

האוניברסיטה הפתוחה
המחלקה למדעי המחשב

פרויקט מתקדם במדעי המחשב - 22997

מערכת FCBM לניתוח אשכולות עמום לתמונות
MRI מוח לצורך זיהוי סרטן מסוג GBM

מגישה: ענת אבידב ת.ז. 028048494

הפרויקט הוכן בהנחייתה של ד"ר מיה הרמן

אוקטובר 2016

תוכן העניינים

3	תקציר	1
4	מבוא	2
5	הגדרת דרישות המערכת	3
5	דרישות פונקציונליות	3.1
5	דרישות לא פונקציונליות	3.2
5	הנחות יסוד	3.3
6	תרחישים	3.4
6	דיאגרמת תרחישים של כלל מערכת FCBM	3.4.1
6	דיאגרמת תרחישים של Preprocess	3.4.2
7	דיאגרמת תרחישים של Image Preprocess	3.4.3
8	דיאגרמת תרחישים של FCM Preprocess	3.4.4
8	דיאגרמת תרחישים של Data Mining	3.4.5
9	תרחיש CheckValidity	3.4.6
9	תרחיש ViewAllPatients	3.4.7
9	תרחיש ViewPatient	3.4.8
10	דיאגרמת מחלקות למערכת FCBM	3.5
11	תהליכים עיקריים במערכת FCBM	3.6
11	תהליך ה-Start Server	3.6.1
12	תהליך ה-Pre Processing	3.6.2
13	תהליך ה-PreProcessBlock	3.6.3
15	תהליך ה-Mining	3.6.4
16	תהליך ה-MineBlock	3.6.5
17	תכן בסיס הנתונים	4
19	ארכיטקטורת מערכת FCBM	5
19	שכבות מערכת FCBM	5.1
20	מבנה מערכת FCBM	5.2
21	שרת	5.2.1
29	לקוח	5.2.2
30	ממשק משתמש	6
39	כלי פיתוח	7
41	בדיקות המערכת	8
41	בדיקות יחידה	8.1
50	בדיקות אינטגרציה	8.2
51	מקרה בוחן	9
51	ביצוע תהליך העיבוד המקדים לתמונה	9.1
56	ביצוע תהליך העיבוד המקדים ל-FCM	9.2
56	ביצוע תהליך כריית המידע באמצעות FCM	9.3
56	תוצאות	9.4
57	סיכום והצעות להמשך מחקר	10
58	נספחים	11
58	נספח א: סקריפט ליצירת הטבלאות	11.1
60	נספח ב: סקריפט ליצירת stored procedures	11.2
68	נספח ג: קוד	11.3
69	נספח ד: הוראות התקנה	11.4
70	נספח ה: הוראות הרצה	11.5
71	מקורות	12

6	איור 1 - דיאגרמת תרחישים ברמת על למערכת FCBM
6	איור 2 - דיאגרמת תרחישים של PrpProcess
7	איור 3 - דיאגרמת תרחישים של Image Preprocess
8	איור 4 - דיאגרמת תרחישים של FCM Preprocess
8	איור 5 - דיאגרמת תרחישים של Data Mining
10	איור 6 - דיאגרמת מחלקות למערכת FCBM
11	איור 7 - דיאגרמת רצף של תהליך ה-Start Server
12	איור 8 - דיאגרמת רצף של תהליך ה-Pre Processing
14	איור 9 - דיאגרמת רצף של תהליך ה-PreProcessBlock
15	איור 10 - דיאגרמת רצף של תהליך ה-Mining
16	איור 11 - דיאגרמת רצף של תהליך ה-MineBlock
19	איור 12 - שכבות מערכת FCBM
20	איור 13 - רכיבי מערכת FCBM
24	איור 14 - wavelet transform של תמונה
25	איור 15 - אילוסטרציה של wavelet transform על תמונה
25	איור 16 - חלוקה של תוצאת wavelet transform ל-sub-bands
30	איור 17 - עץ מסכים
30	איור 18 - מסך נבדקים לפני ביצוע כריית מידע
31	איור 19 - מסך פרטי נבדק לפני ביצוע כריית מידע
32	איור 20 - מסך נתונים מפורט של תמונת MRI לפני ביצוע כריית מידע
33	איור 21 - מסך ניהול תהליכים לפני ביצוע עיבוד מקדים ולפני ביצוע כריית מידע
34	איור 22 - מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ולפני ביצוע כריית מידע
35	איור 23 - מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ואחרי ביצוע כריית מידע
36	איור 24 - מסך נבדקים אחרי ביצוע כריית מידע
37	איור 25 - מסך פרטי נבדק אחרי ביצוע כריית מידע
38	איור 26 - מסך נתונים מפורט של תמונת MRI אחרי ביצוע כריית מידע

1 תקציר

בדיקת MRI מתבצעת באופן שגרתי כאשר קיים חשד להפרעה במבנה התקין של המוח, בין היתר בעקבות סרטן. מבין סוגי הסרטן השונים שמקורם במוח גליובלסטומה מולטיפורמה (Glioblastoma Multiforme) הוא הסוג השכיח ביותר והממאיר ביותר. הטיפול בסרטן מסוג Glioblastoma Multiforme (GBM) הוא לרוב ניתוח להוצאת הגידול ולאחר מכן כימותרפיה והקרנות. במקרים רבים ניתן אמנם לראות גידולי מוח בבדיקת CT, אך כאשר מתכננים ניתוח להוצאת הגידול יש לקבל תמונה ברזולוציה גבוהה ככל שניתן. באמצעות תמונת MRI ניתן לזהות במדויק את גבולות הגידול, לראות על אילו מבנים הוא לוחץ ובהתאמה לתכנן ניתוח להוצאתו. סרטן מסוג GBM לרוב מתחיל בחומר הלבן במוח, גדל בקצב מואץ, ויכול להגיע לגדלים מאוד גדולים. כ- 50% מהסרטנים מסוג GBM מגיעים לשתי האונות במוח (סרטן בילטרלי).

נכון להיום, רדיולוג מפענח את תמונות ה-MRI מוח ונותן הערכה איכותית האם המוח תקין או פתולוגי, האם יש גידול, ומה המיקום והגודל של הגידול. מטרת הפרויקט לפתח מערכת לביצוע אשכול עמום של תמונות ה-MRI למוח תקין או פתולוגי.

בפרויקט הנוכחי בוצע אפיון ועיצוב ראשוני של מערכת FCBM (Fuzzy Clustering of Brain MRI) תוך התמקדות בנושא עיבוד מקדים לתמונות MRI ועיבוד מקדים לאשכול עמום (FCM).

הקלט למערכת FCBM הן תמונות MRI מוח תקין ופתולוגי ממאגרי MIDAS ו-TCIA. במסגרת הפרויקט בוצע עיבוד תמונה מקדים שכלל General registration, Histogram matching, Foreground masking, skull stripping, Mark tumor mask, crop ו-scale, ובוצע עיבוד מקדים ל-FCM שכלל חלוקת כל תמונה לבלוקים, נורמליזציה של נתונים, ו-Feature extraction לכל בלוק.

הפלט של מערכת FCBM היה סיווג מוח לתקין או פתולוגי ברמת דיוק 88.13%.

2 מבווא

כריית מידע מתמונות ככלל ובפרט מתמונות MRI היא טכניקה בה מזהים תבניות בתמונה בעזרת אלגוריתמי כריית מידע עמומה במטרה לזהות מוח פתולוגי שקשה לזהותו בעין אנושית [10], [2], [5], [12], [14], [16], [22], [25], [27], [29], [32], [34]. מקור תמונות MRI מוח הוא שני בסיסי נתונים פתוחים באינטרנט. ממאגרים אלו נבנה בסיס נתונים הכולל תמונות מוח תקין ופתולוגי.

○ מאגר של MRI מוח של מתנדבים בריאים מתוך האתר של MIDAS בקישור הבא:
<http://www.insight-journal.org/midas/community/view/21> [11]

מאגר זה הוא חלק מפרויקט (Imaging Methods Assessment and Reporting) IMAR אשר מטרתו לספק מידע, שיטות ואמצעי מחשוב לצורך השוואה כמותית של סגמנטציה, רגיסטרציה ושיטות אבחון ממוחשבות למוח. על הפרויקט בקישור הבא:

<http://www.insight-journal.org/midas/community/view/15> [15]

המאגר של MIDAS מכיל תמונות MRI מוח של 100 מתנדבים, ואינו מכיל מוחות של אנשים בעלי היסטוריה של סכרת, יתר לחץ דם, טראומה לראש, מחלות פסיכיאטריות, או סימפטומים אחרים אשר עלולים להשפיע על המוח. המאגר נלקח באמצעות פרוטוקולים סטנדרטיים במרווחים של $1 \times 1 \times 1$ mm או במרווחים של $0.5 \times 0.5 \times 0.8$ mm. כל תמונות ה-MRI מפורסמות ללא פרטים מזהים של הנבדקים, וכל המתנדבים חתמו על הסכמה לפרסם את תמונות ה-MRI באינטרנט.

○ מאגר של MRI מוח של חולים בסרטן מתוך האתר של The Cancer Imaging Archive (TCIA), מתוך המחקר של TCGA-GBM בקישור הבא:

<https://public.cancerimagingarchive.net> [8]

תיאור מאגר TCGA-GBM - The Cancer Genome Atlas Glioblastoma Multiforme data collection בקישור הבא:

<https://wiki.cancerimagingarchive.net/display/Public/TCGA-GBM> [28]

המאגר מוגדר public domain והוא פתוח בחינם לצפייה ולהורדה לצורך שימוש מסחרי, מדעי או לימודי. פרטים אישיים של כל הנבדקים במאגר עברו de-identification כדי להפוך את הנתונים במאגר לפתוחים לציבור. לגבי תהליך שמירת המידע וה: de-identification ניתן לקרוא בקישור הבא:

<http://www.cancerimagingarchive.net/about-the-cancer-imaging-archive-tcia/> [9]

כלל התמונות שהתקבלו הן תמונות T2 בלבד וזאת לאור מחקרים שהוכיחו שתמונות אלו טובות יותר לזיהוי מוח.

קיימים מספר שלבים מקדימים להליך כריית המידע:

עיבוד תמונה מקדים שכולל:

- General registration
- Histogram matching
- Foreground masking
- Skull stripping
- Mark tumor mask
- Crop
- Scale

עיבוד מקדים ל-FCM שכולל:

- חלוקת כל תמונה לבלוקים
- נורמליזציה של נתונים
- Feature extraction לכל בלוק.

הפרויקט הנוכחי כולל: עיבוד מקדים לתמונה ועיבוד מקדים ל-FCM, כריית מידע באמצעות FCM, הצגת נתוני נבדקים ותוצאות כריית המידע באמצעות ממשק משתמש, מדידת איכות התוצאות וניתוח התוצאות.

3 הגדרת דרישות המערכת

3.1 דרישות פונקציונליות

- המערכת תקבל נתונים מבסיסי נתונים חופשיים ב-offline בפורמט DICOM.
- המערכת תבצע עיבוד מקדים לנתונים:
 - עיבוד תמונה מקדים:
 - עיבוד תמונות מהמאגרים כולל רגיסטרציה, נירמול היסטוגרמות, השחרת רקע, ו-skull stripping יבוצע בעזרת תוכנת 3D Slicer.
 - שמירת התוצאות כאוסף קבצי jpg יבוצע באמצעות תוכנת MIPAV.
 - זיהוי המסגרת, ביצוע crop, וביצוע scaling יבוצעו בעזרת תוכנת offline שלי.
 - תוכנות: 3D Slicer, MIPAV, תוכנת Offline preprocessing שלי.
 - עיבוד מקדים ל-FCM:
 - חלוקת כל תמונה לבלוקים
 - נורמליזציה של נתונים
 - Feature extraction לכל בלוק.
- המערכת תסווג תמונות MRI מוח לתקין ופתולוגי (סרטן מסוג GBM בלבד).
- המערכת תציג את תוצאות הסיווג באמצעות ממשק משתמש.

3.2 דרישות לא פונקציונליות

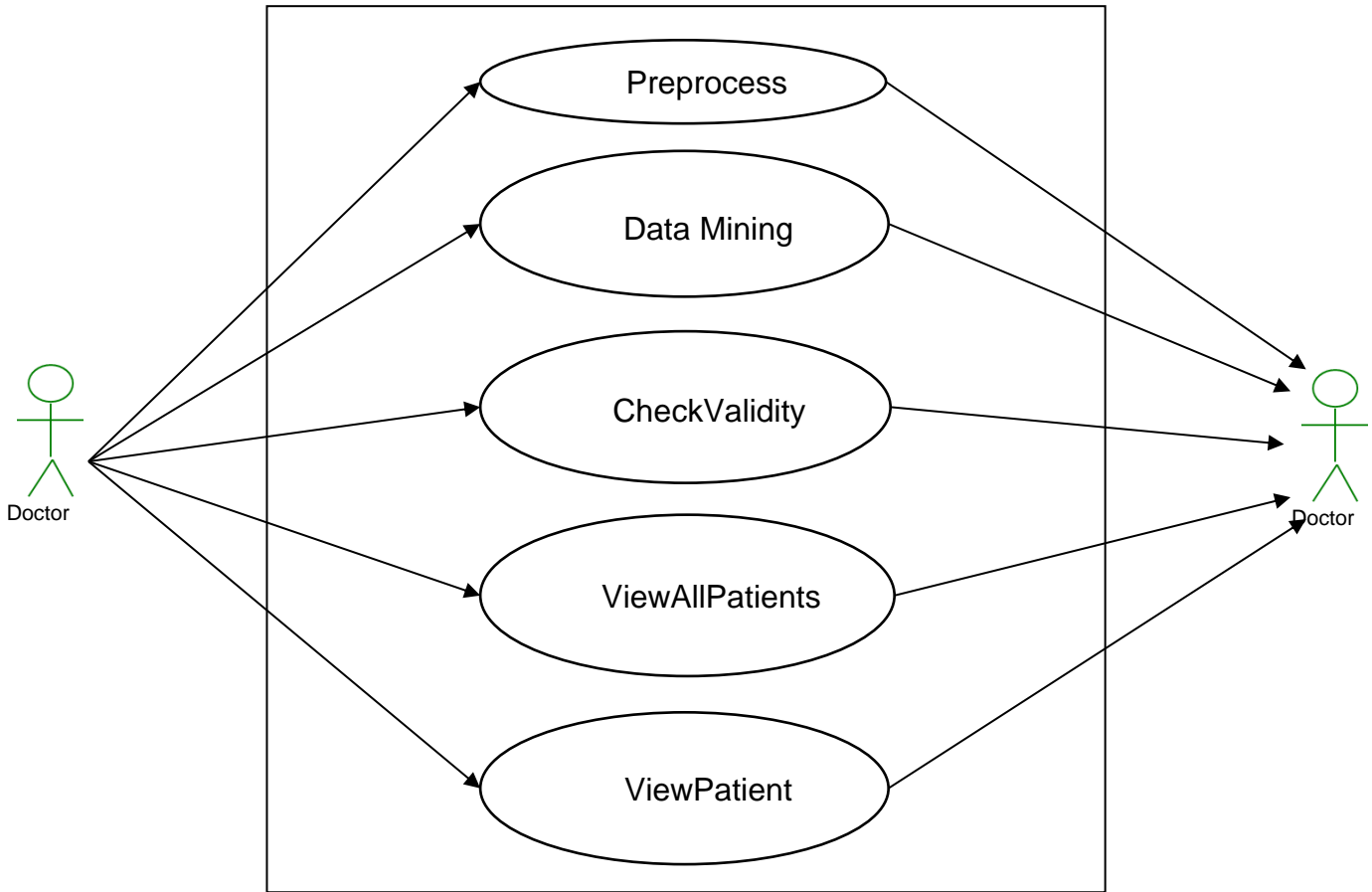
- בחינת איכות האשכול.
- בדיקות:
 - בדיקות יחידה.
 - בדיקת אינטגרציה.

3.3 הנחות יסוד

- בסיס הנתונים יכלול תמונות מוח תקין ופתולוגי.
- התמונות בבסיס הנתונים זה לקוחות מתמונות T2 בחתך אקסיאלי. בסיס נתונים זה יכיל תמונות MRI, אשר מהן התבצע חיתוך של המוח בלבד, ואז התמונות סובבו על ציר ה-X וה-Y כך שכולן מנורמלות לאותו כיוון.
- צורת ההתממשקות למקורות המידע תהיה חד פעמית. הנתונים הרלוונטיים יועברו לבסיס נתונים אשר ישמש את הפרויקט. קבלת הנתונים תהיה מסוג offline. אם בעתיד יהיה צורך להוסיף נתונים תהיה שוב העברת נתונים חד פעמית.
- תוצרי הפרויקט ישמרו בבסיס הנתונים. ניתן יהיה לשלוף אותם ולבחון אותם באמצעות תוכנת הלקוח של הפרויקט, או בכל דרך אחרת אשר תשלוף נתונים מבסיס הנתונים.
- הקוד יהיה כתוב ללא הגבלת הכלליות ויאפשר זיהוי אזורים אבנורמליים במספר רב של תמונות לכל נבדק. במאגרי המידע שהיו בשימוש בפרויקט זה (ראה 2) לכל נבדק היו בממוצע 200 תמונות. במסגרת פרויקט זה המערכת תיבדק עם תמונה אחת פר נבדק מטעמי חסכון בזמן הדרוש לקליברציה ידנית (ראה 3.1 נקודה שניה), בזמן הדרוש לעיבוד מקדים (ראה 3.1 נקודה שניה), ובזמן הדרוש לביצוע האשכול.
- על פי American Brain Tumor Association סרטן מסוג GBM נפוץ ביותר בתוך האונה השמאלית או הימנית של המוח, ומופיע בשתייהן בכ 50% מהמקרים: <http://www.abta.org/brain-tumor-information/types-of-tumors/glioblastoma.html> [3]
- כלומר התצוגה המירבית של סרטן מסוג GBM במוח היא בחתך הנמצא באמצע המוח. מסיבה זאת נבחרה תמונה (slice) אשר נמצאת באמצע המוח לצורך בדיקת המערכת.

3.4 תרחישים

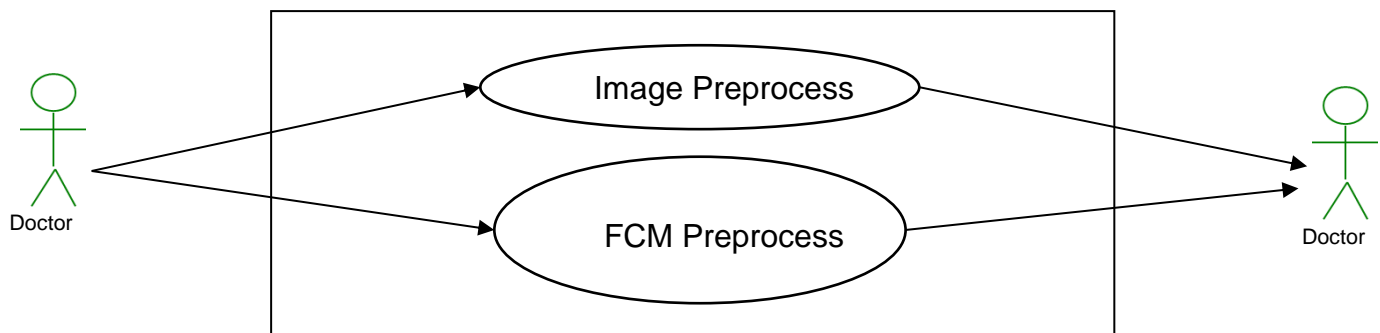
3.4.1 דיאגרמת תרחישים של כלל מערכת FCBM



איור 1 - דיאגרמת תרחישים ברמת על למערכת FCBM

3.4.2 דיאגרמת תרחישים של Preprocess

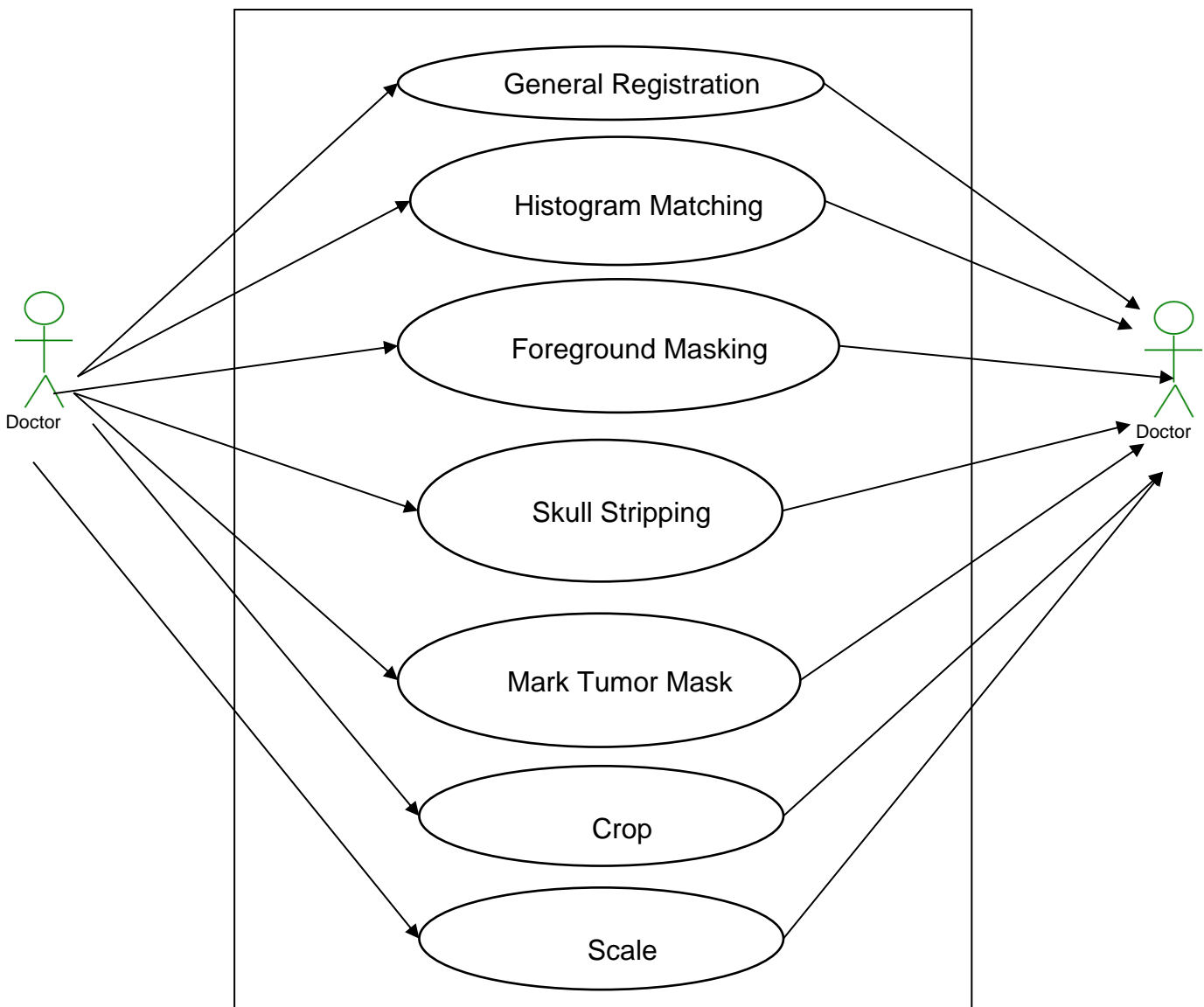
העיבוד המקדים במערכת מתחלק לשני שלבים: עיבוד מקדים לתמונה ועיבוד מקדים ל-FCM.



