

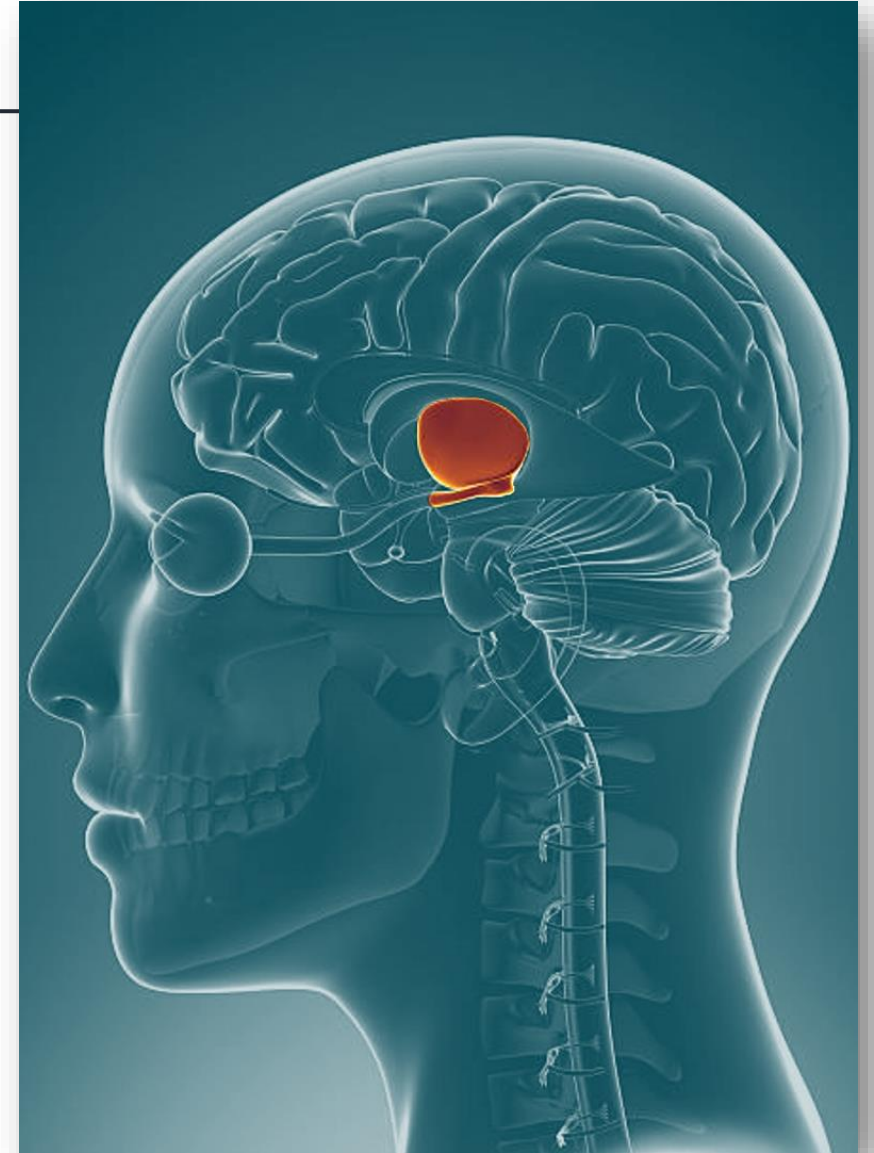
# Which Hand?

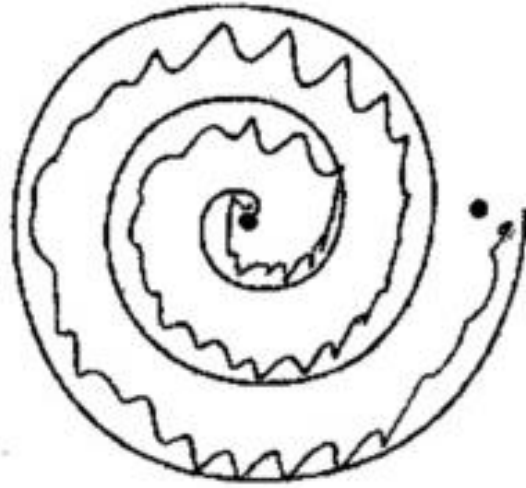
Robyn Gebbie and Jesse van der Merwe  
Project 22P59, Group 22G05



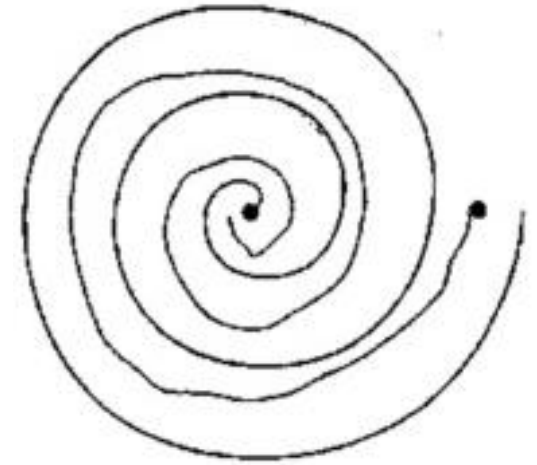
## Focused Ultrasound Treatment

- ☐ Focused Ultrasound is used to treat Parkinson's Disease and Essential Tremor
- ☐ This treatment delivers acoustic energy to one side of the patient's thalamus
- ☐ This process causes small lesions to form on this part of the brain, interrupting abnormal brain activity and causing the patient to have fewer uncontrollable movements, such as tremor
- ☐ This treatment only improves the tremor on one side of the patient's body

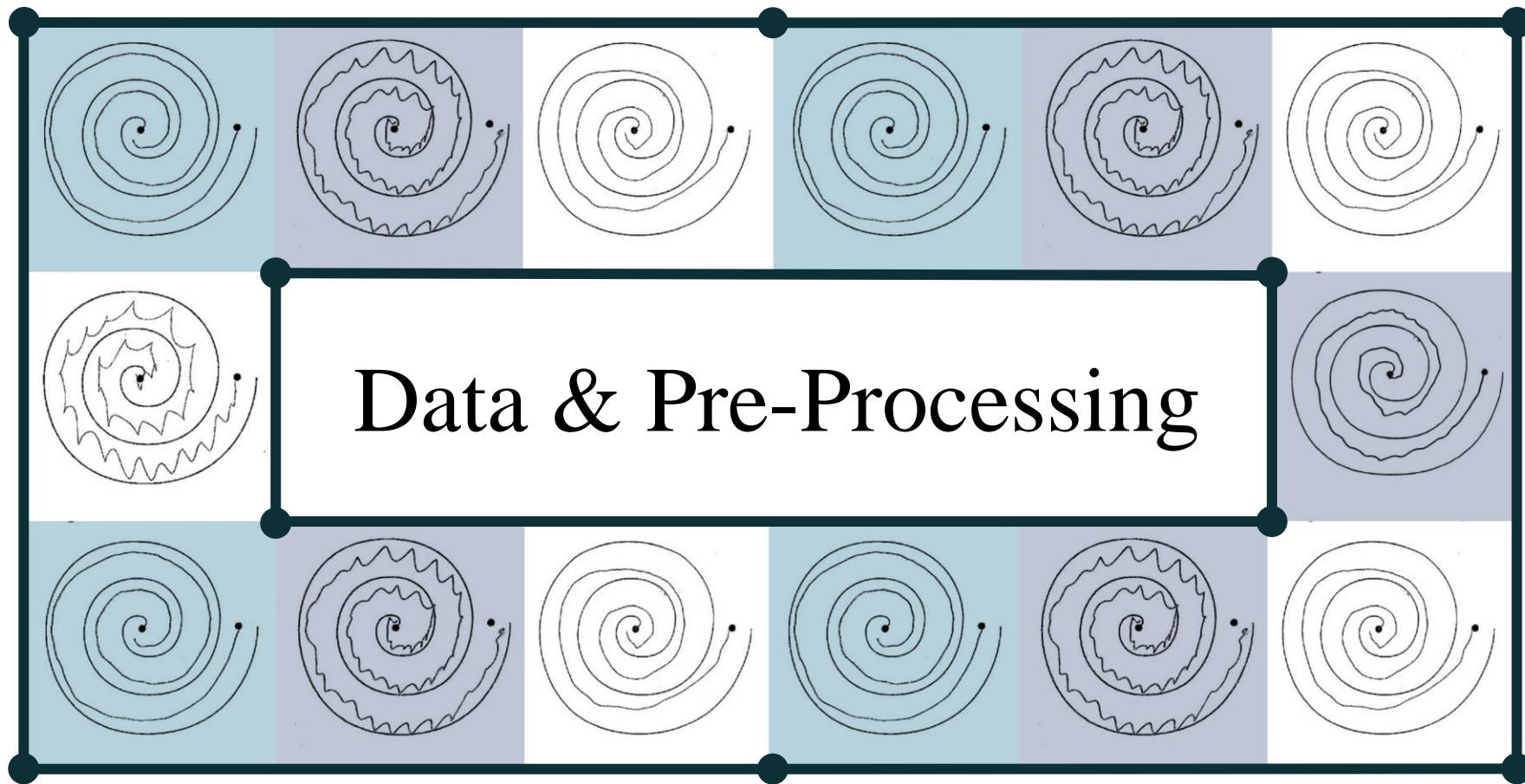




**Before Treatment**



**After Treatment**



# Data & Pre-Processing

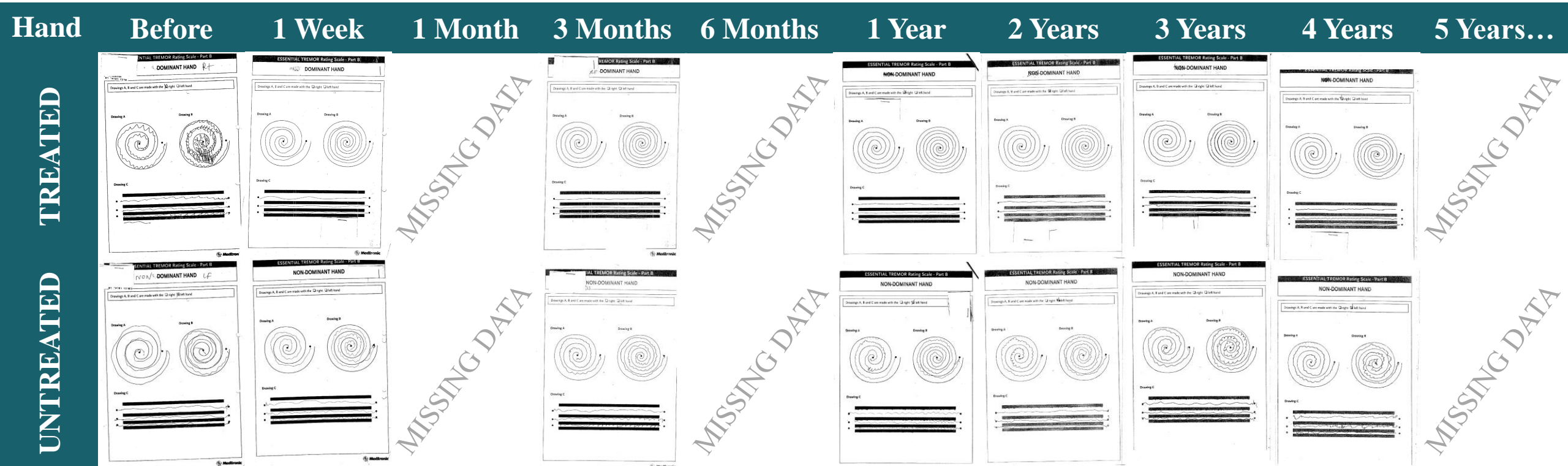


# Data: Overview



- ☐ Fully anonymised data provided by the Rambam Medical Centre, Haifa, Israel.
- ☐ Ethical clearance obtained from University of Witwatersrand HREC Committee (Medical).
- ☐ 122 patients total:
  - 34 with Parkinson's disease
  - 88 with essential tremor

# Data: Patient #5 Example



# Data: Patient #5 Example



# Data: Patient #5 Example

## BEFORE TREATMENT

Hand

Before

UNTREATED TREATED

ESSENTIAL TREMOR Rating Scale - Part B

☒ DOMINANT HAND *Rt*

Drawings A, B and C are made with the ☒ right ☐ left hand

Drawing A

Drawing B

Drawing C

Medtronic

6 Months

1 Year

2 Years

3 Years

4 Years

5 Years...

MISSING DATA

MISSING DATA

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

MISSING DATA

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

ESSENTIAL TREMOR Rating Scale - Part B

☐ NON-DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☒ left hand

Drawing A

Drawing B

Drawing C

MISSING DATA



# Data: Patient #5 Example

**BEFORE TREATMENT**

**AFTER TREATMENT**

Hand

Before

6 Months

Years

5 Years...

UNTREATED TREATED

ESSENTIAL TREMOR Rating Scale - Part B

DOMINANT HAND *Rt*

Drawings A, B and C are made with the ☒ right ☐ left hand

Drawing A

Drawing B

Drawing C

Medtronic

ESSENTIAL TREMOR Rating Scale - Part B

DOMINANT HAND

Drawings A, B and C are made with the ☐ right ☐ left hand

Drawing A

Drawing B

Drawing C

Medtronic

MISSING DATA

MISSING DATA

MISSING DATA

MISSING DATA

# Data Pre-Processing: Overview

1

## **Image Isolation Code**

Kelvin da Silva: “[Tremor Classification](#)”

Adrian Rosebrock: “[OpenCV Text Detection \(EAST text detector\)](#)”

