WHICH HAND? AVERAGE AREA AND NUMBER OF 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:** The purpose of this document is to provide an easy-to-use template/style sheet to enable authors to prepare papers in the correct format and style for the final year laboratory project. This document may be downloaded from the School of Electrical and Information Engineering web site and can be used as a template (MS Word 6.0 or later). To ensure conformity of appearance it is essential that these instructions are followed. The abstract should be limited to 50-200 words, which should concisely summarise the paper.

# Introduction

(AIM) Quantitatively investigate the efficacy of the FUS treatment by investigating computational and statistical analysis of spiral drawings. Provide insight about the extent that these results can be used to assess the severity of tremor on the patient’s treated or untreated side after the treatment.

NEED MORE

TWO METHODS WERE INVESTIGATED AND IMPLEMENTED FURTHER.

# Literature survey

## Focused Ultrasound Treatment

Focused Ultrasound Treatment (FUS) is a new and promising non-invasive treatment for movement disorders. 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 – or temporarily modify the function of – 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, often resulting in unilateral treatment of just treating the dominant hand [2] since FUS is only performed on one side of the brain. This treatment shows immediate reduction in tremor on the treated side of the body. This study will focus on FUS as a treatment for patients with ET and PD in an attempt to determine whether the treatment is successful in reducing tremor, slowing the progression of these conditions.

## Hand-drawn Shapes

The observational analysis of hand-drawn shapes by a neurologist is widely used as a test of severity of movement disorders [3]. Analysis of hand-drawn shapes instead of handwriting is performed to prevent the stylistic differences of handwriting being a contributing factor in the severity tests [4]. An Archimedes spiral drawing is used in particular as it is able to capture the frequency, amplitude, and direction of a tremor [4]. Long, straight vertical, or horizontal line drawings offer similar results. Since these drawings require one continuous pen motion, instead of the broken motions of written words, they are able to emphasise the abnormal movements specific to each of the various movement disorders [4]. The typical characteristics of tremor types seen in writing and drawing tasks can be seen in [ [4], Tab. 1]. While this table does show that ET and PD have similar characteristics, previous studies have shown that computational analysis of such drawings can reliably discriminate between the movement disorders [3]. Further, the combination of traditional and computational analysis has provided significant progress in the classification of disease severity [5]. It is with this in mind that this research of quantitatively investigating the efficacy of FUS treatment is performed.

## Existing Methods

1. ANALYSIS OF PATTERNS IN TREMOR DIAGNOSIS SPIRAL DRAWINGS FOR AUTOMATED CLASSIFICATION [6]
2. Application of machine learning and numerical analysis to classify tremor in patients affected with essential tremor or Parkinson’s disease [7]
3. Quantification of tremor with a digitizing tablet [8]
4. Quantification of the drawing of an Archimedes spiral through the analysis of its digitized picture [9]

Do we need this?

Yes – mention method 1

# project plan

This is a group project, with an individual report component. At the commencement of this project, the submitted Investigation Project plan, found in Appendix X, was reviewed and discussed with Professor Aharonson, the supervisor of this project. It was quickly realised that the scope and complexity of the project was greatly exaggerated and that much change would need be to made to the plan. The biggest error made in scope assumption was that of needing machine learning as a means of answering the investigation question. There is also no need to differentiate between patients with PD or ET using machine learning, as this has already been done and does not help answer the investigation question. Instead, this project focuses on the use of image analysis and data computational analysis. Once these differentiating factors were identified, a new project plan, scope and schedule was created.

## Workload Division

*Research:* Both members of the group conducted sufficient research about the basics of the required medical theory regarding PD, ET and the FUS treatment.

*Pre-processing:* To pre-process the data, it was decided to use existing code and methods from outside sources, with appropriate copyright licenses that allow for changes to be made to the code as long as full credit is maintained to the original author(s). I focused on implementing and improving the pre-processing code and methods.