איור 2 - דיאגרמת תרחישים של PrpProcess

3.4.3 דיאגרמת תרחישים של Image Preprocess

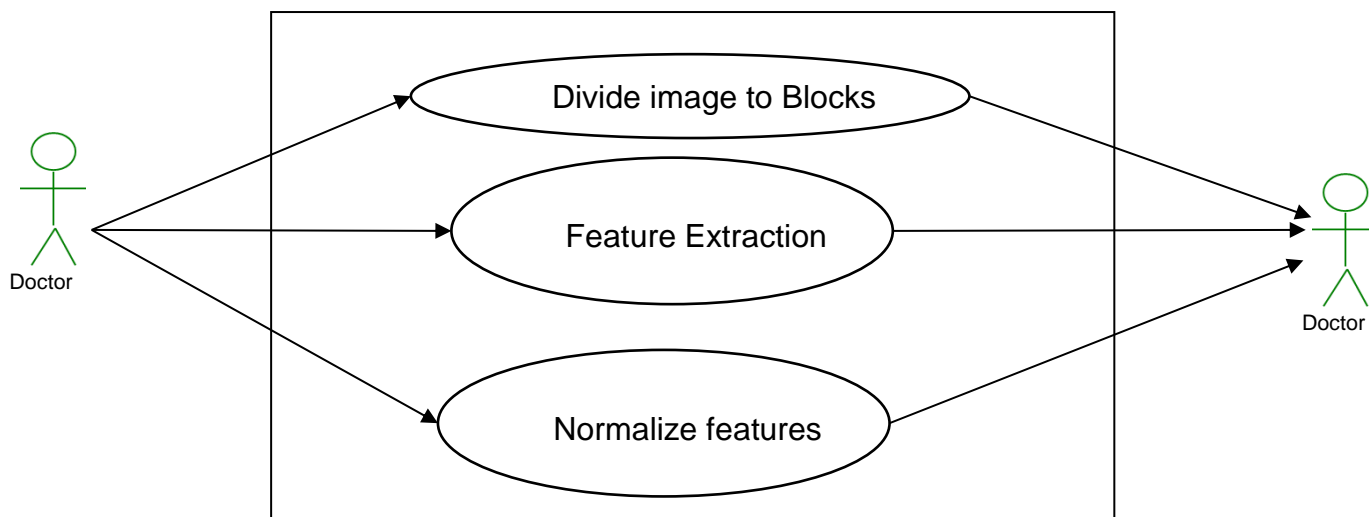
- נשתמש בתוכנה 3D Slicer הניתנת להורדה בחינם בקישור הבא:
https://www.slicer.org/slicerWiki/index.php/Main_Page [1]
על כל התמונות של כל הנבדקים נבצע רגיסטראציה, נירמול היסטוגרמות, השחרת רקע, ו- skull stripping באמצעות 3D Slicer.
- נפתח את הקבצים שהתקבלו באמצעות תוכנת MIPAV - Medical Image Processing, Analysis, and Visualization מהקישור הבא:
<https://mipav.cit.nih.gov/> [21]
באמצעות MIPAV נבצע שמירה של ה-MRI כסדרה של תמונות JPG.
- נריץ תוכנת עזר ל-PreProcessing שאני כתבתי (ומצורפת בדיסקים המכילים את קוד המקור) אשר מזהה את המסגרת הריקה בכל תמונה ומבצעת crop של החלק המשמעותי בכל תמונה, ועושה scaling של כל התמונות לאותו גודל.



איור 3 - דיאגרמת תרחישים של Image Preprocess

3.4.4 דיאגרמת תרחישים של FCM Preprocess

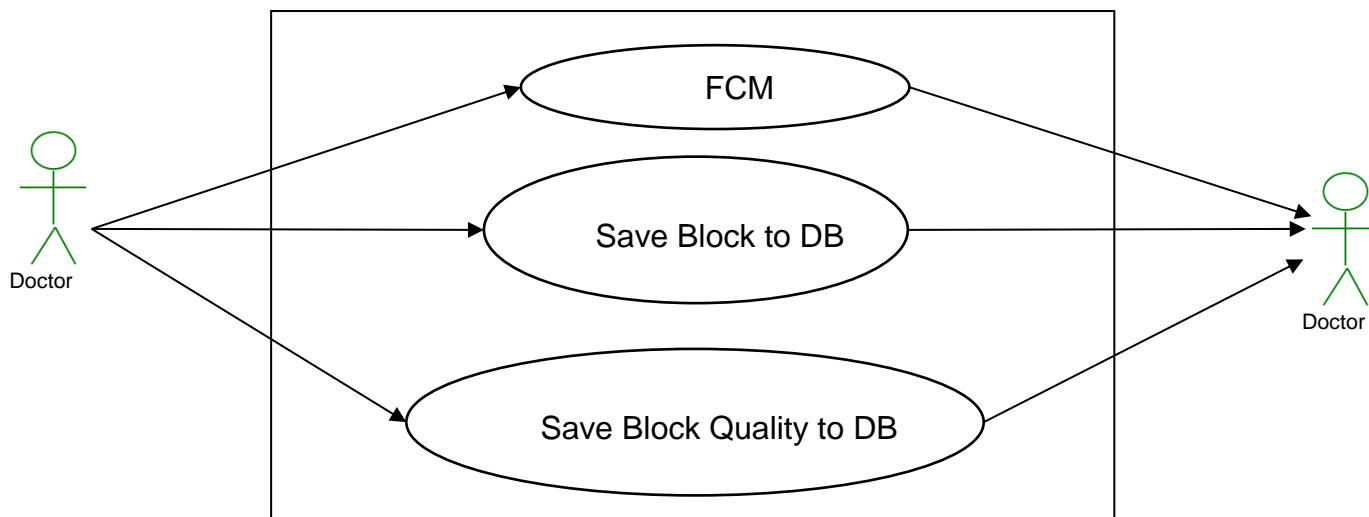
- עיבוד מקדים זה יכול חלוקת כל תמונה לבלוקים, נורמליזציה של נתונים, ו- Feature extraction לכל בלוק.
- שמירת תוצאת העיבוד המקדים בבסיס הנתונים.



איור 4 - דיאגרמת תרחישים של FCM Preprocess

3.4.5 דיאגרמת תרחישים של Data Mining

- סיווג הבלוקים לתקין ופגולוגי.
- עיבוד לתוצאות הסיווג שהתקבלו לכל תמונה ולכל נבדק.
- שמירת התוצאה בבסיס הנתונים.



איור 5 - דיאגרמת תרחישים של Data Mining

3.4.6 תרחיש CheckValidity

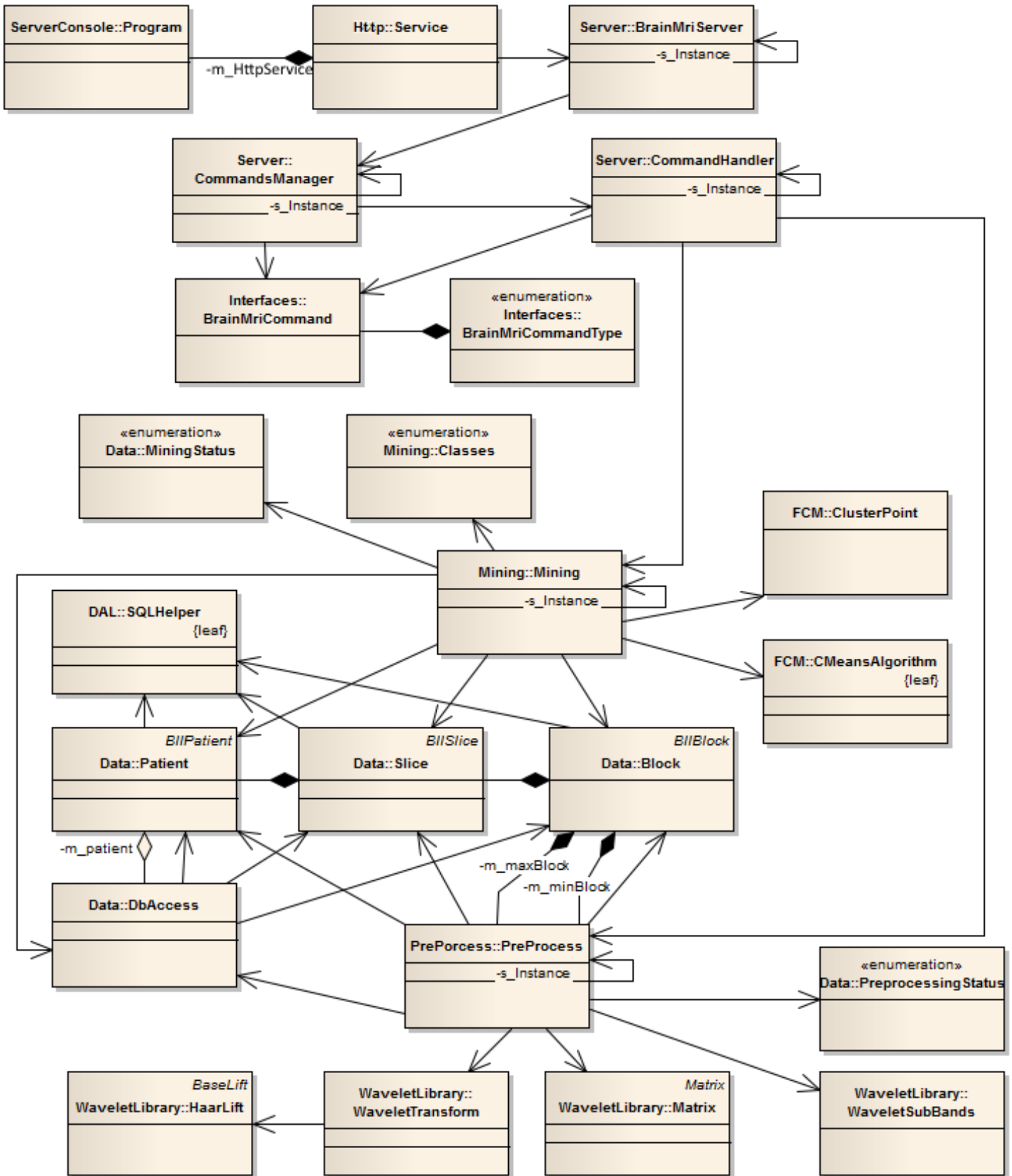
- חישוב מדדים לבחינת איכות האשכולות, ושמירתם בבסיס הנתונים.

3.4.7 תרחיש ViewAllPatients

- הצגת כל הנבדקים ברשימה. לכל הנבדקים יוצג מזהה הנבדק, סיווג – תקין/פתולוגי, ואינדיקציה למידת השייכות למחלקה אליה שויך הנבדק.

3.4.8 תרחיש ViewPatient

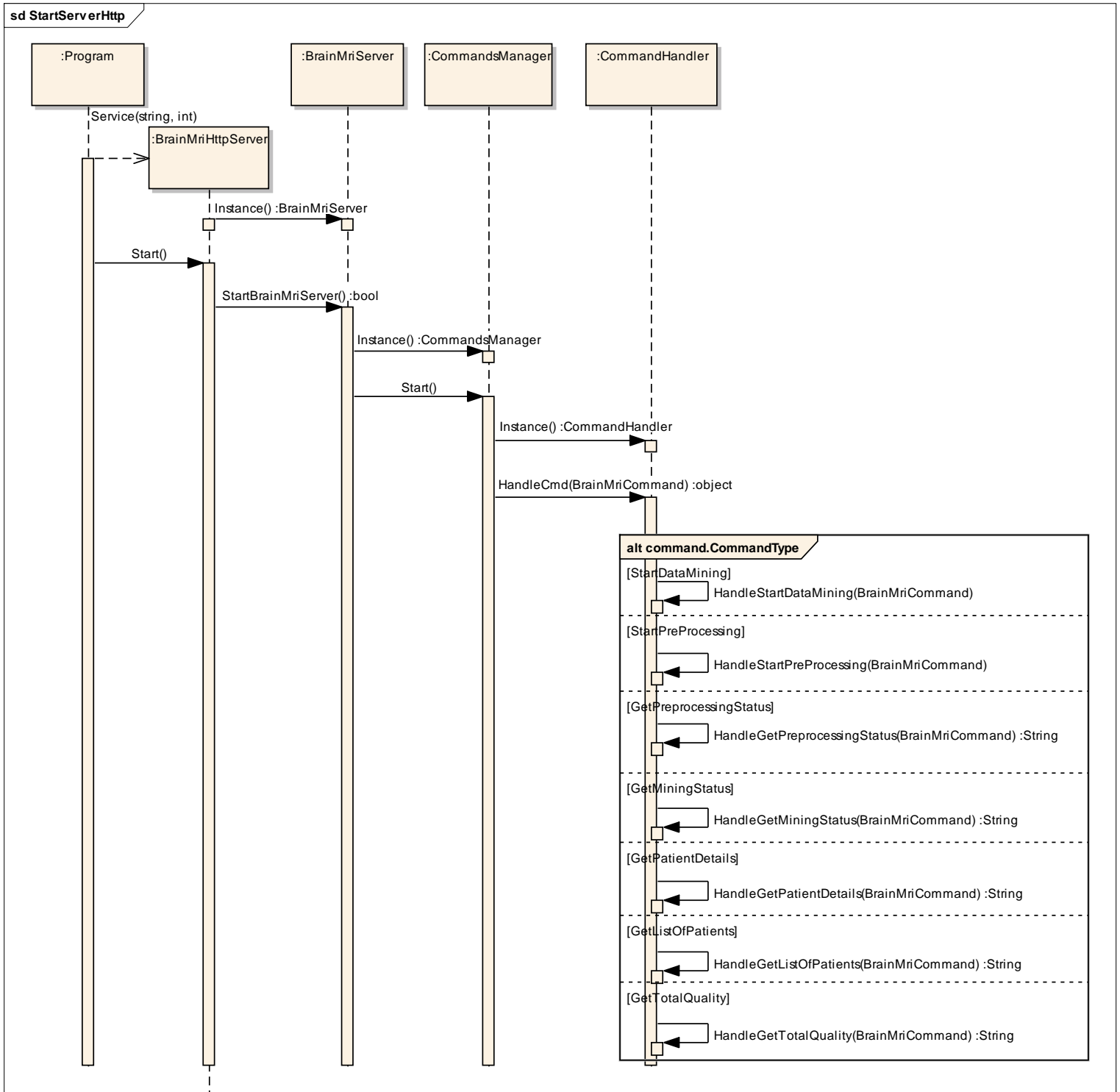
- הצגת נתוני נבדק ספציפי. יוצג מזהה הנבדק, סיווג – תקין/פתולוגי, ואינדיקציה למידת השייכות למחלקה.
- יוצגו תמונות ה-MRI מוח.
- עבור נבדק פתולוגי יוצגו התמונות בעלות בלוקים פתולוגיים בתוספת הדמיה צבעונית של הבלוקים המסווגים כפתולוגיים.



איור 6 - דיאגרמת מחלקות למערכת FCBM

3.6.1 תהליך ה-Start Server

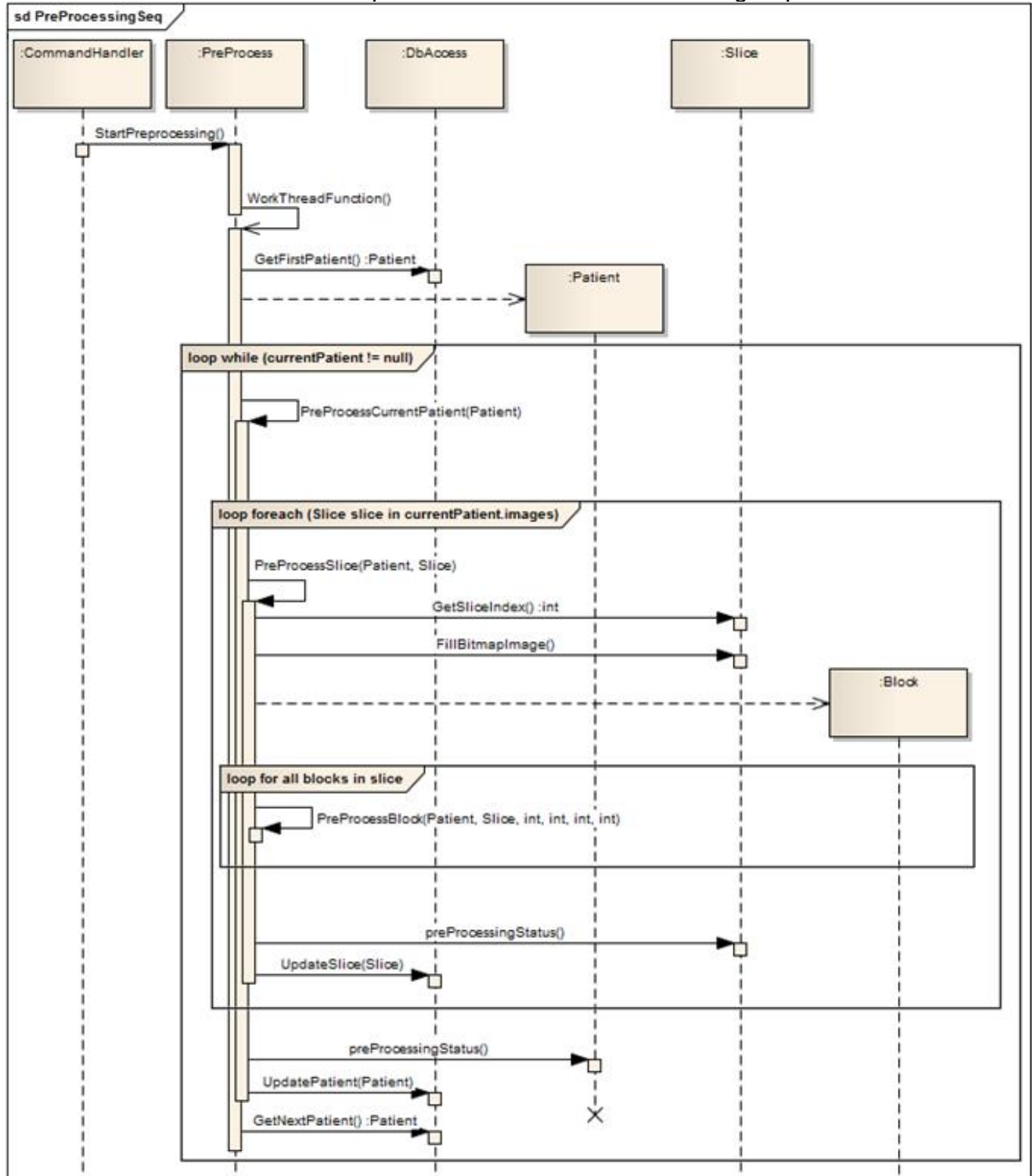
מטרה: התהליך מתרחש בזמן העלאת השרת, ומטרתו להתחיל את Http.Service המאזין להודעות HTTP מהלקוח, ולהתחיל גם את תהליך BrainMriServer המטפל בניהול השירותים המסופקים במערכת באמצעות העברת פקודות מהלקוח לתור לניהול פקודות.
 קלט: העלאת השרת גומרת להתחלת תהליך זה. אין קלט נוסף לתהליך.
 עיבוד: מתחיל תהליכים במערכת. אין לתהליך זה עיבוד עצמאי.
 פלט: ללא פלט. התהליך מסתיים כאשר מפסיקים את ריצת המערכת.