2

## **Extraction of Line-Block C**

Extending original code to also extract line-drawing C

3

## **Improvement of Original Code**

Better logic checks  
Improved error checking

4

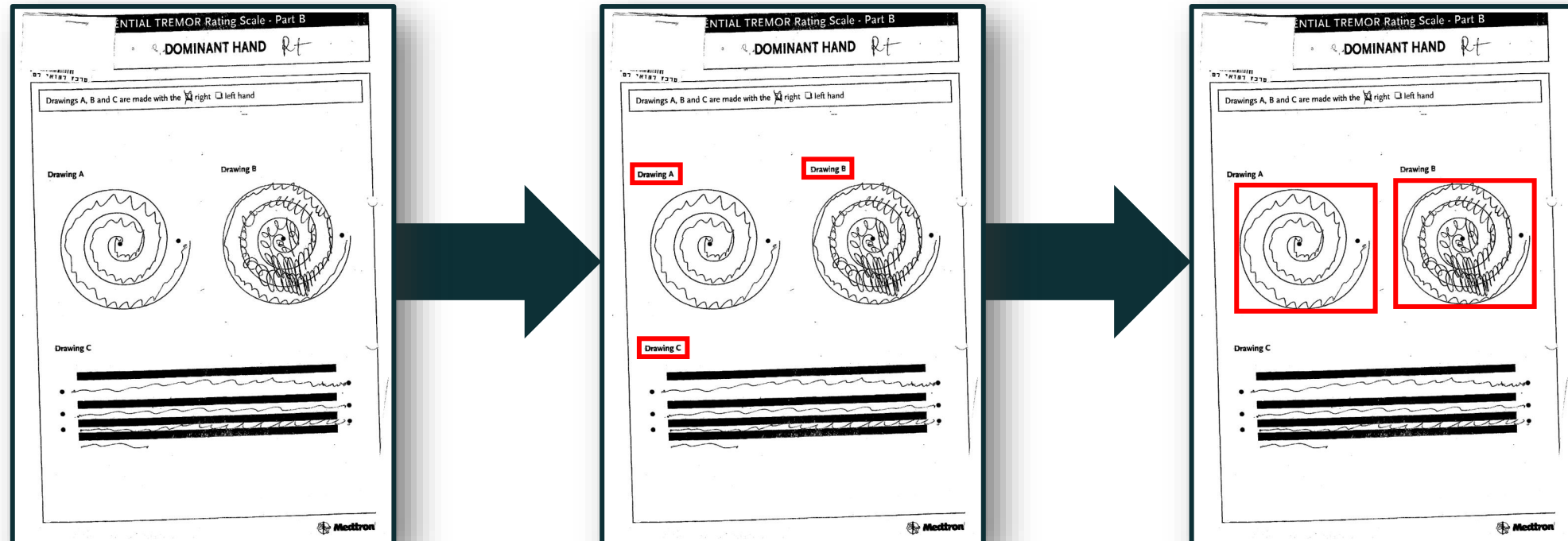
## **Extract Line-C with Rotation/Warp Correction**

Adrian Rosebrock: “[OpenCV shape detection](#)”  
jdhao: “[Cropping Rotated Rectangles from Image with OpenCV](#)”

# Data Pre-Processing: Image Isolation Code

1

Kelvin da Silva' spiral isolation code: Logic

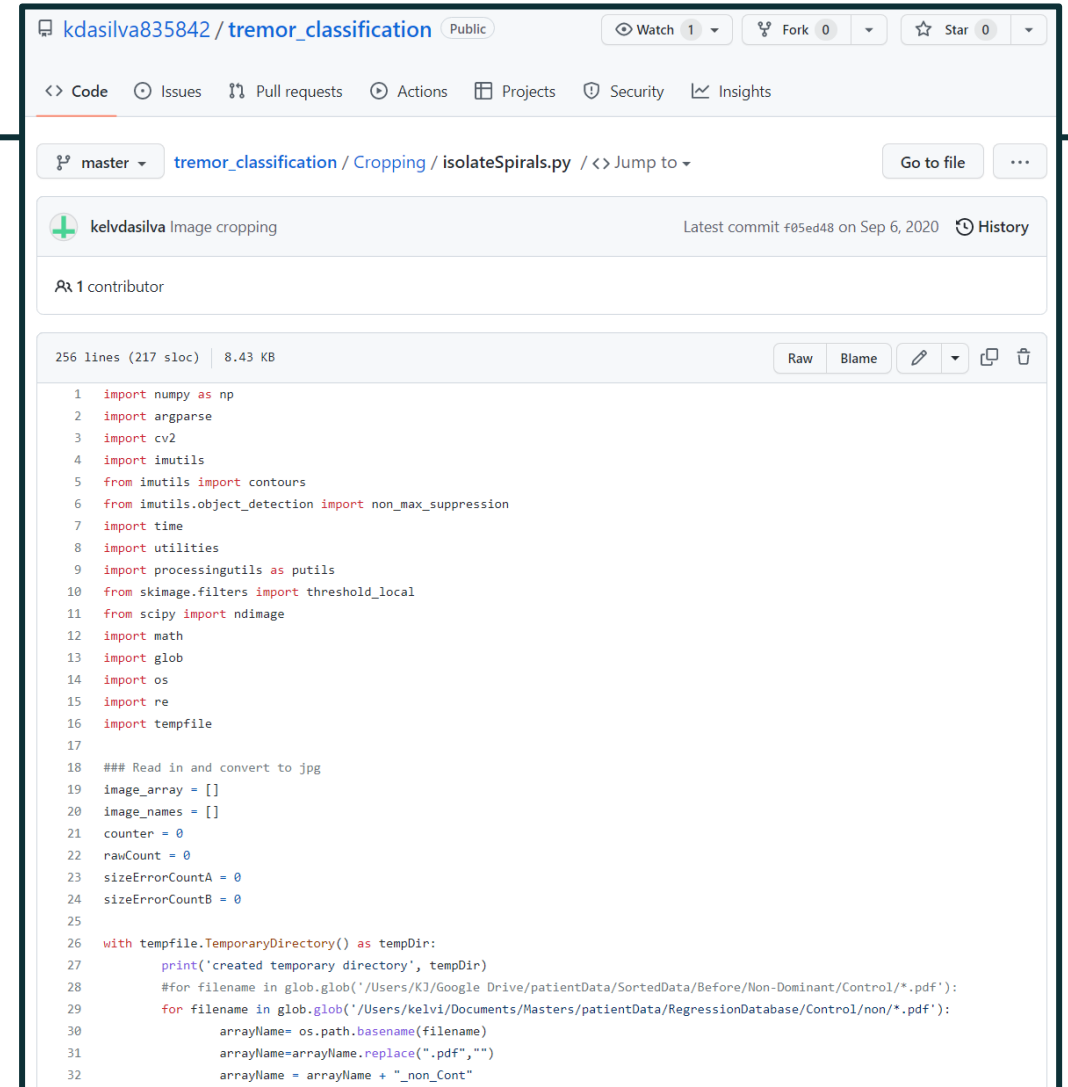


# Data Pre-Processing: Image Isolation Code

1

## Kelvin da Silva' spiral isolation code: Advantages

- Text detection is challenging – not part of project scope
- Based on article by Adrian Rosebrock:  
“[OpenCV Text Detection \(EAST text detector\)](#)”
  - **E**fficient and **A**ccurate **S**cene **T**ext detection pipeline
- Deep learning based text detection

A screenshot of a GitHub repository page for 'kdasilva835842 / tremor\_classification'. The file 'isolateSpirals.py' is selected, showing its code. The code is a Python script for image cropping and spiral isolation, using libraries like numpy, argparse, cv2, imutils, and skimage. It includes comments and variable names related to text detection and image processing.

```
1 import numpy as np
2 import argparse
3 import cv2
4 import imutils
5 from imutils import contours
6 from imutils.object_detection import non_max_suppression
7 import time
8 import utilities
9 import processingutils as putils
10 from skimage.filters import threshold_local
11 from scipy import ndimage
12 import math
13 import glob
14 import os
15 import re
16 import tempfile
17
18 ### Read in and convert to jpg
19 image_array = []
20 image_names = []
21 counter = 0
22 rawCount = 0
23 sizeErrorCountA = 0
24 sizeErrorCountB = 0
25
26 with tempfile.TemporaryDirectory() as tempDir:
27     print('created temporary directory', tempDir)
28     #for filename in glob.glob('/Users/KJ/Google Drive/patientData/SortedData/Before/Non-Dominant/Control/*.pdf'):
29     for filename in glob.glob('/Users/kelvi/Documents/Masters/patientData/RegressionDatabase/Control/non/*.pdf'):
30         arrayName= os.path.basename(filename)
31         arrayName=arrayName.replace(".pdf","")
32         arrayName = arrayName + "_non_Cont"
```

[https://github.com/kdasilva835842/tremor\\_classification/blob/master/Cropping/isolateSpirals.py](https://github.com/kdasilva835842/tremor_classification/blob/master/Cropping/isolateSpirals.py)

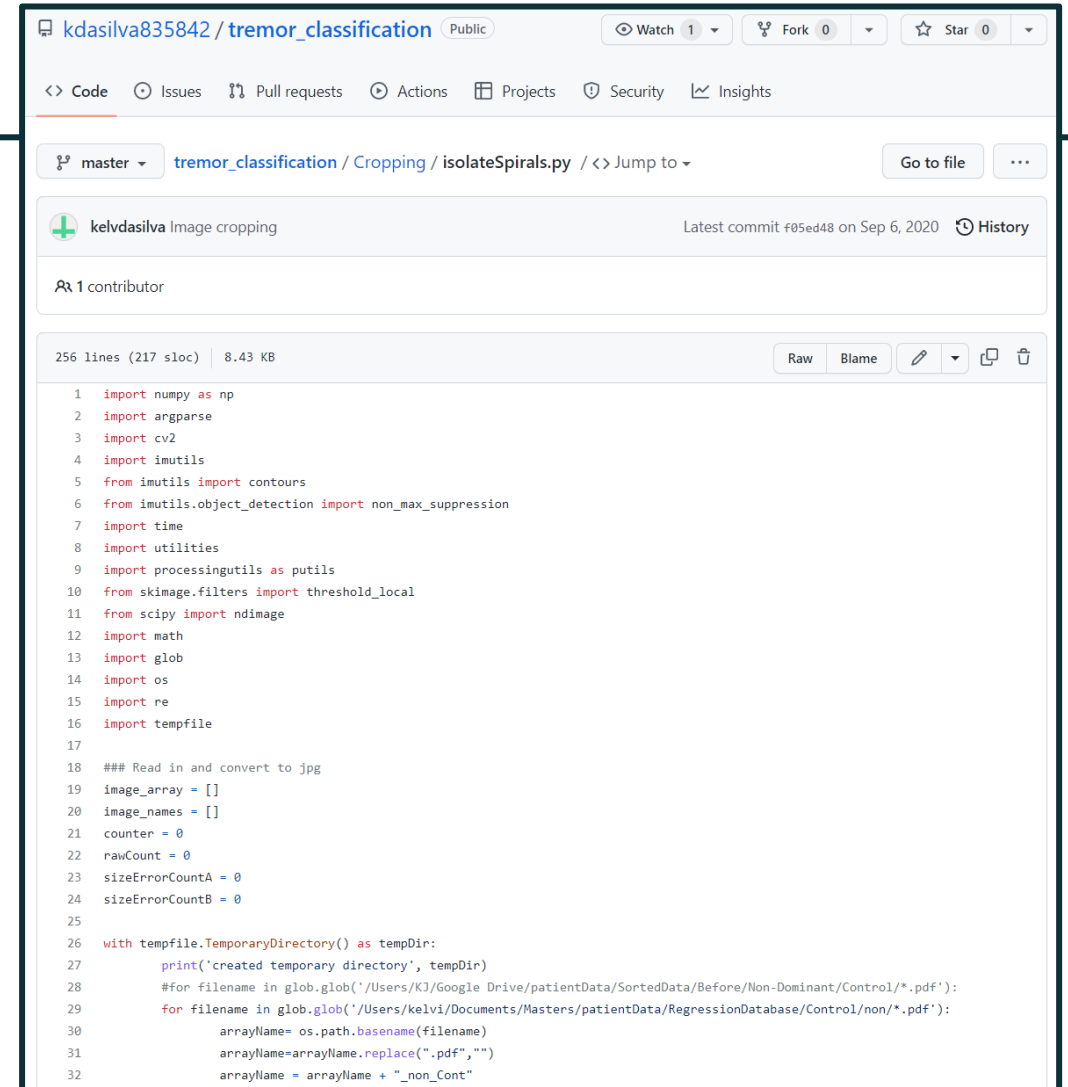


# Data Pre-Processing: Image Isolation Code

1

## Kelvin da Silva' spiral isolation code: Disadvantages

- Does not include sufficient error **correction**, only removal
- Deep learning-based text detection might be overkill considering the template format of each image
- Detects any and all text – picks up erroneous markings
- Simpler and more efficient methods available

A screenshot of a GitHub repository page for 'kdasilva835842 / tremor\_classification'. The file 'isolateSpirals.py' is selected, showing its code. The code is a Python script for image cropping and isolation. It includes imports for numpy, argparse, cv2, imutils, and other libraries. The script defines a function to read and convert images to jpg format. It also includes a main block that processes files in a specific directory structure, replacing '.pdf' extensions with '\_non\_Cont' and adding a '\_non\_Cont' suffix to the filename.