*Methods and results:* It was decided to implement two methods of computational analysis. Each group member focused on one of these methods, and then worked together to discuss, compare results, and draw conclusions.

# data

## Database access and ethical clearance

Fully anonymised data of patients with either PD or ET has been 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 permission letter from Dr Schlesinger as well as the Clearance Certificate for this project can be found in Appendices X and Y respectively.

## Patients

Out of the 122 patients, 34 are undergoing treatment for Parkinson’s Disease, and the remaining 88 for Essential Tremor. This investigation does not need to differentiate between disease, but only determine whether the tremor is reducing. However, future studies might re-look at this data in order to draw better conclusions, especially if machine learning is implemented.

## Data

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

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.

# Pre-Processing of Data

## Original Spiral Isolation Code

A previous master’s research project, by Kelvin da Santos, included a pre-processing python script that was built for isolating the spirals [10]. This code was based on the ‘OpenCV Text Detection (EAST text detector)’ article by Adrian Rosebrock [11]. After meeting and discussing with Kelvin, permission was granted to use and further improve this code to suit the needs of this project. This code originally made use of a text detector library, ‘frozen\_east\_text\_detection’, to detect the labels “Drawing A”, “Drawing B” and “Drawing C”, and save them into a list. This returned the four coordinates of each of these text areas. This process can be seen in figure X, image B. 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.

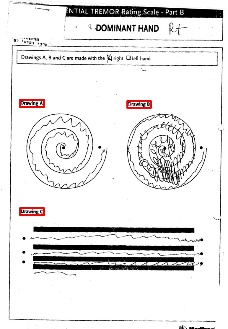
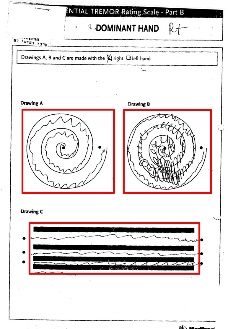
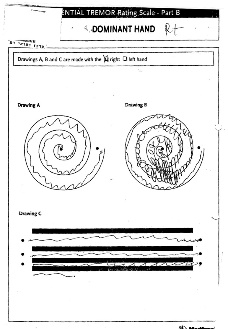


Figure 1: A) Original filled in template, B) text detection and C) image isolation and cropping procedure.

## Improvement of Isolation Code

Most of the scanned drawings are pixelated, slightly rotated, or contain erroneous pen or other markings. This caused the above-mentioned text detection code to sometimes incorrectly detect erroneous markings as text. These would be added to the list of text coordinates, which would lead to inaccurately cropped spirals. This further caused the results of the image analysis to become skewed due to these accidental outliers. Since there are 127 patient’s data, with each patient having anywhere between 2 and 20 scanned drawing, it was imperative to correct the inaccurately cropped spirals. Thus, much time was spent improving the error detection and correction before cropping of the spirals. This code also did not crop the line-drawing, which was decided is a requirement for this investigation in order to be able to see whether analysing the spirals or lines provided more reliable results – and allowing for a combination of the two. It is imperative that as many spirals and line-drawings are isolated and cropped in order to have as much useable data as possible.

*Performing logic checks:* More logic checks were introduced into the code to decide whether the coordinates of text detected were the required ones of “Drawing A”, “Drawing B” and “Drawing C” respectively. Other markings detected were ignored. This was possible since every template is the same and thus the approximate position of the spirals and line-drawings can be determined.

*Line-drawing C:* More code was developed to also crop the line-drawings found at the bottom of each scan. This can be seen in figure X, image B and C above.

*Not relying on specific text:* The previous method relied on the position of “Drawing A” to determine the width of the spirals, and thus the cropping position of both spirals. However, in some instances the “Drawing A” text was not detected, and thus the entire process would fail. Instead, a combination of the relative position of any available text detected would be used to ensure the best possible image isolation and cropping. It no text was detected, average values from all previous scanned drawings are used.