איור 7 - דיאגרמת רצף של תהליך ה-Start Server

3.6.2 תהליך ה-Pre Processing

מטרה: מטרת התהליך לבצע עיבוד מקדים לכל הנבדקים, לקבל כתוצאה מאפיינים לכל הבלוקים בכל התמונות של כל הנבדקים, ולשמור את המאפיינים בבסיס הנתונים. קלט: בסיס נתונים המכיל טבלת נבדקים, ותמונות לכל נבדק. עיבוד: תהליך העיבוד המקדים מבצע מעבר על כל הנבדקים במערכת, כאשר לכל נבדק מתבצע מעבר על כל התמונות שלו. כל תמונה מחולקת לבלוקים. לכל בלוק מבוצע עיבוד מקדים שיתואר בתהליך הבא (3.6.3), ותוצאותיו ישמרו בבסיס הנתונים. פלט: שמירת המאפיינים לכל בלוק בבסיס הנתונים (מתבצע על ידי התהליך הפנימי 3.6.3) ושמירה בבסיס הנתונים של סטטוס תהליך ה-PreProcessing לכל תמונה ולכל נבדק.



איור 8 - דיאגרמת רצף של תהליך ה-Pre Processing

3.6.3 תהליך PreProcessBlock

מטרה: תהליך זה הוא למעשה תת-תהליך המופעל על ידי תהליך ה-Pre Processing. מטרת התהליך לקבל בלוק מתהליך האב (3.6.2) ולבצע feature extraction (חישוב מאפיינים). המאפיינים שיחשבו לכל בלוק יהיו:

Mean, variance, horizontal band of wavelet transform, vertical band of wavelet transform, diagonal band of wavelet transform, contrast.

קלט: הבלוק עליו מחושבים המאפיינים.
עיבוד:

$$mean = \frac{\sum_{i=1}^N Pixel\ i\ Grayscale\ value}{N}$$

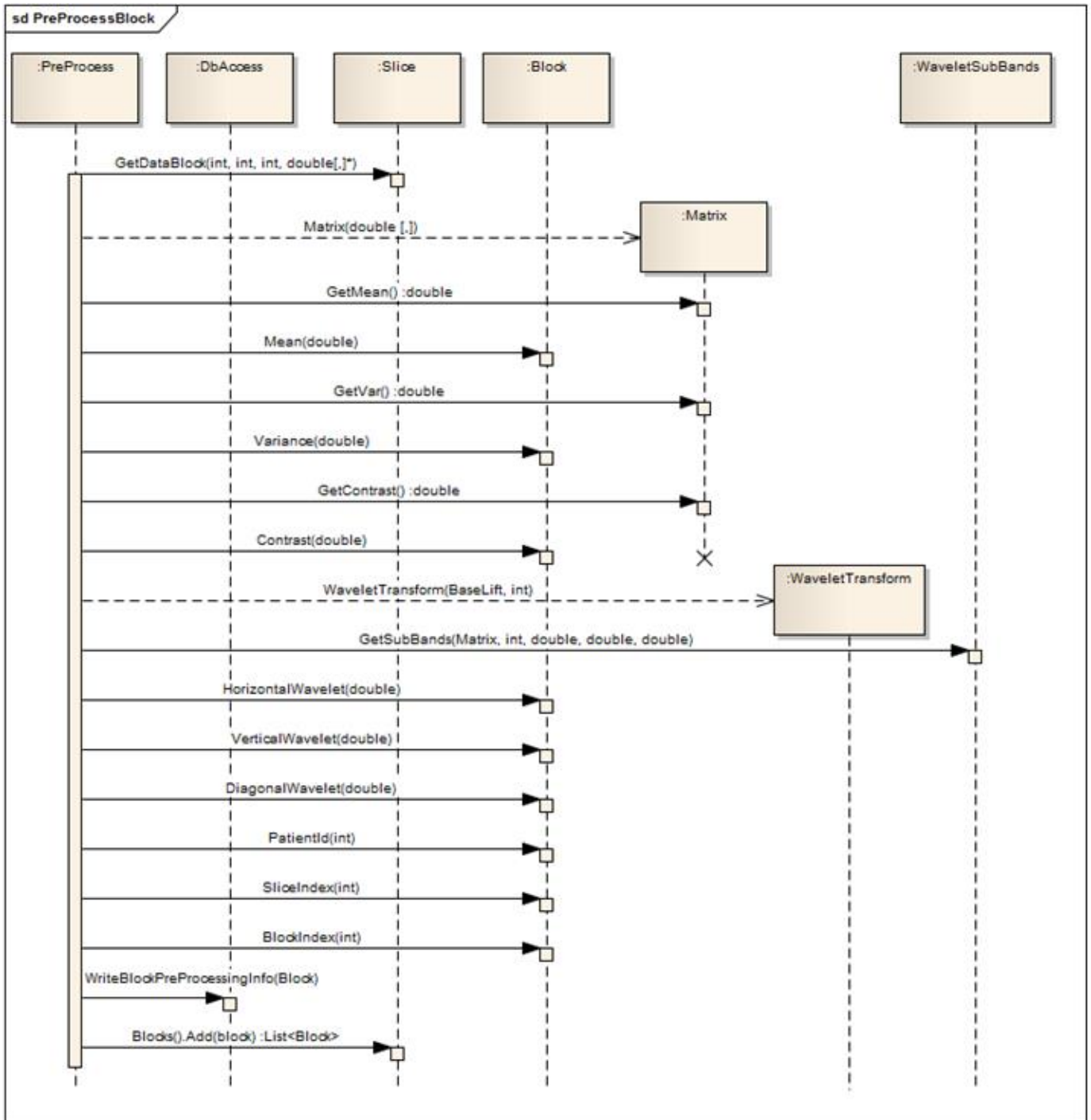
$$Variance = \frac{\sum_{i=1}^N (Pixel\ i\ Grayscale\ value - mean)^2}{N - 1}$$

חישובי horizontal band of wavelet transform, vertical band of wavelet transform, diagonal band of wavelet transform ראה 5.2.1.2.

$$contrast = \frac{Grayscale_{max} - Grayscale_{min}}{Grayscale_{max} + Grayscale_{min}}$$

חישוב ה-contrast נעשה בשיטת Michelson [19], אשר מיועדת לחישוב קונטרסט בתמונות שבהן יש מאפיינים בהירים וגם מאפיינים כהים.
כאשר $Grayscale_{max}$ הוא ערך ה-Grayscale המקסימלי בבלוק, ו- $Grayscale_{min}$ הוא ערך ה-Grayscale המינימלי בבלוק.

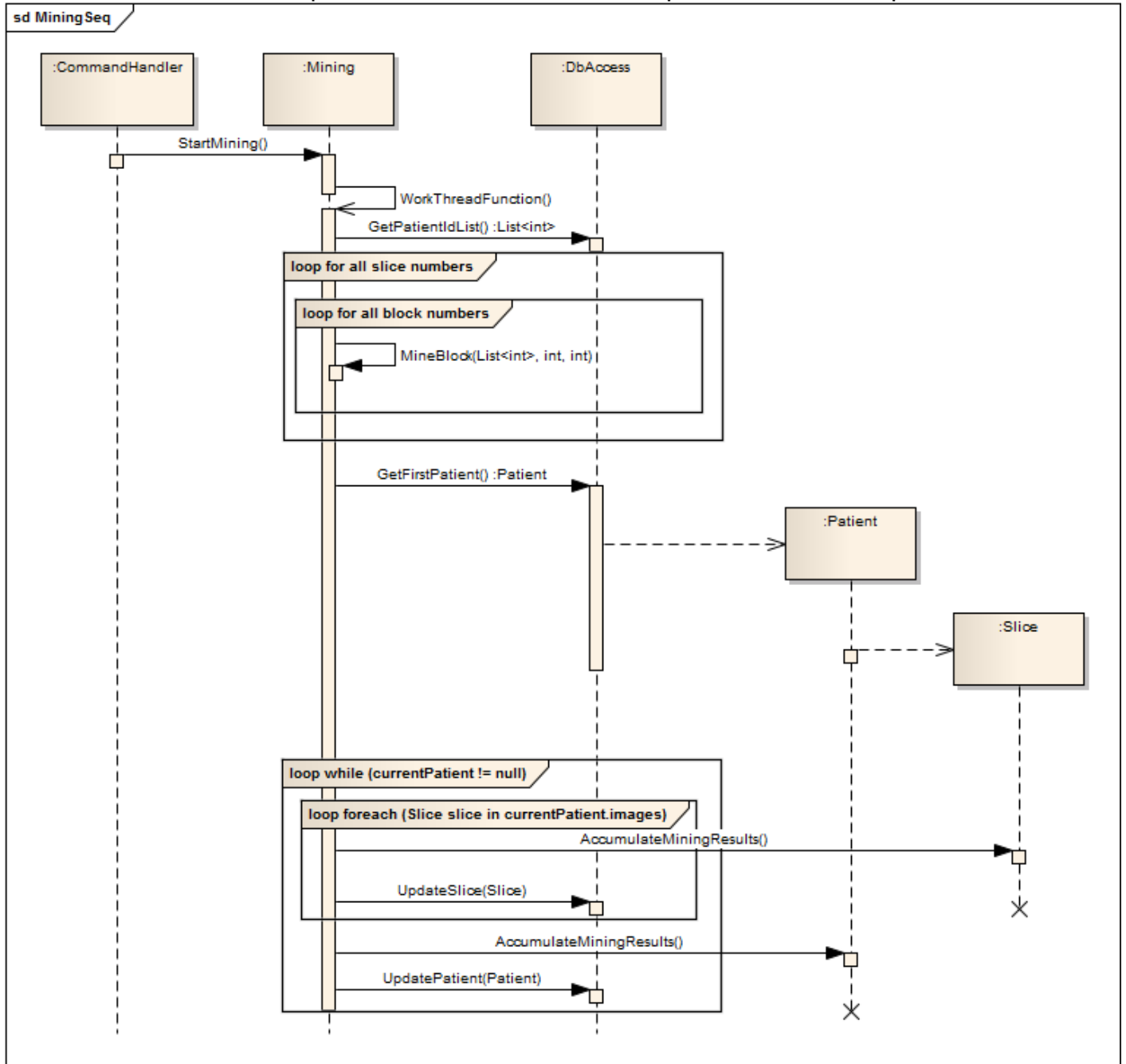
פלט: שמירת המאפיינים לכל בלוק בבסיס הנתונים.



איור 9 - דיאגרמת רצף של תהליך PreProcessBlock

3.6.4 תהליך ה-Mining

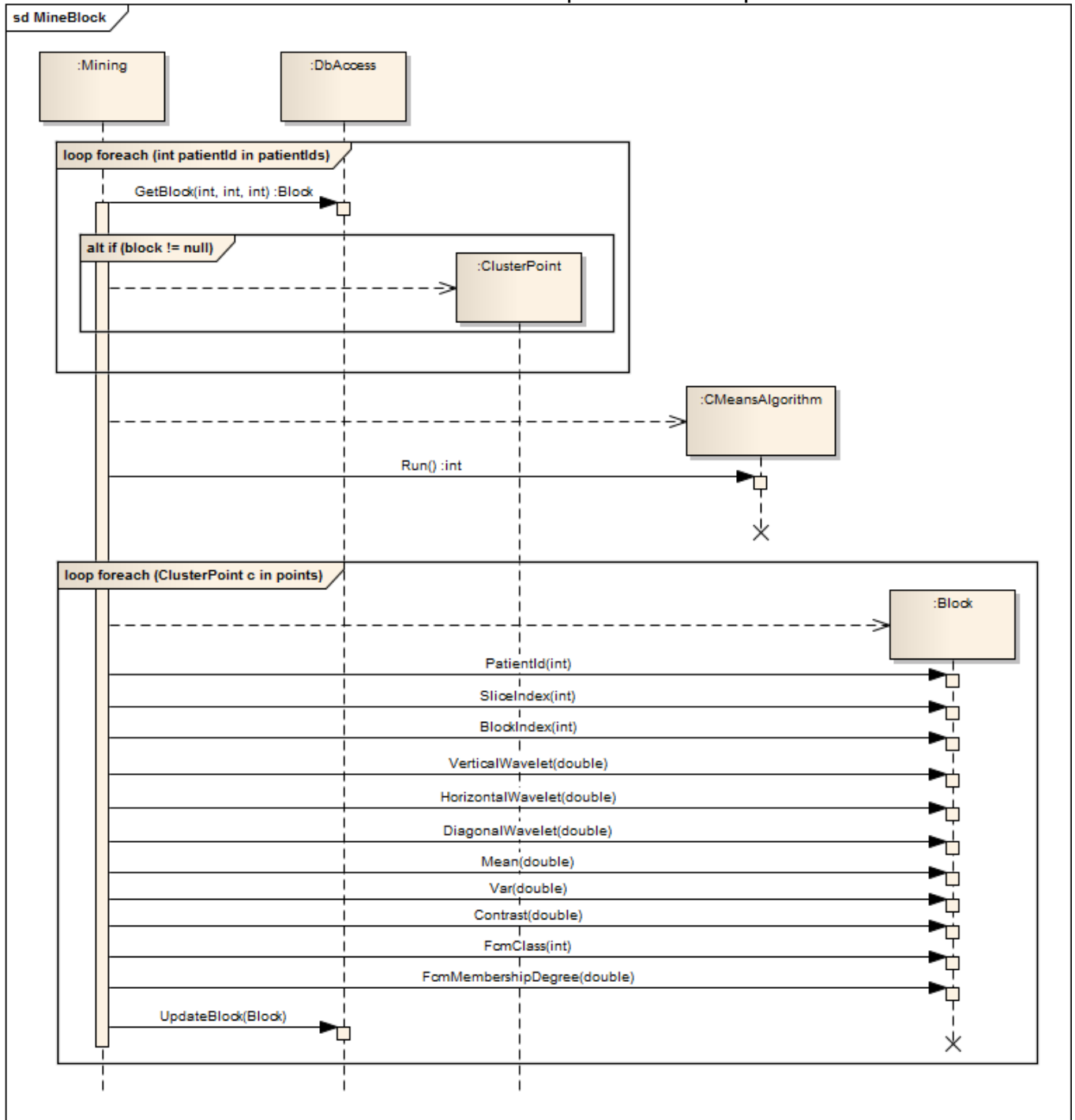
מטרה: מטרת התהליך לבצע כריית מידע לכל הנבדקים, לקבל כתוצאה אשכול לתקין/ פתולוגי ומידת ביטחון באשכול, ולשמור את האשכול ומידת הביטחון בבסיס הנתונים.
 קלט: בסיס נתונים המכיל טבלת נבדקים, תמונות לכל נבדק, ובלוקים שעברו עיבוד מקדים לכל תמונה.
 עיבוד: בתהליך כריית המידע מתבצע מעבר על כל הנבדקים במערכת, כאשר לכל נבדק מתבצע מעבר על כל התמונות שלו. בכל תמונה מתבצע מעבר על כל הבלוקים שלה. לכל בלוק מבוצע כריית מידע לבלוק שיתואר בתהליך הבא (3.6.5). לכל תמונה ולכל נבדק יחושב שקלול של האשכול ומידת הביטחון באשכול.
 פלט: שמירת האשכול לכל בלוק בבסיס הנתונים (מתבצע על ידי התהליך הפנימי 3.6.5) ושמירה בבסיס הנתונים של האשכול, מידת הביטחון באשכול וסטטוס תהליך כריית המידע לכל תמונה ולכל נבדק.



איור 10 - דיאגרמת רצף של תהליך ה-Mining

3.6.5 תהליך MineBlock

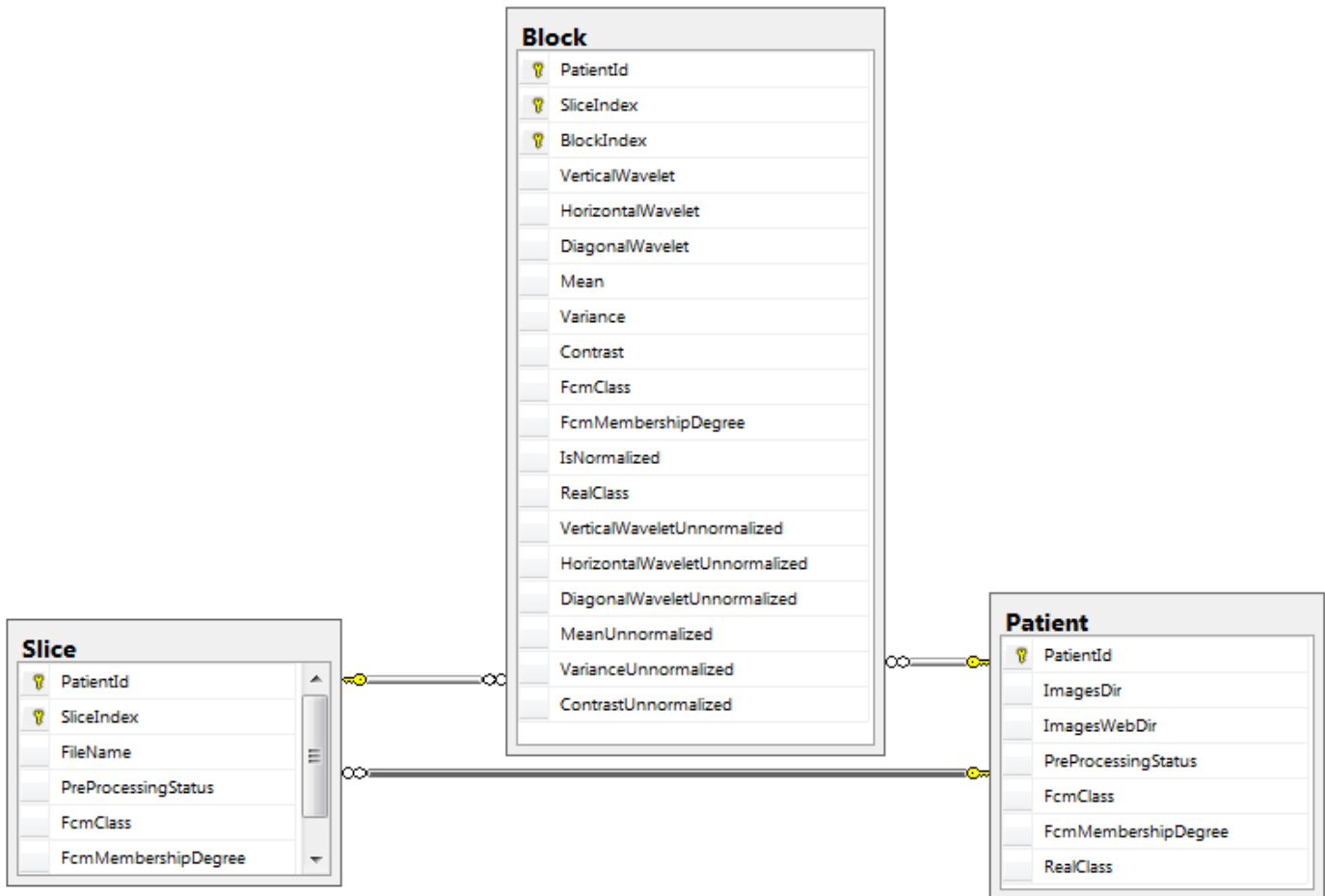
מטרה: תהליך זה הוא למעשה תת-תהליך המופעל על ידי תהליך ה-Mining (3.6.4). מטרת התהליך לבצע כריית מידע לבלוק ספציפי בתמונה ספציפית על כל הנבדקים. קלט: בסיס נתונים המכיל טבלת נבדקים, תמונות לכל נבדק, ובלוקים שעברו עיבוד מקדים לכל תמונה. עיבוד: אלגוריתם FCM מתואר בפסאודו קוד בהמשך באלגוריתם 1 – FCM (5.2.1.3). פלט: שמירת האשכול ומידת הביטחון באשכול לכל בלוק בבסיס הנתונים.



