around the faraday cage, leaving some interference to occur at minimal magnitudes, we will be able to then take induction measurements
-------	---

6. General Outstanding questions

- Adam asked if we will be allowed to make changes on our projects after the 3rd of November 2022. It was advised that the question be taken up with course coordinators and/or supervisors.

7. Closure by Cynthia Kijjambu and Neo Mthimkhulu

MINUTES OF THE MEETING OF ELEN4012 INVESTIGATION COHORT 8,

HELD ON 17 OCTOBER 2022 AT 10:30 AM ON MICROSOFT TEAMS

PRESENT:

- All the groups in Cohort 8
- Hosted by Group 34 & Group 20
- Chairs:
 - Maria Botha
 - Sanet Witbooi
- Secretaries:
 - Lutfiya Charfaray
 - Nomtha Njamela

1. Maria Botha opened the meeting by welcoming the cohort members.
2. Adam Blumenthal sent his apologies as he could not attend the meeting.
- 3.1. Previously raised concerns: Adam previously asked if we are allowed to make changes to our project after the open day (3 November 2022). Maria asked if anyone got feedback regarding this.
- 3.2. New concerns: Whether we should host cohort meetings after the 3rd of November. Sanet has emailed the course coordinators regarding this and will inform the group on WhatsApp. Casey asked where we are supposed to submit the minutes of our meetings. Lungelo and Mukundi responded saying that it should be in the appendix of our reports and Sanet suggested that we should confirm this with our supervisors. Milan asked if the cohort will still have a meeting the week of the 31st of October 2022 and Maria suggested that we should, this will allow Milan's group to fulfil their chairing requirements.

4.

Group Number	Progress and Plan
22G09	Last week: Integrated their systems and completed their circuitry. This week: Testing of sensors and reliability testing.
22G46	Last week: Ordered parts. This week: Busy with circuitry and hardware design for electrode.
22G19	Last week: Settled on reliable sensor configuration. This week: Working on calibration and creating the database.
22G01	At the moment: Combining circuitry, testing of cane and reliability of sensors.
22G73	Last week: Looked at mobile app development and finalizing peak detection on code. This week: Mobile app development and moving circuitry to Veroboard.
22G05	Last week: Result of spirals, indication of the amount of tremor in spirals and lines. This week: Automate results and start writing report.

22G34	Last week: Timing of assessments, created a visual assessment rating tool, packaging of app for android. This week: Complete signal processing and testing of app.
22G02	Last week: Extracted features, tested data. This week: Machine learning and writing of reports.
22G20	Last week: Completed code and built system. This week: Feed in blind tests, calibrate and complete code.
22G50	Last week: Finished testing on all their data and all their algorithms are trained. This week: Complete their reports.
22G85	Last week: Similarity tests. This week: Confirm results using machine learning.
22G58	Last week: Finished entropy model and tested it with a few Covid-19 sequences. This week: Quantification model.
22G03	Last week: Varied distances to see inductive effects. Recorded results. This week: Verify results. Review and compare coaxial cables to nerves. Vary length of the coaxial cable and research the frequencies of lighting.

5.The next cohort meeting will take place on 24 October 2022 at 10:30am on MS Teams

6. Maria and Sanet adjourned the meeting at 10:48 am.

Week 5 Meeting Minutes by Nomtha Njamela

Date and time: 17th October 2022 at 10:30

Chairs: Sanet Witbooi and Maria (Louise) Botha

Secretaries: Nomtha Njamela and Lutfiya Charfaray

Groups: 20 and 34

Cohort 8

1. Opening and welcoming by Louise

2. Apologies

- Apologies from Adam Blumenthal for not attending the cohort meeting.

3. All previous issues were brought up and new raising concerns were addressed:

Previously raised concerns:

- Adam asked if changes are allowed to be made on the projects after the 3rd of November (Open day), Sanet offered to email the course coordinator regarding this concern.

New concerns:

- Casey – asked where minutes should be placed in the report. Meeting members Lungelo and Mukundi says it should be in the report's Appendix
- Milan – asked if a cohort meeting will take place the week of open day. Meeting member Maria confirmed that it will take place for them to fulfil their chairing requirements.

4. Sanet asked each group to present a quick summary of what they have achieved so far and what they are planning to do going forward:

Group	Progress
22G09	Previous week: integrated their mechanical sensory with their power supply. All of their circuitry is done, and they started testing their sensors This week: reliability tests will be done, like the hand holding, to make sure that the measurements are consistent in order for them to start deciding on how to control the actual movement.
22G46	Previous week: they ordered components. This week: boarding their circuitry and working on the hardware design for electrode
22G19	Previous week: settled on a reliable sensor configuration This week: working on calibration and creating their database.
22G01	Previous week: combining circuitry and doing tests This week: do reliability test for chain and the sensors
22G73	Previous week: looking at the mobile app development, learning how to program a mobile app and finalizing the peak detection of their code. This week: log into the mobile app development and finalize and move the circuitry.
22G05	Previous week: they received the result for their spirals. This week: automate spirals so that they can find it for every patient that they have data on and start writing.

22G34	Previous week: timed the assessments and created a visual assessment rating tool. Attempted to package the app for Android, but there are still some issues. <u>This week: to complete the signal processing and start testing the app</u>
22G02	Previous week: tested data and extracted Features <u>This week: machine learning and being with writing the report</u>
22G20	Previous week: built system and complete 90% of the code. <u>This week: feed in blind tests, calibrate and complete the 10% of the code</u>
22G50	Previous week: finished testing on all their data and algorithms are trained. <u>This week: Started writing their reports.</u>
22G85	Previous week: translation comparisons and visual processes to spot differences. <u>This week: confirm results using machine learning methods</u>
22G58	Previous week: received entropy model and tested it out with a few COVID-19 sequences. <u>This week: time evolution quantification model</u>
22G03	Previous week: verified the distance and inductive effect. Results were recorded. This week: verification of the results. Review what a nerve is represented as electrically and compare that with how a coaxial cable is represented electrically. Verification of the length of their coaxial cable, find lightning frequencies that correspond to which lengths to induce voltages.

5. The meeting was adjourned at 10:48 by Maria and Sanet

Meeting Minutes 24 October 2022

Chaired by Groups 01 and 73

Cohort 8

1. Opening and welcome
2. Checked attendance of all groups.
3. Apologies
 - Milan from Group 19.
 - Group 9.
4. Revised last week's minutes.
5. Follow up on last week's queries.
 - Queries on open day were addressed by Dr. Y. Genga. A communication is to be released shortly.
6. Progress from each group:

Group	Progress
22G09	Excused.
22G46	Last week: Completed the 3D printing for the two channels. Waiting for components to duplicate the system for the two channels. This week: Still busy with the signal processing stage. Need to standardize the testing methodology and add the linear envelope to the Veroboard to complete the system for one channel.
22G19	Last week: Sensors are working on the glove and a correct output is obtained. Letters are recognized as the hand gestures change. This week: Busy with the text-to-speech using Bluetooth interface.
22G01	Last week: Took data from the cane in relation to the ultrasonic and accelerometer sensors to detect falling for response generation (a buzzer). Planning on working on sending an SMS to someone (a relative) in case a cane user gets injured. This week: Waiting for components. Working on data analysis and results. To start working on the poster.
22G73	Last week: Busy with the mobile application development. Tried integrating the system on Friday but still experiencing a few bugs in the software. This week: To work on fixing those bugs and try moving to a Veroboard.
22G05	Last week: Worked on two methods, one of the group members had method that performed well. This week: To collect results from the best performing method and start working on the poster. To start working on system validation.
22G34	Last week: Focused on signal and image processing of the tremor drawings. Added analysis to results. Fixed the UX functionalities. Performed unit testing. This week: To complete the testing and work on making the application run on an android device (tablet).
22G02	Last week: Got results that were too good to be true. This week: To test the system using a different dataset for validation.

22G20	Last week: Done with the system and the coding part. Created the control model. This week: To gather data on the abnormalities and decide which ones are to be investigated for the purpose of this project. This will be based on the data available. To perform system validation.
22G50	Last week: Created a GUI for the application to display the different heart auscultations. This week: To start working on the poster.
22G85	Last week: Confirmed that the theory they are using is not true. Still trying to formulate a new one. This week: To come up with a different theory and work on data visualization for past results for the poster.
22G58	Last week: Test functional and random sequences on the model created. Came up with a conclusion to their hypothesis. This week: Results verification with supervisor. To try and finalize the time evolution of the model. To test and get results.
22G03	Last week: Finished taking lab measurements. This week: To work on scaling the model and basing it from using a coaxial cable to a nerve. To model the effect the effects using a real nerve instead of a coaxial cable.

7. Queries:

- Open day (start and end times).

Meeting Minutes 31 October 2022 (Milan Jeevan)

Chaired by Group 19 (Faatimah Mahomed)

Cohort 8

1. Opening and welcome
2. Checked group attendance
3. Apologies
 - Nomtha (Group 20) excused due to a medical appointment.
4. Allowed each group to present their progress (tabulated below) for the previous week and their plans for the current week, as well as allow for feedback or information sharing from other groups

Group	Progress
9	Experienced issues with connections and sensor readings, had to reconnect and remake sensors and subsequently retake measurements. Plan to finalise Open Day presentation this week.
46	Experienced a setback with burnt components, had to redo the circuit setup with new components
19	Received ethics clearance last Friday, working on movement with human hand. Open Day preparation to also be done this week.
1	Have achieved proof of concept. Currently busy with recording and documenting results in preparation of Open Day.
3	Had to replace Bluetooth module and thus adjust the code previously written. Working on web app since issues were experienced with the mobile app. Open Day prep planned for this week.
5	Have finalised results (tremor values and related data), poster is also complete. Open Day presentation to be completed this week.
34	Have tabulated results, have begun and continue to work on unit tests and packaging the app, will also work on the report.
2	Have achieved meaningful results, will complete final testing this week and preparation for Open Day.
20	Has completed system and model along with verified results. Preparation for open day and working on report planned for this week.
50	Poster has been completed, working on packaging application and further preparation for Open Day.
85	Attempted data extraction one final time, will prepare for Open Day this week.
58	Tests done on entropy model. Supervisor verified the methodology and made further suggestions. Currently working on finalising tests and preparing for Open Day.
3	Currently working on model, Open Day preparation planned for this week

5. General questions / any other business
 - The topic of the Open Day poster was brought up, majority of the groups were unclear and seeking clarity with regards to poster format and content. No concrete input could be provided during the meeting, it was decided that should anyone receive any helpful advice it is to be shared on the Cohort 8 WhatsApp group.
6. Meeting closed off by Faatimah Mahomed.

Appendix 5: Ethical Clearance Certificate



Office of the Deputy Vice-Chancellor (Research and Innovation)

TO: Miles J van der Merwe and R Gebbie
School of Electrical and Information Engineering
University

E-mail: 1829172@students.wits.ac.za

CC: Supervisor: Professor V Aharonson
<Vered.Aharonson@wits.ac.za>
and <HREC-Medical Research Office@wits.ac.za>

FROM: Mr Iain Burns
Human Research Ethics Committee (Medical)
Tel: 011 717 1252

E-mail: Iain.Burns@wits.ac.za

DATE: 2022/09/12

REF: R14/49

PROTOCOL NO: **M220883** (This is your ethics application reference number. Please quote it in all enquiries, oral or written, relating to this study.)

PROJECT TITLE: *Which hand?*

Please find attached the Clearance Certificate for the above project. I hope it goes well and that an article in a recognized publication comes out of it. This will reflect well on your professional standing and contribute to Government funding of the University.

A handwritten signature in black ink, appearing to read "Iain Burns".



R49 Mlles J van der Merwe and R Gebbie

**HUMAN RESEARCH ETHICS COMMITTEE (MEDICAL)
CLEARANCE CERTIFICATE NO. M220883**

NAME: Mlles J van der Merwe and R Gebbie
(Principal Investigator)

DEPARTMENT: School of Electrical and Information Engineering
University

PROJECT TITLE: Which hand?

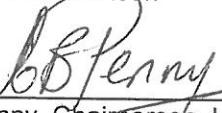
DATE CONSIDERED: Ad hoc

DECISION: Approved unconditionally

CONDITIONS:

NOTE: If contact information regarding student study participants is required,
please contact the Registrar's office - <Nicoleen.Potgieter@wits.ac.za>

SUPERVISOR: Professor V Aharonson

APPROVED BY: 
Dr CB Penny, Chairperson, HREC (Medical)

DATE OF APPROVAL: 2022/09/12

This Clearance Certificate is valid for 5 years from the date of approval. An extension may be applied for.

DECLARATION OF INVESTIGATORS

To be completed in duplicate and ONE COPY returned to the Research Office secretariat on the 3rd floor, Phillip Tobias Building, Parktown, University of the Witwatersrand, Johannesburg.

I/we fully understand the conditions under which I am/we are authorized to carry out the above-mentioned research and I/we undertake to ensure compliance with these conditions. Should any departure be contemplated from the research protocol as approved, I/we undertake to submit details to the Committee. I agree to submit a yearly progress report. When a funder requires annual re-certification, the application date will be one year after the date when the study was initially reviewed. In this case, the study was initially reviewed in August and therefore reports and re-certification will be due in the month of August each year. Unreported changes to the study may invalidate the clearance given by the HREC (Medical).

Signature of Principal Investigator

Date

Appendix 6: Secondary Use of Database Permission Letter (Rambam Medical Center)



31/6/2022

Dear Ms. Van Der Merwe and Ms. Gebbie

Thank you for your enquiry about secondary use of the database relating to "Hand drawing tests of Parkinson's disease and Essential Tremor patients under treatment". This data comprises hand drawn shapes on paper, that patients and control subjects in our hospital drew over time of treatment, and has been fully anonymized. The data also contains MRI scans of the patients, and their medical histories, both of which are anonymized.

As the custodian of this database, I confirm that it was acquired for the experiment "Treatment efficacy in movement disorders patients" which was a study performed at Rambam Medical Center, Haifa, Israel and ethics for the primary collection of the data was obtained at the Medical Center, Haifa, number 4040-17.

I confirm that you may use this database for your project, for computations analysis of the data, subject to you obtaining ethics clearance from the University of the Witwatersrand for secondary use of this data.