*Final cropped images:* The code iterates through the specific folder structure and saves the cropped spiral A, spiral B, and line-block C for each scanned drawing of every patient in JPEG format. Each of these final images are converted to greyscale and resized to 300x300 pixel squares for the spirals, and 600x300 pixel rectangles for the line-block images. This ensures consistent pixel distribution for more accurate comparison further by reducing the chance that an image will be pixelated according to the resolution at which it was scanned.

## Extraction of line-drawing C from line-block C

Since, this report focuses on the second method, which analyses the line-drawings. Thus, much of the work required for method 2 included the pre-processing of this data; will be expanded and explained here. 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. Extracting only the line prevents the large black rectangles from influencing the analysis techniques. However, this provided to not be as simple as detecting the rectangles and extracting the contained line-drawings due to the slight rotation of each page due to human error whilst scanning. Thus, compensation for this rotation needs to occur in order to prevent erroneously cropped lines.

*Rectangle rotation correction:* Using code snippets from the article “Cropping Rotated Rectangles from Image with OpenCV” by jdhao [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 of concern 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, using the OpenCV python package, the function is used in conjunction with the 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 whether 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 X below.

A: Poor detection B: Accurate detection C: Corrected image

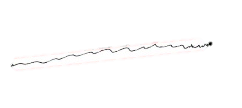


Figure 2: Rectangle detection and rotation correction process

*Final extracted singular lines:* The main goal of this pre-processing section is to successfully extract the most possible images with high precision and very little error. Out of the 1161 total scanned images, a total of 1019 line-drawing C 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.

# 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. This method was implemented on each of the ‘spiral A’ images mentioned above.

## Method

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

*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 in order to take the inverse tangent to produce the ‘pixel angle’ [6].

*Relative orientation:* This quantity is found by subtracting the pixel angle from the edge angle for each pixel. This is plotted as a histogram to visualise and quantify the distribution of edge angles. 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. The normalised standard deviation is found for every spiral of each patient. This allows for comparison between patient’s own hands, as well as between patients in general. The existence of more angles – a higher frequency of edge angles – also indicates a worse tremor.

## Results

The two spirals seen in figure X are used to demonstrate the effectiveness of this method. A normalised standard deviation of 0.64 is calculated for spiral A1, and 0.15 for spiral A2. As seen in figure X, spiral A1 also has a higher frequency of the various edge angles present. These two indicators correctly imply that spiral A1 has a worse tremor than spiral A2.

Figure 3: Spirals A1 and A2 used to demonstrate method 1

Spiral A1: Before treatment Spiral A2: After 1 month

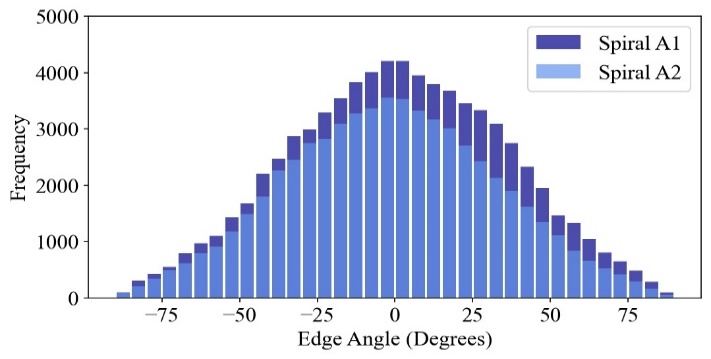
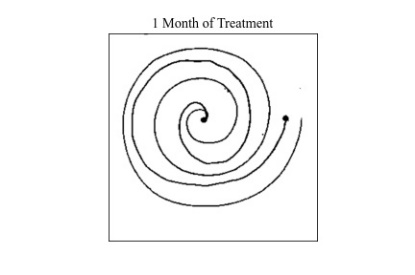
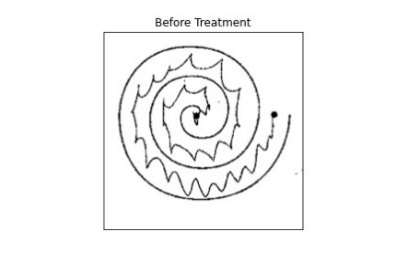