איור 11 - דיאגרמת רצף של תהליך MineBlock

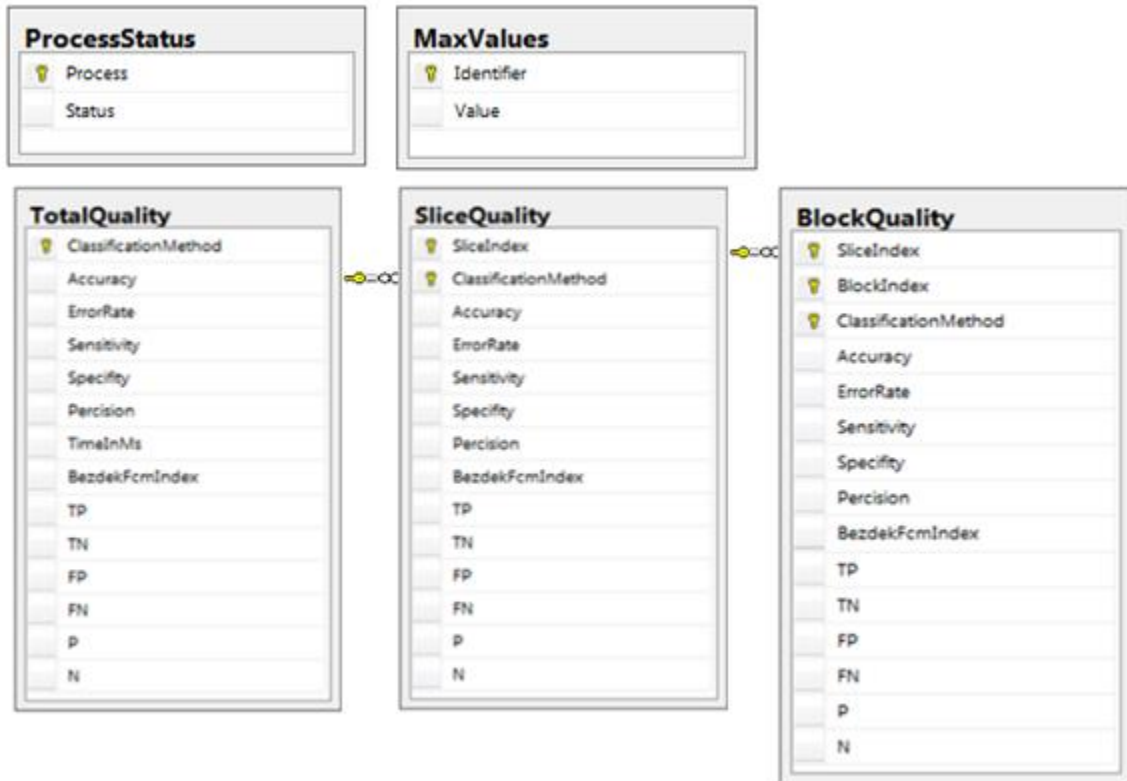
4 תכנן בסיס הנתונים

בסיס הנתונים בפרויקט הוא Microsoft Sql Server [20].
שלוש הישויות המרכזיות הן Patient, Slice, Block.
המפתח המזהה את Patient הוא PatientId,
המפתח המזהה את Slice הוא PatientId+SliceIndex,
והמפתח המזהה את Block הוא PatientId+SliceIndex+BlockIndex.



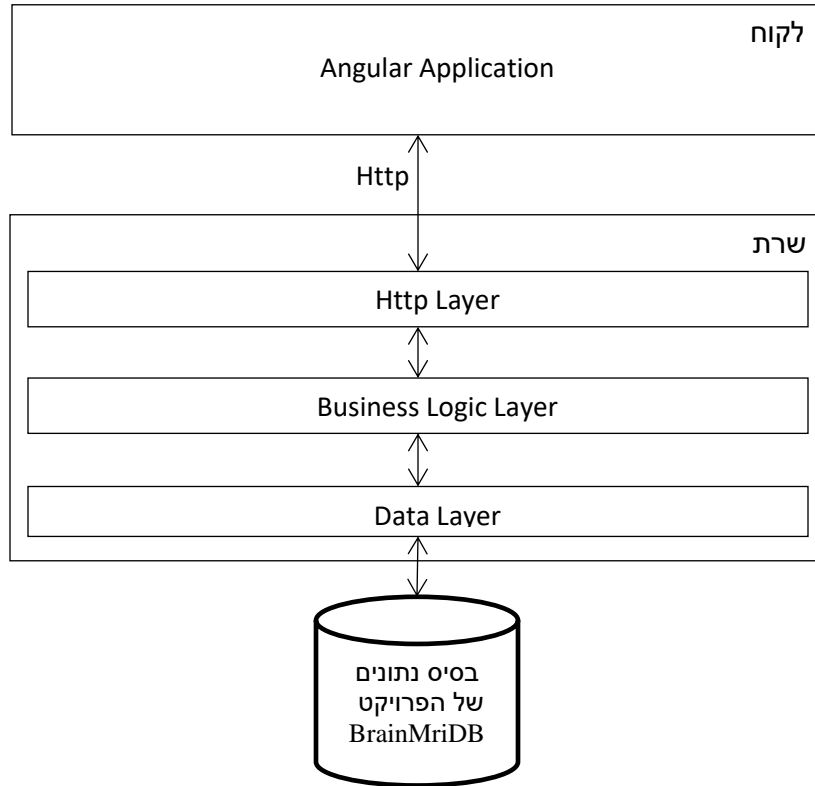
ישויות נוספות במערכת:

ProcessStatus – בטבלה זאת נחזיק שורה לכל תהליך (אחת לתהליך PreProcess ואחת לתהליך Mining). המפתח הוא התהליך, ובנוסף נחזיק סטטוס לכל תהליך.
MaxValues – בטבלה זאת נחזיק את ערכי המקסימום למספר בלוק ולמספר slice. טבלאות Quality לבלוק, ל-slice ו-total: בטבלאות אלו נמדוד את איכות התוצאות של האשכול. לכל בלוק נמדוד את האיכות לעומת ערכי סיווג ידועים מראש. כל slice יכיל ממוצע מדדי איכות של כל הבלוקים באותו slice. בטבלת TotalQuality נשמור את ממוצע מדדי האיכות של כל התמונות במערכת.



5 ארכיטקטורת מערכת FCBM

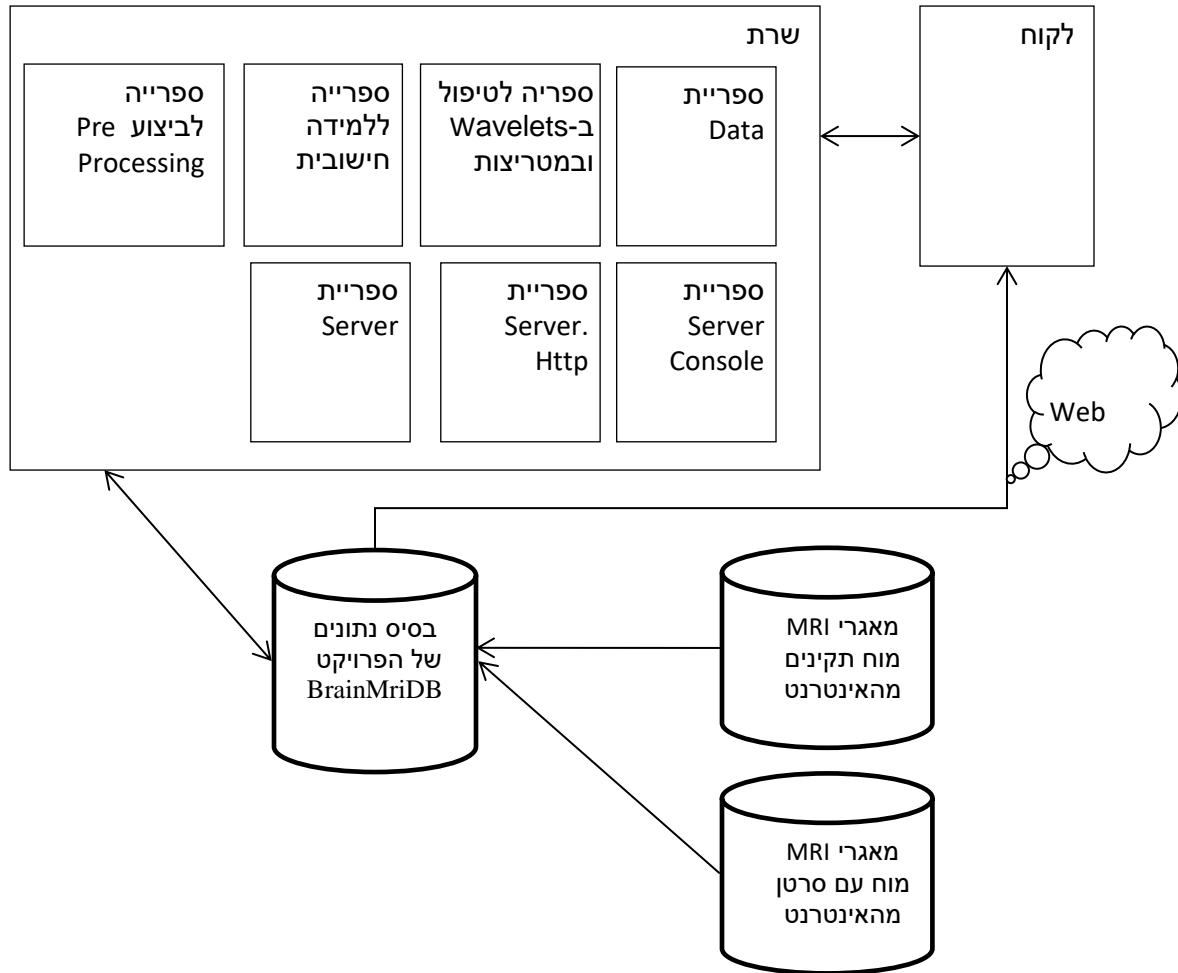
5.1 שכבות מערכת FCBM



איור 12 – שכבות מערכת FCBM

5.2 מבנה מערכת FCBM

מערכת FCBM היא מערכת שרת לקוח. הרכיבים העיקריים של המערכת מתוארים להלן באיור 13 – רכיבי מערכת FCBM.



איור 13 – רכיבי מערכת FCBM

5.2.1 שרת

השרת הוא זה שמבצע את העיבוד המקדים, ואת כריית המידע, ושומר את תוצאות התהליכים לבסיס הנתונים.

בנוסף, השרת מדבר עם הלקוח באמצעות HTTP ומקבל מהלקוח בקשות ומחזיר ללקוח תשובות. הבקשות שהשרת מטפל בהן כוללות בקשה לקבלת נתוני כל הנבדקים, בקשה לקבלת נתוני נבדק מסוים, בקשה לנתוני תמונה מסוימת, בקשה להתחלת תהליך העיבוד המקדים ל-FCM, בקשה להתחלת תהליך כריית המידע, ובקשה לקבלת סטטוס התהליכים במערכת.

רכיבי השרת הם: ספרייה לביצוע Preprocessing, ספרייה לטיפול ב-Wavelets ובמטריצות, ספרייה ללמידה חישובית, ספריית Data, ספריית ServerConsole, ספריית Server.Http, וספריית Server. להלן תיאור רכיבי השרת:

5.2.1.1 ספרייה לביצוע Preprocessing

ספרייה אשר מבצעת עיבוד מקדים על הנתונים בקלט. כל תמונה תחולק לבלוקים. כל בלוק יהיה בגודל $n \times n$. הבלוקים יהיו מסודרים כך שכל בלוק מתאר תמונה מסוימת בתוך הסדרה, ומיקום מסוים בתוך התמונה. לכל בלוק יבוצעו קריאות לספרייה לטיפול ב-Wavelets.

5.2.1.2 ספרייה לטיפול ב-Wavelets ובמטריצות

רקע בנושא [wavelets](#) [18]:

Wavelets הם כלי מתמטי לפירוק היררכי של מידע. הם מאפשרים לתאר מידע כצורה כללית באופן גס, בתוספת פרטים אשר מתוארים במספר רמות מפרטים רחבים ועד פרטים צרים. שימוש ב-wavelets מאפשר טכניקה לייצוג רמות של פרטים לגבי המידע שעליו מפעילים את ה-wavelet transform. המידע בתיאור גס הוא ממוצע שנותן רזולוציה נמוכה של המידע המקורי, והפרטים ברמות השונות הם ה-detail coefficients. כלומר שיטת Discrete wavelet transform היא כלי אשר מחלק מידע למרכיבי תדר שונים. השימוש ב-Discrete wavelet transform הוא לצורך קבלת מאפיינים של מידע שבו התדר משתנה עם הזמן.

השימוש ב-Discrete wavelet transform מאפשר לחלק תמונה למרכיביה מבחינת frequency sub-band וגם בהיררכיה של scales. טרנספורמצית פורייר (Fourier transform) נותנת רפרזנטציה של תמונה על פי ה-frequency וללא התייחסות ללוקליות בתוך התמונה. לעומתה Discrete wavelet transform נותנת רפרזנטציה של תמונה גם על פי frequency וגם בהיררכיה של scales ולכן היא מכילה מידע לגבי מספר רזולוציות של התמונה ולגבי לוקליות. תכונות אלו הפכו את ה-Discrete wavelet transform לכלי מועדף לצורך feature extraction.

לצורך ביצוע טרנספורמציה דו ממדית נשתמש בפונקציונליות של טרנספורמציה חד ממדית. ניתן דוגמה לצורך המחשת טרנספורמציה חד ממדית ולאחריה נביא את הפסאודו קוד של טרנספורמציה חד ממדית. נניח שהמערך המקורי הוא [9 7 3 5]. נחשב ממוצע של כל שני ערכים כדי לקבל רזולוציה נמוכה יותר של המערך המקורי ונקבל את המערך: [8 4]. בתהליך המיצוע איבדנו מידע אשר ניתן לשחזור באמצעות ה-detail coefficients. הראשון יהיה 1 משום שצריך להוסיף 1 ל-8 כדי לקבל את המידע המקורי 9 וצריך להוריד 1 מ-8 כדי לקבל את המידע המקורי 9. ה-coefficient השני יהיה -1 משום שצריך להוסיף -1 ל-4 כדי לקבל את המידע המקורי 3 וצריך להוריד -1 מ-4 כדי לקבל את המידע המקורי 5. חזרה על התהליך על המערך [8 4] תיתן פירוט נוסף לרזולוציה נמוכה יותר. נקבל:

Resolution	Averages	Detail coefficients
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

ה-wavelet transform של המערך המקורי הוא הממוצע הסופי ברזולוציה 1, ולאחריה רשימת ה-coefficients בסדר עולה של רזולוציות, כך שנקבל את המערך: [6 2 1 -1].

להלן פסאודו קוד:

```

procedure DecompositionStep(C: array [1..h] of reals)
  for i=1 to h/2 do
    C0 [i] = (C[2i-1] + C[2i]) / sqrt(2)
    C0 [h/2 + i] = (C[2i-1] - C[2i]) / sqrt(2)
  end for
  C=C0
end procedure

```

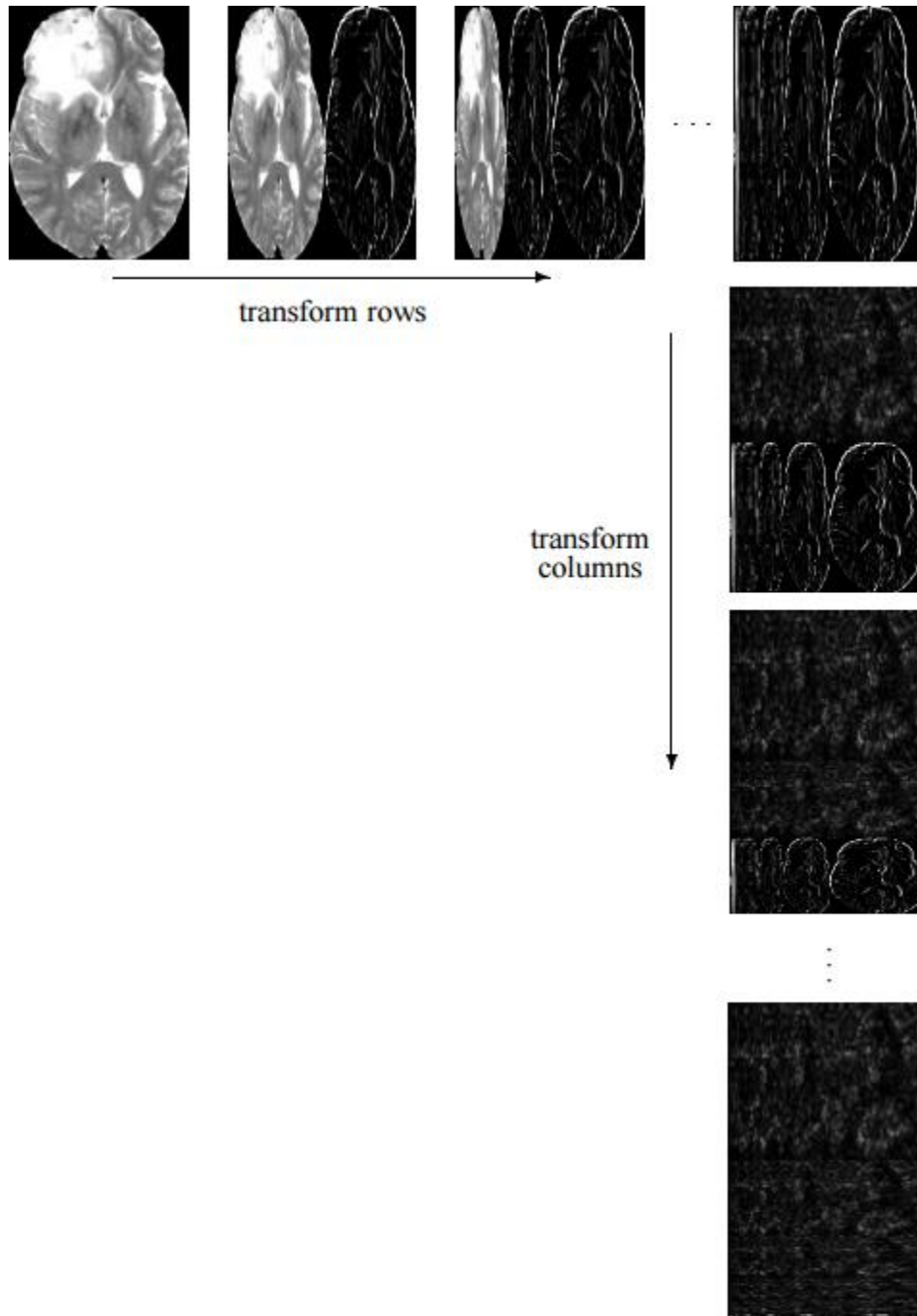
```

procedure WaveletTransform1D (C: array [1..h] of reals)
  C = C / sqrt(h) (normalize input coefficients)
  while h > 1 do
    DecompositionStep(C[1..h])
    h = h/2
  end while
end procedure

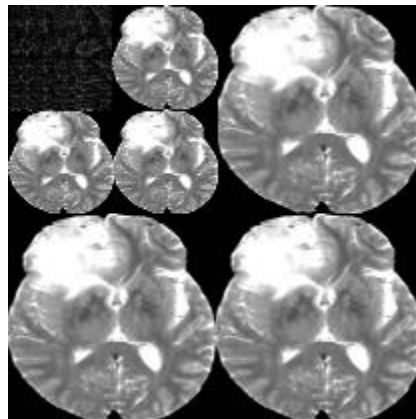
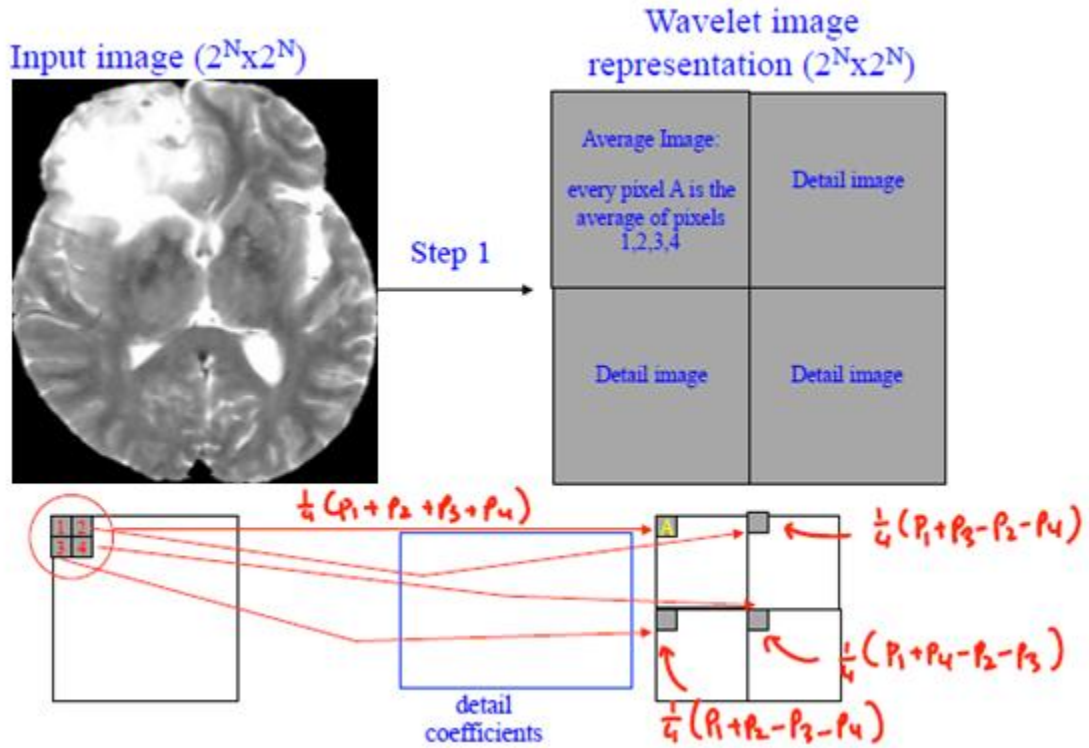
```