Yours sincerely

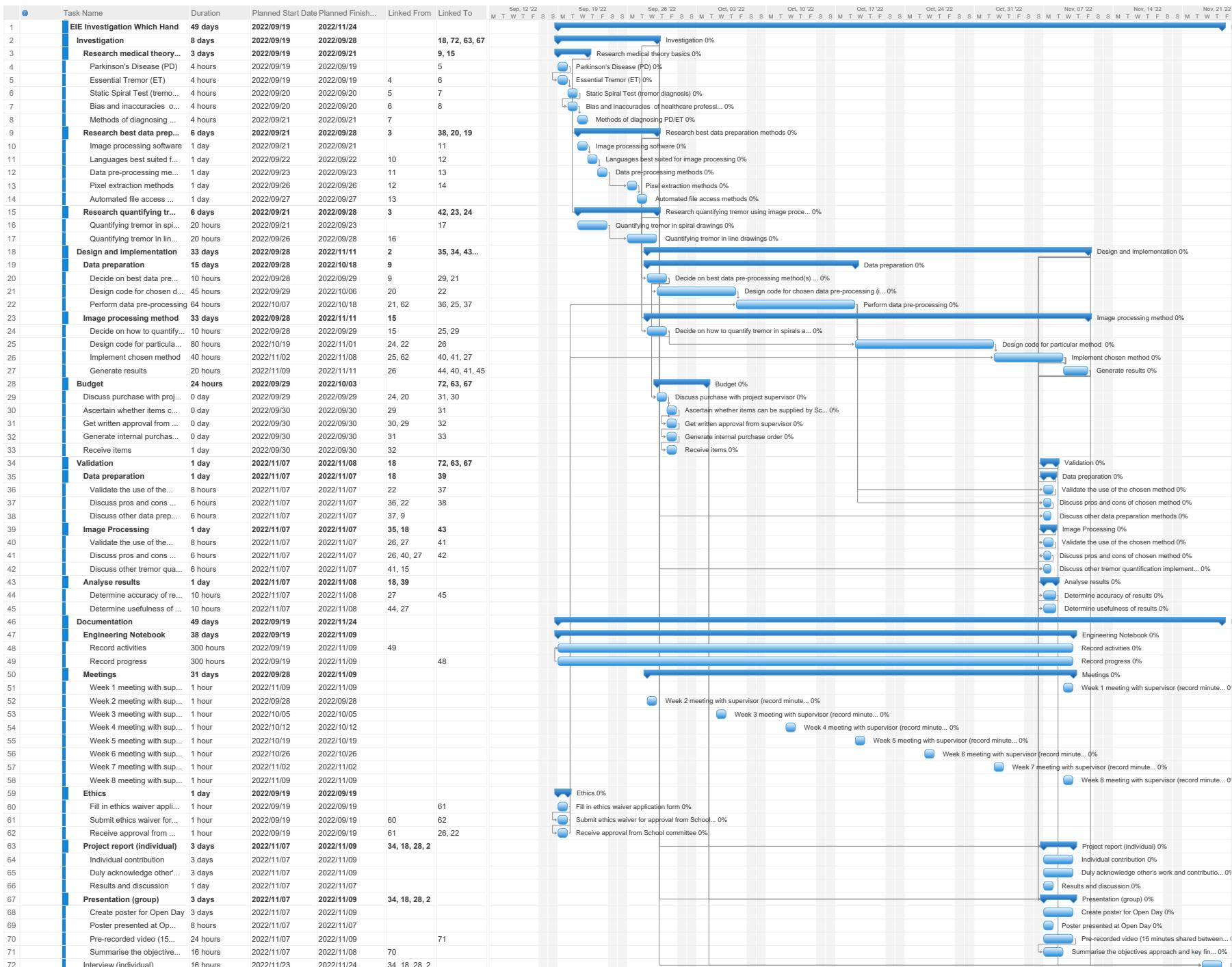
Dr. Ilana Schlesinger
Head of the Gait Disorder Clinic
Rambam Medical Center
i_schles@rambam.health.gov.il

A handwritten signature in blue ink, appearing to read "P. A." followed by a more cursive and illegible name.

Appendix 7: Corrected Gantt Chart for Updated Project Plan

Which Hand? Corrected Gantt Chart

Page 1



Appendix 8: Code Developed for this Investigation Project Compiled and Written by Jesse van der Merwe

This code can be found at: <https://github.com/JessWhosBack/EIE-Investigation-22G05.git>

It should be noted that the appropriate licenses and full credit for all code that has been taken from an outside source is provided at the top of each script.

CONTENTS:

8A. *isolateSpirals.py*

This code is used to accurately extract and save the spirals and line-blocks from each patient's scanned PDF of the filled in template as JPEG images.

8B. *extractRectangle.py*

This code is used to further extract and save the top-most line drawing from each of the line-blocks as a JPEG image.

8C. *method2.py*

This code is used to analyse the JPEG line drawing image and convert it into a useable Python function to calculate various values including average area and number of peaks and save them all in a 2D Python list for further analysis.

8D. *analyseResults.py*

This code takes the list mentioned above and compares, computes, and graphically demonstrates various values, combinations of values as well as combinations between various patients, etc. All to answer the investigation question and provide results in order to determine whether method 2 is of any use.

8A: isolateSpirals.py

```

79. # Arrays used to save the various x and y coordinates, to better calculate the average coordinates
80. array_xStartA = []
81. array_xStartA.append(140)
82. array_xStartB = []
83. array_xStartB.append(710)
84. array_xStartC = []
85. array_xStartC.append(140)
86. array_yEndA = []
87. array_yEndA.append(490)
88. array_yEndB = []
89. array_yEndB.append(490)
90. array_yEndC = []
91. array_yEndC.append(1032)
92.
93. for counter,image in enumerate(image_array):
94.     print("-----")
95.     print("Starting image ", image_names[counter])
96.     copied = image.copy()
97.
98.     orientedImage = copied.copy()
99.     finalA = orientedImage.copy()
100.    finalB = orientedImage.copy()
101.    finalC = orientedImage.copy()
102.    image = orientedImage.copy()
103.
104.    (total_h,total_w) = image.shape[:2]
105.    mask = np.zeros((total_h,total_w), np.uint8)
106.    cv2.rectangle(mask, (0, int(0.245*total_h)), (total_w, int(total_h*0.8)), 255, -1)
107.
108.    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
109.    image = cv2.bitwise_and(image,mask)
110.    kernel = np.ones((3,3),np.uint8)
111.    image = cv2.erode(image,kernel,iterations = 1)
112.    image = cv2.cvtColor(image, cv2.COLOR_BAYER_BG2BGR)
113.
114.    # Important: The EAST text requires that your input image dimensions be multiples of 32
115.    newW = 704
116.    newH = 704
117.    minConf = 0.8
118.    (H,W) = image.shape[:2]
119.    rW = W / float(newW)
120.    rH = H / float(newH)
121.    image = cv2.resize(image, (newW, newH))
122.    (H, W) = image.shape[:2]
123.
124.    # Define the two output layer names for the EAST detector model -- the first is the output probabilities and
125.    # the second can be used to derive the bounding box coordinates of text
126.    layerNames = [
127.        "feature_fusion/Conv_7/Sigmoid",
128.        "feature_fusion(concat_3)"]
129.
130.    # Load the pre-trained EAST text detector
131.    net = cv2.dnn.readNet("Cropping/frozen_east_text_detection.pb")
132.
133.    # Construct a blob from image and then perform a forward pass of the model to obtain the two output layer sets
134.    blob = cv2.dnn.blobFromImage(image, 1.0, (W, H),
135.                                (123.68, 116.78, 103.94), swapRB=True, crop=False)
136.
137.    net.setInput(blob)
138.    (scores, geometry) = net.forward(layerNames)
139.
140.    # Show timing information on text prediction grab the number of rows and columns from the scores volume, then
141.    # initialize our set of bounding box rectangles and corresponding confidence scores
142.    (numRows, numCols) = scores.shape[2:4]
143.    rects = []
144.    confidences = []
145.
146.    for y in range(0, numRows):
147.        # Extract scores (probabilities), followed by geometrical data to derive bounding box coordinates
148.        scoresData = scores[0, 0, y]
149.        xData0 = geometry[0, 0, y]
150.        xData1 = geometry[0, 1, y]
151.        xData2 = geometry[0, 2, y]
152.        xData3 = geometry[0, 3, y]
153.        anglesData = geometry[0, 4, y]
154.
155.        # Loop over the number of columns
156.        for x in range(0, numCols):
157.            # If score does not have sufficient probability, ignore it
158.            if scoresData[x] < minConf:
159.                Continue
160.            # Compute the offset factor as resulting feature maps will be 4x smaller than the input image

```

```

159.     (offsetX, offsetY) = (x * 4.0, y * 4.0)
160.
161.     # Extract the rotation angle for the prediction and then compute the sin and cosine
162.     angle = anglesData[x]
163.     cos = np.cos(angle)
164.     sin = np.sin(angle)
165.
166.     # Use the geometry volume to derive the width and height of the bounding box
167.     h = xData0[x] + xData2[x]
168.     w = xData1[x] + xData3[x]
169.
170.     # Compute both the starting and ending (x, y)-coordinates for the text prediction bounding box
171.     endX = int(offsetX + (cos * xData1[x]) + (sin * xData2[x]))
172.     endY = int(offsetY - (sin * xData1[x]) + (cos * xData2[x]))
173.     startX = int(endX - w)
174.     startY = int(endY - h)
175.
176.     # Add the bounding box coordinates and probability score to respective lists
177.     rects.append((startX, startY, endX, endY))
178.     confidences.append(scoresData[x])
179.
180. # Apply non-maxima suppression to suppress weak, overlapping bounding boxes
181. boxes = non_max_suppression(np.array(rects), probs=confidences)
182.
183. xStart = []
184. xEnd = []
185. yStart = []
186. yEnd = []
187.
188. # Loop over the bounding boxes
189. for (startX, startY, endX, endY) in boxes:
190.     # Scale the bounding box coordinates based on the respective ratios
191.     startX = int(startX * rW)
192.     startY = int(startY * rH)
193.     endX = int(endX * rW)
194.     endY = int(endY * rH)
195.
196.     # Draw the bounding box on the image
197.     xStart.append(startX) # vector of x coords
198.     yStart.append(startY)
199.     xEnd.append(endX)
200.     yEnd.append(endY)
201.
202. (height, width) = finalA.shape[:2]
203. orientation = 0
204.
205. # Set array position of the Drawing's to be -1 (i.e. not found)
206. A_position = -1
207. B_position = -1
208. C_position = -1
209.
210. # If the coordinates match where a drawing should be, save that position to the corresponding drawing's
211. X_position variable
212. for i in range(0, len(xStart)):
213.     if xStart[i] < 400:
214.         if yEnd[i] < total_h/2:
215.             if A_position == -1:
216.                 A_position = i
217.             else:
218.                 if xEnd[i] - xStart[i] < 100:
219.                     A_position = i
220.                 elif yEnd[i] < 1300 and xEnd[i] < 500:
221.                     C_position = i
222.                 elif yStart[i] < total_h/2:
223.                     if B_position == -1:
224.                         B_position = i
225.                     else:
226.                         if xEnd[i] - xStart[i] < 100:
227.                             B_position = i
228.
229. # In case a drawing is not found, use the average values instead
230. width = np.mean(array_width)
231. xStart_A = np.mean(array_xStartA)
232. xStart_B = np.mean(array_xStartB)
233. xStart_C = np.mean(array_xStartC)
234. yEnd_A = np.mean(array_yEndA)
235. yEnd_B = np.mean(array_yEndB)
236. yEnd_C = np.mean(array_yEndC)
237.
238.
239. # Check if the drawing was found, and save the coordinates respectively

```


8B: extractRectangle.py

8C: method2.py

```

80.     else:
81.         try:
82.             output = str(time[0]) + "Y"
83.             return output
84.         except:
85.             print("ERROR - y but no time - " + str(image_name))
86.
87.     if image_name.find("before") != -1:
88.         return "before"
89.     return -1
90.
91. def is_treated_hand(image_name, patient_number):
92.     image_name = str.lower(image_name)
93.
94.     if patient_number == 6 or patient_number == 17 or patient_number == 22 or patient_number == 32 or
patient_number == 74 or patient_number == 85 or patient_number == 87 or patient_number == 109 or patient_number ==
111 or patient_number == 115 or patient_number == 116 or patient_number == 8 or patient_number == 16 or
patient_number == 31 or patient_number == 39 or patient_number == 40 or patient_number == 47 or patient_number ==
56 or patient_number == 57 or patient_number == 124:
95.         left_position = image_name.find("lt")
96.         if left_position == -1:
97.             left_position = image_name.find("left")
98.             if left_position == -1:
99.                 return False
100.            else:
101.                return True
102.            else:
103.                return True
104.        else:
105.            right_position = image_name.find("rt")
106.            if right_position == -1:
107.                right_position = image_name.find("right")
108.                if right_position == -1:
109.                    return False
110.                else:
111.                    return True
112.                else:
113.                    return True
114.
115. def is_dominant_hand(image_name, patient_number):
116.     image_name = str.lower(image_name)
117.
118.     if patient_number == 17 or patient_number == 22 or patient_number == 32 or patient_number == 56 or
patient_number == 74 or patient_number == 87 or patient_number == 102 or patient_number == 109 or patient_number ==
== 111 or patient_number == 115 or patient_number == 116 or patient_number == 117:
119.         left_position = image_name.find("lt")
120.         if left_position == -1:
121.             left_position = image_name.find("left")
122.             if left_position == -1:
123.                 return False
124.             else:
125.                 return True
126.         else:
127.             return True
128.     else:
129.         right_position = image_name.find("rt")
130.         if right_position == -1:
131.             right_position = image_name.find("right")
132.             if right_position == -1:
133.                 return False
134.             else:
135.                 return True
136.         else:
137.             return True
138.
139. def is_PD_hand(patient_number):
140.     if patient_number == 3 or patient_number == 4 or patient_number == 7 or patient_number == 8 or patient_number ==
== 16 or patient_number == 18 or patient_number == 28 or patient_number == 31 or patient_number == 33 or
patient_number == 37 or patient_number == 38 or patient_number == 39 or patient_number == 40 or patient_number ==
43 or patient_number == 44 or patient_number == 47 or patient_number == 48 or patient_number == 55 or
patient_number == 56 or patient_number == 57 or patient_number == 59 or patient_number == 63 or patient_number ==
68 or patient_number == 70 or patient_number == 71 or patient_number == 76 or patient_number == 77 or
patient_number == 92 or patient_number == 100 or patient_number == 102 or patient_number == 112 or patient_number ==
113 or patient_number == 114 or patient_number == 120 or patient_number == 124:
141.         return True
142.     else:
143.         return False
144.
145. for outer_foldername in glob.glob('Data\Cropped\*'):
146.     for foldername in glob.glob(outer_foldername + '\*'):
147.         patient_number = os.path.basename(foldername)
148.         patient_number = patient_number.replace("#", "")
```

```

149.
150.     for filename in glob.glob(foldername + "\DrawingC\Rectangles\*.jpg"):
151.         arrayName = os.path.basename(filename)
152.         arrayName = arrayName.replace(".jpg","");
153.
154.         time_period = get_time_period(arrayName)
155.         is_dominant = is_dominant_hand(arrayName, patient_number)
156.         is_treated = is_treated_hand(arrayName, patient_number)
157.         is_PD = is_PD_hand(patient_number)
158.         if time_period != -1:
159.             im = Image.open(filename).convert('L') # convert image to grayscale
160.             res = im.point((lambda p: 256 if p>=200 else 0)) # convert each pixel into either black or white
161.             res.save(filename)
162.
163.             image = cv2.imread(filename)
164.             image_array.append(image)
165.             image_names.append(arrayName)
166.             image_patient_number.append(patient_number)
167.             image_time_frame.append(time_period)
168.             image_dominant_hand.append(is_dominant)
169.             image_treated_hand.append(is_treated)
170.             image_PD_hand.append(is_PD)
171.             file_count = file_count + 1
172.
173.results_array = [[-1 for x in range(len(Results))] for y in range(file_count)]
174.
175.temp_counter_patient_5 = -1
176.# Loop through the image array to perform image processing-----#
177.for image_counter, image in enumerate(image_array):
178.    temp_counter_patient_5 += 1
179.    temp_title = image_names[image_counter].replace("(5)_C_RECT", "")
180.
181.    results_array[image_counter][Results.FILENAME.value] = image_names[image_counter]
182.    results_array[image_counter][Results.PATIENT_NUMBER.value] = image_patient_number[image_counter]
183.    results_array[image_counter][Results.TIME_PERIOD.value] = image_time_frame[image_counter]
184.    results_array[image_counter][Results.DOMINANT_HAND.value] = image_dominant_hand[image_counter]
185.    results_array[image_counter][Results.TREATED_HAND.value] = image_treated_hand[image_counter]
186.    results_array[image_counter][Results.PD_HAND.value] = image_PD_hand[image_counter]
187.
188.    coordinates = np.argwhere(image < 0.9)
189.    # fig, (axs_original, axs_fft, axs_ifft, axs_area) = plt.subplots(4)
190.
191.    try:
192.        # NOTE: The horizontal axis is denoted by y, and the vertical axis is denoted by x (ACCIDENTAL MISTAKE)
193.        x_tuple, y_tuple, z_tuple = zip(*coordinates)
194.
195.        x = np.array(x_tuple).tolist()
196.        y = np.array(y_tuple).tolist()
197.        z = np.array(z_tuple).tolist()
198.
199.        # Sort the array of pixels into an ordered array according to the horizontal x-axis
200.        for i in range(0, len(y)-1):
201.            for j in range(0, len(y)-i-1):
202.                if y[j] > y[j+1]:
203.                    temp_x = x[j]
204.                    x[j] = x[j+1]
205.                    x[j+1] = temp_x
206.
207.                    temp_y = y[j]
208.                    y[j] = y[j+1]
209.                    y[j+1] = temp_y
210.
211.        # GRAPH 1: ORIGINAL
212.        # axs_original.set_title("ORIGINAL GRAPH: " + str(temp_title))
213.        # axs_original.plot(y, x)
214.
215.        average_x = np.mean(x)
216.        new_array_x = []
217.        new_array_y = []
218.
219.        # Take the absolute value and shift the points down to be centered around the horizontal axis
220.        for i in range(0, len(x)):
221.            new_array_y.append(y[i])
222.            new_array_x.append(x[i]-average_x)
223.
224.        data_step = 0.1
225.        n = len(new_array_y)
226.        yf = rfft(new_array_x)
227.        xf = rfftfreq(n, data_step)
228.
229.        yf_abs = np.abs(yf)
230.        yf_max = np.max(yf_abs)