Figure 4: Relative orientation distributions of spirals A1 and A2

The spirals of each patient after various treatment times were analysed using this method in order to determine whether the patient’s tremor improved. As seen in figure X, the treated hand of most patients improved 75.8% of the time, with the untreated hand improving 53.4% of the time. However, it should be noted that the number of patients with data for later periods of time (2 years and greater) drastically decreased. This calls into question the validity of the later years’ results due to the small amount of data.

# Method 2: Average Area and Number of peaks

After the pre-processing mentioned in section 5.3 above, each of the top-most line drawings have been successfully extracted for each patient’s scanned drawings. These will be refereed to as ‘the line(s)’. There are a total of 1019 useable lines that were extracted and saved as JPEG images. This method relies on converting the line into a python function array on which analysis can be made.

*Line noise reduction:* Since many of the lines still contained erroneous pixels that were not of the original drawing, these need to be removed. Pixelation and blurriness due to the poor quality of each scan also needs to be reduced. All this will allow for high-quality analysis. First, each line is converted to grayscale, and using the function from the Image module from the PIL Python package [16], each pixel above a particular greyscale value threshold is mapped to a black coloured pixel. This allows any light grey erroneous pixels to be removed and only the remaining wanted black pixels are left.

*Converting to function:* Using the NumPy Python package’s function [17], all indices of pixels in the line that are non-zero (black) are read into a multidimensional array. The x- and y-coordinates of each index is then extracted from this array. This has successfully created two arrays of corresponding coordinates for the x- and y-values of each pixel from the line. By sorting these according to the x-values, these arrays can easily be plotted on a graph to show these points. However, it can quickly be seen that these points are extremely scattered and noisy – possibly due to the blurriness and pixelation of the original scanned images. These noisy plots, with hundreds of peaks and troughs, can be seen in figure X. For easier analysis, each graph for each line is shifted down by the average y-value so that it is centred around the x-axis. These arrays will be referred to as the ‘line-function’.

*Graph noise reduction:* To reduce the noise of the resulting graphs, the real one-dimensional Fourier Transform is computed using the SciPy.fft Python package [18]. All entries in the y-value array are real, and thus the faster and more optimized is used. From this transform, it can be seen that there is only a small range of useful frequencies present in the line-function. It was decided to keep only 5% of the frequencies, and discard the higher, unwanted frequencies caused by the pixelated/blurry input or erroneous markings. The inverse real discrete Fourier Transform is then computed, resulting in line-graphs that are noise free.

*Extreme frequency bug fix:* It should be noted that some of the line-graphs produced an extreme range of frequencies, and thus only keeping 5% did not reduce the noise sufficiently. This occurred for a variety of reasons, including having pen markings of very thick width. The graph noise reduction process was repeated until sufficient noise reduction had occurred. It was known when sufficient noise had been removed when the number of peaks was accurate and not due to noisy plotting.

Chart, histogram

Description automatically generated

Chart, line chart

Description automatically generated

To more easily analyse the final results produced from these lines, a 2D array was used to store the following information:

* Name of line (unique field)
* Patient number
* Time frame after treatment when the line was drawn
* Whether the line was drawn by the patient’s dominant or non-dominant hand
* Whether the line was drawn by the patient’s treated or non-treated hand
* Each of the analytical values for each line after the line had been transformed into a function
  + Area under absolute value of function
  + Standard deviation of the area
  + Maximum value of function
  + Average area of small segments along the absolute value of the function
  + Standard deviation of this average area
  + Number of peak and troughs
  + Average distance between all adjacent peaks and troughs

OpenCV EAST text detection [11]

OpenCV shape detection [13]

Cropped rotated rectangles [12]

## Implementation

## Results

## Future Improvements

This method can be applied to spiral images; however it would be much more complicated.