אלגוריתם wavelet transform על תמונה דו ממדית הוא למעשה הכללה של האלגוריתם שתואר לעיל לגבי המקרה החד ממדי.
כדי למצוא ערכי coefficients לתמונה אשר מקפלים בחובם מאפיינים של התמונה כולה, האלגוריתם של wavelet transform על תמונה מבצע חישוב בשני שלבים:
ראשית האלגוריתם מפעיל wavelet transform חד ממדי על כל שורת פיקסלים בתמונה. פעולה זאת נותנת ערך ממוצע וערכי coefficients לכל שורה.
שנית האלגוריתם מתייחס לשורות שעברו טרנספורמציה כאילו הן תמונה בפני עצמה, ומבצע טרנספורמציה חד ממדית על כל טור. התוצאה היא ממוצע כללי אחד וערכי coefficients לתמונה כולה.
להלן פסאודו קוד:

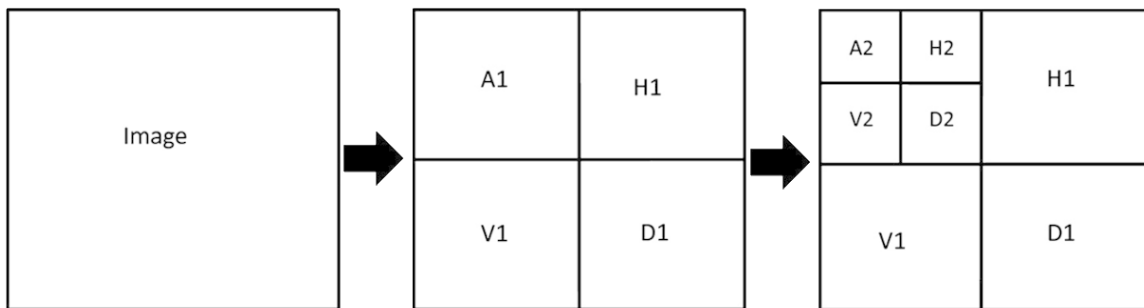
```
procedure WaveletTransform (C: array [1..h, 1..w] of reals)
  for row 1 to h do
    WaveletTransform1D (C[row, 1..w])
  end for
  for col 1 to w do
    WaveletTransform1D (C[1..h, col])
  end for
end procedure
```

איור 14 - wavelet transform של תמונה



איור 15 - אילוסטרציה של wavelet transform על תמונה



איור 16 - חלוקה של תוצאת wavelet transform ל-sub-bands

הספרייה לטיפול ב-Wavelets תחשב מידע לצורך Feature extraction אשר ישמש בהמשך כחלק מהקלט לאלגוריתם FCM. לכל בלוק בכל תמונה נבצע feature extraction (חישוב מאפיינים). המאפיינים שיחושבו לכל בלוק יהיו [14]:

Mean, variance, horizontal band of wavelet transform, vertical band of wavelet transform, diagonal band of wavelet transform, contrast.

חישוב המאפיינים לפי הסימונים באיור 16:

Horizontal band of wavelet transform = Average (H2)

Vertical band of wavelet transform = Average (V2)

Diagonal band of wavelet transform = Average (D2)

5.2.1.3 ספרייה ללמידה חישובית

ספרייה המכילה מימוש לאלגוריתם אשכול עמום FCM clustering. בספרייה זו תהיה גם מחלקה אשר מבצעת את האלגוריתם לאשכול על הנתונים המעובדים במערכת זאת.

נתון N נתונים היוצרים C אשכולות. נגדיר:

- מרכזי אשכולות C_j
- מטריצת שייכות U שבה $U_{i,j}$ הוא מידת השייכות של נתון X_i לאשכול j
- אקספוננט עמימות m מספר ממשי גדול מ-1 אשר קובע את מידת העמימות של תוצאת ההרצה. אם $m=1$ האלגוריתם יצטמצם לאלגוריתם K-Means שמבצע חלוקה קשיחה לאשכולות. במאמר [7] נעשתה בדיקה לגבי השפעת הפרמטר m על תוצאות האשכול. המסקנה הייתה כי יש השפעה על התוצאות אולם קשה לקבוע מראש מהו הפרמטר m האופטימלי, וכי הוא יכול להשתנות עבור נתונים שונים. העמימות בפונקציית המטרה להלן ובאלגוריתם בכלל נקבעת על ידי m .
- אפסילון עצירה ϵ .

אלגוריתם 1 – FCM [6], [13]

1. Randomly select C cluster centers C_j out of the given data.
2. Initialize membership matrix $U = [U_{i,j}]$, for iteration $k=0$: $U^{(0)}$

$$U_{i,j} = \frac{1}{\sum_{k=1}^C \left(\frac{\|X_i - C_j\|}{\|X_i - C_k\|} \right)^{\frac{2}{m-1}}}$$

3. At k iteration: calculate the cluster centers $C^{(k)} = [C_j]$ with $U^{(k)}$

$$C_j = \frac{\sum_{i=1}^N U_{i,j}^m \cdot X_i}{\sum_{i=1}^N U_{i,j}^m}$$

4. Update $U^{(k+1)}$

5. If

$$\max_{i,j} \left\{ \left| U_{i,j}^{(k+1)} - U_{i,j}^{(k)} \right| \right\} < \epsilon$$

then STOP, otherwise return to step 3 at iteration $k+1$.

בספריה ללמידה חישובית מתבצע גם חישוב מדדי איכות לכריית המידע.

הגדרת מדדי האיכות:

נשתמש בסימונים לצורך הסבר המדדים האפשריים לבחינת איכות הסיווג [13]:
Positive (P): מספר הרשומות שהמחלקה שלהם היא המחלקה העיקרית שמעניינת אותנו.
בפרויקט זה P הוא מספר הרשומות שהמחלקה שלהם היא פתולוגית.
Negative (N): מספר הרשומות שהמחלקה שלהם היא מחלקה כלשהי שונה מהמחלקה העיקרית.
בפרויקט זה N הוא מספר הרשומות שהמחלקה שלהם היא נורמלית.
True Positive (TP): מספר הרשומות שהמחלקה שלהם P והמודל סיווג אותם כמחלקה P.
True Negative (TN): מספר הרשומות שהמחלקה שלהם N והמודל סיווג אותם כמחלקה N.
False Positive (FP): מספר הרשומות שהמחלקה שלהם N והמודל סיווג אותם (באופן שגוי) כמחלקה P.
False Negative (FN): מספר הרשומות שהמחלקה שלהם P והמודל סיווג אותם (באופן שגוי) כמחלקה N.
מדדים לבחינת טיב הסיווג:

מדד הדיוק - accuracy

הדיוק של המודל הוא אחוז הרשומות שסווגו נכון מתוך סה"כ הרשומות:

$$\text{Accuracy} = \frac{(TP + TN)}{(P + N)}$$

מדד הדיוק רלוונטי ביותר כאשר התפלגות הרשומות לשתי המחלקות (P, N) היא התפלגות כמעט אחידה.

מדד קצב השגיאות - error rate

קצב השגיאות הוא מדד המשלים ל-1 של מדד הדיוק. המסווג איכותי יותר ככל שקצב השגיאות נמוך יותר. מדד קצב השגיאות רלוונטי ביותר כאשר התפלגות הרשומות לשתי המחלקות (P, N) היא התפלגות כמעט אחידה.

$$\text{Error rate} = 1 - \text{accuracy} = \frac{(FP + FN)}{(P + N)}$$

מדד הרגישות - sensitivity

כאשר המחלקה העיקרית שמעניינת אותנו נדירה בהרבה מכל המחלקות האחרות, אנו עשויים לקבל מדד דיוק טוב, אולם המודל שנבנה עלול לסווג באופן שגוי את אותן רשומות שחשוב לסווג אותן נכון. לדוגמה כאשר אנו בודקים האם נבדק חולה (P) או בריא (N), ואחוז החולים נמוך, אנו מעדיפים לסווג באופן שגוי את הבריאים (ואולי לגרום להם לבצע עוד בדיקות) מאשר לטעות ולסווג כבריא אדם חולה, משום שאם אדם חולה חשוב שנדע זאת כדי שהוא יקבל טיפול. במצבים כאלו נשתמש במדד רגישות. מדד זה מחשב את היכולת של המודל לסווג נכון את המחלקה P. ככל שמדד רגישות גבוה יותר אז הסיווג טוב יותר בזיהוי חולים.

$$\text{Sensitivity} = \frac{(TP)}{(P)} = \frac{TP}{(TP + FN)}$$

מדד הסגוליות (ספציפיות) - specificity

כאשר התפלגות הרשומות לשתי המחלקות (P, N) אינה אחידה אולם מעניין אותנו יותר למדוד את היכולת של המודל לסווג נכון את המחלקה N נשתמש במדד הספציפיות.

$$\text{Specificity} = \frac{(TN)}{(N)} = \frac{TN}{(TN + FP)}$$

מדד ה-precision

מדד שבדוק את דיוק הסיווג של המחלקה P. מתוך כל הפעמים שהמודל סיווג P כמה היו באמת P.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

5.2.1.4 ספריית Data

ספרייה המכילה את כל ישויות המידע במערכת. בספרייה זאת ימומשו גם כל הגישות לבסיס הנתונים.

5.2.1.5 ספריית ServerConsole

זו הספרייה המכילה את הקוד לאתחול השרת. היא המפעילה את האפליקציה (exe) של השרת.

5.2.1.6 ספריית Server.Http

ספרייה המטפלת בקליטת מסרי Http מהלקוח.

5.2.1.7 ספריית Server

ספרייה המטפלת בניהול פקודות והפעלת הפונקציונליות המתאימה.

5.2.2 לקוח

בצד הלקוח המשתמש יכול באמצעות ממשק משתמש להתחיל כל אחד מהתרחישים שתוארו בסעיף 3.4.

רכיבי הלקוח הם:

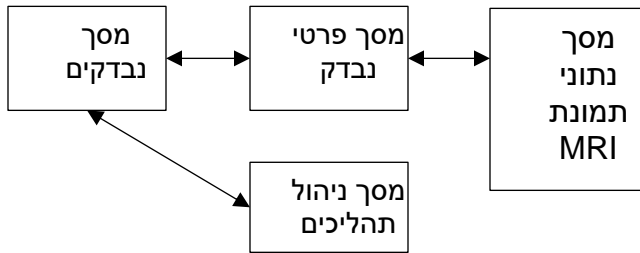
- דפי html להתחלת תהליכים, להצגת רשימת נבדקים ולהצגת פרטי נבדק בודד.
- קבצי javascript המכילים את הלוגיקות בצד הלקוח
- Styles הנמצאים בשימוש בלקוח

בצד הלקוח ניתן לבצע את הפעולות הבאות:

- לפני ביצוע כריית המידע:
 1. צפייה ברשימת נבדקים.
 2. בחירת נבדק מרשימת הנבדקים וצפייה בפרטי נבדק מפורטים – ללא הסיווג שלו לתקין או פתולוגי, וללא אינדיקציה ויזואלית ל-slices הפתולוגיים.
 3. עבור נבדק מסוים בחירת slice וצפייה ב-slice הנבחר ללא אינדיקציה של הבלוקים הפתולוגיים.
- התחלת תהליך עיבוד מקדים ל-FCM.
- כריית מידע
 1. התחלת תהליך כריית מידע. סיום ביצוע PreProcessing חייב להיות קודם לבקשה לביצוע Data mining.
 - לאחר כריית מידע
 1. צפייה במדדי איכות של תוצאות כריית המידע.
 2. צפייה ברשימת נבדקים.
 3. בחירת נבדק מרשימת הנבדקים וצפייה בפרטי נבדק מפורטים כולל הסיווג שלו לתקין או פתולוגי, וכולל אינדיקציה ויזואלית ל-slices הפתולוגיים.
 4. עבור נבדק מסוים בחירת slice וצפייה באינדיקציה ויזואלית לבלוקים הפתולוגיים.

6 ממשק משתמש

להלן עץ מסכים המדגים ברמת על את הניווט בין המסכים:



איור 17 - עץ מסכים

נתאר את מסכי המערכת לפי סדר הפעולות הניתנות לביצוע שתוארו בסעיף 0.

1. **מסך נבדקים לפני ביצוע כריית מידע:** בעת הרצת תוכנת הלקוח לראשונה יוצג מסך נבדקים לפני ביצוע כריית מידע.

הטור FCM Predict מציין את השיוך לאשכול על פי כריית המידע.

הטור FCM Certainty % מציין את % הביטחון בשיוך לאשכול.

הטור Actual מציין האם נבדק זה תקין או פתולוגי לפי ידע קודם.

מכיוון שעדיין לא בוצעה כריית מידע, לא ידוע השיוך לאשכול לאף נבדק ולכן FCM Predict יהיה Unknown ו-FCM Certainty יהיה 0.00% בכל השורות.

ניתן להגיע למסך זה בכל זמן על ידי לחיצה על "List Patients" בשורת התפריט העליונה.

List Patients Manage Processes			
Patients			
id	FCM Predict	FCM Certainty %	Actual
1	Unknown	0.00%	Normal
2	Unknown	0.00%	Normal
4	Unknown	0.00%	Normal
5	Unknown	0.00%	Normal
6	Unknown	0.00%	Normal
10	Unknown	0.00%	Normal
11	Unknown	0.00%	Normal
12	Unknown	0.00%	Normal
13	Unknown	0.00%	Normal
17	Unknown	0.00%	Normal
18	Unknown	0.00%	Normal
19	Unknown	0.00%	Normal
23	Unknown	0.00%	Normal

איור 18 - מסך נבדקים לפני ביצוע כריית מידע

2. **מסך פרטי נבדק לפני ביצוע כריית מידע:** מתוך מסך רשימת נבדקים, ניתן לבחור נבדק על ידי לחיצה על id ולעבור למסך פרטי נבדק. במסך זה ניתן לראות את פרטי הנבדק ואת תמונות ה-MRI שלו. מכיוון שעדיין לא בוצעה כריית מידע הנבדק יוצג ללא הסיווג שלו לתקין או פתולוגי, וללא אינדיקציה ויזואלית ל-slices הפתולוגיים, כלומר FCM Class יהיה Unknown, ו-FCM Certainty יהיה 0.00%, ותמונות ה-MRI יהיו עם מסגרת לבנה.

Id	FCM Class	FCM Certainty %	Real Class
101	Unknown	0.00%	Abnormal

Filter: Show All ▼

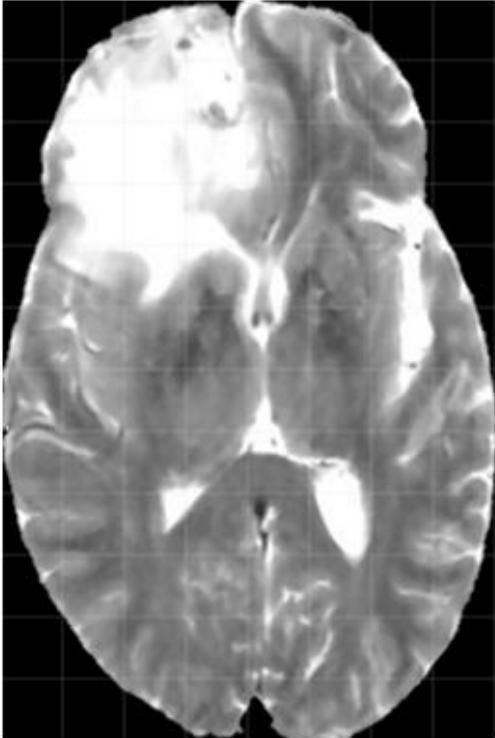
איור 19 - מסך פרטי נבדק לפני ביצוע כריית מידע

3. **מסך נתונים מפורט של תמונת MRI לפני ביצוע כריית מידע:** מתוך מסך פרטי נבדק בחירת slice ולחיצה עליו תעביר אותנו למסך מפורט של תמונת MRI. במסך זה מוצגת תמונת MRI בודדת, ומכיוון שעדיין לא בוצעה כריית מידע לא יהיו בלוקים אדומים המציינים פתולוגיה. ניתן לעבור עם החיצים (ימינה ושמאלה) בין התמונות של אותו נבדק במידה ויש לו מספר תמונות MRI.

Patient Id: 101 (slice #: 100) ×

Patient	Slice#
101	100

Toggle suspected area

<

>

איור 20 - מסך נתונים מפורט של תמונת MRI לפני ביצוע כריית מידע

4. **מסך ניהול תהליכים לפני ביצוע עיבוד מקדים ולפני ביצוע כריית מידע:** ניתן להגיע למסך זה על ידי לחיצה על "Manage Processes" בשורת התפריט העליונה. כאשר לא בוצע עיבוד מקדים ולא בוצעה כריית מידע סטטוס שני התהליכים יהיה Not Started ותוצאות כריית המידע (Data Mining Results) יהיו ריקות.

The screenshot shows a web interface with a green header containing a logo and navigation links for "List Patients" and "Manage Processes". Below the header, a "Processing" status bar is visible. The main content area is divided into three sections:

- Pre Processing:** Contains a green button labeled "START PRE PROCESSING" and the text "Pre processing status: NotStarted".
- Data Mining:** Contains a green button labeled "START DATA MINING" and the text "Data mining status: NotStarted".
- Data Mining Results:** Lists several metrics with their respective units:
 - Accuracy: %
 - Error Rate: %
 - Sensitivity: %
 - Specifity: %
 - Percision: %
 - Time: sec
 - Bezdek FCM Index: %

איור 21 - מסך ניהול תהליכים לפני ביצוע עיבוד מקדים ולפני ביצוע כריית מידע

לחיצה על "START PRE PROCESSING" במסך זה תגרום ל התחלת תהליך העיבוד המקדים.

5. **מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ולפני ביצוע כריית מידע:** לאחר ביצוע העיבוד המקדים ולפני ביצוע כריית המידע סטטוס ה-Pre processing יהיה Complete, סטטוס כריית המידע יהיה Not Started ותוצאות כריית המידע (Data Mining Results) יהיו ריקות.

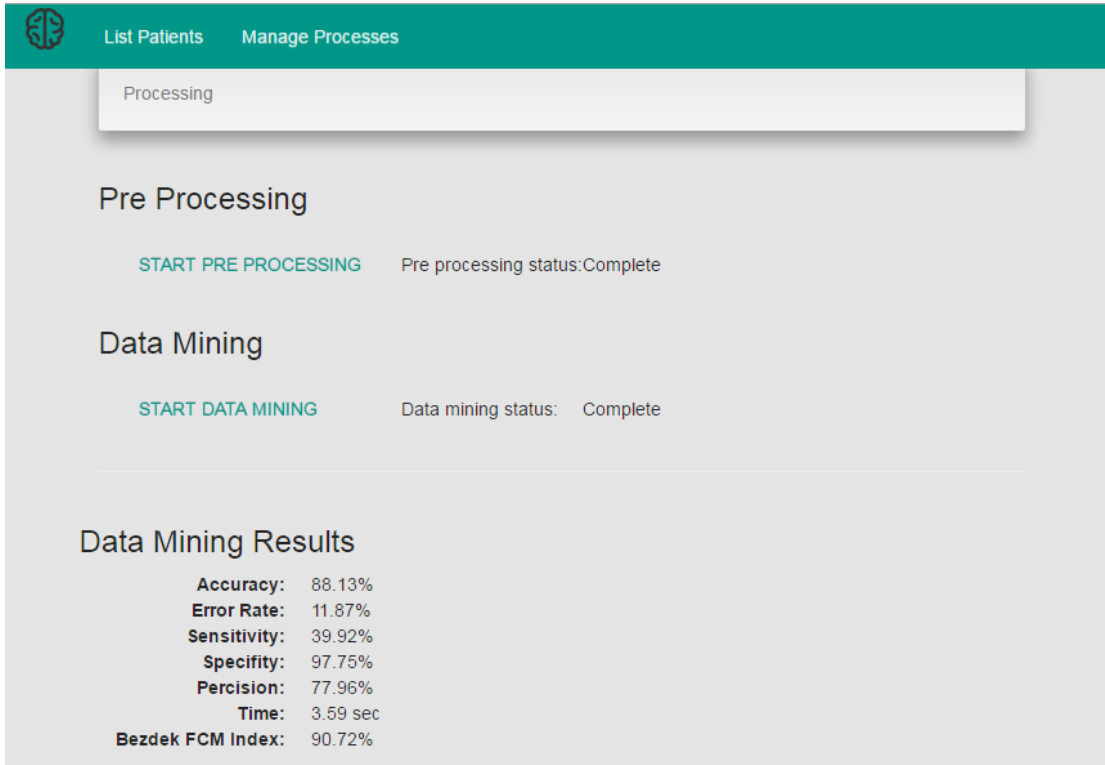
The screenshot shows a web application interface with a teal header containing 'List Patients' and 'Manage Processes' buttons. Below the header is a 'Processing' status bar. The main content area is divided into three sections:

- Pre Processing:** Contains a 'START PRE PROCESSING' button and the text 'Pre processing status: Complete'.
- Data Mining:** Contains a 'START DATA MINING' button and the text 'Data mining status: NotStarted'.
- Data Mining Results:** Lists the following metrics:
 - Accuracy: %
 - Error Rate: %
 - Sensitivity: %
 - Specificity: %
 - Percision: %
 - Time: sec
 - Bezdek FCM Index: %

איור 22 - מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ולפני ביצוע כריית מידע

לחיצה על "START DATA MINING" במסך זה תגרום ל התחלת תהליך כריית המידע.