```

```

231.
232. # # GRAPH 2: FFT
233. # axs_fft.set_title("FFT: " + str(temp_title))
234. # axs_fft.plot(xf, yf_abs)
235. # axs_fft.set_xlim(0, 0.5)
236.
237. multiplier = 5
238. MIN_multiplier = 5
239. indices = yf_abs > (multiplier/100*yf_max)
240. yf_clean = indices*yf
241. new_f_clean = irfft(yf_clean)
242. x_peaks = signal.find_peaks(np.array(new_f_clean))
243. MIN_x_peaks = signal.find_peaks(np.array(-new_f_clean))
244.
245. while len(x_peaks[0]) > 50 or len(MIN_x_peaks[0]) > 50:
246.     indices = yf_abs > (multiplier/100*yf_max)
247.     yf_clean = indices*yf
248.     new_f_clean = irfft(yf_clean)
249.     x_peaks = signal.find_peaks(np.array(new_f_clean))
250.     MIN_x_peaks = signal.find_peaks(np.array(-new_f_clean))
251.
252.     multiplier = multiplier+5
253.
254. # # GRAPH 3: IFFT
255. # axs_ifft.set_title("IFFT: " + str(temp_title))
256. # axs_ifft.plot(new_array_y, new_array_x)
257.
258. # if len(new_array_y) > len(new_f_clean):
259. #     new_new_array_y = new_array_y
260. #     new_new_array_y.pop(0)
261. #     axs_ifft.plot(new_new_array_y, new_f_clean)
262. # else:
263. #     axs_ifft.plot(new_array_y, new_f_clean)
264.
265. # axs_ifft.set_xlim(0, 600)
266. # axs_ifft.set_ylim(-20, 20)
267.
268. # Quantile values of the data
269. min, q1, q2, q3, q90, max = np.quantile(new_array_x, [0, 0.25, 0.5, 0.75, 0.9, 1])
270. iqr = q3-q1
271. std = np.std(new_array_x)
272.
273. x_peaks = signal.find_peaks(np.array(new_f_clean))
274. num_peaks = len(x_peaks[0])
275. y_peaks_points = []
276. x_peaks_points = []
277. sum_peaks = 0
278.
279. for p in x_peaks[0]:
280.     x_peaks_points.append(new_f_clean[p])
281.     y_peaks_points.append(new_array_y[p])
282.
283. MIN_x_peaks = signal.find_peaks(np.array(-new_f_clean))
284. MIN_num_peaks = len(MIN_x_peaks[0])
285. MIN_y_peaks_points = []
286. MIN_x_peaks_points = []
287.
288. for p in MIN_x_peaks[0]:
289.     MIN_x_peaks_points.append(new_f_clean[p])
290.     MIN_y_peaks_points.append(new_array_y[p])
291.
292. peakmin_distance_array = []
293. for i in range(len(x_peaks_points)):
294.     if y_peaks_points[i] < MIN_y_peaks_points[i]:
295.         if i == 0:
296.             peakmin_distance_array.append(abs(x_peaks_points[i] - MIN_x_peaks_points[i]))
297.         else:
298.             peakmin_distance_array.append(abs(x_peaks_points[i] - MIN_x_peaks_points[i-1]))
299.             peakmin_distance_array.append(abs(x_peaks_points[i] - MIN_x_peaks_points[i]))
300.         else:
301.             peakmin_distance_array.append(abs(x_peaks_points[i] - MIN_x_peaks_points[i]))
302.             if i < len(x_peaks_points)-1:
303.                 peakmin_distance_array.append(abs(x_peaks_points[i] - MIN_x_peaks_points[i+1]))
304.
305. average_peakmin_distance = np.mean(peakmin_distance_array)
306.
307. # METHOD: TRAPZ FORMULA (NUMPY)
308. new_array_x_ABS = [abs(x) for x in new_f_clean]
309. total_area_trapz_x = trapz(new_array_x_ABS) # Area under the curve, using numpy's trapz formula
310.
311.
312.

```


8D: analyseResults.py

```

79.         patient_area_trapz[j] = patient_area_trapz[j+1]
80.         patient_area_trapz[j+1] = temp
81.
82.         temp = patient_avg_area_trapz[j]
83.         patient_avg_area_trapz[j] = patient_avg_area_trapz[j+1]
84.         patient_avg_area_trapz[j+1] = temp
85.
86.         temp = patient_avg_std_dev_array[j]
87.         patient_avg_std_dev_array[j] = patient_avg_std_dev_array[j+1]
88.         patient_avg_std_dev_array[j+1] = temp
89.
90.         temp = patient_num_peaks[j]
91.         patient_num_peaks[j] = patient_num_peaks[j+1]
92.         patient_num_peaks[j+1] = temp
93.
94.         temp = patient_avg_peak_dist[j]
95.         patient_avg_peak_dist[j] = patient_avg_peak_dist[j+1]
96.         patient_avg_peak_dist[j+1] = temp
97.
98.         temp = patient_determinator_1[j]
99.         patient_determinator_1[j] = patient_determinator_1[j+1]
100.        patient_determinator_1[j+1] = temp
101.
102.        temp = patient_determinator_2[j]
103.        patient_determinator_2[j] = patient_determinator_2[j+1]
104.        patient_determinator_2[j+1] = temp
105.
106.D_patient_number_array_avg_std_dev = []
107.D_patient_number_array_avg_std_dev.append(1)
108.D_patient_counter_avg_std_dev = 0
109.
110.D_time_before_avg_std_dev = []
111.D_time_1W_avg_std_dev = []
112.D_time_1M_avg_std_dev = []
113.D_time_3M_avg_std_dev = []
114.D_time_6M_avg_std_dev = []
115.D_time_1Y_avg_std_dev = []
116.D_time_2Y_avg_std_dev = []
117.D_time_3Y_avg_std_dev = []
118.D_time_4Y_avg_std_dev = []
119.
120.D_time_before_avg_std_dev.append(0)
121.D_time_1W_avg_std_dev.append(0)
122.D_time_1M_avg_std_dev.append(0)
123.D_time_3M_avg_std_dev.append(0)
124.D_time_6M_avg_std_dev.append(0)
125.D_time_1Y_avg_std_dev.append(0)
126.D_time_2Y_avg_std_dev.append(0)
127.D_time_3Y_avg_std_dev.append(0)
128.D_time_4Y_avg_std_dev.append(0)
129.
130.ND_patient_number_array_avg_std_dev = []
131.ND_patient_number_array_avg_std_dev.append(1)
132.ND_patient_counter_avg_std_dev = 0
133.
134.ND_time_before_avg_std_dev = []
135.ND_time_1W_avg_std_dev = []
136.ND_time_1M_avg_std_dev = []
137.ND_time_3M_avg_std_dev = []
138.ND_time_6M_avg_std_dev = []
139.ND_time_1Y_avg_std_dev = []
140.ND_time_2Y_avg_std_dev = []
141.ND_time_3Y_avg_std_dev = []
142.ND_time_4Y_avg_std_dev = []
143.
144.ND_time_before_avg_std_dev.append(0)
145.ND_time_1W_avg_std_dev.append(0)
146.ND_time_1M_avg_std_dev.append(0)
147.ND_time_3M_avg_std_dev.append(0)
148.ND_time_6M_avg_std_dev.append(0)
149.ND_time_1Y_avg_std_dev.append(0)
150.ND_time_2Y_avg_std_dev.append(0)
151.ND_time_3Y_avg_std_dev.append(0)
152.ND_time_4Y_avg_std_dev.append(0)
153.
154.max_std_dev = 0.0
155.min_std_dev = 100.0
156.
157.
158.
159.
160.
```

```

161. for i in range(0, n):
162.     if float(patient_avg_std_dev_array[i]) > float(max_std_dev):
163.         max_std_dev = patient_avg_std_dev_array[i]
164.
165.     if float(patient_avg_std_dev_array[i])<float(min_std_dev) and float(patient_avg_std_dev_array[i])>=float(0.0):
166.         min_std_dev = patient_avg_std_dev_array[i]
167.
168.     if patient_treated_hand[i] == 'True':
169.         if int(patient_number_array[i]) != int(D_patient_number_array_avg_std_dev[D_patient_counter_avg_std_dev]):
170.             D_patient_number_array_avg_std_dev.append(patient_number_array[i])
171.             D_patient_counter_avg_std_dev = D_patient_counter_avg_std_dev + 1
172.             D_time_before_avg_std_dev.append(0)
173.             D_time_1W_avg_std_dev.append(0)
174.             D_time_1M_avg_std_dev.append(0)
175.             D_time_3M_avg_std_dev.append(0)
176.             D_time_6M_avg_std_dev.append(0)
177.             D_time_1Y_avg_std_dev.append(0)
178.             D_time_2Y_avg_std_dev.append(0)
179.             D_time_3Y_avg_std_dev.append(0)
180.             D_time_4Y_avg_std_dev.append(0)
181.
182.         if patient_time_array[i] == "before":
183.             D_time_before_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
184.         elif patient_time_array[i] == "1W":
185.             D_time_1W_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
186.         elif patient_time_array[i] == "1M":
187.             D_time_1M_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
188.         elif patient_time_array[i] == "3M":
189.             D_time_3M_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
190.         elif patient_time_array[i] == "6M":
191.             D_time_6M_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
192.         elif patient_time_array[i] == "1Y":
193.             D_time_1Y_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
194.         elif patient_time_array[i] == "2Y":
195.             D_time_2Y_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
196.         elif patient_time_array[i] == "3Y":
197.             D_time_3Y_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
198.         elif patient_time_array[i] == "4Y":
199.             D_time_4Y_avg_std_dev[D_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
200.
201.     else:
202.         if int(patient_number_array[i])!=int(ND_patient_number_array_avg_std_dev[ND_patient_counter_avg_std_dev]):
203.             ND_patient_number_array_avg_std_dev.append(patient_number_array[i])
204.             ND_patient_counter_avg_std_dev = ND_patient_counter_avg_std_dev + 1
205.             ND_time_before_avg_std_dev.append(0)
206.             ND_time_1W_avg_std_dev.append(0)
207.             ND_time_1M_avg_std_dev.append(0)
208.             ND_time_3M_avg_std_dev.append(0)
209.             ND_time_6M_avg_std_dev.append(0)
210.             ND_time_1Y_avg_std_dev.append(0)
211.             ND_time_2Y_avg_std_dev.append(0)
212.             ND_time_3Y_avg_std_dev.append(0)
213.             ND_time_4Y_avg_std_dev.append(0)
214.
215.         if patient_time_array[i] == "before":
216.             ND_time_before_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
217.         elif patient_time_array[i] == "1W":
218.             ND_time_1W_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
219.         elif patient_time_array[i] == "1M":
220.             ND_time_1M_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
221.         elif patient_time_array[i] == "3M":
222.             ND_time_3M_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
223.         elif patient_time_array[i] == "6M":
224.             ND_time_6M_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
225.         elif patient_time_array[i] == "1Y":
226.             ND_time_1Y_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
227.         elif patient_time_array[i] == "2Y":
228.             ND_time_2Y_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
229.         elif patient_time_array[i] == "3Y":
230.             ND_time_3Y_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
231.         elif patient_time_array[i] == "4Y":
232.             ND_time_4Y_avg_std_dev[ND_patient_counter_avg_std_dev] = patient_avg_std_dev_array[i]
233.
234. D_data_avg_std_dev = {'Patient': D_patient_number_array_avg_std_dev, 'Before' : D_time_before_avg_std_dev, '1 Week' : D_time_1W_avg_std_dev, '1 Month' : D_time_1M_avg_std_dev, '3 Months': D_time_3M_avg_std_dev, '6 Months': D_time_6M_avg_std_dev, '1 Year': D_time_1Y_avg_std_dev, '2 Years': D_time_2Y_avg_std_dev, '3 Years':D_time_3Y_avg_std_dev, '4 Years':D_time_4Y_avg_std_dev }
235. D_df_avg_std_dev = pd.DataFrame(D_data_avg_std_dev)
236. D_df_avg_std_dev.to_csv('RESULTS/D_AvgStdDev.csv', index=False)
237.
238. ND_data_avg_std_dev = {'Patient': ND_patient_number_array_avg_std_dev, 'Before' : ND_time_before_avg_std_dev, '1 Week' : ND_time_1W_avg_std_dev, '1 Month' : ND_time_1M_avg_std_dev, '3 Months': ND_time_3M_avg_std_dev, '6