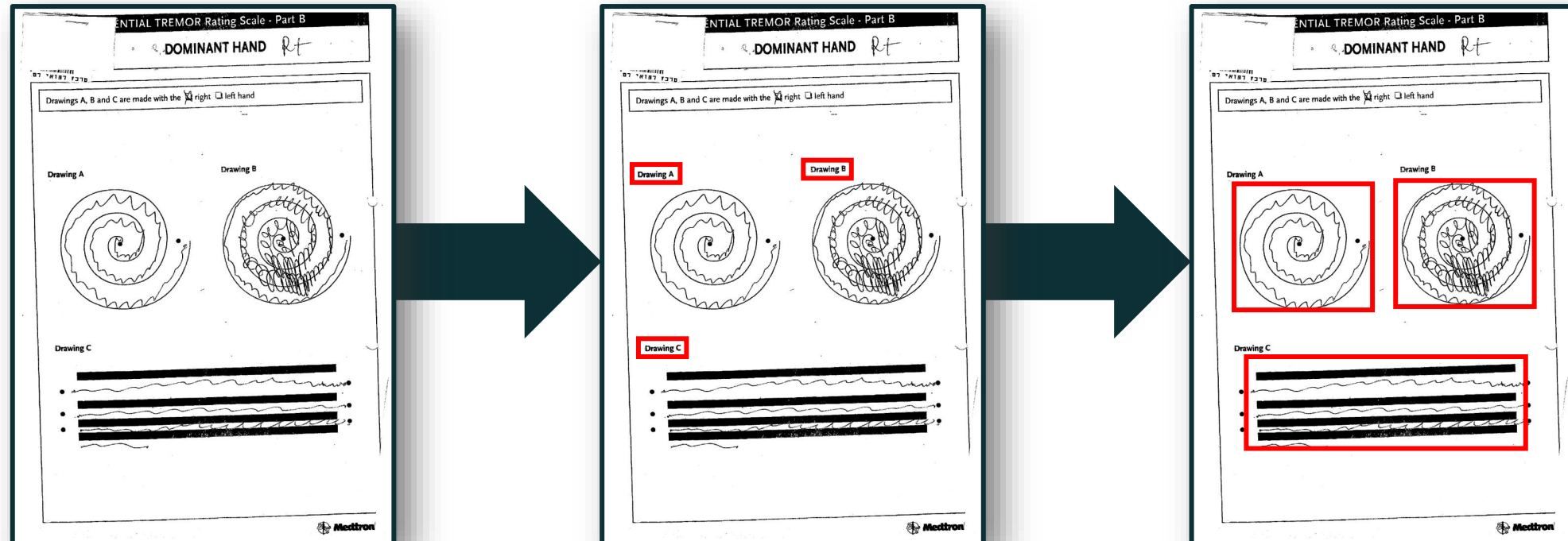
```
1 import numpy as np
2 import argparse
3 import cv2
4 import imutils
5 from imutils import contours
6 from imutils.object_detection import non_max_suppression
7 import time
8 import utilities
9 import processingutils as putils
10 from skimage.filters import threshold_local
11 from scipy import ndimage
12 import math
13 import glob
14 import os
15 import re
16 import tempfile
17
18 ### Read in and convert to jpg
19 image_array = []
20 image_names = []
21 counter = 0
22 rawCount = 0
23 sizeErrorCountA = 0
24 sizeErrorCountB = 0
25
26 with tempfile.TemporaryDirectory() as tempDir:
27     print('created temporary directory', tempDir)
28     #for filename in glob.glob('/Users/KJ/Google Drive/patientData/SortedData/Before/Non-Dominant/Control/*.pdf'):
29     for filename in glob.glob('/Users/kelvi/Documents/Masters/patientData/RegressionDatabase/Control/non/*.pdf'):
30         arrayName= os.path.basename(filename)
31         arrayName=arrayName.replace(".pdf","")
32         arrayName = arrayName + "_non_Cont"
```

[https://github.com/kdasilva835842/tremor\\_classification/blob/master/Cropping/isolateSpirals.py](https://github.com/kdasilva835842/tremor_classification/blob/master/Cropping/isolateSpirals.py)

# Data Pre-Processing: Extracting Line-Block C

2

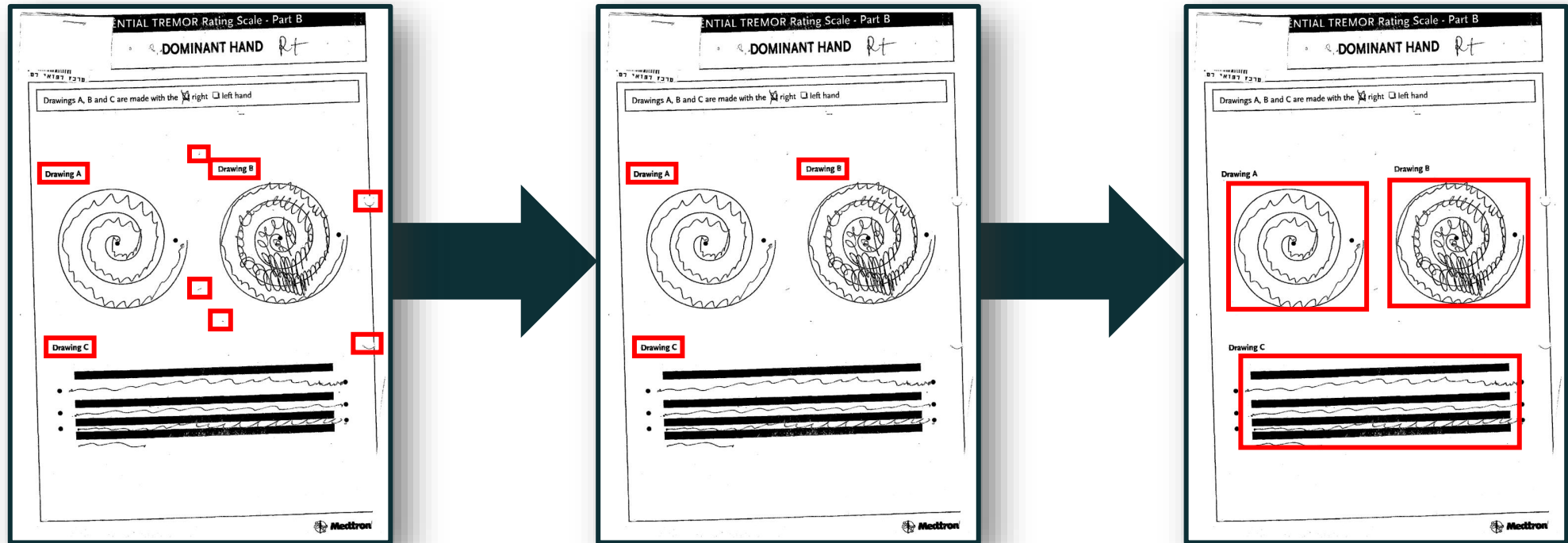
Expanding the code to extract the line-block C



# Data Pre-Processing: Improving the Code

3

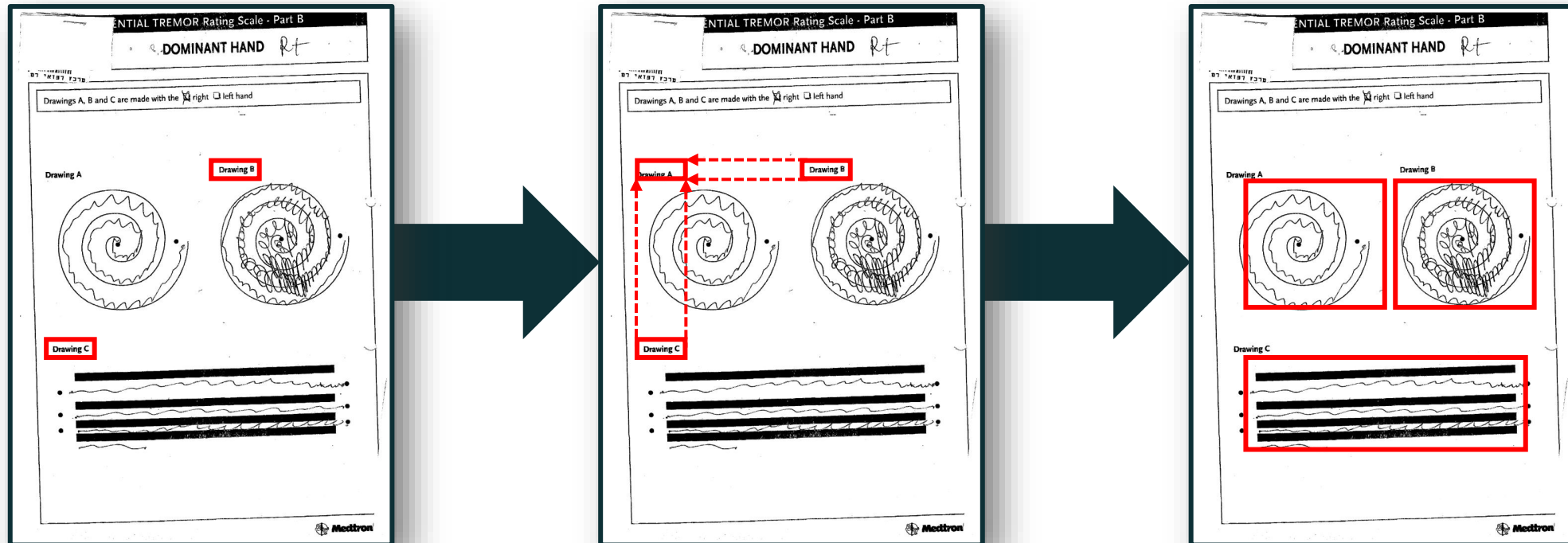
Logic checks: Remove extra detected text



# Data Pre-Processing: Improving the Code

3

Error correction: Missing “Drawing A”

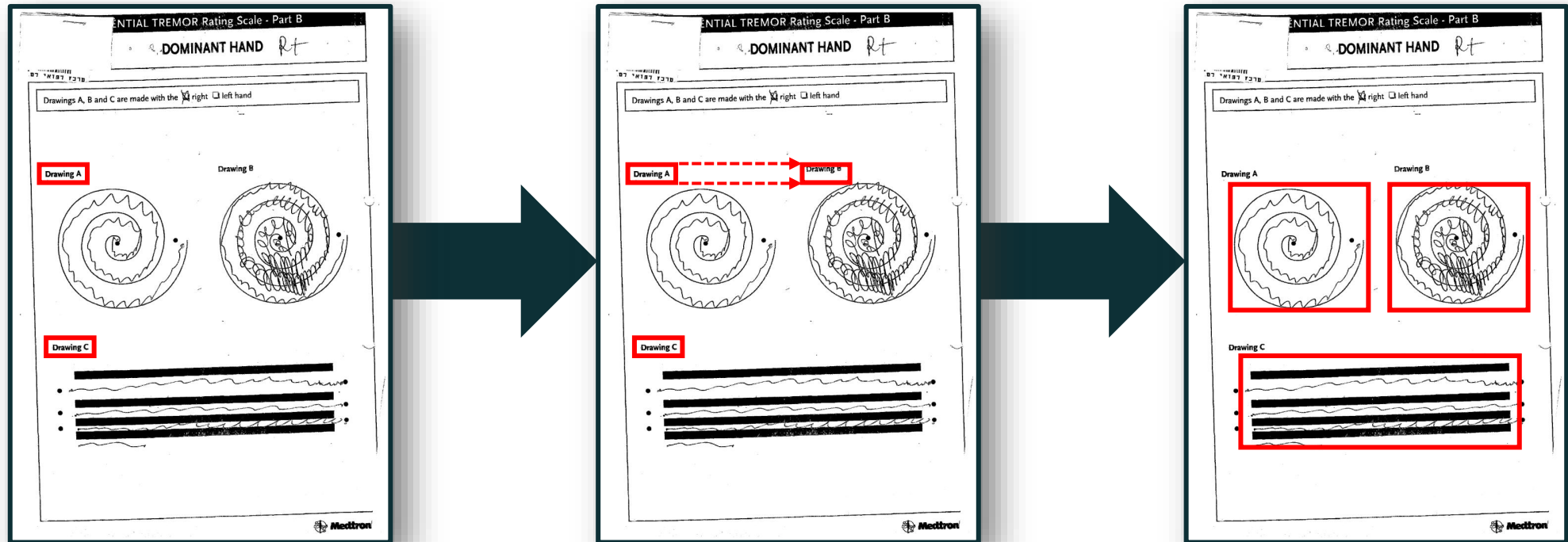




# Data Pre-Processing: Improving the Code

3

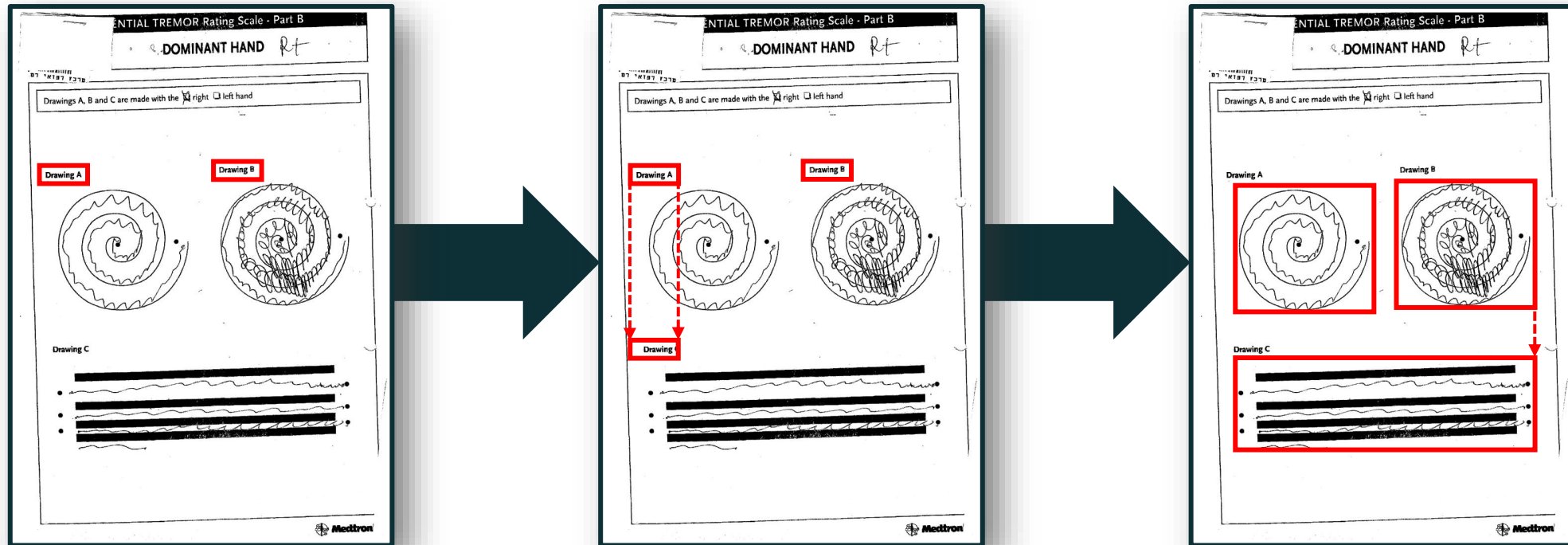
Error correction: Missing “Drawing B”