6. **מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ואחרי ביצוע כריית מידע:** לאחר ביצוע כריית המידע סטטוס ה-Pre processing וסטטוס כריית המידע יהיה Complete, ותוצאות כריית המידע (Data Mining Results) יהיו מלאות. ניתן לצפות במדדי האיכות של תוצאות כריית המידע. אופן חישוב מדדי האיכות של כריית המידע בסעיף 5.2.1.3.



The screenshot displays a web application interface with a teal header containing a logo and navigation links for 'List Patients' and 'Manage Processes'. A 'Processing' status bar is visible. The main content area is divided into three sections: 'Pre Processing' with a 'START PRE PROCESSING' button and 'Pre processing status: Complete'; 'Data Mining' with a 'START DATA MINING' button and 'Data mining status: Complete'; and 'Data Mining Results' which lists the following metrics: Accuracy: 88.13%, Error Rate: 11.87%, Sensitivity: 39.92%, Specificity: 97.75%, Percision: 77.96%, Time: 3.59 sec, and Bezdek FCM Index: 90.72%.

Metric	Value
Accuracy	88.13%
Error Rate	11.87%
Sensitivity	39.92%
Specificity	97.75%
Precision	77.96%
Time	3.59 sec
Bezdek FCM Index	90.72%

איור 23 - מסך ניהול תהליכים אחרי ביצוע עיבוד מקדים ואחרי ביצוע כריית מידע

7. **מסך נבדקים אחרי ביצוע כריית מידע:** לאחר ביצוע כריית המידע ניתן להגיע למסך זה על ידי לחיצה על "List Patients" בשורת התפריט העליונה. תוצאות FCM Predict, FCM Certainty יהיו בעלות ערכים משמעותיים לאחר ביצוע כריית המידע.

id	FCM Predict	FCM Certainty %	Actual
1	Normal	97.05%	Normal
2	Normal	96.17%	Normal
4	Normal	95.86%	Normal
5	Normal	97.16%	Normal
6	Normal	97.80%	Normal
10	Normal	95.16%	Normal
11	Normal	96.19%	Normal
12	Normal	97.19%	Normal
13	Normal	96.42%	Normal
17	Normal	96.50%	Normal
18	Normal	96.28%	Normal
19	Normal	96.72%	Normal
23	Normal	97.54%	Normal
100	Abnormal	95.12%	Abnormal
101	Abnormal	89.42%	Abnormal
102	Abnormal	95.69%	Abnormal
103	Abnormal	74.39%	Abnormal
104	Abnormal	85.20%	Abnormal
105	Abnormal	85.78%	Abnormal
106	Abnormal	95.98%	Abnormal
107	Abnormal	72.58%	Abnormal
108	Abnormal	84.23%	Abnormal
109	Abnormal	84.29%	Abnormal
110	Abnormal	92.22%	Abnormal
111	Abnormal	97.05%	Abnormal
112	Abnormal	83.64%	Abnormal
114	Abnormal	90.18%	Abnormal
115	Abnormal	82.43%	Abnormal
116	Abnormal	83.10%	Abnormal

איור 24 - מסך נבדקים אחרי ביצוע כריית מידע

8. **מסך פרטי נבדק אחרי ביצוע כריית מידע:** מתוך מסך רשימת נבדקים, ניתן לבחור נבדק על ידי לחיצה על id ולעבור למסך פרטי נבדק. במסך זה ניתן לראות את פרטי הנבדק ואת תמונות ה-MRI שלו. לאחר ביצוע כריית מידע, פרטי הנבדק יכללו את הסיווג שלו לתקין או פתולוגי, ויכללו אינדיקציה ויזואלית ל-slices הפתולוגיים. תמונה בעלת מחלקה לא ידועה תופיע ללא מסגרת. תמונה נורמלית תופיע עם מסגרת ירוקה. תמונה פתולוגית תופיע עם מסגרת אדומה.

מימין יש combo box עם אפשרות לבחירת Filter: כל התמונות, תמונות נורמליות, תמונות פתולוגיות או תמונות שמחלקתן אינה ידועה. מופיע בצילום המסך שלהלן בצבע תכלת עם הבחירה "Show All".

Id	FCM Class	FCM Cerainty %	Real Class
101	Abnormal	89.42%	Abnormal

Filter: Show All

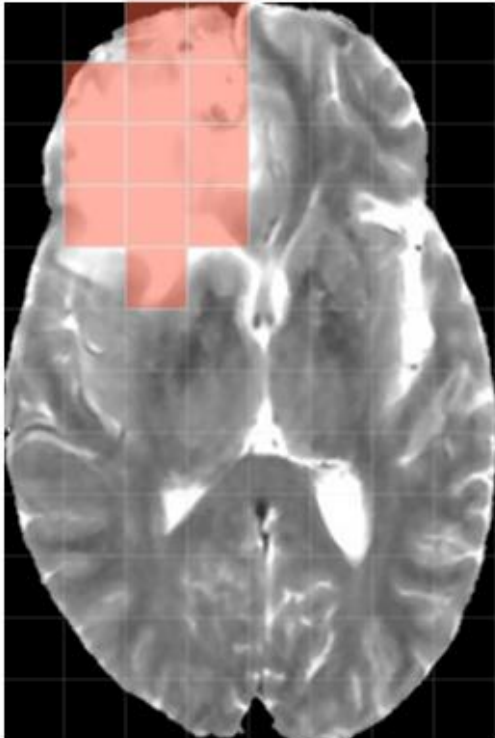
איור 25 - מסך פרטי נבדק אחרי ביצוע כריית מידע

9. **מסך נתונים מפורט של תמונת MRI אחרי ביצוע כריית מידע:** מתוך מסך פרטי נבדק בחירת slice ולחיצה עליו תעביר אותנו למסך מפורט של תמונת MRI. במסך זה מוצגת תמונת MRI בודדת ומוצגת אינדיקציה ויזואלית לבלוקים הפתולוגיים. האינדיקציה לבלוק פתולוגי היא צביעת הבלוק באדום כמתואר בתמונה. יהיו בלוקים אדומים רק בתמונות שזוהו בהן בלוקים אבנורמליים. באמצעות ה-Toggle יש אפשרות להציג או להסתיר את הבלוקים המזוהים כפתולוגיים. בנוסף ניתן לעבור עם החיצים בין התמונות של אותו נבדק במידה ויש לו מספר תמונות MRI.

Patient Id: 101 (slice #: 100) ×

Patient	Slice#
101	100

Toggle suspected area

<

>

איור 26 - מסך נתונים מפורט של תמונת MRI אחרי ביצוע כריית מידע

Server

ספריות הקוד של השרת מפורטות ב-5.2.1.

השרת הוא זה שמבצע את העיבוד המקדים, ואת כריית המידע, ושומר את תוצאות התהליכים לבסיס הנתונים. בנוסף, השרת מדבר עם הלקוח באמצעות HTTP ומקבל מהלקוח בקשות ומחזיר ללקוח תשובות. הבקשות שהשרת מטפל בהן כוללות בקשה לקבלת נתוני כל הנבדקים, בקשה לקבלת נתוני נבדק מסוים, בקשה לנתוני תמונה מסוימת, בקשה להתחלת תהליך העיבוד המקדים ל-FCM, בקשה להתחלת תהליך כריית המידע, ובקשה לקבלת סטטוס התהליכים במערכת.

בנוסף, כדי לאפשר לתמונות להיות מאוחסנות בדיסק במחשב השרת או בשרת נפרד כלשהו, יש צורך להעביר ללקוח את התמונות המוצגות באמצעות פרוטוקול HTTP. לצורך כך נעשה שימוש בשרת פשוט וחינמי הנקרא http-server אשר משמש בפרויקט זה להעברת תמונות מהשרת ללקוח באמצעות HTTP. הנחיות להתקנת והרצת http-server נמצאות בספחים 11.4, 11.5.

- Editor and Debugger: Microsoft Visual Studio 2015 [30]
- C#
- HTTP server (Node.js)

Client

פלטפורמת AngularJS היא תשתית open source לפיתוח צד הלקוח של אפליקציות Web, שפותחה על ידי חברת גוגל ומתוחזקת על ידי גוגל ועל ידי קהילה רחבה של מפתחים. הפלטפורמה בנויה לפי design pattern של MVC (Model View Controller). מערכות קוד פתוח מאפשרות למאות אלפי מתכנתים מכל העולם לתרום מניסיונם ומהידע המקצועי שלהם. מערכות אלה נחשבות אמינות בהשוואה למערכות סגורות, ולכן הן משמשות גם לטובת בניית אתרי אינטרנט. ספריית AngularJS עובדת על ידי קריאת דף ה-HTML, כאשר אל התגיות שבו נוספו תכונות נוספות. פלטפורמת Angular תפרש את התכונות הללו כהנחיות לקשר את אזורי הקלט או הפלט בדף למודל שמיוצג על ידי משתני JavaScript פשוטים. הערכים של משתני ה-JavaScript האלו ניתנים לשינוי על ידי קוד, וניתן לגשת אליהם בצורה סטטית, או בצורה דינאמית בעזרת JSON (הרחבה בהמשך). שימוש בקוד HTML הוא מצוין לעבודה עם דפי אינטרנט סטטיים אך הוא מאתגר בכל פעם שנדרשת דינמיות באותם הדפים. פלטפורמת Angularjs מאפשרת להרחיב את אוצר המילים של קוד ה-HTML וכתוצאה היא סביבת עבודה מהירה ונוחה. עקרון העבודה המנחה באמצעות AngularJS הוא לקיחת יותר החלטות עבור המתכנת, דבר הבא לידי ביטוי במספר שורות הקוד הקצר הנדרש. שימוש ב-Angularjs הופך להיות שכיח יותר ויותר עקב התכונות המתקדמות שלו וקלות השימוש בו.

ספריה הנמצאת באינטראקציה עם Angularjs היא Twitter Bootstrap, המשלימה את הפלטפורמה בעזרת UI עשיר ויפה. Bootstrap היא ספריה חגיגית המכילה אוסף של templates לפיתוח אפליקציות web. Bootstrap פותחה על ידי טוויטר והפכה להיות נפוצה מאוד. ה-stylesheets המוכללים ב-bootstrap הופכים את הנראות של האתר למודרני ואחיד ללא מאמץ מצד המפתח.

פורמט JSON (ראשי תיבות של JavaScript Object Notation) הוא פורמט טקסטואלי המיועד להעברת מידע בין שרת ללקוח. בכתיבת אתר באמצעות AngularJS משתמשים ב-JSON לצורך עדכון דינמי של משתני JavaScript בדפי ה-HTML. jQuery היא ספריית JavaScript הנתמכת על ידי דפדפנים שונים (cross-browser), אשר מספקת כלים המפשטים עבודה עם ה-DOM ומספקת פונקציונליות עבור פעולות נפוצות ב-JavaScript, ובכך מקלה על כתיבת סקריפטים לצד הלקוח. זוהי ספריית JavaScript הפופולרית ביותר שנמצאת בשימוש כיום. השימוש ב-jQuery הוא בעיקר לצורך ביצוע מניפולציות על רכיבי ה-DOM בדף ה-HTML, למשל עדכון דינמי של צבע של כפתור לאחר שלחצו עליו.

- Editor: Sublime text
- HTML
- CSS 3
- Bootstrap (CSS + Javascript)
 - Paper theme
- AngularJS - SPA (Single Page Application)
- JQuery
- JSON

8 בדיקות המערכת

8.1 בדיקות יחידה

כל בדיקות היחידה נכתבו בסביבת הפיתוח והורצו באמצעות Nunit. תוצאת הרצת כל הבדיקות היתה תוצאה עוברת (הכל ירוק).

בדיקת Wavelet transform על בלוק שכולו שחור

מטרת הבדיקה: לוודא כי wavelet transform על בלוק ידוע נותנת תוצאה תקינה. שלבי הבדיקה: נבחר בלוק שכולו שחור ונצפה לתוצאת טרנספורמציה שהיא בלוק שכולו שחור. קוד הבדיקה:

```
[TestMethod]
public void TestWaveletTransform()
{
    // בלולאה הראשונה נייצר בלוק שכולו שחור.
    double[,] dataBlock = new double[BLOCKSIZE, BLOCKSIZE];
    for (int i = 0; i < dataBlock.GetLength(0); i++)
    {
        for (int j = 0; j < dataBlock.GetLength(1); j++)
        {
            dataBlock[i, j] = 0.0; // black
        }
    }

    // נפעיל טרנספורמצית wavelet transform.

    Matrix dataMatrix = new Matrix(dataBlock);
    int levels = 2;
    var transform = new WaveletTransform(new HaarLift(), levels);
    //Console.WriteLine("Wavelet transform...");
    dataMatrix = transform.DoForward(dataMatrix);

    // בלולאה השניה נוודא שהתוצאה היא בלוק שכולו שחור.

    for (int i = 0; i < dataBlock.GetLength(0); i++)
    {
        for (int j = 0; j < dataBlock.GetLength(1); j++)
        {
            Assert.AreEqual(dataBlock[i, j], 0.0);
        }
    }
}
```

תוצרים: נוצר בלוק שחור במלואו.

בדיקת feature extraction על בלוק שכולו שחור

מטרת הבדיקה: לוודא כי feature extraction על בלוק ידוע נותן תוצאה נכונה של כל ה-features.

שלבי הבדיקה: נבחר בלוק שכולו שחור ונצפה לתוצאה 0 בכל ה-features הבאים:

Horizontal band of wavelet transform, vertical band of wavelet transform, diagonal band of wavelet transform.

קוד הבדיקה:

```
[TestMethod]
public void TestFeatureExtraction()
{
    // בלולאה הראשונה נייצר בלוק שכולו שחור.
    double[,] dataBlock = new double[BLOCKSIZE, BLOCKSIZE];
    for (int i = 0; i < dataBlock.GetLength(0); i++)
    {
        for (int j = 0; j < dataBlock.GetLength(1); j++)
        {
            dataBlock[i, j] = 0.0; // black
        }
    }

    // נפעיל טרנספורמציה wavelet transform.
    Matrix dataMatrix = new Matrix(dataBlock);
    int levels = 2;
    var transform = new WaveletTransform(new HaarLift(), levels);
    //Console.WriteLine("Wavelet transform...");
    dataMatrix = transform.DoForward(dataMatrix);

    double V = 0.0;
    double H = 0.0;
    double D = 0.0;

    // נחשב את ה-features.
    WaveletSubBands.GetSubBands(dataMatrix, levels, ref V, ref H, ref D);

    // נוודא שקיבלנו בכל ה-features תוצאה 0.
    Assert.AreEqual(V, 0.0);
    Assert.AreEqual(H, 0.0);
    Assert.AreEqual(D, 0.0);
}
```

תוצרים: כל ה-features קיבלו ערך 0.

שליחת מסר מהלקוח לשרת ובדיקה האם הגיע לשרת

מטרת הבדיקה: לוודא שהשרת יכול לקבל מסרי Http.
שלבי הבדיקה: בבדיקה זאת נריץ את השרת (מתוך הקוד של בדיקת היחידה) ונשלח לו מסר Http (כמו זה שמגיע מדפדפן). נוודא שהשרת מקבל את המסר.
קוד הבדיקה:

```
[TestMethod]
public void TestServerReceiveMessage()
{
    try
    {
        נריץ את השרת
        HttpServer httpService = new BrainMriHttpServer("127.0.0.1", 8080);
        httpService.Start();

        נשלח הודעת startpreprocessing לשרת
        RequestManager request = new RequestManager();
        HttpWebResponse response =
            request.SendGETRequest("http://localhost:8080/startpreprocessing",
                null, null, true);

        נוודא שההודעה הגיעה לשרת והחזירה תגובה
        Assert.IsTrue(response.StatusCode == HttpStatusCode.OK);
        // Releases the resources of the response.
        response.Close();
        httpService.Stop();
    }
    catch (SocketException ex)
    {
        Assert.Fail();
    }
}
```

תוצרים: ההודעה הגיעה לשרת.

שליחת מסר מהשרת ללקוח ובדיקה האם הגיע ללקוח

מטרת הבדיקה: לוודא שהשרת מטפל נכון במסר `getpreprocessingstatus` ומחזיר תשובה נכונה ללקוח.

שלבי הבדיקה: בבדיקה זאת נריץ את השרת (מתוך הקוד של בדיקת היחידה) ונשלח לו מסר `Http` אשר ידוע מה התגובה אליו. נוודא שהלקוח מקבל את התגובה אשר ציפינו לה. המסר שבחרנו הוא `getpreprocessingstatus` ונריץ את בדיקות היחידה לפני תהליך העיבוד המקדים, לכן התגובה הצפויה למסר זה היא `"NotStarted"`.
קוד הבדיקה:

```
[TestMethod]
public void TestClientReceiveResponse()
{
    try
    {
        // נריץ את השרת
        HttpServer httpService = new BrainMriHttpServer("127.0.0.1", 8080);
        httpService.Start();

        RequestManager request = new RequestManager();
        // נשלח הודעת getpreprocessingstatus לשרת
        HttpResponseMessage response =
            request.SendHttpRequest("http://localhost:8080/getpreprocessingstatus",
                null, null, true);
        // נוודא שההודעה החזירה תגובה תקינה לשליחת ההודעה
        Assert.IsTrue(response.StatusCode == HttpStatusCode.OK);

        string responseString = GetResponseString(response);

        // Releases the resources of the response.
        response.Close();
        // נוודא שהשרת שלח תגובה ידועה שתהליך העיבוד המקדים עדיין לא התחיל
        Assert.IsTrue(responseString.Contains("NotStarted"));
        httpService.Stop();
    }
    catch (SocketException ex)
    {
        Assert.Fail();
    }
    catch (Exception ex)
    {
        Assert.Fail();
    }
}
```

תוצרים: ההודעה הגיעה לשרת וקיבלנו תגובה נכונה.

אשכול בו זמני של שני תהליכי אשכול שונים

מטרת הבדיקה: לוודא שהשרת לא מאפשר להתחיל תהליך של אשכול בזמן שמתרחש ברקע תהליך אשכול שהותחל ועדיין לא הסתיים.
שלבי הבדיקה: בדיקת היחידה תנסה לבצע אשכול בו זמני, השרת לא יאפשר לבצע אשכול נוסף בזמן שמתבצע הליך אשכול.
קוד הבדיקה:

```
[TestMethod]
public void TestDoubleDataMiningActions()
{
    נריץ את השרת

    IBrainMriServer server;
    server = BrainMriServer.Server.BrainMriServer.Instance;
    server.StartBrainMriServer();

    נריץ פעמיים ברצף את תהליך האשכול

    string returnString1 = server.StartDataMining();
    string returnString2 = server.StartDataMining();
    server.StopBrainMriServer();

    Debug.WriteLine(returnString1);
    Debug.WriteLine(returnString2);

    נוודא שרק אחד מתהליכי האשכול התחיל והשני נחסם
    // only one of the two commands was accepted (according to whichever command
    // started first).
    // This Assert will only work if the mining process is long enough.
    // If we have almost no records it may be that this first process
    // has finished when the second one request to start.
    Assert.IsTrue(
        (returnString1.Equals("started") && returnString2.Equals("notallowed")) ||
        (returnString2.Equals("started") && returnString1.Equals("notallowed"))
    );
}
```

תוצרים: תהליך אשכול יחיד מתחיל ואילו השני נחסם.