```

```

    Months': ND_time_6M_avg_std_dev, '1 Year': ND_time_1Y_avg_std_dev, '2 Years': ND_time_2Y_avg_std_dev, '3
    Years':ND_time_3Y_avg_std_dev, '4 Years':ND_time_4Y_avg_std_dev }
239.ND_df_avg_std_dev = pd.DataFrame(ND_data_avg_std_dev)
240.ND_df_avg_std_dev.to_csv('RESULTS/ND_AvgStdDev.csv', index=False)
241.
242.#-----
243.
244.D_patient_number_array_avg_area_trapz = []
245.D_patient_number_array_avg_area_trapz.append(1)
246.D_patient_counter_avg_area_trapz = 0
247.
248.D_time_before_avg_area_trapz = []
249.D_time_1W_avg_area_trapz = []
250.D_time_1M_avg_area_trapz = []
251.D_time_3M_avg_area_trapz = []
252.D_time_6M_avg_area_trapz = []
253.D_time_1Y_avg_area_trapz = []
254.D_time_2Y_avg_area_trapz = []
255.D_time_3Y_avg_area_trapz = []
256.D_time_4Y_avg_area_trapz = []
257.
258.D_time_before_avg_area_trapz.append(0)
259.D_time_1W_avg_area_trapz.append(0)
260.D_time_1M_avg_area_trapz.append(0)
261.D_time_3M_avg_area_trapz.append(0)
262.D_time_6M_avg_area_trapz.append(0)
263.D_time_1Y_avg_area_trapz.append(0)
264.D_time_2Y_avg_area_trapz.append(0)
265.D_time_3Y_avg_area_trapz.append(0)
266.D_time_4Y_avg_area_trapz.append(0)
267.
268.ND_patient_number_array_avg_area_trapz = []
269.ND_patient_number_array_avg_area_trapz.append(1)
270.ND_patient_counter_avg_area_trapz = 0
271.
272.ND_time_before_avg_area_trapz = []
273.ND_time_1W_avg_area_trapz = []
274.ND_time_1M_avg_area_trapz = []
275.ND_time_3M_avg_area_trapz = []
276.ND_time_6M_avg_area_trapz = []
277.ND_time_1Y_avg_area_trapz = []
278.ND_time_2Y_avg_area_trapz = []
279.ND_time_3Y_avg_area_trapz = []
280.ND_time_4Y_avg_area_trapz = []
281.
282.ND_time_before_avg_area_trapz.append(0)
283.ND_time_1W_avg_area_trapz.append(0)
284.ND_time_1M_avg_area_trapz.append(0)
285.ND_time_3M_avg_area_trapz.append(0)
286.ND_time_6M_avg_area_trapz.append(0)
287.ND_time_1Y_avg_area_trapz.append(0)
288.ND_time_2Y_avg_area_trapz.append(0)
289.ND_time_3Y_avg_area_trapz.append(0)
290.ND_time_4Y_avg_area_trapz.append(0)
291.
292.max_area = 0.0
293.min_area = 100.0
294.
295.for i in range(0, n):
296.    if float(patient_avg_area_trapz[i]) > float(max_area):
297.        max_area = patient_avg_area_trapz[i]
298.
299.    if float(patient_avg_area_trapz[i]) < float(min_area) and float(patient_avg_area_trapz[i]) >= float(0.0):
300.        min_area = patient_avg_area_trapz[i]
301.
302.    if patient_treated_hand[i] == 'True':
303.        if int(patient_number_array[i]) !=
            int(D_patient_number_array_avg_area_trapz[D_patient_counter_avg_area_trapz]):
304.            D_patient_number_array_avg_area_trapz.append(patient_number_array[i])
305.            D_patient_counter_avg_area_trapz = D_patient_counter_avg_area_trapz + 1
306.            D_time_before_avg_area_trapz.append(0)
307.            D_time_1W_avg_area_trapz.append(0)
308.            D_time_1M_avg_area_trapz.append(0)
309.            D_time_3M_avg_area_trapz.append(0)
310.            D_time_6M_avg_area_trapz.append(0)
311.            D_time_1Y_avg_area_trapz.append(0)
312.            D_time_2Y_avg_area_trapz.append(0)
313.            D_time_3Y_avg_area_trapz.append(0)
314.            D_time_4Y_avg_area_trapz.append(0)
315.
316.
317.
```

```

318.     if patient_time_array[i] == "before":
319.         D_time_before_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
320.
321.     elif patient_time_array[i] == "1W":
322.         D_time_1W_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
323.
324.     elif patient_time_array[i] == "1M":
325.         D_time_1M_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
326.
327.     elif patient_time_array[i] == "3M":
328.         D_time_3M_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
329.
330.     elif patient_time_array[i] == "6M":
331.         D_time_6M_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
332.
333.     elif patient_time_array[i] == "1Y":
334.         D_time_1Y_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
335.
336.     elif patient_time_array[i] == "2Y":
337.         D_time_2Y_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
338.
339.     elif patient_time_array[i] == "3Y":
340.         D_time_3Y_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
341.
342.     elif patient_time_array[i] == "4Y":
343.         D_time_4Y_avg_area_trapz[D_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
344.
345. else:
346.     if int(patient_number_array[i]) != 1:
347.         ND_patient_number_array_avg_area_trapz[ND_patient_counter_avg_area_trapz].append(patient_number_array[i])
348.         ND_patient_counter_avg_area_trapz = ND_patient_counter_avg_area_trapz + 1
349.         ND_time_before_avg_area_trapz.append(0)
350.
351.         ND_time_1W_avg_area_trapz.append(0)
352.         ND_time_1M_avg_area_trapz.append(0)
353.         ND_time_3M_avg_area_trapz.append(0)
354.         ND_time_6M_avg_area_trapz.append(0)
355.         ND_time_1Y_avg_area_trapz.append(0)
356.         ND_time_2Y_avg_area_trapz.append(0)
357.         ND_time_3Y_avg_area_trapz.append(0)
358.         ND_time_4Y_avg_area_trapz.append(0)
359.
360.     if patient_time_array[i] == "before":
361.         ND_time_before_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
362.
363.     elif patient_time_array[i] == "1W":
364.         ND_time_1W_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
365.
366.     elif patient_time_array[i] == "1M":
367.         ND_time_1M_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
368.
369.     elif patient_time_array[i] == "3M":
370.         ND_time_3M_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
371.     elif patient_time_array[i] == "6M":
372.         ND_time_6M_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
373.
374.     elif patient_time_array[i] == "1Y":
375.         ND_time_1Y_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
376.
377.     elif patient_time_array[i] == "2Y":
378.         ND_time_2Y_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
379.
380.     elif patient_time_array[i] == "3Y":
381.         ND_time_3Y_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
382.
383.     elif patient_time_array[i] == "4Y":
384.         ND_time_4Y_avg_area_trapz[ND_patient_counter_avg_area_trapz] = patient_avg_area_trapz[i]
385.
386. D_data_avg_area_trapz = {'Patient': D_patient_number_array_avg_area_trapz, 'Before' : D_time_before_avg_area_trapz, '1 Week' : D_time_1W_avg_area_trapz, '1 Month' : D_time_1M_avg_area_trapz, '3 Months' : D_time_3M_avg_area_trapz, '6 Months' : D_time_6M_avg_area_trapz, '1 Year': D_time_1Y_avg_area_trapz, '2 Years': D_time_2Y_avg_area_trapz, '3 Years':D_time_3Y_avg_area_trapz, '4 Years':D_time_4Y_avg_area_trapz }
387. D_df_avg_area_trapz = pd.DataFrame(D_data_avg_area_trapz)
388. D_df_avg_area_trapz.to_csv('RESULTS/D_AreaTrapz.csv', index=False)
389.
390. ND_data_avg_area_trapz = {'Patient': ND_patient_number_array_avg_area_trapz, 'Before' : ND_time_before_avg_area_trapz, '1 Week' : ND_time_1W_avg_area_trapz, '1 Month' : ND_time_1M_avg_area_trapz, '3 Months' : ND_time_3M_avg_area_trapz, '6 Months' : ND_time_6M_avg_area_trapz, '1 Year': ND_time_1Y_avg_area_trapz, '2 Years': ND_time_2Y_avg_area_trapz, '3 Years':ND_time_3Y_avg_area_trapz, '4 Years':ND_time_4Y_avg_area_trapz }
391. ND_df_avg_area_trapz = pd.DataFrame(ND_data_avg_area_trapz)
392. ND_df_avg_area_trapz.to_csv('RESULTS/ND_AreaTrapz.csv', index=False)
393.
394. #-----#
395.
396. D_patient_number_array_det_2 = []
397. D_patient_number_array_det_2.append(1)
398. D_patient_counter_det_2 = 0
399.
400. D_time_before_det_2 = []
401. D_time_1W_det_2 = []
402. D_time_1M_det_2 = []
403. D_time_3M_det_2 = []
404. D_time_6M_det_2 = []
405. D_time_1Y_det_2 = []
406. D_time_2Y_det_2 = []
407. D_time_3Y_det_2 = []
408. D_time_4Y_det_2 = []

```

```

393.
394.D_time_before_det_2.append(0)
395.D_time_1W_det_2.append(0)
396.D_time_1M_det_2.append(0)
397.D_time_3M_det_2.append(0)
398.D_time_6M_det_2.append(0)
399.D_time_1Y_det_2.append(0)
400.D_time_2Y_det_2.append(0)
401.D_time_3Y_det_2.append(0)
402.D_time_4Y_det_2.append(0)
403.
404.ND_patient_number_array_det_2 = []
405.ND_patient_number_array_det_2.append(1)
406.ND_patient_counter_det_2 = 0
407.
408.ND_time_before_det_2 = []
409.ND_time_1W_det_2 = []
410.ND_time_1M_det_2 = []
411.ND_time_3M_det_2 = []
412.ND_time_6M_det_2 = []
413.ND_time_1Y_det_2 = []
414.ND_time_2Y_det_2 = []
415.ND_time_3Y_det_2 = []
416.ND_time_4Y_det_2 = []
417.
418.ND_time_before_det_2.append(0)
419.ND_time_1W_det_2.append(0)
420.ND_time_1M_det_2.append(0)
421.ND_time_3M_det_2.append(0)
422.ND_time_6M_det_2.append(0)
423.ND_time_1Y_det_2.append(0)
424.ND_time_2Y_det_2.append(0)
425.ND_time_3Y_det_2.append(0)
426.ND_time_4Y_det_2.append(0)
427.
428.D_patient_hand_array_det_2 = []
429.ND_patient_hand_array_det_2 = []
430.
431.max_det_2 = 0.0
432.min_det_2 = 100.0
433.
434.for i in range(0, n):
435.    if float(patient_determinator_2[i]) > float(max_det_2):
436.        max_det_2 = patient_determinator_2[i]
437.
438.    if float(patient_determinator_2[i]) < float(min_det_2) and float(patient_determinator_2[i]) >= float(0.0):
439.        min_det_2 = patient_determinator_2[i]
440.
441.    if patient_treated_hand[i] == 'True':
442.        if int(patient_number_array[i]) != int(D_patient_number_array_det_2[D_patient_counter_det_2]):
443.            D_patient_hand_array_det_2.append(patient_dominant_hand[i])
444.            D_patient_number_array_det_2.append(patient_number_array[i])
445.            D_patient_counter_det_2 = D_patient_counter_det_2 + 1
446.            D_time_before_det_2.append(0)
447.            D_time_1W_det_2.append(0)
448.            D_time_1M_det_2.append(0)
449.            D_time_3M_det_2.append(0)
450.            D_time_6M_det_2.append(0)
451.            D_time_1Y_det_2.append(0)
452.            D_time_2Y_det_2.append(0)
453.            D_time_3Y_det_2.append(0)
454.            D_time_4Y_det_2.append(0)
455.
456.        if patient_time_array[i] == "before":
457.            D_time_before_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
458.        elif patient_time_array[i] == "1W":
459.            D_time_1W_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
460.        elif patient_time_array[i] == "1M":
461.            D_time_1M_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
462.        elif patient_time_array[i] == "3M":
463.            D_time_3M_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
464.        elif patient_time_array[i] == "6M":
465.            D_time_6M_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
466.        elif patient_time_array[i] == "1Y":
467.            D_time_1Y_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
468.        elif patient_time_array[i] == "2Y":
469.            D_time_2Y_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
470.        elif patient_time_array[i] == "3Y":
471.            D_time_3Y_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
472.        elif patient_time_array[i] == "4Y":
473.            D_time_4Y_det_2[D_patient_counter_det_2] = patient_determinator_2[i]
474.