# Data Pre-Processing: Improving the Code

3

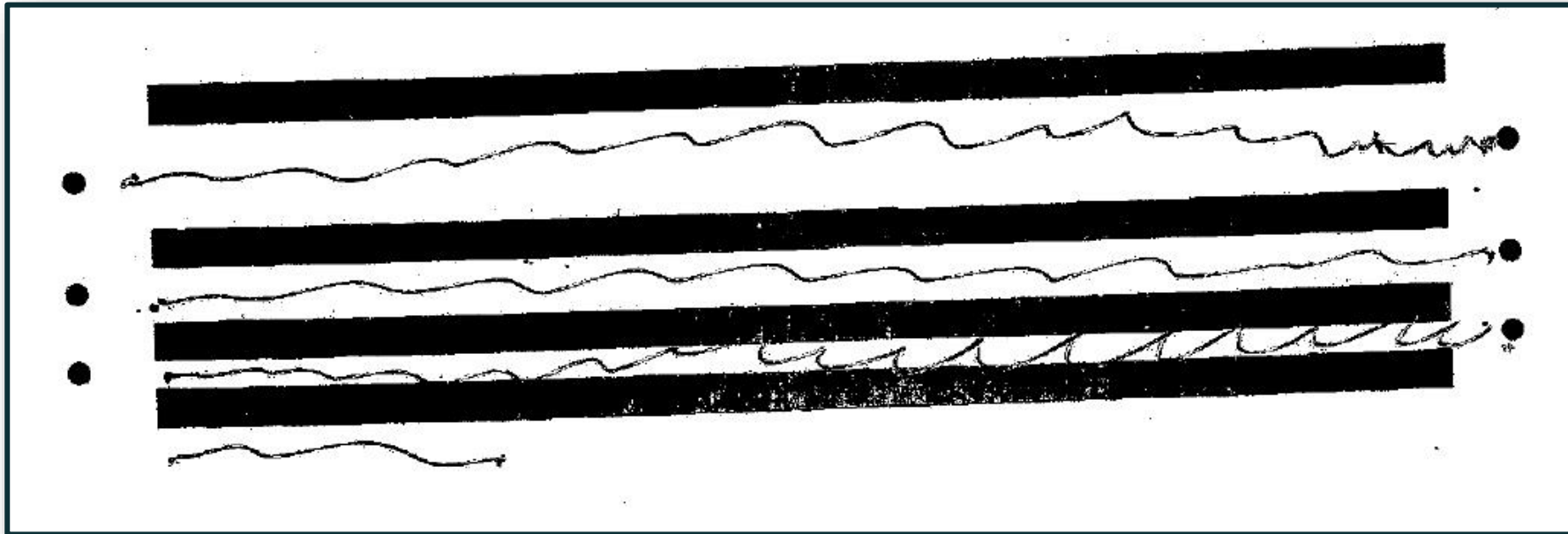
Error correction: Missing “Drawing C”



# Data Pre-Processing: Extracting Line C with Warp Correction

4

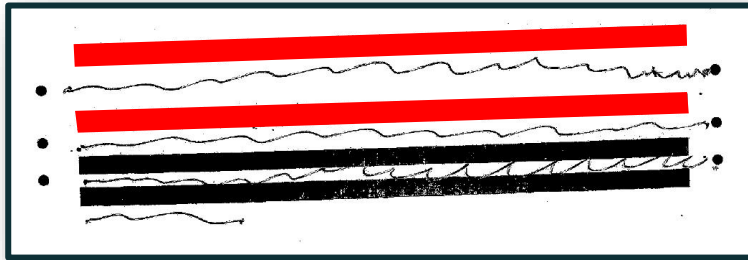
The extracted line-block C



# Data Pre-Processing: Extracting Line C with Warp Correction

4

Extracting line-drawing C with rotation/warp correction



Poor detection

## REFERENCES:

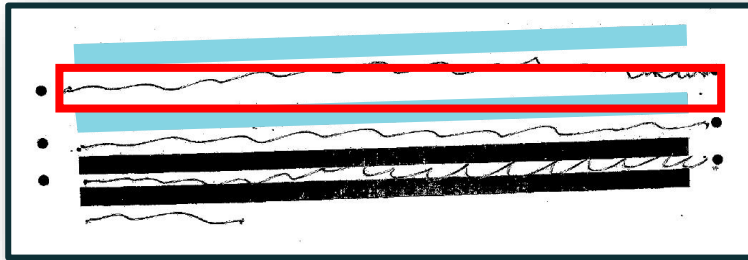
Adrian Rosebrock: “OpenCV shape detection” – <https://pyimagesearch.com/2016/02/08/opencv-shape-detection/>



# Data Pre-Processing: Extracting Line C with Warp Correction

4

Extracting line-drawing C with rotation/warp correction



Poor detection

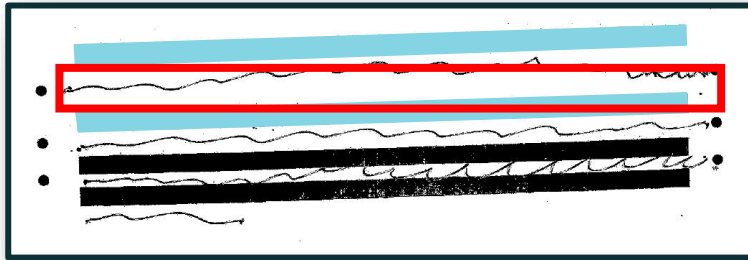
## REFERENCES:

Adrian Rosebrock: “OpenCV shape detection” – <https://pyimagesearch.com/2016/02/08/opencv-shape-detection/>

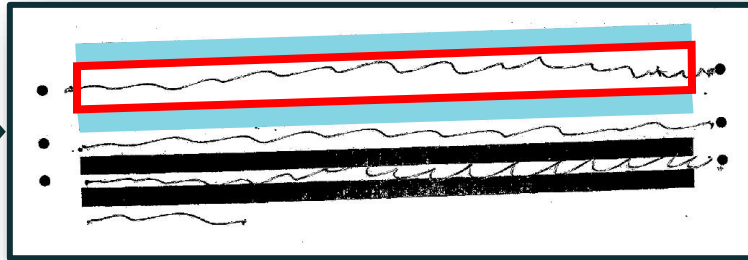
# Data Pre-Processing: Extracting Line C with Warp Correction

4

Extracting line-drawing C with rotation/warp correction



Poor detection

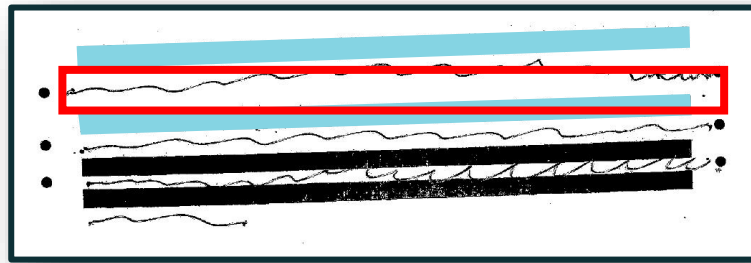


More accurate detection

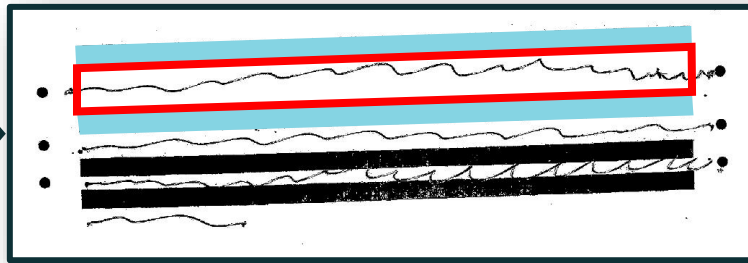
# Data Pre-Processing: Extracting Line C with Warp Correction

4

Extracting line-drawing C with rotation/warp correction



Poor detection



More accurate detection



Corrected image

## REFERENCES:

jdhao: "Cropping Rotated Rectangles from Image with OpenCV" – [https://jdhao.github.io/2019/02/23/crop\\_rotated\\_rectangle\\_opencv/](https://jdhao.github.io/2019/02/23/crop_rotated_rectangle_opencv/)

# Data Pre-Processing: Overview

1

## **Image Isolation Code**

Kelvin da Silva: “[Tremor Classification](#)”

Adrian Rosebrock: “[OpenCV Text Detection \(EAST text detector\)](#)”

2

## **Extraction of Line-Block C**

Extending original code to also extract line-drawing C

3

## **Improvement of Original Code**

Better logic checks  
Improved error checking

4

## **Extract Line-C with Rotation/Warp Correction**

Adrian Rosebrock: “[OpenCV shape detection](#)”  
jdhao: “[Cropping Rotated Rectangles from Image with OpenCV](#)”



# Data Pre-Processing: Overview

1

## Image Isolation Code

Kelvin da Silva: "[Tremor Classification](#)"

Adrian Rosebrock: "[OpenCV Text Detection \(EAST text detector\)](#)"

2

## Extraction of Line-Block C

Extending original code to also extract line-drawing C

3

## Improvement of Original Code

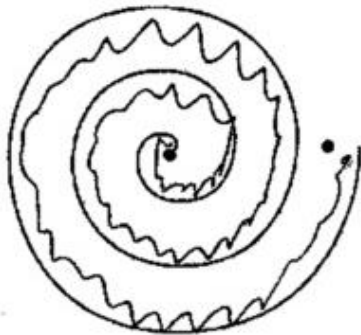
Better logic checks  
Improved error checking

4

## Extract Line-C with Rotation/Warp Correction

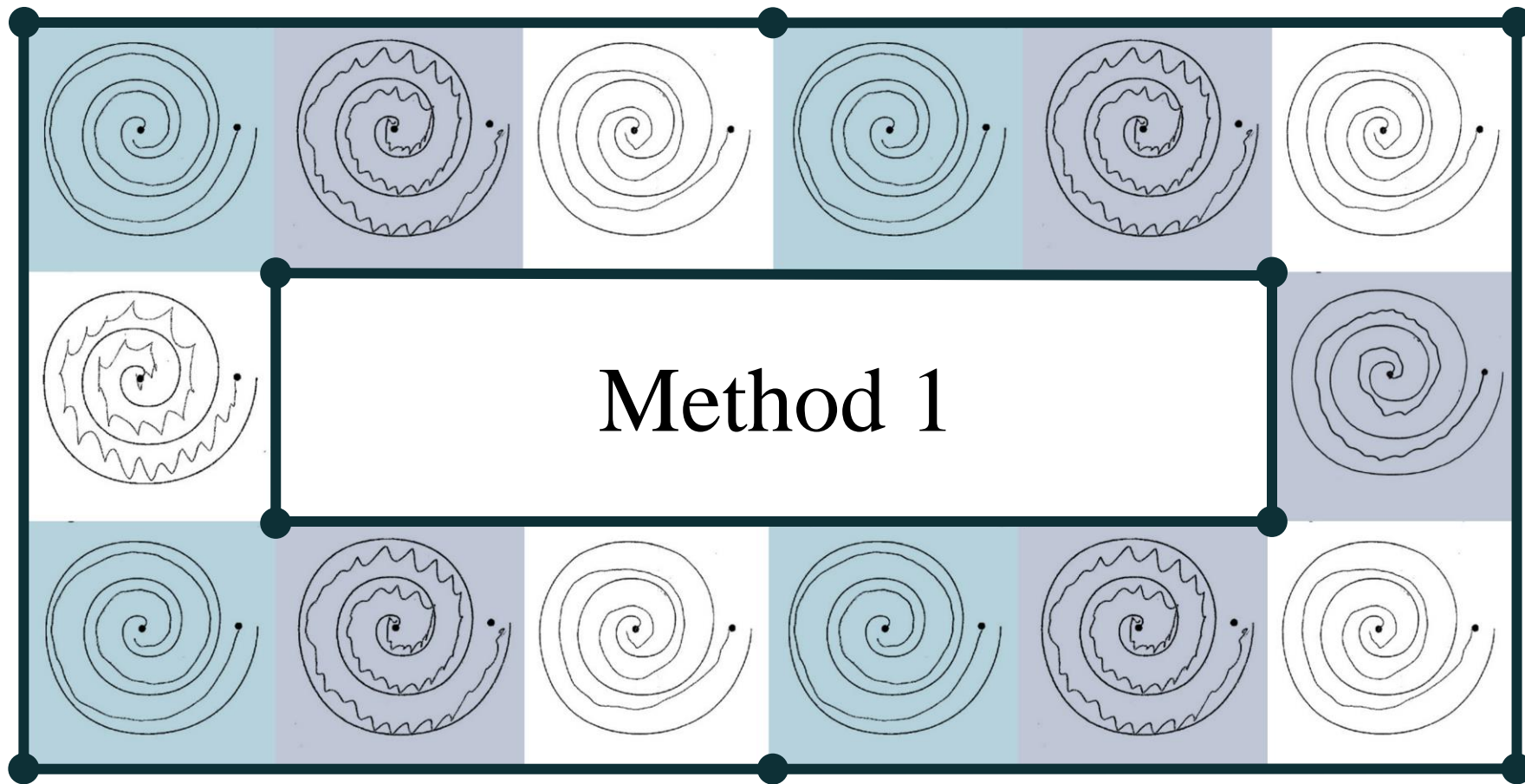
Adrian Rosebrock: "[OpenCV shape detection](#)"  
jdhao: "[Cropping Rotated Rectangles from Image with OpenCV](#)"

Method 1:  
Spiral A



Method 2:  
Line-drawing C





# Spiral Tremor Quantification

## Preparation

Images were resized.  
x and y coordinates of  
the spiral pixels were  
extracted.