בדיקת נכונות השקלול של תוצאות האשכול בתמונה שבה כל הבלוקים מאותו אשכול מטרת הבדיקה: לוודא שהשרת משקלל נכון את האשכול של כל הבלוקים בתמונה. שלבי הבדיקה: בבדיקה זאת נקים אובייקטים של תמונה וכל הבלוקים שבה, ונוודא שאם בכל הבלוקים אותה תוצאת אשכול גם התמונה כולה תקבל אותה תוצאת אשכול. בנוסף ניתן לכל הבלוקים מידת שייכות 1 ונוודא כי מידת השייכות של התמונה כולה גם היא 1. קוד הבדיקה:

```
[TestMethod]
public void TestAccumulateMiningResults_AllBlocksNormal()
{
    // נייצר slice עם מספר בלוקים, וניתן לכל הבלוקים אשכול למחלקה 0 ומידת שייכות 1.
    Slice s = new Slice();
    s.PatientId = 1;
    s.SliceIndex = 1;

    Block b1 = new Block();
    b1.BlockIndex = 1;
    b1.FcmClass = 0;
    b1.FcmMembershipDegree = 1;
    b1.PatientId = 1;
    b1.SliceIndex = 1;
    s.Blocks.Add(b1);

    Block b2 = new Block();
    b2.BlockIndex = 1;
    b2.FcmClass = 0;
    b2.FcmMembershipDegree = 1;
    b2.PatientId = 1;
    b2.SliceIndex = 1;
    s.Blocks.Add(b2);

    Block b3 = new Block();
    b3.BlockIndex = 1;
    b3.FcmClass = 0;
    b3.FcmMembershipDegree = 1;
    b3.PatientId = 1;
    b3.SliceIndex = 1;
    s.Blocks.Add(b3);

    // נריץ את הפונקציה המשקללת את תוצאות האשכול לתמונה
    s.AccumulateMiningResults();
    // נוודא כי תוצאת האשכול המשוקללת של התמונה כולה היא 0 ומידת השייכות היא 1.
    Assert.IsTrue(s.FcmClass == 0 && s.FcmMembershipDegree == 1);
}

// תוצרים: תוצאת האשכול המשוקללת של התמונה כולה היא 0 ומידת השייכות היא 1.
```

בדיקת נכונות השקלול של תוצאות האשכול לנבדק כאשר כל התמונות מאותו אשכול מטרת הבדיקה: לוודא שהשרת משקלל נכון את האשכול של כל התמונות לנבדק. שלבי הבדיקה: בבדיקה זאת נקים אובייקטים של נבדק וכל התמונות שלו, ונוודא שאם בכל התמונות אותה תוצאת אשכול ואותה מידת שייכות גם הנבדק יקבל אותה תוצאת אשכול, ובאותה מידת שייכות. קוד הבדיקה:

```
[TestMethod]
public void TestAccumulateMiningResults_AllSlicesNormal()
{
    // נייצר patient עם מספר slices, וניתן לכל ה-slices אשכול למחלקה 0 ומידת שייכות 1.
    Patient p = new Patient();
    p.PatientId = 1;

    Slice s1 = new Slice();
    s1.PatientId = 1;
    s1.SliceIndex = 1;
    s1.FcmClass = 0;
    s1.FcmMembershipDegree = 1;
    p.Images.Add(s1);

    Slice s2 = new Slice();
    s2.PatientId = 1;
    s2.SliceIndex = 1;
    s2.FcmClass = 0;
    s2.FcmMembershipDegree = 1;
    p.Images.Add(s2);

    Slice s3 = new Slice();
    s3.PatientId = 1;
    s3.SliceIndex = 1;
    s3.FcmClass = 0;
    s3.FcmMembershipDegree = 1;
    p.Images.Add(s3);

    // נריץ את הפונקציה שמשקללת את תוצאות האשכול לנבדק
    p.AccumulateMiningResults();
    // נוודא כי תוצאת האשכול המשוקללת של הנבדק היא 0 ושמידת השייכות היא 1.
    Assert.IsTrue(p.FcmClass == 0 && p.FcmMembershipDegree == 1);
}
```

תוצרים: תוצאת האשכול המשוקללת של הנבדק היא 0 ומידת השייכות היא 1.

בדיקת פורמט ה-serialization של נבדק

מטרת הבדיקה: לוודא שהשרת מייצר serialization נכון של נבדק.
שלי הבדיקה: בבדיקה זאת נייצר נבדק פיקטיבי ונבצע serialization שלו. נבדוק כי צורת ביצוע ה-serialization היא בפורמט הנכון.
קוד הבדיקה:

```
[TestMethod]
public void TestPatientSerializeToJson()
{
    // נייצר patient עם מספר תכונות ועם מספר slices ולהם מספר בלוקים
    Patient p = new Patient();
    p.PatientId = 102;
    p.ImagesDir = @"c:\Anat";
    p.PreProcessingStatus = PreprocessingStatus.Complete;
    p.FcmClass = 1;
    p.FcmMembershipDegree = 5.2;
    p.RealClass = 0;

    Slice s1 = new Slice();
    s1.PatientId = 102;
    s1.SliceIndex = 3;
    s1.FileName = "s1.bmp";
    s1.PreProcessingStatus = PreprocessingStatus.InProgress;
    s1.FcmClass = 0;
    s1.FcmMembershipDegree = 0.98;
    p.Images.Add(s1);
    Block b1 = new Block();
    b1.PatientId = 102;
    b1.SliceIndex = 3;
    b1.BlockIndex = 1;
    b1.VerticalWavelet = 52;
    b1.HorizontalWavelet = 86;
    b1.DiagonalWavelet = 88;
    b1.Mean = 55;
    b1.Variance = 5;
    b1.Contrast = 78.6;
    b1.FcmClass = 1;
    b1.FcmMembershipDegree = 99.9;
    b1.IsNormalized = false;
    s1.Blocks.Add(b1);
    Block b2 = new Block();
    b2.PatientId = 102;
    b2.SliceIndex = 3;
    b2.BlockIndex = 2;
    b2.VerticalWavelet = 66;
    b2.HorizontalWavelet = 98.4;
    b2.DiagonalWavelet = 85.2;
    b2.Mean = 45.2;
    b2.Variance = 89;
    b2.Contrast = 54.1;
    b2.FcmClass = 0;
    b2.FcmMembershipDegree = 75.8;
    b2.IsNormalized = true;
    s1.Blocks.Add(b2);

    // נבצע serialization
    string json = JsonConvert.SerializeObject(p, Formatting.Indented);
}
```

נוודא שתוצאת ה-serialization היא אכן התוצאה הצפויה.

```
    string expected = @"{
    ""preProcessingStatus"": 0,
    ""Images"": [
    {
    ""PreProcessingStatus"": 1,
    ""Blocks"": [
    {
        ""PatientId"": 102,
        ""SliceIndex"": 3,
        ""BlockIndex"": 1,
        ""VerticalWavelet"": 52.0,
        ""HorizontalWavelet"": 86.0,
        ""DiagonalWavelet"": 88.0,
        ""Mean"": 55.0,
        ""Variance"": 5.0,
        ""Contrast"": 78.6,
        ""FcmClass"": 1,
        ""FcmMembershipDegree"": 99.9,
        ""IsNormalized"": false
    },
    {
        ""PatientId"": 102,
        ""SliceIndex"": 3,
        ""BlockIndex"": 2,
        ""VerticalWavelet"": 66.0,
        ""HorizontalWavelet"": 98.4,
        ""DiagonalWavelet"": 85.2,
        ""Mean"": 45.2,
        ""Variance"": 89.0,
        ""Contrast"": 54.1,
        ""FcmClass"": 0,
        ""FcmMembershipDegree"": 75.8,
        ""IsNormalized"": true
    }
    ],
    ""Height"": 0,
    ""Width"": 0,
    ""PatientId"": 102,
    ""SliceIndex"": 3,
    ""FileName"": ""s1.bmp"",
    ""FcmClass"": 0,
    ""FcmMembershipDegree"": 0.98
    }
    ],
    ""PatientId"": 102,
    ""ImagesDir"": ""c:\\Anat"",
    ""PreProcessingStatus"": 3,
    ""FcmClass"": 1,
    ""FcmMembershipDegree"": 5.2,
    ""RealClass"": 0
    }";

    StringAssert.Equals(expected, json);
}
```

תוצרים: תוצאת ה-serialization היא התוצאה הצפויה.

8.2 בדיקות אינטגרציה

בבדיקת האינטגרציה המטרה היא לבדוק את התנהגות המערכת כולה. נבנה Database מתוך הקלט, נריץ את תהליך העיבוד המקדים ואת תהליך כריית המידע ונבדוק את איכות התוצאות.

קלט: מאגר תמונות של נבדקים בעלי בדיקת MRI מוח תקינה ופתולוגית.

תוצאות צפויות לאחר ביצוע הבדיקה: כל הנבדקים הבריאים מקוטלגים כבריאים, וכל הנבדקים החולים מקוטלגים כאבנורמליים. בנוסף בדיקת תמונה של נבדק חולה תציג את האזור האבנורמלי במוח בהדגשה באדום. דיוק של לפחות 80% במדד הדיוק.

תוצאות:

מדדים לבחינת איכות האשכולות שהיו בשימוש הם:

Accuracy, Error Rate, Sensitivity, Specifity, Precision [13], Time, Bezdek FCM Index [31], [23], [26], [33], [17], [4], [24].

Num Abnormal Patients	total num patients	Accuracy (%)	Error Rate (%)	Sensitivity (%)	Specifity (%)	Precision (%)	Time (sec)	Bezdek FCM Index (%)
10	23	78	22	77	78	14	2.96	69
36	49	88.13	11.87	39.92	97.75	77.96	3.59	90.72

9 מקרה בוחן

כמקרה בוחן: בתחילה היה ניסיון להריץ את המערכת עם 23 מוחות, ולאחר מכן הוגדל מקרה הבוחן ל-49 מוחות. נבחרה תמונה אחת מאמצע המוח ונבנה database לכל הנבדקים עם התמונה שנבחרה.

9.1 ביצוע תהליך העיבוד המקדים לתמונה

העיבוד המקדים לתמונה מתבצע באמצעות תוכנת 3D Slicer [1], באמצעות תוכנת MIPAV [21] ובאמצעות תוכנת PreProcessing שאני כתבתי (ומצורפת בדיסקים המכילים את קוד המקור).

- עיבוד מקדים לתמונה באמצעות 3D Slicer:

העיבוד המקדים לתמונה באמצעות 3D Slicer כולל שלבים שמתבצעים לכל התמונות במערכת, ושלבים שמתבצעים רק על תמונות ששייכות לנבדקים בעלי תמונת מוח המכילות פתולוגיה.

עיבודי תמונה שמתבצעים לכל התמונות:

- General registration
- Histogram matching
- Foreground masking
- Skull stripping
- Crop
- Scale

עיבודי תמונה שמתבצעים לתמונות המכילות פתולוגיה:

- Mark tumor mask

- עיבוד מקדים לתמונה באמצעות MIPAV:

באמצעות MIPAV (Medical Image Processing, Analysis, and Visualization) נבצע שמירה של ה-MRI כסדרה של תמונות JPG. עיבוד זה מתבצע על כל התמונות במערכת.

- עיבוד מקדים לתמונה באמצעות תוכנת עזר ל-PreProcessing:

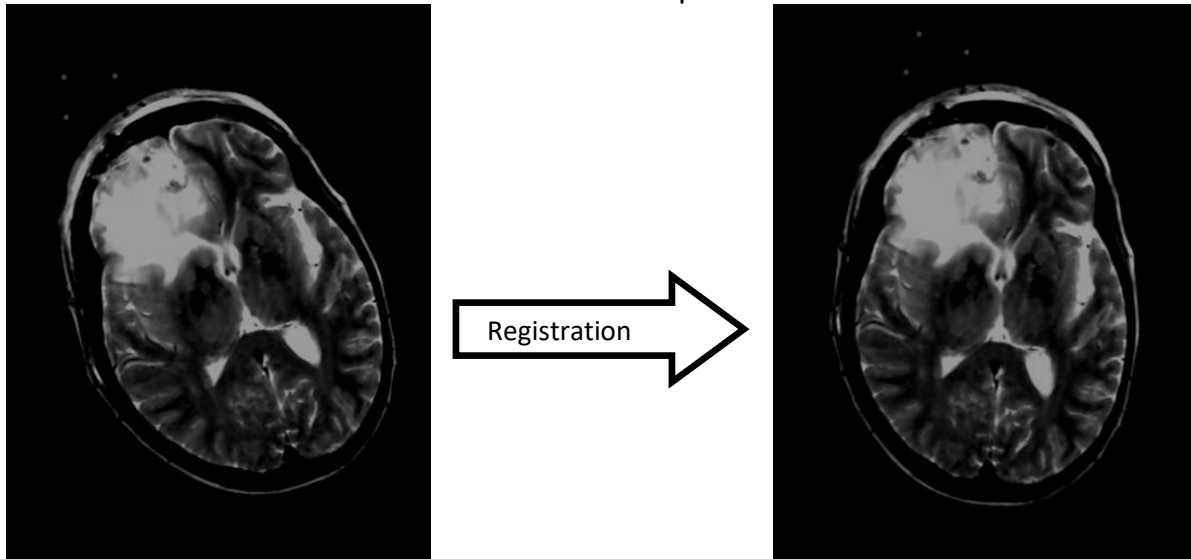
עיבוד זה מתבצע על כל התמונות במערכת.

- זיהוי המסגרת הריקה
- ביצוע crop של החלק המשמעותי
- ביצוע scaling של כל התמונות לאותו גודל

נסביר כל שלב, ונדגים בכל שלב תמונה אחת לפני ואחרי העיבוד.

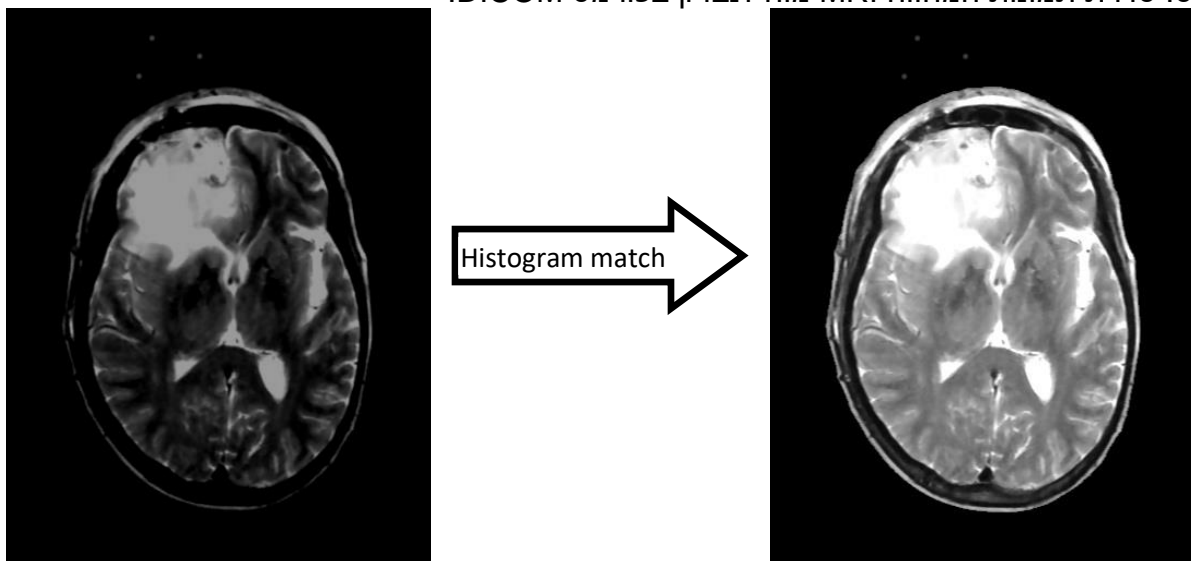
General registration BRAINS

מטרה: תהליך זה נועד לסובב את כל התמונות כך שיהיו באותו מנח.
קלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM, בנוסף סדרת תמונות המהווה MRI מוח לייחוס שעל פיה מיישרים את כל התמונות בסדרת הקלט.
עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מייצרת תמונה מיושרת לפי סדרת הייחוס, ומפיקה סדרת תמונות מיושרת בפורמט DICOM.
פלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.



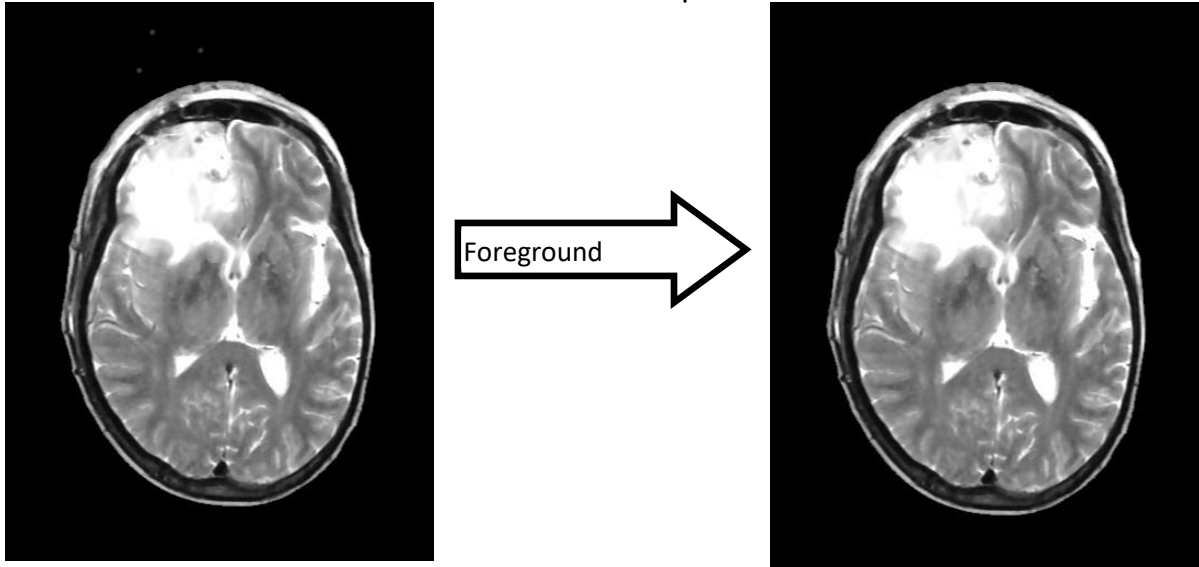
Histogram matching

מטרה: תהליך זה נועד להביא את כל התמונות לאותה סקלת בהירות.
קלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM, בנוסף סדרת תמונות המהווה MRI מוח לייחוס.
עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מייצרת תמונה באותה סקלת בהירות לפי סדרת הייחוס, ומפיקה סדרת תמונות בפורמט DICOM.
פלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.



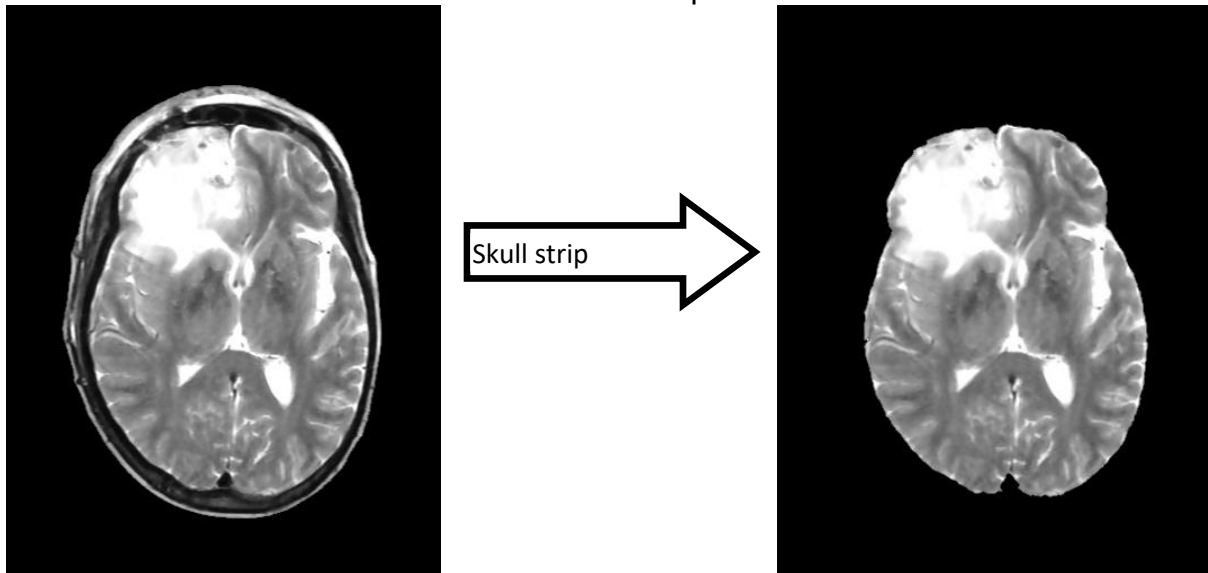
Foreground masking BRAINS

מטרה: תהליך זה נועד לזהות את הרקע ולהשחיר את כולו כדי שארטיפקטים ברקע לא ישפיעו על הזיהוי של הסרטן.
קלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.
עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מייצרת תמונה בעלת רקע שחור, ומפיקה סדרת תמונות בפורמט DICOM.
פלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.