```



```

547.D_both_2Y, D_improved_2Y, D_avg_amount_2Y, D_avg_diff_2Y, D_avg_diff_improved_2Y =
    checkImproved(D_time_before_det_2, D_time_2Y_det_2)
548.D_both_3Y, D_improved_3Y, D_avg_amount_3Y, D_avg_diff_3Y, D_avg_diff_improved_3Y =
    checkImproved(D_time_before_det_2, D_time_3Y_det_2)
549.D_both_4Y, D_improved_4Y, D_avg_amount_4Y, D_avg_diff_4Y, D_avg_diff_improved_4Y =
    checkImproved(D_time_before_det_2, D_time_4Y_det_2)
550.
551.D_improved_percentage = [D_improved_1W/D_both_1W*100, D_improved_1M/D_both_1M*100, D_improved_3M/D_both_3M*100,
    D_improved_6M/D_both_6M*100, D_improved_1Y/D_both_1Y*100, D_improved_2Y/D_both_2Y*100,
    D_improved_3Y/D_both_3Y*100, D_improved_4Y/D_both_4Y*100]
552.D_avg_improved_amount = [D_avg_amount_1W, D_avg_amount_1M, D_avg_amount_3M, D_avg_amount_6M, D_avg_amount_1Y,
    D_avg_amount_2Y, D_avg_amount_3Y, D_avg_amount_4Y]
553.D_num_improved = [D_improved_1W, D_improved_1M, D_improved_3M, D_improved_6M, D_improved_1Y, D_improved_2Y,
    D_improved_3Y, D_improved_4Y]
554.D_total_num = [D_both_1W, D_both_1M, D_both_3M, D_both_6M, D_both_1Y, D_both_2Y, D_both_3Y, D_both_4Y]
555.D_avg_diff = [D_avg_diff_1W, D_avg_diff_1M, D_avg_diff_3M, D_avg_diff_6M, D_avg_diff_1Y, D_avg_diff_2Y,
    D_avg_diff_3Y, D_avg_diff_4Y]
556.D_avg_diff_improved = [D_avg_diff_improved_1W, D_avg_diff_improved_1M, D_avg_diff_improved_3M,
    D_avg_diff_improved_6M, D_avg_diff_improved_1Y, D_avg_diff_improved_2Y, D_avg_diff_improved_3Y,
    D_avg_diff_improved_4Y]
557.
558.ND_both_1W, ND_improved_1W, ND_avg_amount_1W, ND_avg_diff_1W, ND_avg_diff_improved_1W =
    checkImproved(ND_time_before_det_2, ND_time_1W_det_2)
559.ND_both_1M, ND_improved_1M, ND_avg_amount_1M, ND_avg_diff_1M, ND_avg_diff_improved_1M =
    checkImproved(ND_time_before_det_2, ND_time_1M_det_2)
560.ND_both_3M, ND_improved_3M, ND_avg_amount_3M, ND_avg_diff_3M, ND_avg_diff_improved_3M =
    checkImproved(ND_time_before_det_2, ND_time_3M_det_2)
561.ND_both_6M, ND_improved_6M, ND_avg_amount_6M, ND_avg_diff_6M, ND_avg_diff_improved_6M =
    checkImproved(ND_time_before_det_2, ND_time_6M_det_2)
562.ND_both_1Y, ND_improved_1Y, ND_avg_amount_1Y, ND_avg_diff_1Y, ND_avg_diff_improved_1Y =
    checkImproved(ND_time_before_det_2, ND_time_1Y_det_2)
563.ND_both_2Y, ND_improved_2Y, ND_avg_amount_2Y, ND_avg_diff_2Y, ND_avg_diff_improved_2Y =
    checkImproved(ND_time_before_det_2, ND_time_2Y_det_2)
564.ND_both_3Y, ND_improved_3Y, ND_avg_amount_3Y, ND_avg_diff_3Y, ND_avg_diff_improved_3Y =
    checkImproved(ND_time_before_det_2, ND_time_3Y_det_2)
565.ND_both_4Y, ND_improved_4Y, ND_avg_amount_4Y, ND_avg_diff_4Y, ND_avg_diff_improved_4Y =
    checkImproved(ND_time_before_det_2, ND_time_4Y_det_2)
566.
567.ND_improved_percentage = [ND_improved_1W/ND_both_1W*100, ND_improved_1M/ND_both_1M*100,
    ND_improved_3M/ND_both_3M*100, ND_improved_6M/ND_both_6M*100, ND_improved_1Y/ND_both_1Y*100,
    ND_improved_2Y/ND_both_2Y*100, ND_improved_3Y/ND_both_3Y*100, ND_improved_4Y/ND_both_4Y*100]
568.ND_avg_improved_amount = [ND_avg_amount_1W, ND_avg_amount_1M, ND_avg_amount_3M, ND_avg_amount_6M,
    ND_avg_amount_1Y, ND_avg_amount_2Y, ND_avg_amount_3Y, ND_avg_amount_4Y]
569.ND_num_improved = [ND_improved_1W, ND_improved_1M, ND_improved_3M, ND_improved_6M, ND_improved_1Y, ND_improved_2Y,
    ND_improved_3Y, ND_improved_4Y]
570.ND_total_num = [ND_both_1W, ND_both_1M, ND_both_3M, ND_both_6M, ND_both_1Y, ND_both_2Y, ND_both_3Y, ND_both_4Y]
571.ND_avg_diff = [ND_avg_diff_1W, ND_avg_diff_1M, ND_avg_diff_3M, ND_avg_diff_6M, ND_avg_diff_1Y, ND_avg_diff_2Y,
    ND_avg_diff_3Y, ND_avg_diff_4Y]
572.ND_avg_diff_improved = [ND_avg_diff_improved_1W, ND_avg_diff_improved_1M, ND_avg_diff_improved_3M,
    ND_avg_diff_improved_6M, ND_avg_diff_improved_1Y, ND_avg_diff_improved_2Y, ND_avg_diff_improved_3Y,
    ND_avg_diff_improved_4Y]
573.
574.avg_total_num = [(D_both_1W+ND_both_1W)/2, (D_both_1M+ND_both_1M)/2, (D_both_3M+ND_both_3M)/2,
    (D_both_6M+ND_both_6M)/2, (D_both_1Y+ND_both_1Y)/2, (D_both_2Y+ND_both_2Y)/2, (D_both_3Y+ND_both_3Y)/2,
    (D_both_4Y+ND_both_4Y)/2]
575.
576.D_both_1W_AVG_AREA, D_improved_1W_AVG_AREA, D_avg_amount_1W_AVG_AREA, D_avg_diff_1W_AVG_AREA,
    D_avg_diff_improved_1W_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_1W_avg_area_trapz)
577.D_both_1M_AVG_AREA, D_improved_1M_AVG_AREA, D_avg_amount_1M_AVG_AREA, D_avg_diff_1M_AVG_AREA,
    D_avg_diff_improved_1M_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_1M_avg_area_trapz)
578.D_both_3M_AVG_AREA, D_improved_3M_AVG_AREA, D_avg_amount_3M_AVG_AREA, D_avg_diff_3M_AVG_AREA,
    D_avg_diff_improved_3M_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_3M_avg_area_trapz)
579.D_both_6M_AVG_AREA, D_improved_6M_AVG_AREA, D_avg_amount_6M_AVG_AREA, D_avg_diff_6M_AVG_AREA,
    D_avg_diff_improved_6M_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_6M_avg_area_trapz)
580.D_both_1Y_AVG_AREA, D_improved_1Y_AVG_AREA, D_avg_amount_1Y_AVG_AREA, D_avg_diff_1Y_AVG_AREA,
    D_avg_diff_improved_1Y_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_1Y_avg_area_trapz)
581.D_both_2Y_AVG_AREA, D_improved_2Y_AVG_AREA, D_avg_amount_2Y_AVG_AREA, D_avg_diff_2Y_AVG_AREA,
    D_avg_diff_improved_2Y_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_2Y_avg_area_trapz)
582.D_both_3Y_AVG_AREA, D_improved_3Y_AVG_AREA, D_avg_amount_3Y_AVG_AREA, D_avg_diff_3Y_AVG_AREA,
    D_avg_diff_improved_3Y_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_3Y_avg_area_trapz)
583.D_both_4Y_AVG_AREA, D_improved_4Y_AVG_AREA, D_avg_amount_4Y_AVG_AREA, D_avg_diff_4Y_AVG_AREA,
    D_avg_diff_improved_4Y_AVG_AREA = checkImproved(D_time_before_avg_area_trapz, D_time_4Y_avg_area_trapz)
584.
585.ND_both_1W_AVG_AREA, ND_improved_1W_AVG_AREA, ND_avg_amount_1W_AVG_AREA, ND_avg_diff_1W_AVG_AREA,
    ND_avg_diff_improved_1W_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_1W_avg_area_trapz)
586.ND_both_1M_AVG_AREA, ND_improved_1M_AVG_AREA, ND_avg_amount_1M_AVG_AREA, ND_avg_diff_1M_AVG_AREA,
    ND_avg_diff_improved_1M_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_1M_avg_area_trapz)
587.ND_both_3M_AVG_AREA, ND_improved_3M_AVG_AREA, ND_avg_amount_3M_AVG_AREA, ND_avg_diff_3M_AVG_AREA,
    ND_avg_diff_improved_3M_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_3M_avg_area_trapz)
588.ND_both_6M_AVG_AREA, ND_improved_6M_AVG_AREA, ND_avg_amount_6M_AVG_AREA, ND_avg_diff_6M_AVG_AREA,
    ND_avg_diff_improved_6M_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_6M_avg_area_trapz)

```

```

589.ND_both_1Y_AVG_AREA, ND_improved_1Y_AVG_AREA, ND_avg_amount_1Y_AVG_AREA, ND_avg_diff_1Y_AVG_AREA,
ND_avg_diff_improved_1Y_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_1Y_avg_area_trapz)
590.ND_both_2Y_AVG_AREA, ND_improved_2Y_AVG_AREA, ND_avg_amount_2Y_AVG_AREA, ND_avg_diff_2Y_AVG_AREA,
ND_avg_diff_improved_2Y_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_2Y_avg_area_trapz)
591.ND_both_3Y_AVG_AREA, ND_improved_3Y_AVG_AREA, ND_avg_amount_3Y_AVG_AREA, ND_avg_diff_3Y_AVG_AREA,
ND_avg_diff_improved_3Y_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_3Y_avg_area_trapz)
592.ND_both_4Y_AVG_AREA, ND_improved_4Y_AVG_AREA, ND_avg_amount_4Y_AVG_AREA, ND_avg_diff_4Y_AVG_AREA,
ND_avg_diff_improved_4Y_AVG_AREA = checkImproved(ND_time_before_avg_area_trapz, ND_time_4Y_avg_area_trapz)
593.
594.D_improved_percentage_AVG_AREA = [D_improved_1W_AVG_AREA/D_both_1W_AVG_AREA*100,
D_improved_1M_AVG_AREA/D_both_1M_AVG_AREA*100, D_improved_3M_AVG_AREA/D_both_3M_AVG_AREA*100,
D_improved_6M_AVG_AREA/D_both_6M_AVG_AREA*100, D_improved_1Y_AVG_AREA/D_both_1Y_AVG_AREA*100,
D_improved_2Y_AVG_AREA/D_both_2Y_AVG_AREA*100, D_improved_3Y_AVG_AREA/D_both_3Y_AVG_AREA*100,
D_improved_4Y_AVG_AREA/D_both_4Y_AVG_AREA*100]
595.ND_improved_percentage_AVG_AREA = [ND_improved_1W_AVG_AREA/ND_both_1W_AVG_AREA*100,
ND_improved_1M_AVG_AREA/ND_both_1M_AVG_AREA*100, ND_improved_3M_AVG_AREA/ND_both_3M_AVG_AREA*100,
ND_improved_6M_AVG_AREA/ND_both_6M_AVG_AREA*100, ND_improved_1Y_AVG_AREA/ND_both_1Y_AVG_AREA*100,
ND_improved_2Y_AVG_AREA/ND_both_2Y_AVG_AREA*100, ND_improved_3Y_AVG_AREA/ND_both_3Y_AVG_AREA*100,
ND_improved_4Y_AVG_AREA/ND_both_4Y_AVG_AREA*100]
596.
597.## GENERAL GRAPH SETTINGS
598.plt.rcParams["font.family"] = "Times New Roman"
599.plt.rc('axes', axisbelow=True)
600.plt.rcParams.update({'font.size': 16})
601.
602.## GRAPH 1: PERCENTAGE OF PATIENTS WITH TREMOR BEFORE TREATMENT THAT IMPROVED AFTER VARIOUS TREATMENT TIMES
603.x = ['1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y']
604.x_axis = np.arange(len(x))
605.fig, ax1 = plt.subplots(figsize = (8,4))
606.plt.title('Percentage of Patients with Tremor Before Treatment that Improved\nAfter Various Treatment Times - PEAK DISTANCE (METHOD 2B)')
607.plt.grid(linestyle = '-', linewidth=0.5, axis='y')
608.plt.tight_layout()
609.
610.ax1.bar(x_axis - 0.15, D_improved_percentage, 0.3, label = 'Treated Hand', color = 'darkblue')
611.ax1.bar(x_axis + 0.15, ND_improved_percentage, 0.3, label = 'Treated Hand', color = 'cornflowerblue')
612.ax1.set_ylabel('Percentage Improved')
613.ax1.set_xlabel('Time')
614.ax1.legend(['Treated Hand', 'Untreated Hand'], loc="upper right", prop={'size': 14})
615.ax1.set_ylim(0, 100)
616.ax2 = ax1.twinx()
617.ax2.plot(x, avg_total_num, color = 'red')
618.ax2.set_ylabel('Number of Patients')
619.ax2.legend(['Number of Patients'], loc="upper center", prop={'size': 14})
620.ax2.set_ylim(0, 100)
621.print("AVERAGE PEAK DISTANCE DOMINANT: " + str(np.average(D_improved_percentage)))
622.print("AVERAGE PEAK DISTANCE NON-DOMINANT: " + str(np.average(ND_improved_percentage)))
623.plt.savefig('RESULTS\GRAPHS\PercentageOfPatients_Det2.png', bbox_inches='tight', dpi=150)
624.
625.## GRAPH 2: PERCENTAGE OF PATIENTS WITH TREMOR BEFORE TREATMENT THAT IMPROVED AFTER VARIOUS TREATMENT TIMES
626.fig, ax1 = plt.subplots(figsize = (8,4))
627.plt.grid(linestyle = '-', linewidth=0.5, axis='y')
628.plt.title('Difference Between Tremor Severity Before Treatment\nand After Various Treatment Periods')
629.
630.ax1.bar(x_axis - 0.15, D_avg_diff, 0.3, label = 'Treated Hand', color = 'darkblue')
631.ax1.bar(x_axis + 0.15, ND_avg_diff, 0.3, label = 'Treated Hand', color = 'cornflowerblue')
632.ax1.set_ylabel('Difference in Tremor')
633.ax1.legend(['Treated Hand', 'Untreated Hand'], loc="upper right")
634.ax1.set_xlabel('Time')
635.
636.plt.xticks(x_axis, x)
637.plt.tight_layout()
638.plt.show()
639.
640.## GRAPH 3: PERCENT OF PATIENTS WITH TREMOR BEFORE TREATMENT THAT IMPROVED AFTER VARIOUS TREATMENT TIMES -AVG AREA
641.x = ['1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y']
642.x_axis = np.arange(len(x))
643.fig, ax1 = plt.subplots(figsize = (8,4))
644.plt.title('Percentage of Patients with Tremor Before Treatment that Improved\nAfter Various Treatment Times - AVERAGE AREA (METHOD 2A)')
645.plt.grid(linestyle = '-', linewidth=0.5, axis='y')
646.plt.tight_layout()
647.
648.ax1.bar(x_axis - 0.15, D_improved_percentage_AVG_AREA, 0.3, label = 'Treated Hand', color = 'darkblue')
649.ax1.bar(x_axis + 0.15, ND_improved_percentage_AVG_AREA, 0.3, label = 'Treated Hand', color = 'cornflowerblue')
650.ax1.set_ylabel('Percentage Improved')
651.ax1.set_xlabel('Time')
652.ax1.legend(['Treated Hand', 'Untreated Hand'], loc="upper right", prop={'size': 14})
653.ax1.set_ylim(0, 100)
654.ax2 = ax1.twinx()
655.ax2.plot(x, avg_total_num, color = 'red')
656.ax2.set_ylabel('Number of Patients')

```

```

657.ax2.legend(['Number of\npatients'], loc="upper center", prop={'size': 14})
658.ax2.set_ylimit(0, 100)
659=plt.rcParams["figure.figsize"] = (8,4)
660.print("AVERAGE AREA METHOD DOM: " + str(np.average(D_improved_percentage_AVG_AREA)))
661.print("AVERAGE AREA METHOD NON-DOM: " + str(np.average(ND_improved_percentage_AVG_AREA)))
662=plt.savefig('RESULTS\GRAPHS\PercentageOfPatients_AvgArea.png', bbox_inches='tight', dpi=150)
663.
664.# START OF "WHICH HAND?" GRAPH ONE
665.combined_improved_array = []
666.which_hand = []
667.combined_patient_number_array = []
668.dominant_hand = []
669.longest_array = 0
670.
671.if len(D_patient_number_array_det_2) > len(ND_patient_number_array_det_2):
672.    longest_array = len(D_patient_number_array_det_2)
673.else:
674.    longest_array = len(ND_patient_number_array_det_2)
675.
676.D_counter = -1
677.ND_counter = -1
678.
679._before = 1
680._1W = 7
681._1M = 30
682._3M = 91
683._6M = 183
684._1Y = 365
685._2Y = 730
686._3Y = 1095
687._4Y = 1460
688.
689._before_linear = 1
690._1W_linear = 2
691._1M_linear = 3
692._3M_linear = 4
693._6M_linear = 5
694._1Y_linear = 6
695._2Y_linear = 7
696._3Y_linear = 8
697._4Y_linear = 9
698.
699.fig_ALLPATIENTS, ax1_ALLPATIENTS = plt.subplots(figsize = (8,4))
700.
701.for i in range(0, longest_array):
702.    D_counter += 1
703.    ND_counter += 1
704.
705.    if D_counter >= len(D_patient_number_array_det_2) or ND_counter >= len(ND_patient_number_array_det_2):
706.        break
707.    else:
708.        while D_patient_number_array_det_2[D_counter] != ND_patient_number_array_det_2[ND_counter]:
709.            if D_patient_number_array_det_2[D_counter+1] == ND_patient_number_array_det_2[ND_counter]:
710.                D_counter += 1
711.            elif D_patient_number_array_det_2[D_counter] == ND_patient_number_array_det_2[ND_counter + 1]:
712.                ND_counter += 1
713.            else:
714.                D_counter += 1
715.                ND_counter += 1
716.
717.    D_x_array = []
718.    D_x_array_linear = []
719.    D_y_array = []
720.
721.    if D_time_before_det_2[D_counter] != 0 and D_time_before_det_2[D_counter] != -1:
722.        D_y_array.append(D_time_before_det_2[D_counter])
723.        D_x_array.append(_before)
724.        D_x_array_linear.append(_before_linear)
725.    if D_time_1W_det_2[D_counter] != 0 and D_time_1W_det_2[D_counter] != -1:
726.        D_y_array.append(D_time_1W_det_2[D_counter])
727.        D_x_array.append(_1W)
728.        D_x_array_linear.append(_1W_linear)
729.    if D_time_1M_det_2[D_counter] != 0 and D_time_1M_det_2[D_counter] != -1:
730.        D_y_array.append(D_time_1M_det_2[D_counter])
731.        D_x_array.append(_1M)
732.        D_x_array_linear.append(_1M_linear)
733.    if D_time_3M_det_2[D_counter] != 0 and D_time_3M_det_2[D_counter] != -1:
734.        D_y_array.append(D_time_3M_det_2[D_counter])
735.        D_x_array.append(_3M)
736.        D_x_array_linear.append(_3M_linear)
737.    if D_time_6M_det_2[D_counter] != 0 and D_time_6M_det_2[D_counter] != -1:
738.        D_y_array.append(D_time_6M_det_2[D_counter])