One method would be to ‘unwrap’ the spirals and then perform this analysis. Would require more hectic pre-processing.

# results

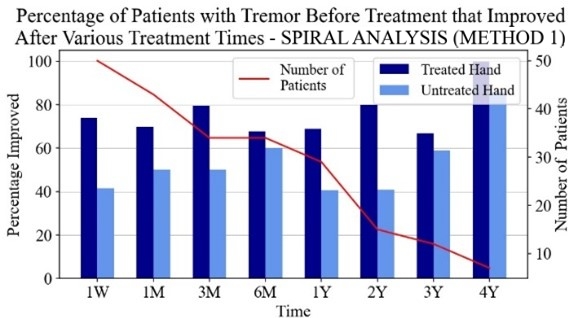


Figure 5: Percentage of improved patients according to method 1

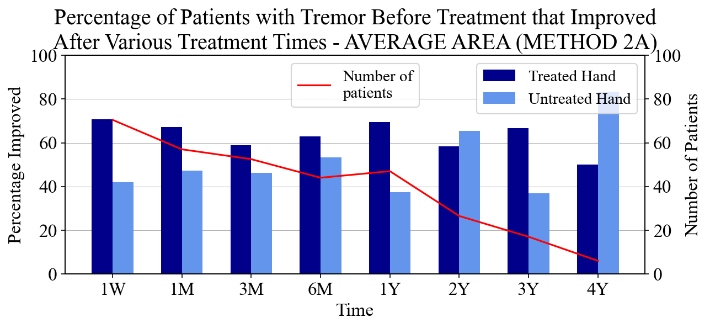


Figure 6: Percentage of improved patients according to average area result from method 2A

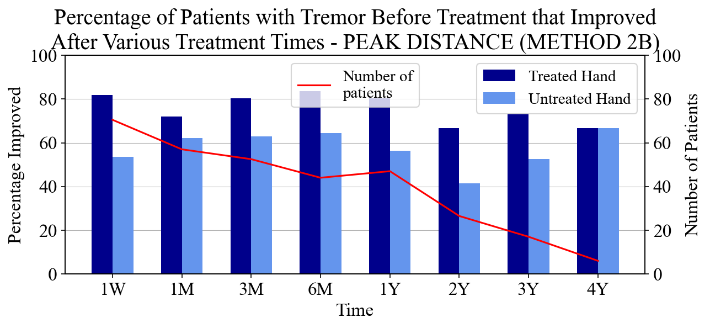


Figure 7: Percentage of improved patients according to peak distance result from method 2B