## Sobel Edge Detection

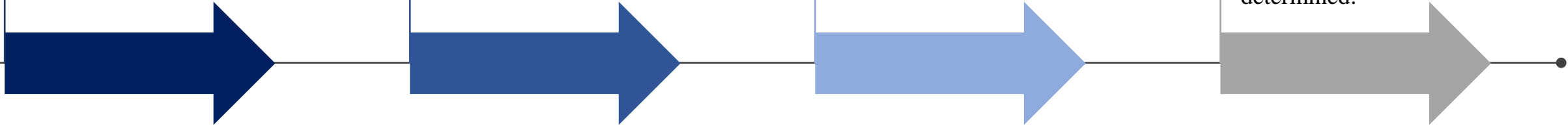
The angles made by the  
gradients of the spiral  
edges were determined

## Pixel Angles

The angle that each pixel  
made with a horizontal  
line through the spiral  
centre was determined

## Relative Orientations

The angles made by the  
gradients of the spiral  
edges relative to the  
centre of the spiral were  
determined.



# Spiral Tremor Quantification

## **Preparation**

Images were resized.  
x and y coordinates of  
the spiral pixels were  
extracted.

## **Sobel Edge Detection**

The angles made by the  
gradients of the spiral  
edges were determined

## **Pixel Angles**

The angle that each pixel  
made with a horizontal  
line through the spiral  
centre was determined

## **Relative Orientations**

The angles made by the  
gradients of the spiral  
edges relative to the  
centre of the spiral were  
determined.



# Spiral Tremor Quantification

## Preparation

Images were resized.  
x and y coordinates of  
the spiral pixels were  
extracted.

## Sobel Edge Detection

The angles made by the  
gradients of the spiral  
edges were determined

## Pixel Angles

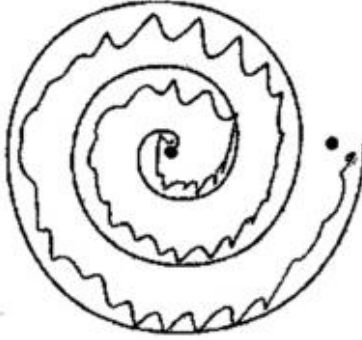
The angle that each pixel  
made with a horizontal  
line through the spiral  
centre was determined

## Relative Orientations

The angles made by the  
gradients of the spiral  
edges relative to the  
centre of the spiral were  
determined.

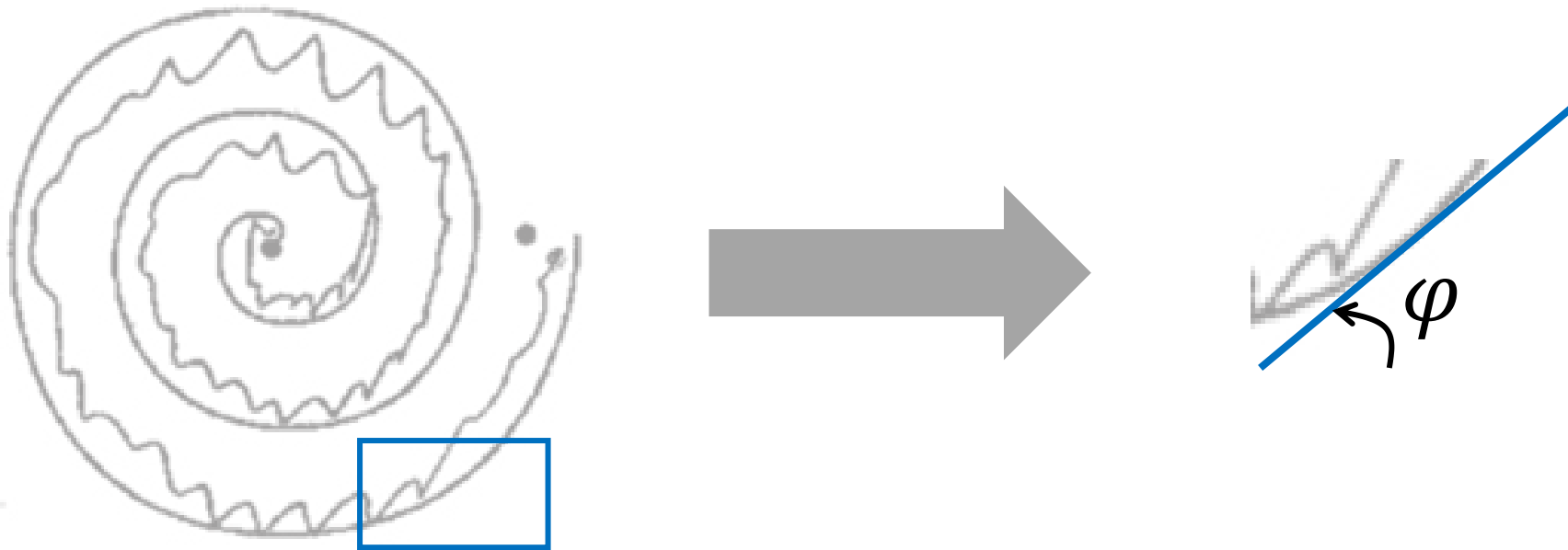


# Vertical Sobel Filter Convolved with Image Array

$$G_x = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ \hline -5 & -4 & -3 & -2 & 0 & 2 & 3 & 4 & 5 \\ \hline -6 & -5 & -4 & -3 & 0 & 3 & 4 & 5 & 6 \\ \hline -7 & -6 & -5 & -4 & 0 & 4 & 5 & 6 & 7 \\ \hline -8 & -7 & -6 & -5 & 0 & 5 & 6 & 7 & 8 \\ \hline -7 & -6 & -5 & -4 & 0 & 4 & 5 & 6 & 7 \\ \hline -6 & -5 & -4 & -3 & 0 & 3 & 4 & 5 & 6 \\ \hline -5 & -4 & -3 & -2 & 0 & 2 & 3 & 4 & 5 \\ \hline -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ \hline \end{array} *$$




# Calculation of the Orientation of an Edge Gradient



# Spiral Tremor Quantification

## Preparation

Images were resized.  
x and y coordinates of  
the spiral pixels were  
extracted.

## Sobel Edge Detection

The angles made by the  
gradients of the spiral  
edges were determined

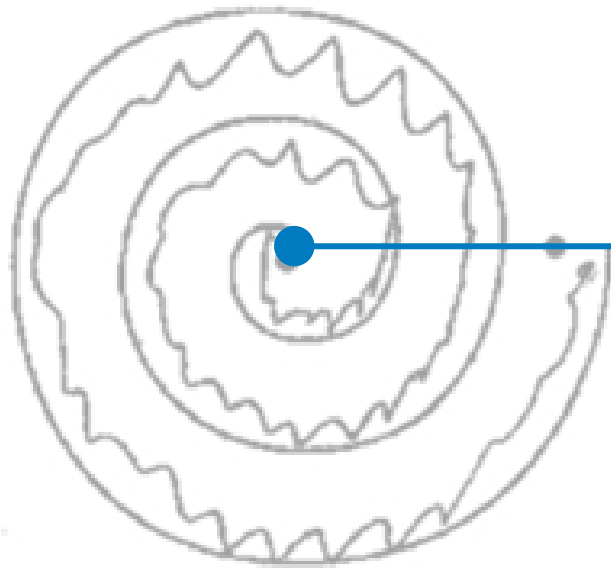
## Pixel Angles

The angle that each pixel  
made with a horizontal  
line through the spiral  
centre was determined

## Relative Orientations

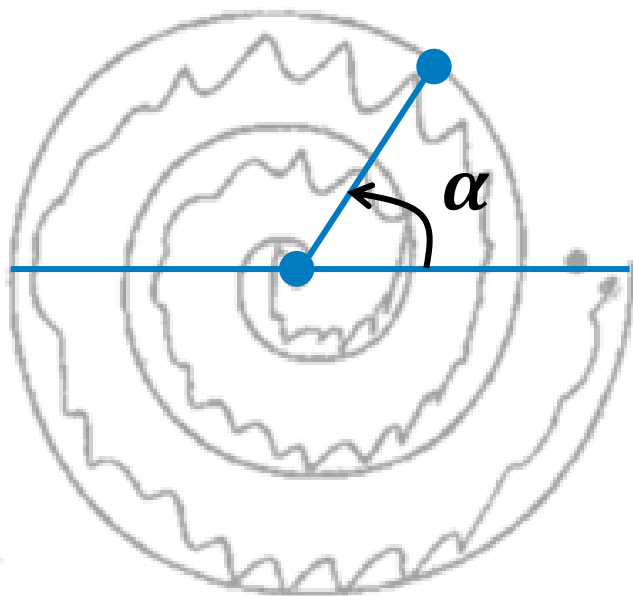
The angles made by the  
gradients of the spiral  
edges relative to the  
centre of the spiral were  
determined.





**Finding the Spiral Centre**

# Finding the Angle a Pixel Makes with the Spiral Centre



# Spiral Tremor Quantification

## Preparation

Images were resized.  
x and y coordinates of  
the spiral pixels were  
extracted.

## Sobel Edge Detection

The angles made by the  
gradients of the spiral  
edges were determined

## Pixel Angles

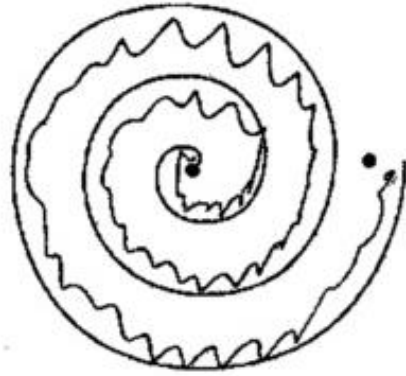
The angle that each pixel  
made with a horizontal  
line through the spiral  
centre was determined

## Relative Orientations

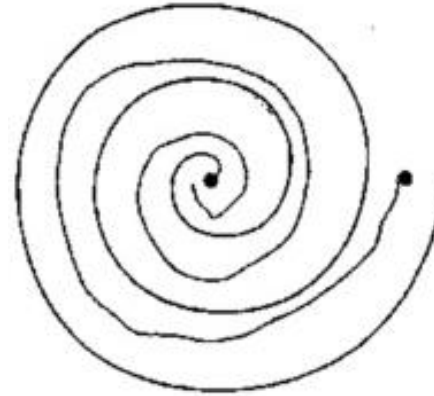
The angles made by the  
gradients of the spiral  
edges relative to the  
centre of the spiral were  
determined.