```

```

739.     D_x_array.append(_6M)
740.     D_x_array_linear.append(_6M_linear)
741.     if D_time_1Y_det_2[D_counter] != 0 and D_time_1Y_det_2[D_counter] != -1:
742.         D_y_array.append(D_time_1Y_det_2[D_counter])
743.         D_x_array.append(_1Y)
744.         D_x_array_linear.append(_1Y_linear)
745.     if D_time_2Y_det_2[D_counter] != 0 and D_time_2Y_det_2[D_counter] != -1:
746.         D_y_array.append(D_time_2Y_det_2[D_counter])
747.         D_x_array.append(_2Y)
748.         D_x_array_linear.append(_2Y_linear)
749.     if D_time_3Y_det_2[D_counter] != 0 and D_time_3Y_det_2[D_counter] != -1:
750.         D_y_array.append(D_time_3Y_det_2[D_counter])
751.         D_x_array.append(_3Y)
752.         D_x_array_linear.append(_3Y_linear)
753.     if D_time_4Y_det_2[D_counter] != 0 and D_time_4Y_det_2[D_counter] != -1:
754.         D_y_array.append(D_time_4Y_det_2[D_counter])
755.         D_x_array.append(_4Y)
756.         D_x_array_linear.append(_4Y_linear)
757.
758. ND_x_array = []
759. ND_x_array_linear = []
760. ND_y_array = []
761. if ND_time_before_det_2[ND_counter] != 0 and ND_time_before_det_2[ND_counter] != -1:
762.     ND_y_array.append(ND_time_before_det_2[ND_counter])
763.     ND_x_array.append(_before)
764.     ND_x_array_linear.append(_before_linear)
765. if ND_time_1W_det_2[ND_counter] != 0 and ND_time_1W_det_2[ND_counter] != -1:
766.     ND_y_array.append(ND_time_1W_det_2[ND_counter])
767.     ND_x_array.append(_1W)
768.     ND_x_array_linear.append(_1W_linear)
769. if ND_time_1M_det_2[ND_counter] != 0 and ND_time_1M_det_2[ND_counter] != -1:
770.     ND_y_array.append(ND_time_1M_det_2[ND_counter])
771.     ND_x_array.append(_1M)
772.     ND_x_array_linear.append(_1M_linear)
773. if ND_time_3M_det_2[ND_counter] != 0 and ND_time_3M_det_2[ND_counter] != -1:
774.     ND_y_array.append(ND_time_3M_det_2[ND_counter])
775.     ND_x_array.append(_3M)
776.     ND_x_array_linear.append(_3M_linear)
777. if ND_time_6M_det_2[ND_counter] != 0 and ND_time_6M_det_2[ND_counter] != -1:
778.     ND_y_array.append(ND_time_6M_det_2[ND_counter])
779.     ND_x_array.append(_6M)
780.     ND_x_array_linear.append(_6M_linear)
781. if ND_time_1Y_det_2[ND_counter] != 0 and ND_time_1Y_det_2[ND_counter] != -1:
782.     ND_y_array.append(ND_time_1Y_det_2[ND_counter])
783.     ND_x_array.append(_1Y)
784.     ND_x_array_linear.append(_1Y_linear)
785. if ND_time_2Y_det_2[ND_counter] != 0 and ND_time_2Y_det_2[ND_counter] != -1:
786.     ND_y_array.append(ND_time_2Y_det_2[ND_counter])
787.     ND_x_array.append(_2Y)
788.     ND_x_array_linear.append(_2Y_linear)
789. if ND_time_3Y_det_2[ND_counter] != 0 and ND_time_3Y_det_2[ND_counter] != -1:
790.     ND_y_array.append(ND_time_3Y_det_2[ND_counter])
791.     ND_x_array.append(_3Y)
792.     ND_x_array_linear.append(_3Y_linear)
793. if ND_time_4Y_det_2[ND_counter] != 0 and ND_time_4Y_det_2[ND_counter] != -1:
794.     ND_y_array.append(ND_time_4Y_det_2[ND_counter])
795.     ND_x_array.append(_4Y)
796.     ND_x_array_linear.append(_4Y_linear)
797.
798. if len(D_x_array) > 4 and len(D_y_array) > 4 and len(ND_x_array) > 4 and len(ND_y_array) > 4:
799.
800.     combined_patient_number_array.append(D_patient_number_array_det_2[D_counter])
801.
802.     D_gradient, D_intercept = np.polyfit(D_x_array, D_y_array, 1)
803.     ND_gradient, ND_intercept = np.polyfit(ND_x_array, ND_y_array, 1)
804.
805.     plt.title('Results of Patients\' Most Improved Hand')
806.     plt.grid(linestyle = '-', linewidth=0.5, axis='y')
807.     plt.xticks([1,2,3,4,5,6,7,8,9], ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
808.
809.     temp_D_y_array = np.array(D_y_array.copy())
810.     D_y_array_max = max(temp_D_y_array)
811.     temp_D_y_array = temp_D_y_array/D_y_array_max*100
812.     temp_ND_y_array = np.array(ND_y_array.copy())
813.     ND_y_array_max = max(temp_ND_y_array)
814.     temp_ND_y_array = temp_ND_y_array/ND_y_array_max*100
815.
816.     if D_gradient < 0 and ND_gradient < 0:
817.         combined_improved_array.append('BOTH')
818.         if D_gradient < ND_gradient:
819.             which_hand.append('TREATED')
820.             ax1_ALLPATIENTS.plot(D_x_array_linear, temp_D_y_array)

```

```

821.
822.         else:
823.             which_hand.append('NON-TREATED')
824.             ax1_ALLPATIENTS.plot(ND_x_array_linear, temp_ND_y_array)
825.
826.
827.         elif D_gradient < 0:
828.             combined_improved_array.append('TREATED')
829.             which_hand.append('TREATED')
830.             ax1_ALLPATIENTS.plot(D_x_array_linear, temp_D_y_array)
831.
832.         elif ND_gradient < 0:
833.             combined_improved_array.append('NON-TREATED')
834.             which_hand.append('NON-TREATED')
835.             ax1_ALLPATIENTS.plot(ND_x_array_linear, temp_ND_y_array)
836.
837.     else:
838.         combined_improved_array.append('NEITHER')
839.         if D_gradient < ND_gradient:
840.             which_hand.append('TREATED')
841.         else:
842.             which_hand.append('NON-TREATED')
843.
844.plt.tight_layout()
845.plt.show()
846.
847.# START OF "WHICH HAND?" TWO
848.combined_improved_array = []
849.which_hand = []
850.combined_patient_number_array = []
851.dominant_hand = []
852.
853.longest_array = 0
854.
855.if len(D_patient_number_array_det_2) > len(ND_patient_number_array_det_2):
856.    longest_array = len(D_patient_number_array_det_2)
857.else:
858.    longest_array = len(ND_patient_number_array_det_2)
859.
860.D_counter = -1
861.ND_counter = -1
862.
863._before = 1
864._1W = 7
865._1M = 30
866._3M = 91
867._6M = 183
868._1Y = 365
869._2Y = 730
870._3Y = 1095
871._4Y = 1460
872.
873._before_linear = 1
874._1W_linear = 2
875._1M_linear = 3
876._3M_linear = 4
877._6M_linear = 5
878._1Y_linear = 6
879._2Y_linear = 7
880._3Y_linear = 8
881._4Y_linear = 9
882.
883.fig_ALLPATIENTS2, ax1_ALLPATIENTS2 = plt.subplots(figsize = (8,4))
884.plt.title('Results of Patients Whose \'Before\' Drawings are the Most Severe')
885.plt.grid(linestyle = '-', linewidth=0.5, axis='y')
886.plt.xticks([1,2,3,4,5,6,7,8,9], ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
887.
888.for i in range(0, longest_array):
889.    D_counter += 1
890.    ND_counter += 1
891.
892.    if D_counter >= len(D_patient_number_array_det_2) or ND_counter >= len(ND_patient_number_array_det_2):
893.        break
894.    else:
895.        while D_patient_number_array_det_2[D_counter] != ND_patient_number_array_det_2[ND_counter]:
896.            if D_patient_number_array_det_2[D_counter+1] == ND_patient_number_array_det_2[ND_counter]:
897.                D_counter += 1
898.            elif D_patient_number_array_det_2[D_counter] == ND_patient_number_array_det_2[ND_counter + 1]:
899.                ND_counter += 1
900.            else:
901.                D_counter += 1
902.                ND_counter += 1

```

```

903.
904. D_x_array = []
905. D_x_array_linear = []
906. D_y_array = []
907.
908. if D_time_before_det_2[D_counter] != 0 and D_time_before_det_2[D_counter] != -1:
909.     D_y_array.append(D_time_before_det_2[D_counter])
910.     D_x_array.append(_before)
911.     D_x_array_linear.append(_before_linear)
912.     if D_time_1W_det_2[D_counter] != 0 and D_time_1W_det_2[D_counter] != -1:
913.         D_y_array.append(D_time_1W_det_2[D_counter])
914.         D_x_array.append(_1W)
915.         D_x_array_linear.append(_1W_linear)
916.     if D_time_1M_det_2[D_counter] != 0 and D_time_1M_det_2[D_counter] != -1:
917.         D_y_array.append(D_time_1M_det_2[D_counter])
918.         D_x_array.append(_1M)
919.         D_x_array_linear.append(_1M_linear)
920.     if D_time_3M_det_2[D_counter] != 0 and D_time_3M_det_2[D_counter] != -1:
921.         D_y_array.append(D_time_3M_det_2[D_counter])
922.         D_x_array.append(_3M)
923.         D_x_array_linear.append(_3M_linear)
924.     if D_time_6M_det_2[D_counter] != 0 and D_time_6M_det_2[D_counter] != -1:
925.         D_y_array.append(D_time_6M_det_2[D_counter])
926.         D_x_array.append(_6M)
927.         D_x_array_linear.append(_6M_linear)
928.     if D_time_1Y_det_2[D_counter] != 0 and D_time_1Y_det_2[D_counter] != -1:
929.         D_y_array.append(D_time_1Y_det_2[D_counter])
930.         D_x_array.append(_1Y)
931.         D_x_array_linear.append(_1Y_linear)
932.     if D_time_2Y_det_2[D_counter] != 0 and D_time_2Y_det_2[D_counter] != -1:
933.         D_y_array.append(D_time_2Y_det_2[D_counter])
934.         D_x_array.append(_2Y)
935.         D_x_array_linear.append(_2Y_linear)
936.     if D_time_3Y_det_2[D_counter] != 0 and D_time_3Y_det_2[D_counter] != -1:
937.         D_y_array.append(D_time_3Y_det_2[D_counter])
938.         D_x_array.append(_3Y)
939.         D_x_array_linear.append(_3Y_linear)
940.     if D_time_4Y_det_2[D_counter] != 0 and D_time_4Y_det_2[D_counter] != -1:
941.         D_y_array.append(D_time_4Y_det_2[D_counter])
942.         D_x_array.append(_4Y)
943.         D_x_array_linear.append(_4Y_linear)
944.
945. ND_x_array = []
946. ND_x_array_linear = []
947. ND_y_array = []
948. if ND_time_before_det_2[ND_counter] != 0 and ND_time_before_det_2[ND_counter] != -1:
949.     ND_y_array.append(ND_time_before_det_2[ND_counter])
950.     ND_x_array.append(_before)
951.     ND_x_array_linear.append(_before_linear)
952.
953.     if ND_time_1W_det_2[ND_counter] != 0 and ND_time_1W_det_2[ND_counter] != -1:
954.         ND_y_array.append(ND_time_1W_det_2[ND_counter])
955.         ND_x_array.append(_1W)
956.         ND_x_array_linear.append(_1W_linear)
957.     if ND_time_1M_det_2[ND_counter] != 0 and ND_time_1M_det_2[ND_counter] != -1:
958.         ND_y_array.append(ND_time_1M_det_2[ND_counter])
959.         ND_x_array.append(_1M)
960.         ND_x_array_linear.append(_1M_linear)
961.     if ND_time_3M_det_2[ND_counter] != 0 and ND_time_3M_det_2[ND_counter] != -1:
962.         ND_y_array.append(ND_time_3M_det_2[ND_counter])
963.         ND_x_array.append(_3M)
964.         ND_x_array_linear.append(_3M_linear)
965.     if ND_time_6M_det_2[ND_counter] != 0 and ND_time_6M_det_2[ND_counter] != -1:
966.         ND_y_array.append(ND_time_6M_det_2[ND_counter])
967.         ND_x_array.append(_6M)
968.         ND_x_array_linear.append(_6M_linear)
969.     if ND_time_1Y_det_2[ND_counter] != 0 and ND_time_1Y_det_2[ND_counter] != -1:
970.         ND_y_array.append(ND_time_1Y_det_2[ND_counter])
971.         ND_x_array.append(_1Y)
972.         ND_x_array_linear.append(_1Y_linear)
973.     if ND_time_2Y_det_2[ND_counter] != 0 and ND_time_2Y_det_2[ND_counter] != -1:
974.         ND_y_array.append(ND_time_2Y_det_2[ND_counter])
975.         ND_x_array.append(_2Y)
976.         ND_x_array_linear.append(_2Y_linear)
977.     if ND_time_3Y_det_2[ND_counter] != 0 and ND_time_3Y_det_2[ND_counter] != -1:
978.         ND_y_array.append(ND_time_3Y_det_2[ND_counter])
979.         ND_x_array.append(_3Y)
980.         ND_x_array_linear.append(_3Y_linear)
981.     if ND_time_4Y_det_2[ND_counter] != 0 and ND_time_4Y_det_2[ND_counter] != -1:
982.         ND_y_array.append(ND_time_4Y_det_2[ND_counter])
983.         ND_x_array.append(_4Y)
984.         ND_x_array_linear.append(_4Y_linear)