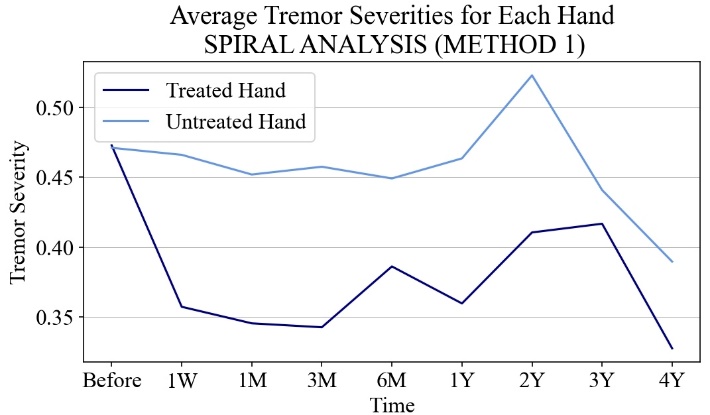


Figure 8: Average tremor severity according to method 1

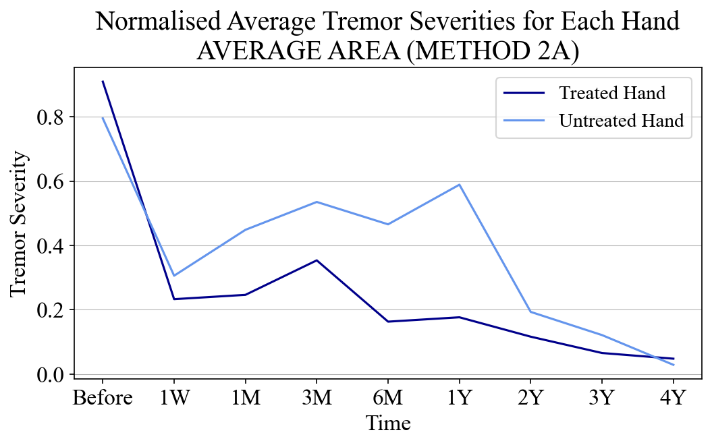


Figure 9: Average tremor severity according to average area results from method 2A

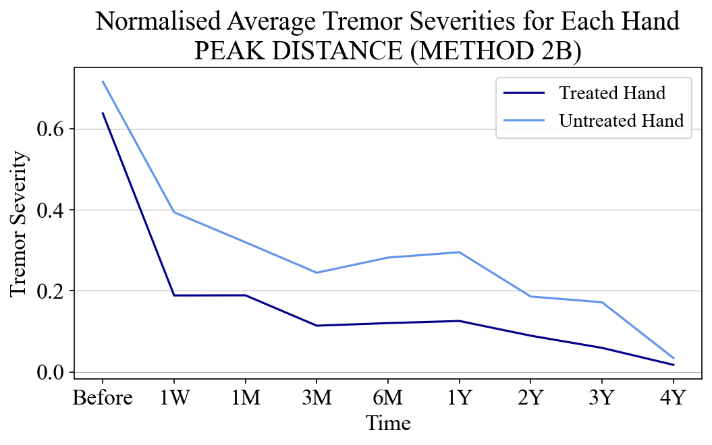


Figure 10: Average tremor severity according to peak distance result from method 2B

# Discussion

# Future improvements

# Conclusion

Table 1: Font size and styles for laboratory project papers.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Method 1 – Spiral Analysis | Method 2A – Average Area | Method 2B – Peak Distance |
| Average percentage of improved patients DOMINANT HAND | 75.8% | 63.0% | 75.7% |
| Average percentage of improved patients NON-DOMINANT HAND | 53.4 | 51.5% | 57.5% |

# equations and references

## Equations

Number the equations consecutively with equation numbers in parentheses flush with the right margin as in (1).

 (1)

Where:

 peak magnitude of current [A]  
 the per unit slip of harmonic *q* the supply frequency [rad/s]  
 phase angle for harmonic *q* [rad]

And:

 (2)

1. rect = cv2.minAreaRect(lineIMG\_coord\_array)
2. box = cv2.boxPoints(rect)
3. rect\_width = int(rect[1][0])
4. rect\_height = int(rect[1][1])
5. new\_rect = np.array([[0, rect\_height-1],
6. [0, 0],
7. [rect\_width-1, 0],
8. [rect\_width-1, rect\_height-1]],   
    dtype="float32")
10. # the perspective transformation matrix
11. M = cv2.getPerspectiveTransform(box, new\_rect)
13. Warped\_rect = cv2.warpPerspective(image, M, (rect\_width, rect\_height))

# 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 Hyperkinet 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 Neroscience 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 Neroscience 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] | jdhao, “Cropping Rotated Rectangles from Image with OpenCV,” jdhao's digital space, 23 February 2019. [Online]. Available: https://jdhao.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, “EIE Investigation: Which Hand?,” 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]. |

CODE CAN BE FOUND AT:

<https://github.com/JessWhosBack/EIE-Investigation-22G05.git>