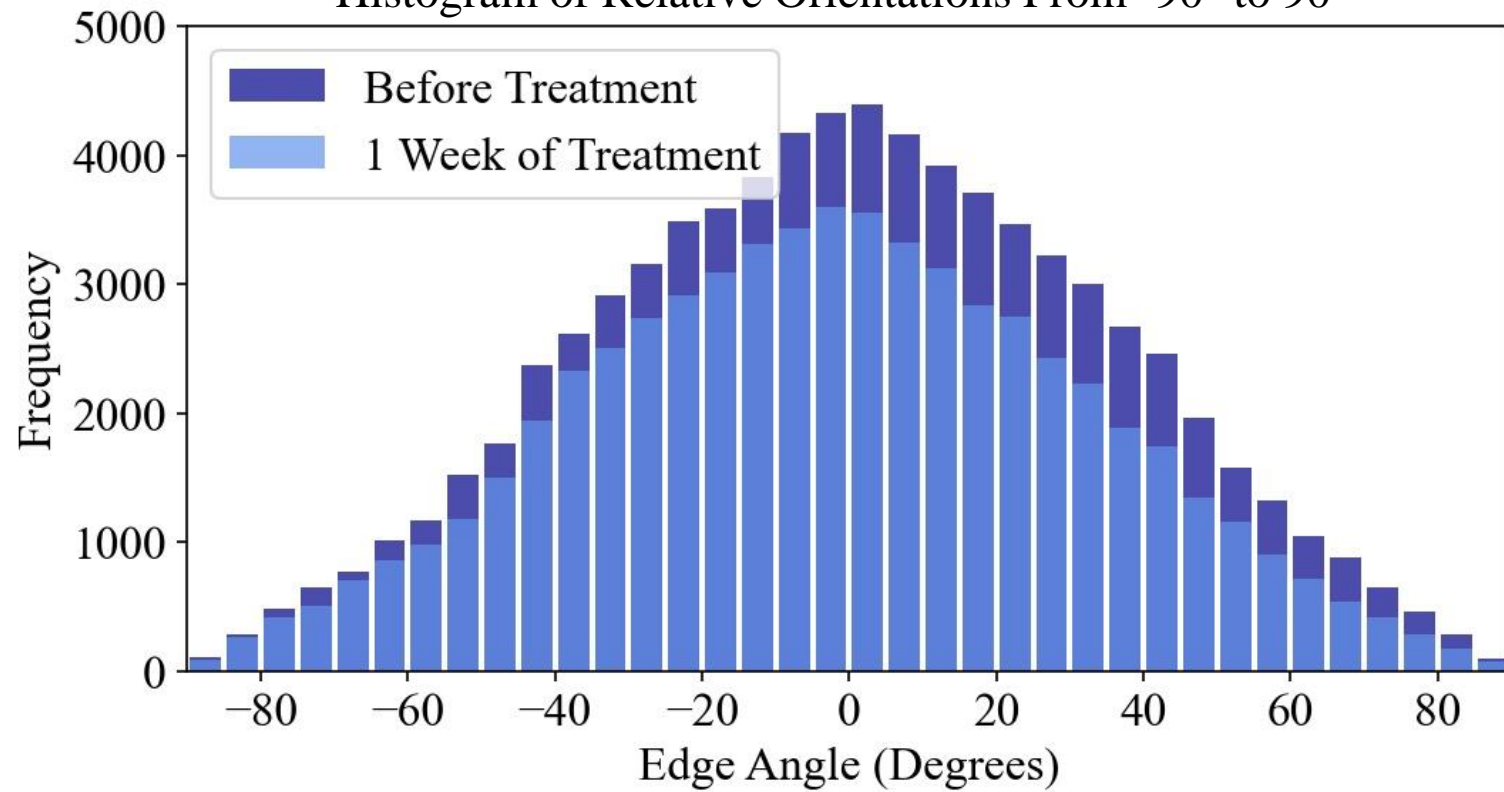
Before Treatment



After 1 Week of Treatment



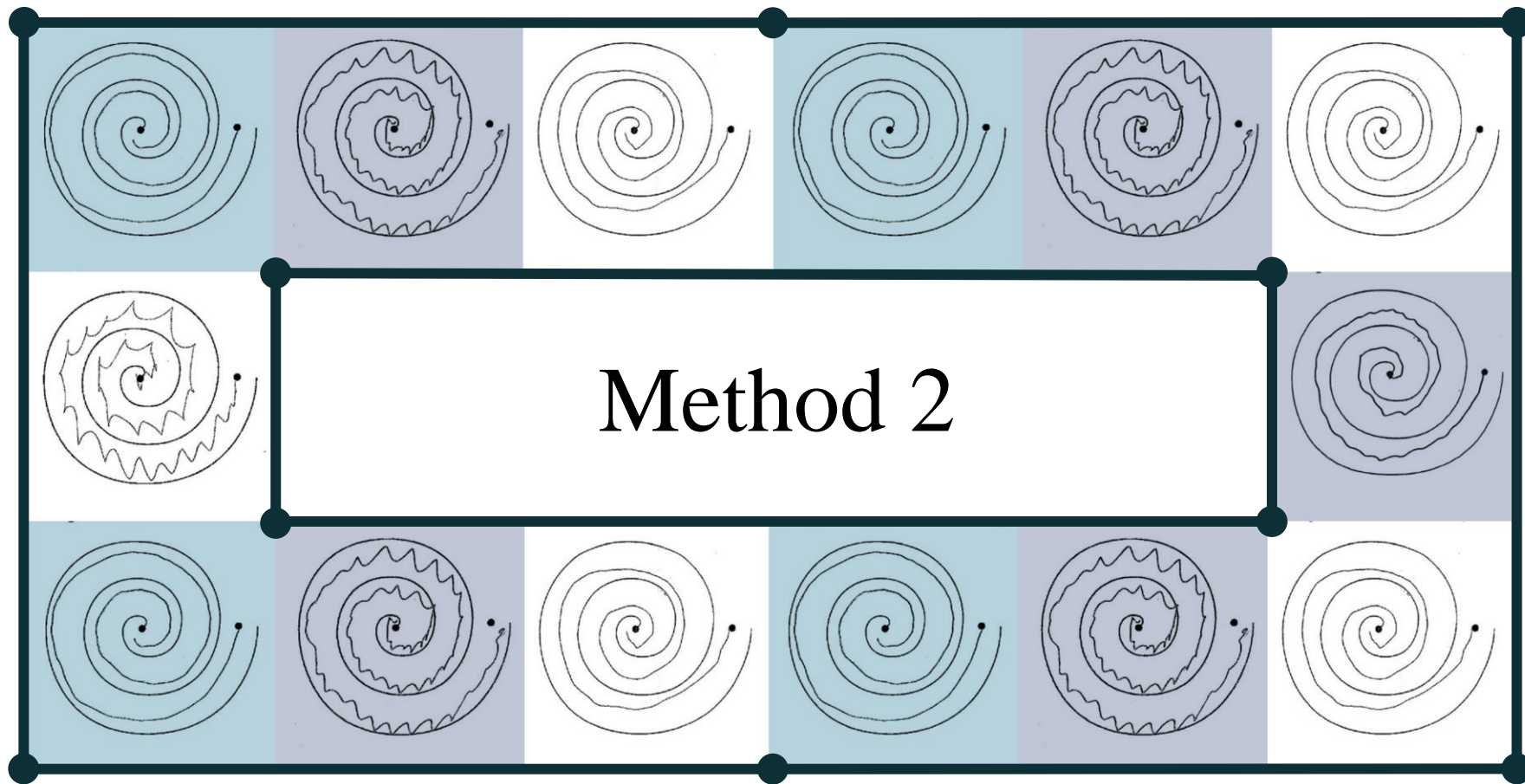
Histogram of Relative Orientations From  $-90^\circ$  to  $90^\circ$





# Generated Dataset – Normalised Standard Deviations of Relative Orientations

Treated Hand										Untreated Hand									
Patient	Before	1 Week	1 Month	3 Months	6 Months	1 Year	2 Years	3 Years	4 Years	Patient	Before	1 Week	1 Month	3 Months	6 Months	1 Year	2 Years	3 Years	4 Years
1	0,69804	0	0,23772	0	0	0	0,29025	0	0	1	0,40887	0	0,36628	0	0	0,25913	0	0,50348	0
2	0	0,28156	0	0	0,43467	0,36525	0,50393	0	0,42806	2	0	0,22898	0	0	0,42819	0	0,61826	0	0,5444
3	0	0	0	0	0,25260	0,38433	0,33108	0	0,30270	3	0,59158	0	0	0	0,34025	0,45434	0,20970	0,27805	0,46490
4	0,58526	0	0,14992	0	0	0,29416	0	0,25833	0,27149	4	0,27908	0	0,38202	0	0	0,34371	0,37276	0,45124	0,25776
5	0,67957	0,36255	0	0,34376	0	0,40972	0,47997	0,39660	0,43313	5	0,28789	0,50874	0	0,41490	0	0,57673	0,54196	0,48261	0,65204
6	0,71281	0	0,30393	0,49919	0	0,47245	0,50885	0	0,48108	6	0,46463	0	0,27557	0,48461	0	0	0,47006	0	0,43042
7	0	0	0,27176	0	0,21851	0,33371	0,46089	0,25351	0,31964	7	0	0	0,38798	0	0,31109	0,42080	0,50225	0,16933	0,33612
8	0,52692	0	0	0	0	0,51366	0	0	0,33659	8	0,28709	0	0	0	0	0,47970	0	0	0,11800
9	0	0,49157	0,47242	0	0,76578	0	0	0	0	9	0	0	0	0	0	0,87702	0	0	0
10	0,53985	0,60501	0,61649	0	0	0	0,66800	0,66869	0	10	0,71581	0,51664	0,54556	0,36939	0	0	0,44500	0,35811	0
11	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0
12	0,81357	0	0,65765	0	0,52252	0,46261	0	0	0	12	0,81730	0	0,56598	0	0,66104	0	0	0	0
13	0,64515	0	0,11996	0,22116	0	0	0,28093	0	0	13	0,43488	0	0	0,49257	0	0,58450	0	0,48633	0
14	0,58010	0	0	0,54242	0	0	0,49908	0,50493	0	14	0,74771	0	0	0	0	0,55042	0,76094	0	0
15	0,64107	0	0	0,38314	0	0	0	0,39679	0	15	0,56392	0	0	0,39887	0	0	0	0,45953	0



# Method 2: Overview

1

**Reduce noise**

PIL Image Python package

2

**Convert to array of  
coordinates**

NumPy Python package

3

**RFFT to remove  
unwanted frequencies**

SciPy Python package

4

**Apply methods 2A  
and 2B**

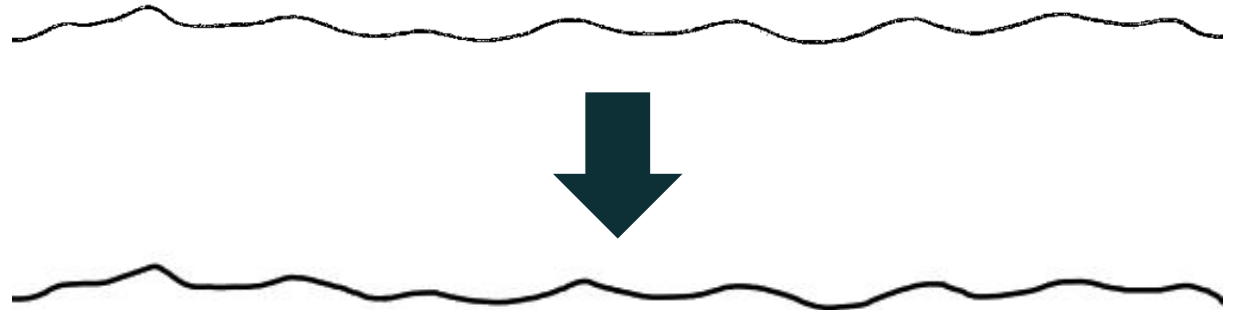
NumPy and SciPy  
Python packages

# Method 2: Reduce Noise

1

Reduce noise by:

- Converting to grayscale
- Mapping all pixels to either black or white



```
im = Image.open(filename).convert('L') # convert image to grayscale
res = im.point((lambda p: 256 if p>=200 else 0)) # convert each pixel into either black or white
```

<https://pillow.readthedocs.io/en/stable/reference/Image.html>

# Method 2: Convert to Array

2

- Use NumPy's *argwhere()* to detect indices of all black pixels and save to multidimensional array
- Extract the x- and y-coordinates
- Sort array by y-coordinates
- Shift y-values down to centre line around the x-axis

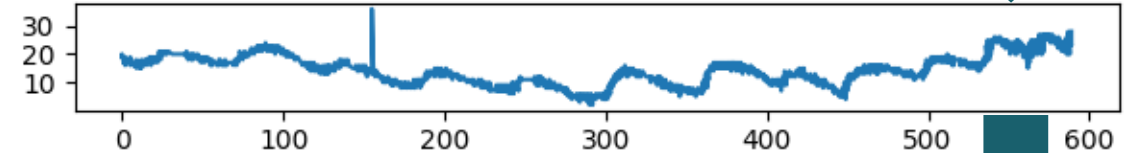
```
coordinates = np.argwhere(image < 0.9)  
x_tuple, y_tuple, z_tuple = zip(*coordinates)
```

<https://numpy.org/doc/stable/index.html>

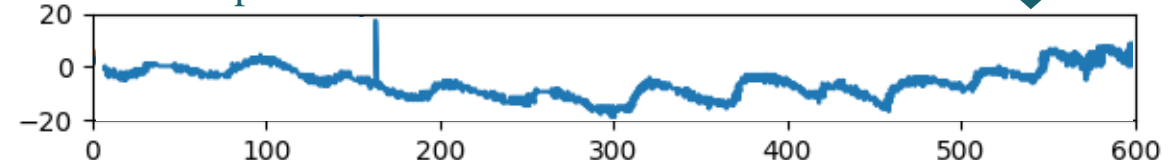
Original line-drawing C



Pixels detected



Shifted pixels

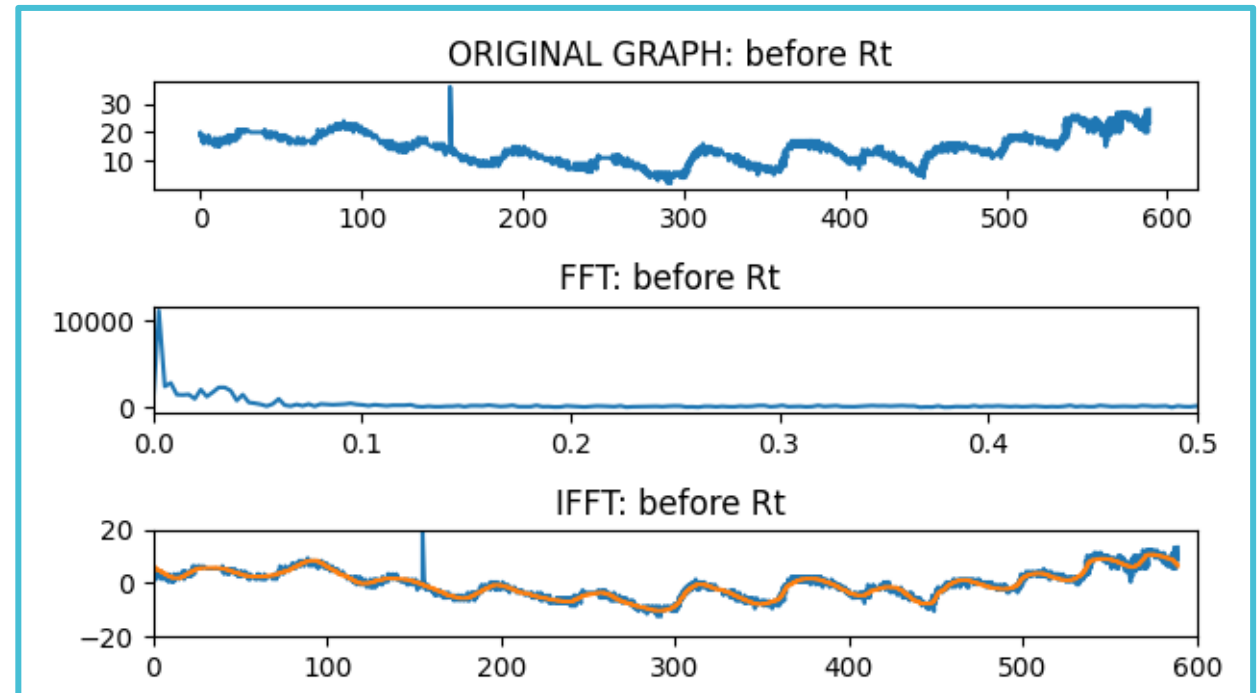


# Method 2: RFFT

3

Using SciPy.fft package:

- Transform to frequency domain ( $RFFT()$ )
- Remove unwanted high frequencies
- Inverse transform ( $IRFFT()$ )



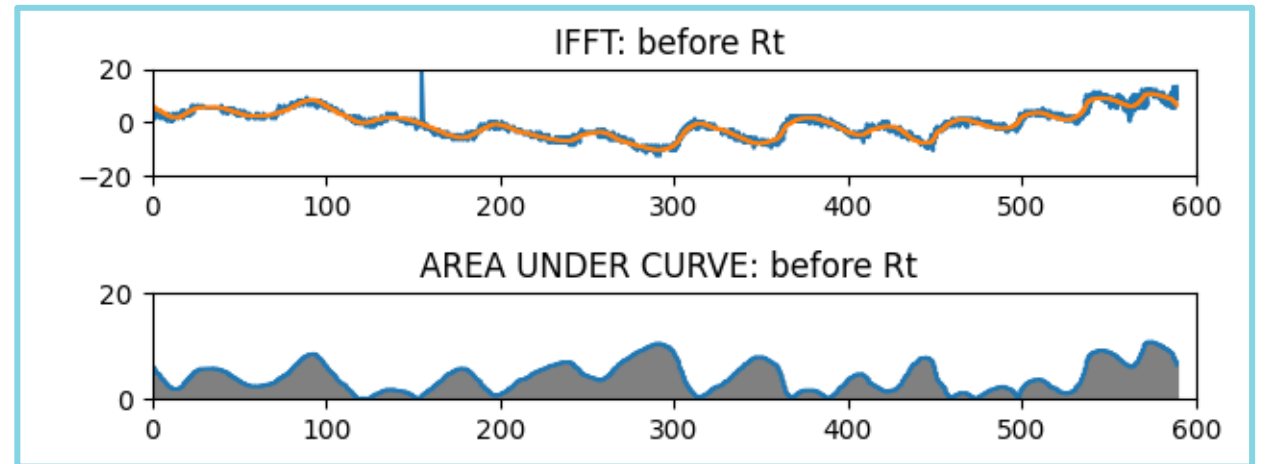


# Method 2: Applying Method 2A

4

## Method 2A: Average area under line

- Take absolute value of array
- Use NumPy *trapz()* function:
  - Calculate average area over array

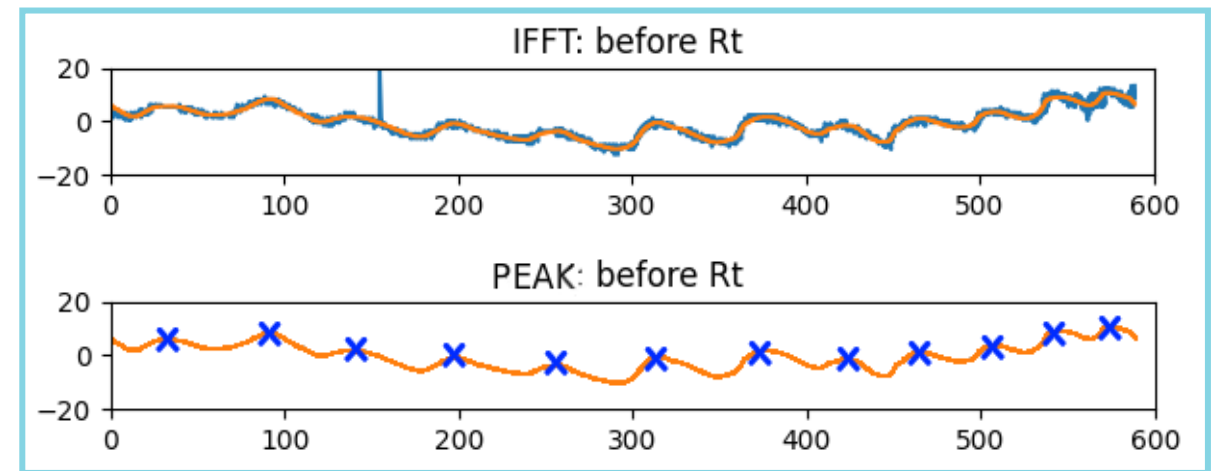


# Method 2: Applying Method 2B

4

## Method 2B: Average peaks $\times$ peak-trough distance

- SciPy *find\_peaks()* returns coordinates of peaks
- Loop through array and calculate distance between all peaks and adjacent troughs
- Multiply these two values

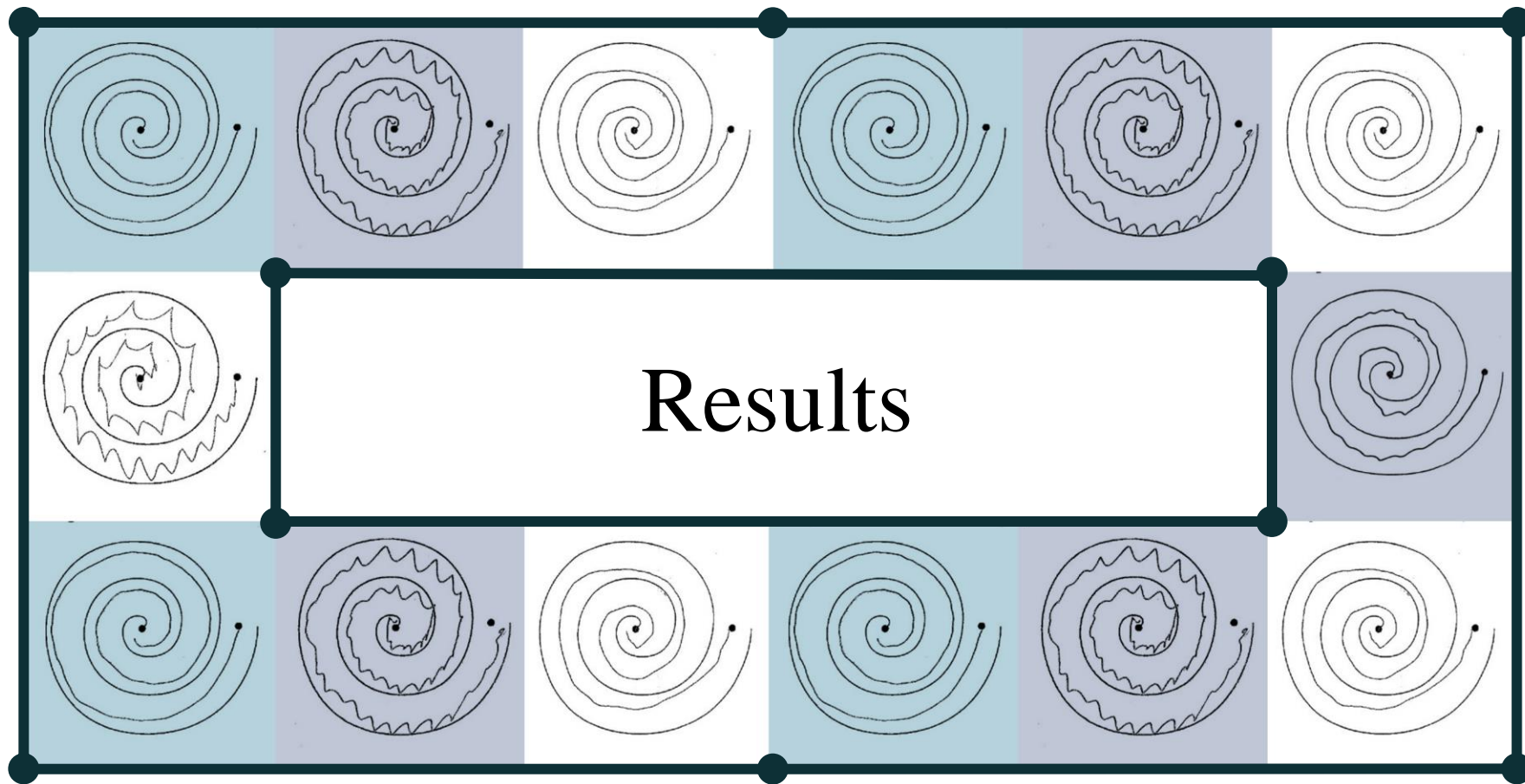


# Method 2: Results

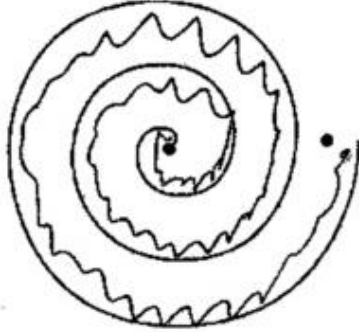
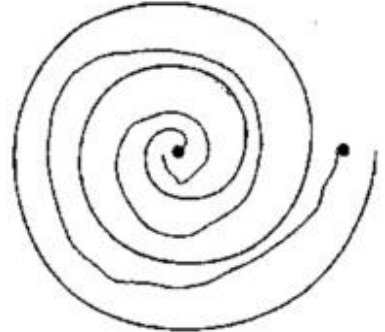

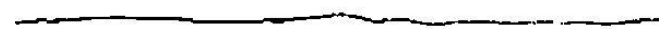
Calculated Value	Before	After
Total area ( <i>pixels</i> <sup>2</sup> )	15 442.10	4 316.75
Standard deviation of total area	5.52	2.10
Maximum value ( <i>pixel</i> )	21.42	4.17
Maximum frequency ( <i>Hz</i> )	11 087.03	2 093.07
<b>METHOD 2A:</b> Average area ( <i>pixels</i> <sup>2</sup> )	13.44	0.86
Standard deviation of average area	122.74	51.09
Number of peaks	12	7
Average peak-trough distance ( <i>pixels</i> )	5.81	2.43
<b>METHOD 2B:</b> Num. peaks × avg. peak-trough distance	69.67	17.02

# Method 2: Results

Calculated Value	Before	After
Total area ( <i>pixels</i> <sup>2</sup> )	15 442.10	4 316.75
Standard deviation of total area	5.52	2.10
Maximum value ( <i>pixel</i> )	21.42	4.17
Maximum frequency ( <i>Hz</i> )	11 087.03	2 093.07
<b>METHOD 2A:</b> Average area ( <i>pixels</i> <sup>2</sup> )	13.44	0.86
Standard deviation of average area	122.74	51.09
Number of peaks	12	7
Average peak-trough distance ( <i>pixels</i> )	5.81	2.43
<b>METHOD 2B:</b> Num. peaks × avg. peak-trough distance	69.67	17.02



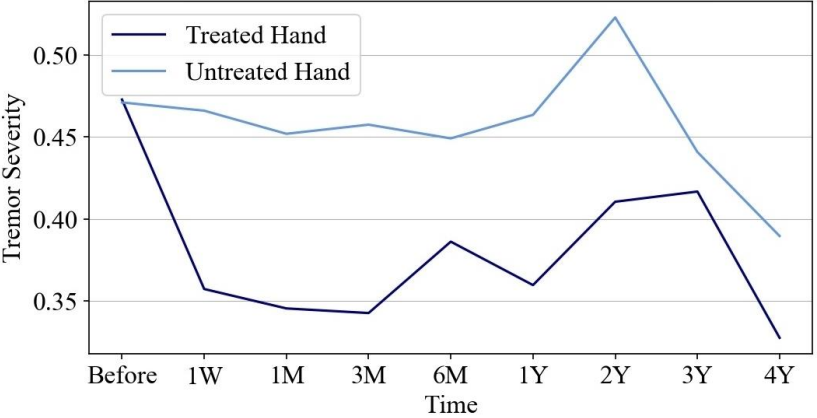
# Results: Patient #5 Summary

Patient #5		Before Treatment	After Treatment
METHOD 1	Spiral A		
	Normalised Standard Deviation	0.64	0.15
METHOD 2	Line C		
	Average Area (TRAPZ)	13.44	0.86
	Num. Peaks $\times$ Avg. Peak-Trough Distance	69.67	17.02