```

```

985.
986.     if len(D_x_array) > 4 and len(D_y_array) > 4 and len(ND_x_array) > 4 and len(ND_y_array) > 4:
987.
988.         combined_patient_number_array.append(D_patient_number_array_det_2[D_counter])
989.
990.         D_gradient, D_intercept = np.polyfit(D_x_array, D_y_array, 1)
991.         ND_gradient, ND_intercept = np.polyfit(ND_x_array, ND_y_array, 1)
992.
993.         temp_D_y_array = np.array(D_y_array.copy())
994.         D_y_array_max = max(temp_D_y_array)
995.         temp_D_y_array = temp_D_y_array/D_y_array_max*100
996.         temp_ND_y_array = np.array(ND_y_array.copy())
997.         ND_y_array_max = max(temp_ND_y_array)
998.         temp_ND_y_array = temp_ND_y_array/ND_y_array_max*100
999.
1000.        if D_time_before_det_2[D_counter] < 600 and ND_time_before_det_2[ND_counter] < 600: # remove outlier
1001.            if D_gradient < 0 and ND_gradient < 0:
1002.                combined_improved_array.append('BOTH')
1003.                if D_gradient < ND_gradient:
1004.                    which_hand.append('TREATED')
1005.                    if D_y_array[0] == D_y_array_max:
1006.                        ax1_ALLPATIENTS2.plot(D_x_array_linear, D_y_array)
1007.                    else:
1008.                        which_hand.append('NON-TREATED')
1009.                        if ND_y_array[0] == ND_y_array_max:
1010.                            ax1_ALLPATIENTS2.plot(ND_x_array_linear, ND_y_array)
1011.
1012.                elif D_gradient < 0:
1013.                    combined_improved_array.append('TREATED')
1014.                    which_hand.append('TREATED')
1015.                    if D_y_array[0] == D_y_array_max:
1016.                        ax1_ALLPATIENTS2.plot(D_x_array_linear, D_y_array)
1017.
1018.                elif ND_gradient < 0:
1019.                    combined_improved_array.append('NON-TREATED')
1020.                    which_hand.append('NON-TREATED')
1021.                    if ND_y_array[0] == ND_y_array_max:
1022.                        ax1_ALLPATIENTS2.plot(ND_x_array_linear, ND_y_array)
1023.
1024.            else:
1025.                combined_improved_array.append('NEITHER')
1026.                if D_gradient < ND_gradient:
1027.                    which_hand.append('TREATED')
1028.                else:
1029.                    which_hand.append('NON-TREATED')
1030. plt.tight_layout()
1031. plt.show()
1032.
1033. # START OF "WHICH HAND?" THREE
1034. x_array = [1,2,3,4,5,6,7,8,9]
1035. fig_ALLPATIENTS3, ax1_ALLPATIENTS3 = plt.subplots(figsize = (8,4))
1036. plt.title('Average Tremor Severities for Each Hand\nnPEAK DISTANCE (METHOD 2B)')
1037. plt.grid(linestyle = '--', linewidth=0.5, axis='y')
1038. plt.xticks(x_array, ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
1039.
1040. D_average_before_det_2 = np.mean(D_time_before_det_2)
1041. D_average_1W_det_2 = np.mean(D_time_1W_det_2)
1042. D_average_1M_det_2 = np.mean(D_time_1M_det_2)
1043. D_average_3M_det_2 = np.mean(D_time_3M_det_2)
1044. D_average_6M_det_2 = np.mean(D_time_6M_det_2)
1045. D_average_1Y_det_2 = np.mean(D_time_1Y_det_2)
1046. D_average_2Y_det_2 = np.mean(D_time_2Y_det_2)
1047. D_average_3Y_det_2 = np.mean(D_time_3Y_det_2)
1048. D_average_4Y_det_2 = np.mean(D_time_4Y_det_2)
1049.
1050. ND_average_before_det_2 = np.mean(ND_time_before_det_2)
1051. ND_average_1W_det_2 = np.mean(ND_time_1W_det_2)
1052. ND_average_1M_det_2 = np.mean(ND_time_1M_det_2)
1053. ND_average_3M_det_2 = np.mean(ND_time_3M_det_2)
1054. ND_average_6M_det_2 = np.mean(ND_time_6M_det_2)
1055. ND_average_1Y_det_2 = np.mean(ND_time_1Y_det_2)
1056. ND_average_2Y_det_2 = np.mean(ND_time_2Y_det_2)
1057. ND_average_3Y_det_2 = np.mean(ND_time_3Y_det_2)
1058. ND_average_4Y_det_2 = np.mean(ND_time_4Y_det_2)
1059.
1060. D_averages = [D_average_before_det_2, D_average_1W_det_2, D_average_1M_det_2, D_average_3M_det_2,
1061. D_average_6M_det_2, D_average_1Y_det_2, D_average_2Y_det_2, D_average_3Y_det_2, D_average_4Y_det_2]
1062. ND_averages = [ND_average_before_det_2, ND_average_1W_det_2, ND_average_1M_det_2, ND_average_3M_det_2,
1063. ND_average_6M_det_2, ND_average_1Y_det_2, ND_average_2Y_det_2, ND_average_3Y_det_2, ND_average_4Y_det_2]
1064. plt.legend()
1065. ax1_ALLPATIENTS3.plot(x_array, D_averages, color = 'darkblue')

```

```

1065. ax1_ALLPATIENTS3.plot(x_array, ND_averages, color = 'cornflowerblue')
1066. ax1_ALLPATIENTS3.set_ylabel('Tremor Severity')
1067. ax1_ALLPATIENTS3.set_xlabel('Time')
1068. ax1_ALLPATIENTS3.legend(['Treated Hand', 'Untreated Hand'], loc="upper right")
1069. plt.tight_layout()
1070. plt.show()
1071.
1072. # AVERAGE TREMOR SEVERITY USING AVG AREA TRAPZ METHOD
1073. x_array = [1,2,3,4,5,6,7,8,9]
1074. fig_ALLPATIENTS3B, ax1_ALLPATIENTS3B = plt.subplots(figsize = (8,4))
1075. plt.title('Average Tremor Severities for Each Hand\nAVERAGE AREA (METHOD 2A)')
1076. plt.grid(linestyle = '--', linewidth=0.5, axis='y')
1077. plt.xticks(x_array, ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
1078.
1079. D_average_before_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_before_avg_area_trapz])
1080. D_average_1W_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_1W_avg_area_trapz])
1081. D_average_1M_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_1M_avg_area_trapz])
1082. D_average_3M_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_3M_avg_area_trapz])
1083. D_average_6M_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_6M_avg_area_trapz])
1084. D_average_1Y_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_1Y_avg_area_trapz])
1085. D_average_2Y_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_2Y_avg_area_trapz])
1086. D_average_3Y_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_3Y_avg_area_trapz])
1087. D_average_4Y_avg_area_trapz = np.mean([abs(float(x)) for x in D_time_4Y_avg_area_trapz])
1088.
1089. ND_average_before_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_before_avg_area_trapz])
1090. ND_average_1W_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_1W_avg_area_trapz])
1091. ND_average_1M_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_1M_avg_area_trapz])
1092. ND_average_3M_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_3M_avg_area_trapz])
1093. ND_average_6M_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_6M_avg_area_trapz])
1094. ND_average_1Y_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_1Y_avg_area_trapz])
1095. ND_average_2Y_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_2Y_avg_area_trapz])
1096. ND_average_3Y_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_3Y_avg_area_trapz])
1097. ND_average_4Y_avg_area_trapz = np.mean([abs(float(x)) for x in ND_time_4Y_avg_area_trapz])
1098.
1099. D_averages = [D_average_before_avg_area_trapz, D_average_1W_avg_area_trapz, D_average_1M_avg_area_trapz,
    D_average_3M_avg_area_trapz, D_average_6M_avg_area_trapz, D_average_1Y_avg_area_trapz,
    D_average_2Y_avg_area_trapz, D_average_3Y_avg_area_trapz, D_average_4Y_avg_area_trapz]
1100. ND_averages = [ND_average_before_avg_area_trapz, ND_average_1W_avg_area_trapz, ND_average_1M_avg_area_trapz,
    ND_average_3M_avg_area_trapz, ND_average_6M_avg_area_trapz, ND_average_1Y_avg_area_trapz,
    ND_average_2Y_avg_area_trapz, ND_average_3Y_avg_area_trapz, ND_average_4Y_avg_area_trapz]
1101.
1102. plt.legend()
1103. ax1_ALLPATIENTS3B.plot(x_array, D_averages, color = 'darkblue')
1104. ax1_ALLPATIENTS3B.plot(x_array, ND_averages, color = 'cornflowerblue')
1105. ax1_ALLPATIENTS3B.set_ylabel('Tremor Severity')
1106. ax1_ALLPATIENTS3B.set_xlabel('Time')
1107. ax1_ALLPATIENTS3B.legend(['Treated Hand', 'Untreated Hand'], loc="upper right")
1108. plt.savefig('RESULTS\GRAPHS\AverageTremSev_AvgArea.png', bbox_inches='tight', dpi=150)
1109.
1110. plt.tight_layout()
1111. plt.show()
1112.
1113. # START OF "WHICH HAND?" FOUR
1114.
1115. x_array = [1,2,3,4,5,6,7,8,9]
1116. fig_ALLPATIENTS4, ax1_ALLPATIENTS4 = plt.subplots(figsize = (8,4))
1117. plt.title('Normalised Average Tremor Severities for Each Hand\nPEAK DISTANCE (METHOD 2B)')
1118. plt.grid(linestyle = '--', linewidth=0.5, axis='y')
1119. plt.xticks(x_array, ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
1120.
1121. all_arrays = [*D_time_before_det_2, *D_time_1W_det_2, *D_time_1M_det_2, *D_time_3M_det_2, *D_time_6M_det_2,
    *D_time_1Y_det_2, *D_time_2Y_det_2, *D_time_3Y_det_2, *D_time_4Y_det_2, *ND_time_before_det_2, *ND_time_1W_det_2,
    *ND_time_1M_det_2, *ND_time_3M_det_2, *ND_time_6M_det_2, *ND_time_1Y_det_2, *ND_time_2Y_det_2, *ND_time_3Y_det_2,
    *ND_time_4Y_det_2]
1122. all_min, q10, q90, q95, q98, q999, all_max = np.quantile(all_arrays, [0, 0.1, 0.9, 0.95, 0.98, 0.99, 1])
1123. denom = q95 - all_min
1124.
1125. D_time_before_normalised = [(x - all_min)/denom for x in D_time_before_det_2]
1126. D_time_1W_normalised = [(x - all_min)/denom for x in D_time_1W_det_2]
1127. D_time_1M_normalised = [(x - all_min)/denom for x in D_time_1M_det_2]
1128. D_time_3M_normalised = [(x - all_min)/denom for x in D_time_3M_det_2]
1129. D_time_6M_normalised = [(x - all_min)/denom for x in D_time_6M_det_2]
1130. D_time_1Y_normalised = [(x - all_min)/denom for x in D_time_1Y_det_2]
1131. D_time_2Y_normalised = [(x - all_min)/denom for x in D_time_2Y_det_2]
1132. D_time_3Y_normalised = [(x - all_min)/denom for x in D_time_3Y_det_2]
1133. D_time_4Y_normalised = [(x - all_min)/denom for x in D_time_4Y_det_2]
1134.
1135. D_times_normalised = [D_time_before_normalised, D_time_1W_normalised, D_time_1M_normalised,
    D_time_3M_normalised, D_time_6M_normalised, D_time_1Y_normalised, D_time_2Y_normalised, D_time_3Y_normalised,
    D_time_4Y_normalised]
1136.
1137. ND_time_before_normalised = [(x - all_min)/denom for x in ND_time_before_det_2]

```

```

1138. ND_time_1W_normalised = [(x - all_min)/denom for x in ND_time_1W_det_2]
1139. ND_time_1M_normalised = [(x - all_min)/denom for x in ND_time_1M_det_2]
1140. ND_time_3M_normalised = [(x - all_min)/denom for x in ND_time_3M_det_2]
1141. ND_time_6M_normalised = [(x - all_min)/denom for x in ND_time_6M_det_2]
1142. ND_time_1Y_normalised = [(x - all_min)/denom for x in ND_time_1Y_det_2]
1143. ND_time_2Y_normalised = [(x - all_min)/denom for x in ND_time_2Y_det_2]
1144. ND_time_3Y_normalised = [(x - all_min)/denom for x in ND_time_3Y_det_2]
1145. ND_time_4Y_normalised = [(x - all_min)/denom for x in ND_time_4Y_det_2]
1146.
1147. D_average_before_det_2 = np.mean(D_time_before_normalised)
1148. D_average_1W_det_2 = np.mean(D_time_1W_normalised)
1149. D_average_1M_det_2 = np.mean(D_time_1M_normalised)
1150. D_average_3M_det_2 = np.mean(D_time_3M_normalised)
1151. D_average_6M_det_2 = np.mean(D_time_6M_normalised)
1152. D_average_1Y_det_2 = np.mean(D_time_1Y_normalised)
1153. D_average_2Y_det_2 = np.mean(D_time_2Y_normalised)
1154. D_average_3Y_det_2 = np.mean(D_time_3Y_normalised)
1155. D_average_4Y_det_2 = np.mean(D_time_4Y_normalised)
1156.
1157. ND_average_before_det_2 = np.mean(ND_time_before_normalised)
1158. ND_average_1W_det_2 = np.mean(ND_time_1W_normalised)
1159. ND_average_1M_det_2 = np.mean(ND_time_1M_normalised)
1160. ND_average_3M_det_2 = np.mean(ND_time_3M_normalised)
1161. ND_average_6M_det_2 = np.mean(ND_time_6M_normalised)
1162. ND_average_1Y_det_2 = np.mean(ND_time_1Y_normalised)
1163. ND_average_2Y_det_2 = np.mean(ND_time_2Y_normalised)
1164. ND_average_3Y_det_2 = np.mean(ND_time_3Y_normalised)
1165. ND_average_4Y_det_2 = np.mean(ND_time_4Y_normalised)
1166.
1167. D_averages = [D_average_before_det_2, D_average_1W_det_2, D_average_1M_det_2, D_average_3M_det_2,
    D_average_6M_det_2, D_average_1Y_det_2, D_average_2Y_det_2, D_average_3Y_det_2, D_average_4Y_det_2]
1168. ND_averages = [ND_average_before_det_2, ND_average_1W_det_2, ND_average_1M_det_2, ND_average_3M_det_2,
    ND_average_6M_det_2, ND_average_1Y_det_2, ND_average_2Y_det_2, ND_average_3Y_det_2, ND_average_4Y_det_2]
1169.
1170. plt.legend()
1171. ax1_ALLPATIENTS4.plot(x_array, D_averages, color = 'darkblue')
1172. ax1_ALLPATIENTS4.plot(x_array, ND_averages, color = 'cornflowerblue')
1173. ax1_ALLPATIENTS4.set_ylabel('Tremor Severity')
1174. ax1_ALLPATIENTS4.set_xlabel('Time')
1175.
1176. ax1_ALLPATIENTS4.legend(['Treated Hand', 'Untreated Hand'], loc="upper right", prop={'size': 14})
1177. plt.savefig('RESULTS\GRAPHS\AverageTremSev_Det2.png', bbox_inches='tight', dpi=150)
1178. plt.show()
1179.
1180. # START OF "WHICH HAND?" FOUR B
1181. x_array = [1,2,3,4,5,6,7,8,9]
1182. fig_ALLPATIENTS4B, ax1_ALLPATIENTS4B = plt.subplots(figsize = (8,4))
1183. plt.title('Normalised Average Tremor Severities for Each Hand\nAVERAGE AREA (METHOD 2A)')
1184. plt.grid(linestyle = '--', linewidth=0.5, axis='y')
1185. plt.xticks(x_array, ['Before', '1W', '1M', '3M', '6M', '1Y', '2Y', '3Y', '4Y'])
1186.
1187.
1188. copy_D_time_before_avg_area_trapz = [abs(float(x)) for x in D_time_before_avg_area_trapz]
1189. for c in copy_D_time_before_avg_area_trapz:
1190.     if c == 0.0:
1191.         copy_D_time_before_avg_area_trapz.remove(c)
1192. copy_D_time_1W_avg_area_trapz = [abs(float(x)) for x in D_time_1W_avg_area_trapz]
1193. for c in copy_D_time_1W_avg_area_trapz:
1194.     if c == 0.0:
1195.         copy_D_time_1W_avg_area_trapz.remove(c)
1196. copy_D_time_1M_avg_area_trapz = [abs(float(x)) for x in D_time_1M_avg_area_trapz]
1197. for c in copy_D_time_1M_avg_area_trapz:
1198.     if c == 0.0:
1199.         copy_D_time_1M_avg_area_trapz.remove(c)
1200. copy_D_time_3M_avg_area_trapz = [abs(float(x)) for x in D_time_3M_avg_area_trapz]
1201. for c in copy_D_time_3M_avg_area_trapz:
1202.     if c == 0.0:
1203.         copy_D_time_3M_avg_area_trapz.remove(c)
1204. copy_D_time_6M_avg_area_trapz = [abs(float(x)) for x in D_time_6M_avg_area_trapz]
1205. for c in copy_D_time_6M_avg_area_trapz:
1206.     if c == 0.0:
1207.         copy_D_time_6M_avg_area_trapz.remove(c)
1208. copy_D_time_1Y_avg_area_trapz = [abs(float(x)) for x in D_time_1Y_avg_area_trapz]
1209. for c in copy_D_time_1Y_avg_area_trapz:
1210.     if c == 0.0:
1211.         copy_D_time_1Y_avg_area_trapz.remove(c)
1212. copy_D_time_2Y_avg_area_trapz = [abs(float(x)) for x in D_time_2Y_avg_area_trapz]
1213. for c in copy_D_time_2Y_avg_area_trapz:
1214.     if c == 0.0:
1215.         copy_D_time_2Y_avg_area_trapz.remove(c)
1216. copy_D_time_3Y_avg_area_trapz = [abs(float(x)) for x in D_time_3Y_avg_area_trapz]
1217. for c in copy_D_time_3Y_avg_area_trapz:

```

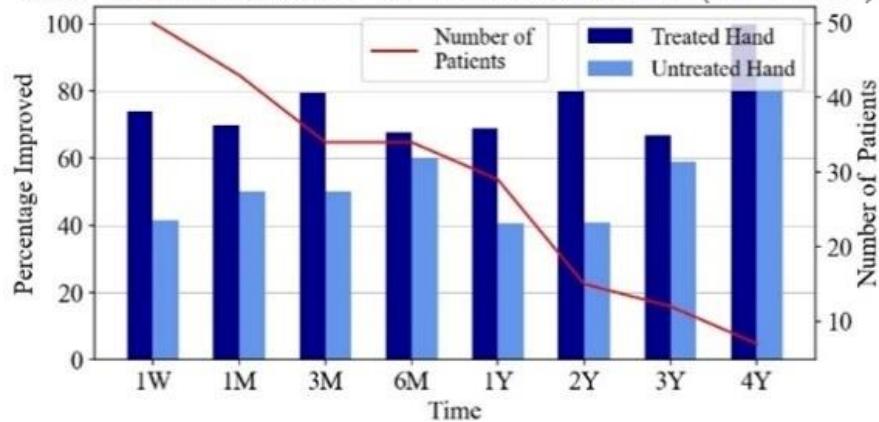
```

1218.     if c == 0.0:
1219.         copy_D_time_3Y_avg_area_trapz.remove(c)
1220. copy_D_time_4Y_avg_area_trapz = [abs(float(x)) for x in D_time_4Y_avg_area_trapz]
1221. for c in copy_D_time_4Y_avg_area_trapz:
1222.     if c == 0.0:
1223.         copy_D_time_4Y_avg_area_trapz.remove(c)
1224.
1225. copy_ND_time_before_avg_area_trapz = [abs(float(x)) for x in ND_time_before_avg_area_trapz]
1226. for c in copy_ND_time_before_avg_area_trapz:
1227.     if c == 0.0:
1228.         copy_ND_time_before_avg_area_trapz.remove(c)
1229. copy_ND_time_1W_avg_area_trapz = [abs(float(x)) for x in ND_time_1W_avg_area_trapz]
1230. for c in copy_ND_time_1W_avg_area_trapz:
1231.     if c == 0.0:
1232.         copy_ND_time_1W_avg_area_trapz.remove(c)
1233. copy_ND_time_1M_avg_area_trapz = [abs(float(x)) for x in ND_time_1M_avg_area_trapz]
1234. for c in copy_ND_time_1M_avg_area_trapz:
1235.     if c == 0.0:
1236.         copy_ND_time_1M_avg_area_trapz.remove(c)
1237. copy_ND_time_3M_avg_area_trapz = [abs(float(x)) for x in ND_time_3M_avg_area_trapz]
1238. for c in copy_ND_time_3M_avg_area_trapz:
1239.     if c == 0.0:
1240.         copy_ND_time_3M_avg_area_trapz.remove(c)
1241. copy_ND_time_6M_avg_area_trapz = [abs(float(x)) for x in ND_time_6M_avg_area_trapz]
1242. for c in copy_ND_time_6M_avg_area_trapz:
1243.     if c == 0.0:
1244.         copy_ND_time_6M_avg_area_trapz.remove(c)
1245. copy_ND_time_1Y_avg_area_trapz = [abs(float(x)) for x in ND_time_1Y_avg_area_trapz]
1246. for c in copy_ND_time_1Y_avg_area_trapz:
1247.     if c == 0.0:
1248.         copy_ND_time_1Y_avg_area_trapz.remove(c)
1249. copy_ND_time_2Y_avg_area_trapz = [abs(float(x)) for x in ND_time_2Y_avg_area_trapz]
1250. for c in copy_ND_time_2Y_avg_area_trapz:
1251.     if c == 0.0:
1252.         copy_ND_time_2Y_avg_area_trapz.remove(c)
1253. copy_ND_time_3Y_avg_area_trapz = [abs(float(x)) for x in ND_time_3Y_avg_area_trapz]
1254. for c in copy_ND_time_3Y_avg_area_trapz:
1255.     if c == 0.0:
1256.         copy_ND_time_3Y_avg_area_trapz.remove(c)
1257. copy_ND_time_4Y_avg_area_trapz = [abs(float(x)) for x in ND_time_4Y_avg_area_trapz]
1258. for c in copy_ND_time_4Y_avg_area_trapz:
1259.     if c == 0.0:
1260.         copy_ND_time_4Y_avg_area_trapz.remove(c)
1261.
1262. print(copy_D_time_before_avg_area_trapz)
1263. print(copy_ND_time_before_avg_area_trapz)
1264. m1 = max(copy_D_time_before_avg_area_trapz)
1265. m2 = max(copy_D_time_1W_avg_area_trapz)
1266. m3 = max(copy_D_time_1M_avg_area_trapz)
1267. m4 = max(copy_D_time_3M_avg_area_trapz)
1268. m5 = max(copy_D_time_6M_avg_area_trapz)
1269. m6 = max(copy_D_time_1Y_avg_area_trapz )
1270. m7 = max(copy_D_time_2Y_avg_area_trapz )
1271. m8 = max(copy_D_time_3Y_avg_area_trapz )
1272. m9 = max(copy_D_time_4Y_avg_area_trapz )
1273.
1274. m10 = max(copy_ND_time_before_avg_area_trapz)
1275. m11 = max(copy_ND_time_1W_avg_area_trapz)
1276. m12 = max(copy_ND_time_1M_avg_area_trapz)
1277. m13 = max(copy_ND_time_3M_avg_area_trapz)
1278. m14 = max(copy_ND_time_6M_avg_area_trapz)
1279. m15 = max(copy_ND_time_1Y_avg_area_trapz)
1280. m16 = max(copy_ND_time_2Y_avg_area_trapz)
1281. m17 = max(copy_ND_time_3Y_avg_area_trapz)
1282. m18 = max(copy_ND_time_4Y_avg_area_trapz)
1283.
1284. m1 = min(copy_D_time_before_avg_area_trapz)
1285. m2 = min(copy_D_time_1W_avg_area_trapz)
1286. m3 = min(copy_D_time_1M_avg_area_trapz)
1287. m4 = min(copy_D_time_3M_avg_area_trapz)
1288. m5 = min(copy_D_time_6M_avg_area_trapz)
1289. m6 = min(copy_D_time_1Y_avg_area_trapz )
1290. m7 = min(copy_D_time_2Y_avg_area_trapz )
1291. m8 = min(copy_D_time_3Y_avg_area_trapz )
1292. m9 = min(copy_D_time_4Y_avg_area_trapz )
1293.
1294. m10 = min(copy_ND_time_before_avg_area_trapz)
1295. m11 = min(copy_ND_time_1W_avg_area_trapz)
1296. m12 = min(copy_ND_time_1M_avg_area_trapz)
1297. m13 = min(copy_ND_time_3M_avg_area_trapz)
1298. m14 = min(copy_ND_time_6M_avg_area_trapz)
1299. m15 = min(copy_ND_time_1Y_avg_area_trapz)

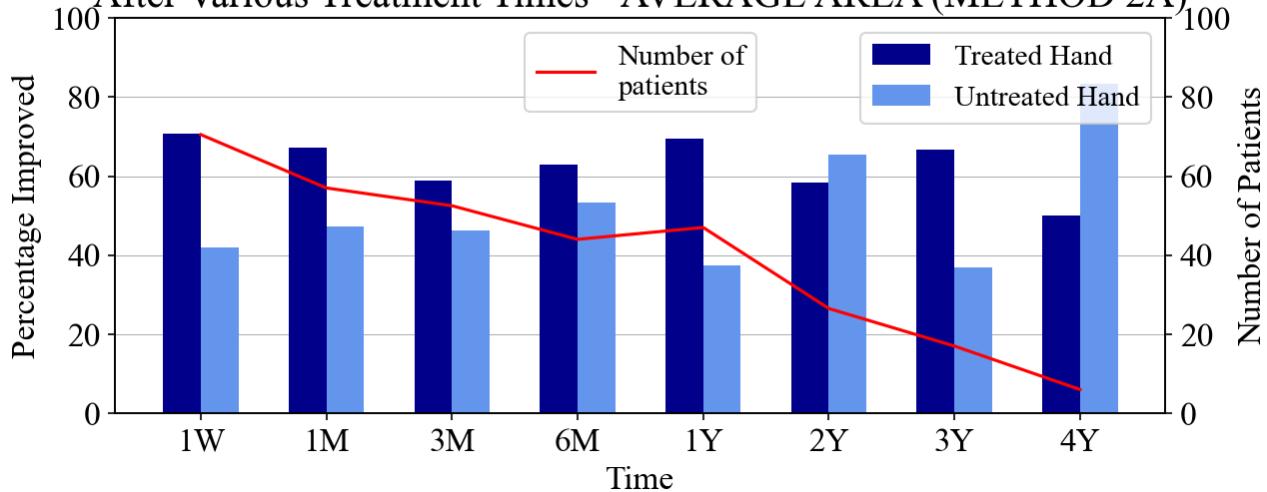
```


Appendix 9: Full Sized Result Graphs

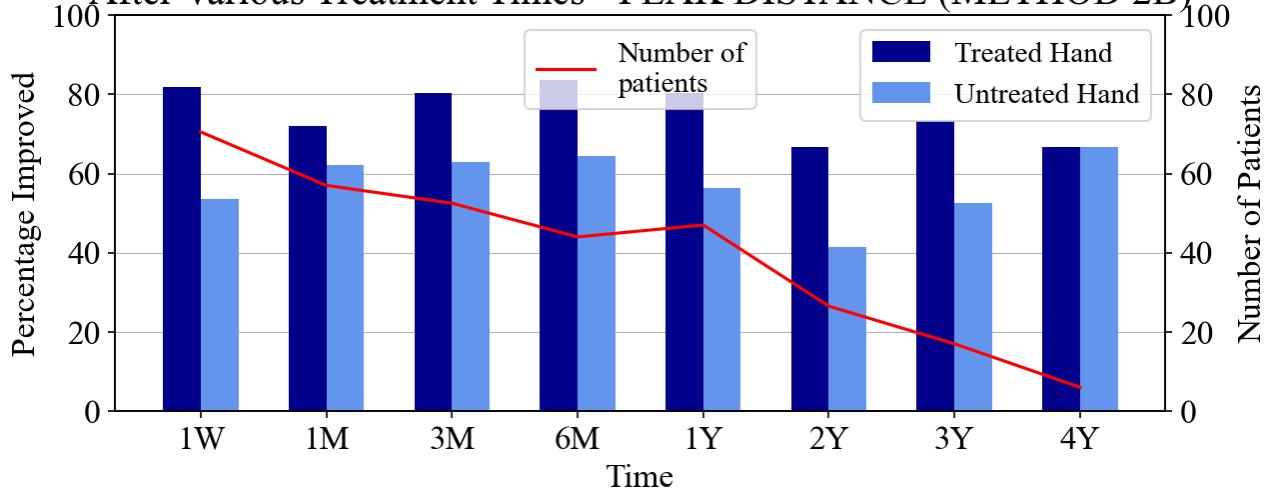
Percentage of Patients with Tremor Before Treatment that Improved After Various Treatment Times - SPIRAL ANALYSIS (METHOD 1)



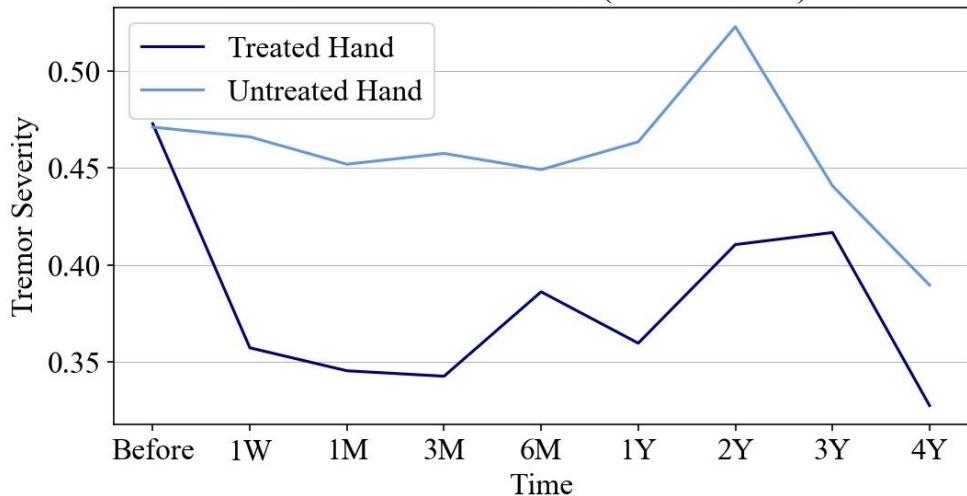
Percentage of Patients with Tremor Before Treatment that Improved After Various Treatment Times - AVERAGE AREA (METHOD 2A)



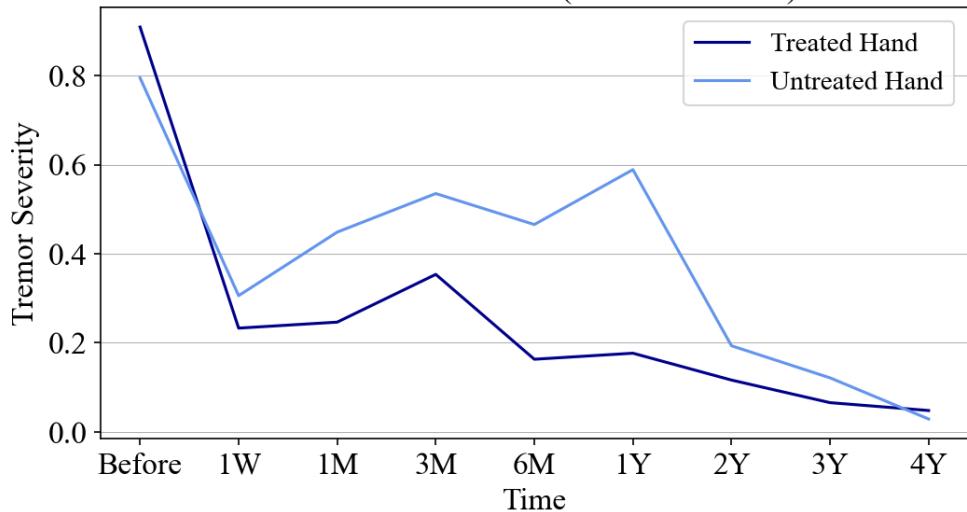
Percentage of Patients with Tremor Before Treatment that Improved After Various Treatment Times - PEAK DISTANCE (METHOD 2B)



**Normalised Average Tremor Severities for Each Hand
SPIRAL ANALYSIS (METHOD 1)**



**Normalised Average Tremor Severities for Each Hand
AVERAGE AREA (METHOD 2A)**



**Normalised Average Tremor Severities for Each Hand
PEAK DISTANCE (METHOD 2B)**