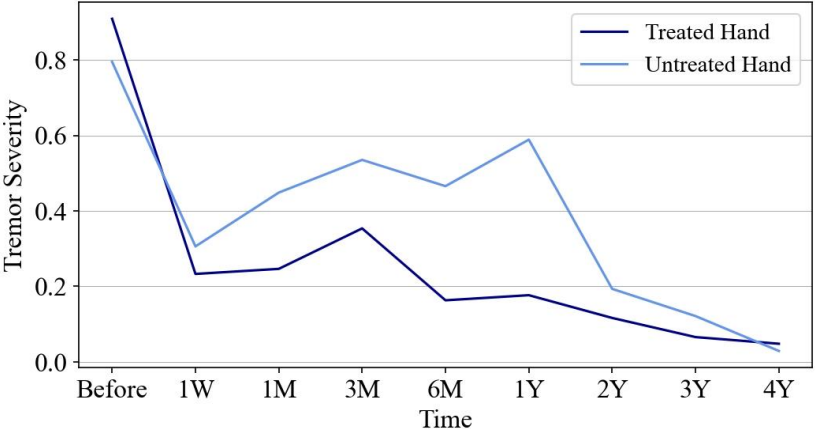


# Results: Normalised Average Tremor Severity

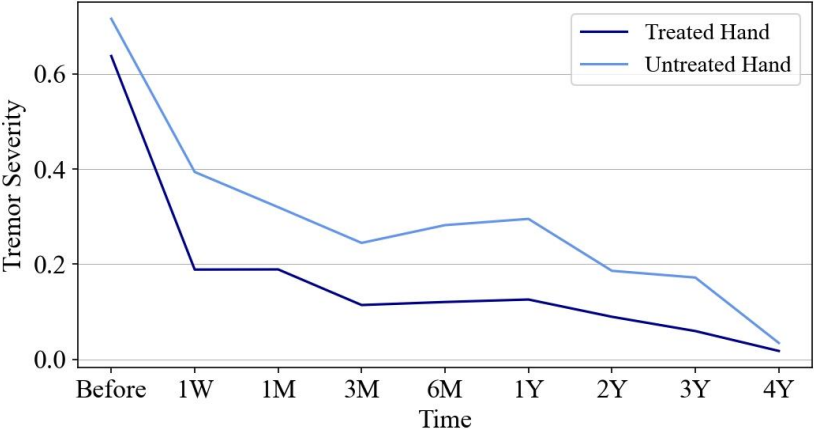
Method 1: Spiral Analysis



Method 2A: Average Area

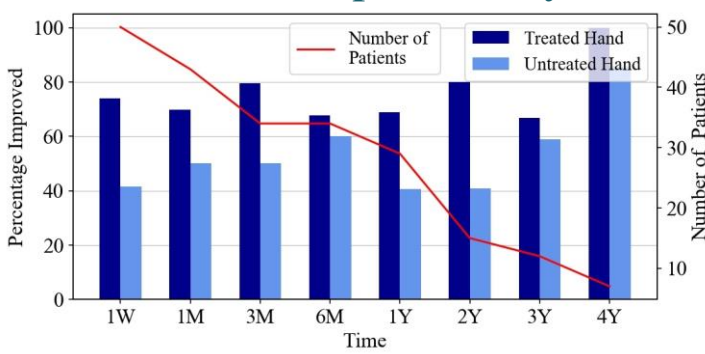


Method 2B: Num Peak  $\times$  Avg.  
Peak-Trough Distance

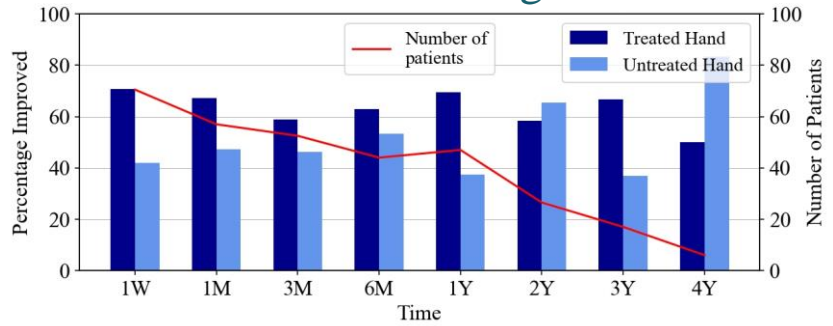


# Results: Percentage of Patients with Tremor Before Treatment that Improved After Various Treatment Times

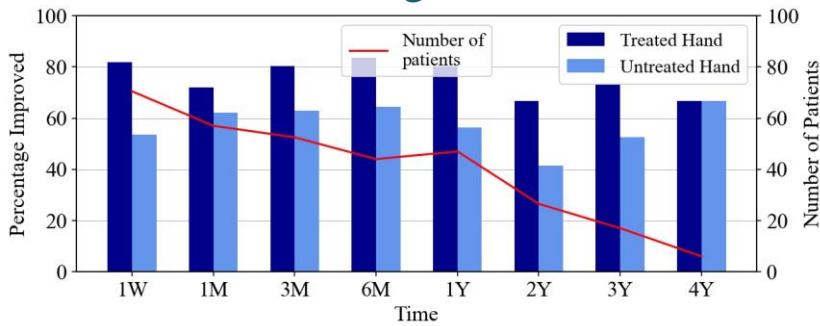
Method 1: Spiral Analysis



Method 2A: Average Area

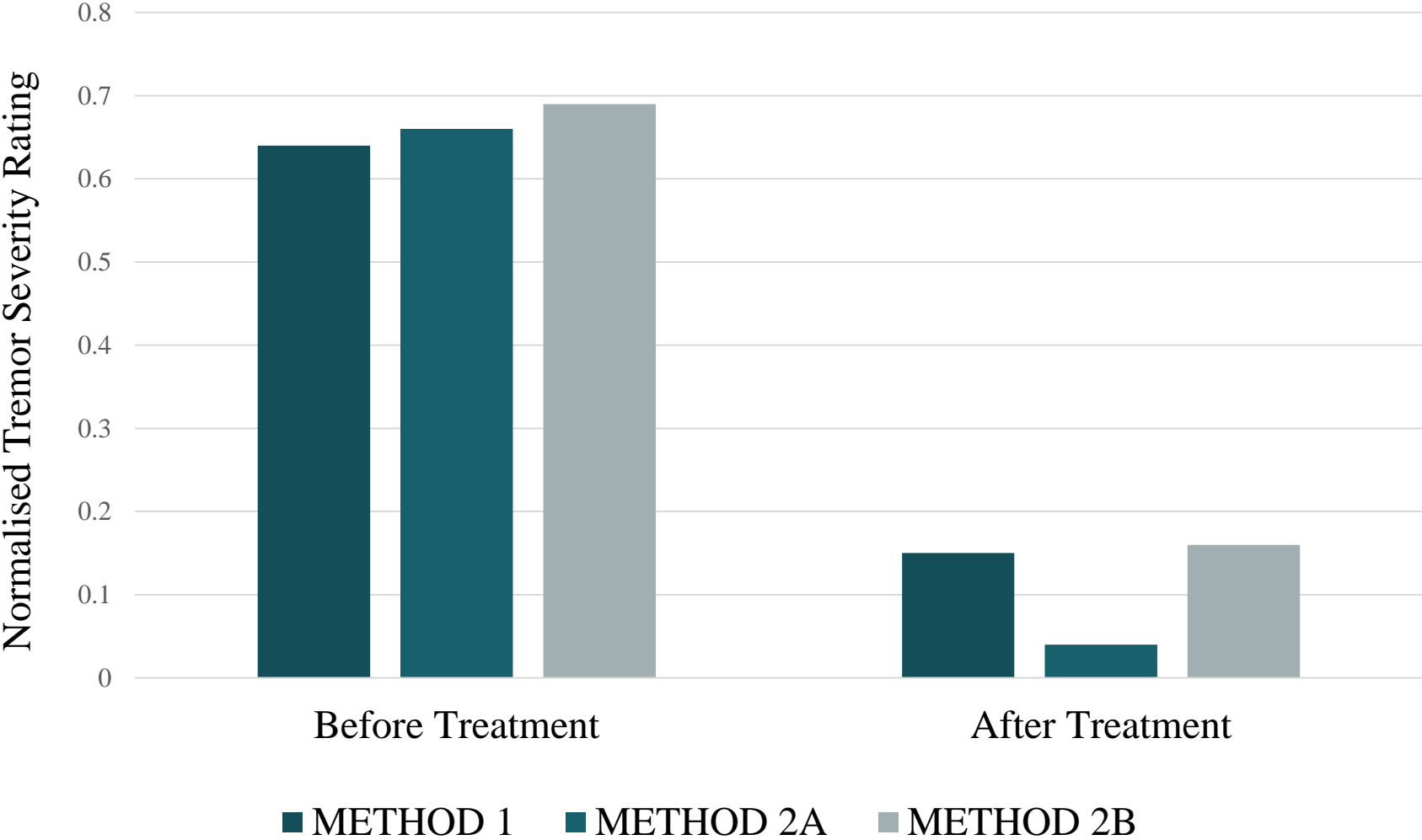


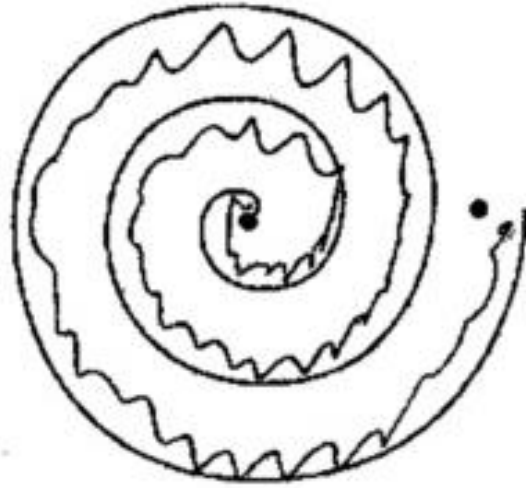
Method 2B: Num Peak  $\times$  Avg. Peak-Trough Distance



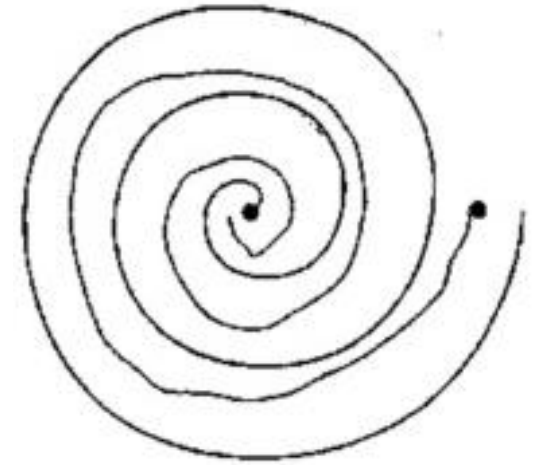
Percentage of patients that improved	Method 1	Method 2A	Method 2B
Treated hand	75.8%	63.5%	75.7%
Untreated hand	53.4%	51.4%	57.4%

# Results: Patient #5 Normalised Tremor Severity Rating

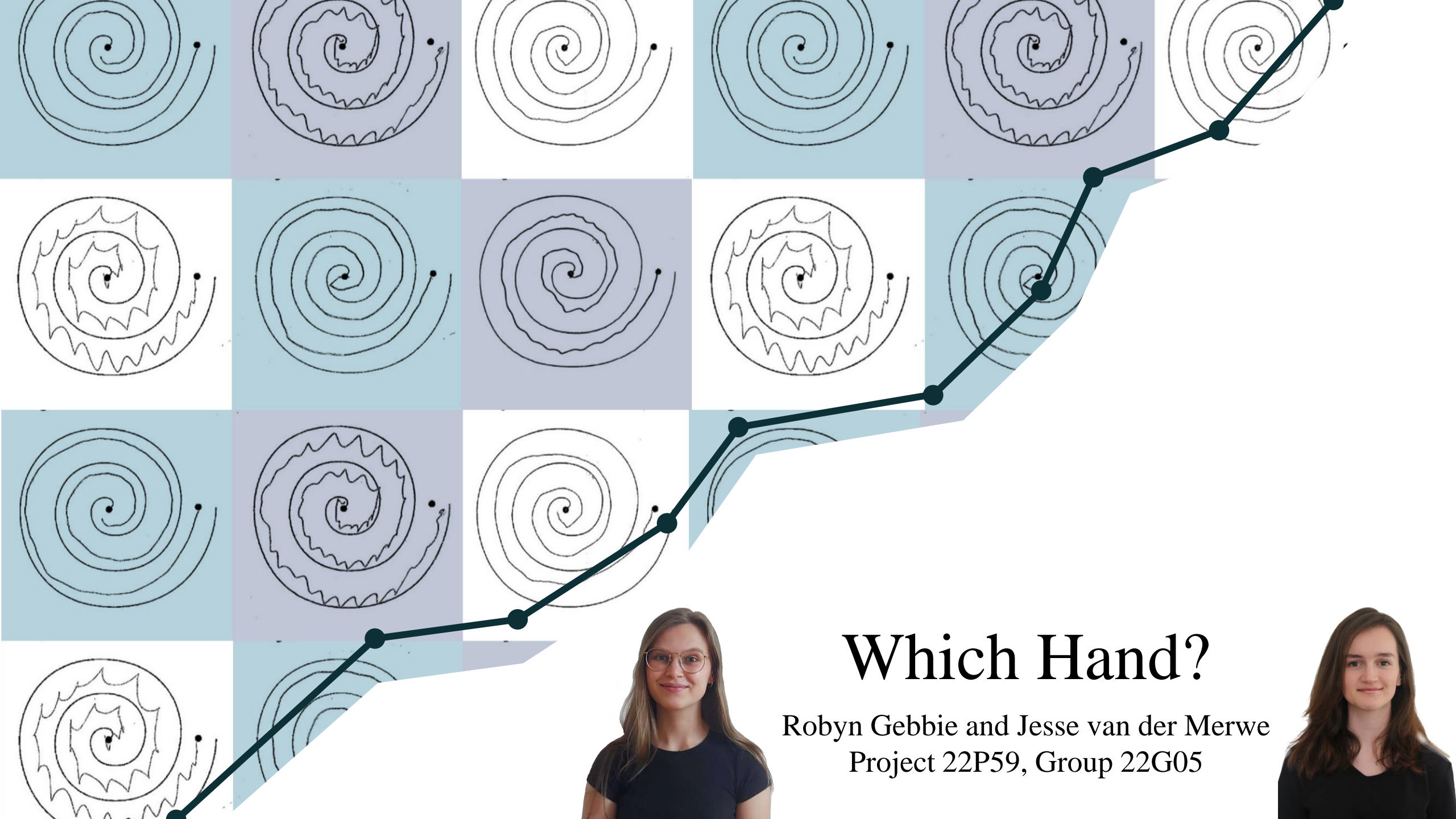




**Before Treatment**



**After Treatment**



# Which Hand?

Robyn Gebbie and Jesse van der Merwe  
Project 22P59, Group 22G05