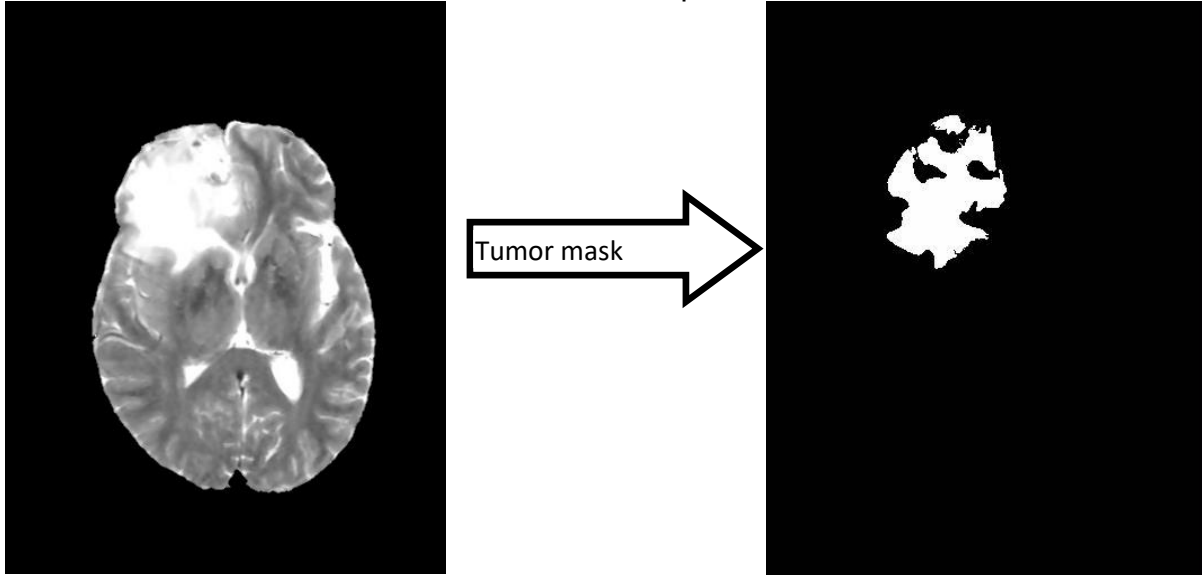
Swiss skull stripping

מטרה: תהליך זה נועד לזהות את המוח ללא הגולגולת והנוזל CSF.
קלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.
עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מייצרת תמונה ללא גולגולת וללא נוזל CSF, ומפיקה סדרת תמונות בפורמט DICOM.
פלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.



Mark tumor mask

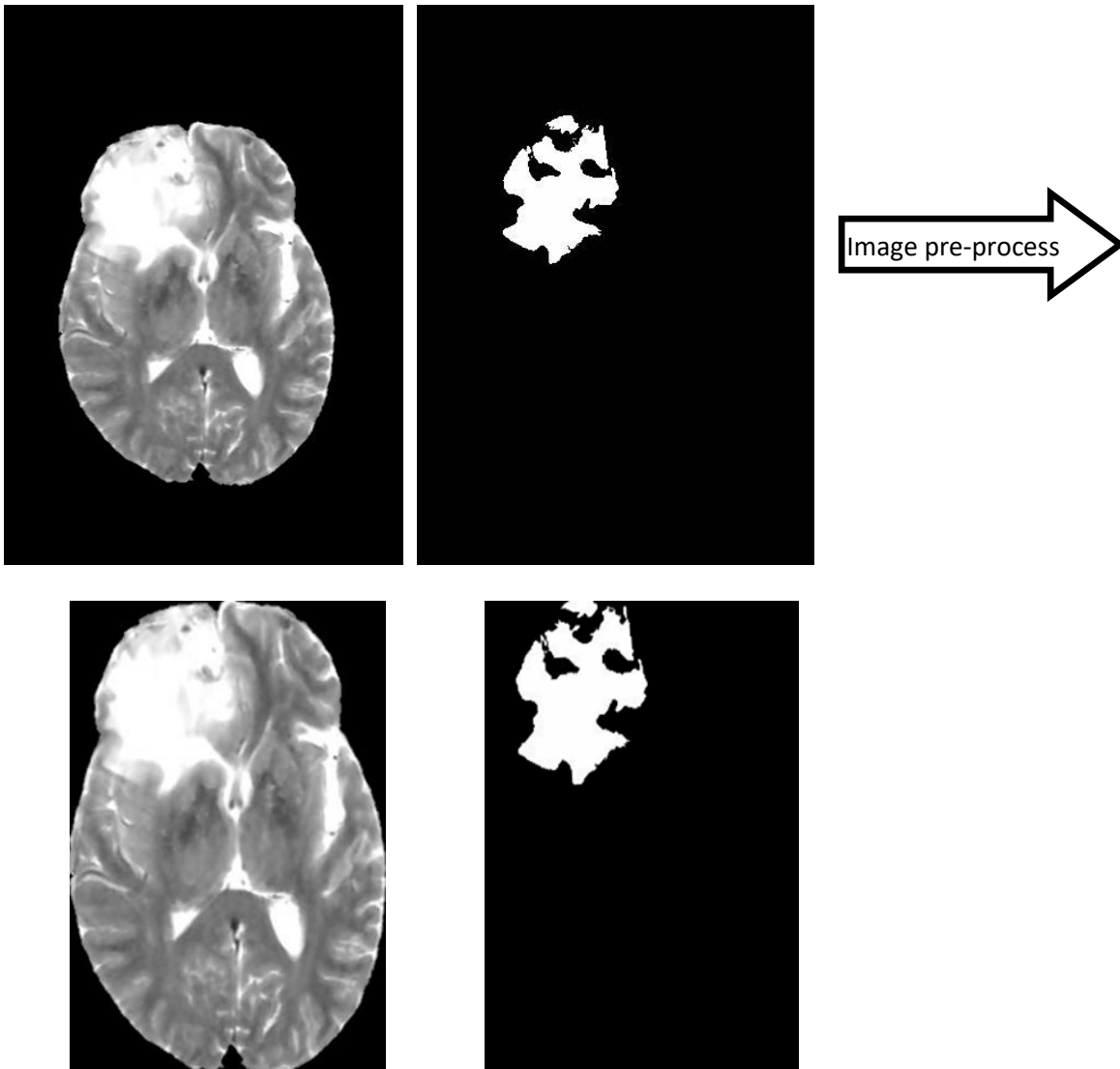
מטרה: תהליך זה לוקח את זיהוי הסרטן בתמונה ספציפית בתוך סדרת תמונות ב-MRI כלשהו ומזהה את הסרטן בכל התמונות באותה סדרת תמונות.
קלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.
עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מייצרת תמונה המכילה רק את האזור הסרטני בצבע לבן, ומפיקה סדרת תמונות בפורמט DICOM.
פלט: סדרת תמונות המהווה MRI מוח לנבדק בפורמט DICOM.



תוכנת PreProcessing

מטרה: מטרת תהליך זה לייצר אוסף תמונות שניתן להשוות ביניהן. לצורך השוואה נדרש שכל התמונות יהיו באותו גודל וכן כי אזורי המוח השונים יהיו ממוקמים באותו אזור בכל התמונות.

קלט: סדרת תמונות בודדות בפורמט JPG, ומיקום הספרייה בדיסק שהתמונות נמצאות בה. עיבוד: התוכנה עוברת תמונה אחר תמונה על כל התמונות בסדרה, מזהה את המסגרת הריקה, מבצעת crop של החלק המשמעותי בתמונה ובהתאמה מבצעת crop זהה של תמונת הסרטן באם קיימת תמונה כזאת, ובסוף מבצעת scaling של כל התמונות לאותו גודל. התוכנה מפיקה סדרת תמונות בפורמט JPG. פלט: סדרת תמונות בודדות בפורמט JPG.



9.2 ביצוע תהליך העיבוד המקדים ל-FCM

תהליך זה מתואר בפירוט בסעיפים: 3.6.2, 3.6.3. מטרה: מטרת התהליך לבצע עיבוד מקדים לכל הנבדקים, לקבל כתוצאה מאפיינים לכל הבלוקים בכל התמונות של כל הנבדקים, ולשמור את המאפיינים בבסיס הנתונים. קלט: בסיס נתונים המכיל טבלת נבדקים, ותמונות לכל נבדק. עיבוד: תהליך העיבוד המקדים מבצע מעבר על כל הנבדקים במערכת, כאשר לכל נבדק מתבצע מעבר על כל התמונות שלו. כל תמונה מחולקת לבלוקים. לכל בלוק מבוצע עיבוד מקדים שתואר בתהליך (3.6.3), ותוצאותיו ישמרו בבסיס הנתונים. פלט: שמירת המאפיינים לכל בלוק בבסיס הנתונים (מתבצע על ידי התהליך הפנימי 3.6.3) ושמירה בבסיס הנתונים של סטטוס תהליך ה-PreProcessing לכל תמונה ולכל נבדק.

9.3 ביצוע תהליך כריית המידע באמצעות FCM

תהליך זה מתואר בפירוט בסעיפים: 3.6.4, 3.6.5. מטרה: מטרת התהליך לבצע כריית מידע לכל הנבדקים, לקבל כתוצאה אשכול לתקין/פתולוגי ומידת ביטחון באשכול, ולשמור את האשכול ומידת הביטחון בבסיס הנתונים. קלט: בסיס נתונים המכיל טבלת נבדקים, תמונות לכל נבדק, ובלוקים שעברו עיבוד מקדים לכל תמונה. עיבוד: בתהליך כריית המידע מתבצע מעבר על כל הנבדקים במערכת, כאשר לכל נבדק מתבצע מעבר על כל התמונות שלו. בכל תמונה מתבצע מעבר על כל הבלוקים שלה. לכל בלוק מבוצע כריית מידע לבלוק שתואר בתהליך (3.6.5). אלגוריתם FCM מתואר בפסאודו קוד באלגוריתם 1 – FCM (5.2.1.3). לכל תמונה ולכל נבדק מחושב שקלול של האשכול ומידת הביטחון באשכול. פלט: שמירת האשכול לכל בלוק בבסיס הנתונים (מתבצע על ידי התהליך הפנימי 3.6.5) ושמירה בבסיס הנתונים של האשכול, מידת הביטחון באשכול וסטטוס תהליך כריית המידע לכל תמונה ולכל נבדק.

9.4 תוצאות

התוצאות שקיבלנו עבור מקרה הבוחן היו:

- כל הנבדקים הבריאים קוטלגו כבריאים.
- כל הנבדקים החולים קוטלגו כאבנורמליים.
- כל התמונות של כל הנבדקים החולים הכילו זיהוי של האזור האבנורמלי במוח.
- דיוק של 88%.

Num Abnormal Patients	total num patients	Accuracy (%)	Error Rate (%)	Sensitivity (%)	Specifity (%)	Precision (%)	Time (sec)	Bezdek FCM Index [31] (%)
10	23	78	22	77	78	14	2.96	69
36	49	88.13	11.87	39.92	97.75	77.96	3.59	90.72

בנוסף ראינו כי קיבלנו FCM Certainty גבוהה כאשר הגוש הסרטני היה בהיר ואילו FCM Certainty נמוך יותר כאשר הגוש הסרטני היה כהה יותר, ולכן דומה יותר לחומר האפור הנורמלי.

ניתן בחקר עתידי (ראה סעיף 10) לנסות למצוא מאפיינים נוספים ל-FCM כך שסרטן הנראה כהה בתמונה יזוהה ברמת ביטחון גבוהה יותר.

10 סיכום והצעות להמשך מחקר

במסגרת פרויקט מתקדם במדעי המחשב פותחה מערכת המסווגת אוטומטית תמונות MRI מוח לתקין או פתולוגי, בציון מדד כמותי המציין את מידת הביטחון באשכול זה. כמו כן, המערכת מבצעת דיטקציה אוטומטית של אזורים ב-MRI המעידים על הפתולוגיה.

המערכת משלבת טכנולוגיות שונות לצד השרת ולצד הלקוח. השרת כתוב כולו בשפת C#. הלקוח משלב טכנולוגיות מוכרות לפיתוח web: הפלטפורמה היא Angularjs, התשתית היא Bootstrap וה- templates גם כן לקוחים מ-Bootstrap, הפורמט להעברת נתונים בין השרת ללקוח הוא JSON. נעשה גם שימוש בספריית jQuery, ובשרת http-server להגשת תמונות מהשרת ללקוח באמצעות HTTP.

חשיבות הפרויקט בהיבט מדעי המחשב מבחינה אישית היא ראשית התנסות בפיתוח מערכת מורכבת כל כך, שנית התנסות ב-data mining, ובנוסף חשיפה מעניינת ומאתגרת לכלים ולטכנולוגיות לפיתוח מערכות web, שלא היו מוכרים לי כלל.

חשיבות הפרויקט בהיבט הרפואי היא בכך שהוא מאפשר לקבל הערכה לגבי נבדקים שצריך לאבחן האם הם חולים בסרטן מסוג GBM, ואם כן באיזה אזור במוח ממוקם הסרטן, ומה היקפו. הערכה זאת יכולה לעזור לכוון את הרופא ולחסוך לו זמן יקר. סדרת תמונות MRI מכילה מאות תמונות, ומעבר של רופא מומחה עליהם דורשת זמן רב ומאמץ רב. הפרויקט יכול לעזור לרופא לדעת באילו תמונות להתמקד מתוך הסדרה, וכן באילו אזורים בכל תמונה להתמקד.

בנוסף, לצורך ניתוח להוצאת סרטן מסוג GBM הערכת מיקום ונפח הסרטן היא קריטית. הדיוק הנדרש הוא גבוה, והפרויקט יכול לעזור למקד את הרופא כאשר הוא יראה באופן יזואלי סימון של הסרטן.

הפרויקט מהווה בסיס להצעות מחקר שונות ומתקדמות כדוגמת:

1. קליברציה אוטומטית של תמונות MRI. צילומי MRI של המוח נבדלים אלו מאלו הן במרחק במילימטרים בין התמונות ולעיתים בכיוון ביצוע הסריקה. המטרה של קליברציה אוטומטית תהיה לקבל MRI המכיל מאות תמונות, ולאנדקס אוטומטית את התמונות ב-DB לפי המיקום שלהן במוח, כך שכל התמונות בעלות אותו אינדקס ישקפו תמונות ברות השוואה.
2. הרחבת הליך הסיווג לכלל התמונות של הנבדקים.
3. הרחבת סיווג לסוגי פתולוגיות שונות. הרחבת מגוון הבדיקות על תמונות ה-MRI כדי לזהות חריגות נוספות בצורה, בנפח ובחלוקת סוגי החומר במוח. חקר עתידי של מידת השיוך של כל מוח למחלקת הפתולוגיים עשוי להקנות תובנות נוספות בנושא התפתחות של מצבים אבנורמליים במוח וההשפעות של סוגי הפתולוגיות על תמונת המוח. בפרט יתכן כי ניתן יהיה להסיק לגבי גרורות משניות לסרטן הראשוני לפי מידת השיוך למחלקת הפתולוגיים.
4. שיפור אלגוריתם FCM לדיוק גבוה יותר. למשל מציאת מאפיינים נוספים ל-FCM כך שסרטן הנראה כה יזוהה ברמת ביטחון גבוהה.
5. התייחסות לנתונים נוספים כגון נתונים דמוגרפיים או נתוני מחלה ידועים לצורך אשכול לתקין או פתולוגי.
6. פיתוח שיטות לאחסון נתונים קומפקטי.
7. שילוב שיטות כריית מידע שונות לבעיה הנתונה והשוואת תוצאות הכרייה, במיוחד עבור המקרים בהם הסרטן נראה כהה.

11 נספחים

11.1 נספח א: סקריפט ליצירת הטבלאות

```
USE [BrainMri]
GO
/** Object: Table [dbo].[Patient] Script Date: 04/10/2015 14:45:24 ***/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Patient](
    [PatientId] [numeric](18, 0) NOT NULL,
    [ImagesDir] [nchar](200) NOT NULL,
    [PreProcessingStatus] [numeric](18, 0) NULL,
    [FcmClass] [numeric](18, 0) NULL,
    [FcmMembershipDegree] [real] NULL,
    CONSTRAINT [PK_Patient] PRIMARY KEY CLUSTERED
(
    [PatientId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/** Object: Table [dbo].[Slice] Script Date: 04/10/2015 14:45:24 ***/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Slice](
    [PatientId] [numeric](18, 0) NOT NULL,
    [SliceIndex] [numeric](18, 0) NOT NULL,
    [FileName] [nchar](200) NOT NULL,
    [PreProcessingStatus] [numeric](18, 0) NOT NULL,
    [FcmClass] [numeric](18, 0) NULL,
    [FcmMembershipDegree] [real] NULL,
    CONSTRAINT [PK_Slice] PRIMARY KEY CLUSTERED
(
    [PatientId] ASC,
    [SliceIndex] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```

/**** Object: Table [dbo].[Block] Script Date: 04/10/2015 14:45:24 ****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Block] (
    [PatientId] [numeric](18, 0) NOT NULL,
    [SliceIndex] [numeric](18, 0) NOT NULL,
    [BlockIndex] [numeric](18, 0) NOT NULL,
    [VerticalWavelet] [real] NOT NULL,
    [HorizontalWavelet] [real] NOT NULL,
    [DiagonalWavelet] [real] NOT NULL,
    [Mean] [real] NOT NULL,
    [Variance] [real] NOT NULL,
    [Contrast] [real] NOT NULL,
    [FcmClass] [numeric](18, 0) NULL,
    [FcmMembershipDegree] [real] NULL,
    [IsNormalized] [bit] NOT NULL,
    CONSTRAINT [PK_Block] PRIMARY KEY CLUSTERED
(
    [PatientId] ASC,
    [SliceIndex] ASC,
    [BlockIndex] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

/* Object: ForeignKey [FK_Slice_Patient] Script Date: 04/10/2015
14:45:24 */
ALTER TABLE [dbo].[Slice] WITH CHECK ADD CONSTRAINT
[FK_Slice_Patient] FOREIGN KEY([PatientId])
REFERENCES [dbo].[Patient] ([PatientId])
GO
ALTER TABLE [dbo].[Slice] CHECK CONSTRAINT [FK_Slice_Patient]
GO

/* Object: ForeignKey [FK_Block_Patient] Script Date: 04/10/2015
14:45:24 */
ALTER TABLE [dbo].[Block] WITH CHECK ADD CONSTRAINT
[FK_Block_Patient] FOREIGN KEY([PatientId])
REFERENCES [dbo].[Patient] ([PatientId])
GO
ALTER TABLE [dbo].[Block] CHECK CONSTRAINT [FK_Block_Patient]
GO

/* Object: ForeignKey [FK_Block_Slice] Script Date: 04/10/2015 14:45:24
*/
ALTER TABLE [dbo].[Block] WITH CHECK ADD CONSTRAINT [FK_Block_Slice]
FOREIGN KEY([PatientId], [SliceIndex])
REFERENCES [dbo].[Slice] ([PatientId], [SliceIndex])
GO
ALTER TABLE [dbo].[Block] CHECK CONSTRAINT [FK_Block_Slice]
GO

```

11.2 גספח ב: סקריפט ליצירת stored procedures

```
--
-- Dropping stored procedure PatientSelectAll :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[PatientSelectAll]') AND OBJECTPROPERTY(id, N'IsProcedure')
= 1)
    DROP PROCEDURE [dbo].[PatientSelectAll]
GO

--
=====
-- Entity Name: PatientSelectAll
-- Create date: 4/11/2015 12:48:31 PM
-- Description: Select all rows form Patient
--
=====

Create Procedure PatientSelectAll
AS
BEGIN
SELECT * FROM Patient
END
GO

--
-- Dropping stored procedure PatientSelectRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[PatientSelectRow]') AND OBJECTPROPERTY(id, N'IsProcedure')
= 1)
    DROP PROCEDURE [dbo].[PatientSelectRow]
GO

--
=====
-- Entity Name: PatientSelectRow
-- Create date: 4/11/2015 12:48:31 PM
-- Description: This SP select a specify row from Patient
--
=====

Create Procedure PatientSelectRow
@PatientId numeric
AS
BEGIN
SELECT * FROM Patient WHERE [PatientId]=@PatientId
END
GO
```

```

-- Dropping stored procedure PatientInsertRow :
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[PatientInsertRow]') AND OBJECTPROPERTY(id, N'IsProcedure')
= 1)
    DROP PROCEDURE [dbo].[PatientInsertRow]
GO

-- =====
-- Entity Name: PatientInsertRow
-- Create date: 4/11/2015 12:48:31 PM
-- Description: This SP Inserts value to Patient table
-- =====

Create Procedure PatientInsertRow
@PatientId numeric(18,0),
@ImagesDir nchar(200),
@PreProcessingStatus numeric(18,0),
@FcmClass numeric(18,0),
@FcmMembershipDegree real,
@RealClass numeric(18,0)
AS
BEGIN
    INSERT INTO Patient
        ([PatientId], [ImagesDir], [PreProcessingStatus],
        [FcmClass], [FcmMembershipDegree], [RealClass])
    VALUES
        (@PatientId, @ImagesDir, @PreProcessingStatus,
        @FcmClass, @FcmMembershipDegree, @RealClass)
END
GO

-- Dropping stored procedure PatientUpdateRow :
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[PatientUpdateRow]') AND OBJECTPROPERTY(id, N'IsProcedure')
= 1)
    DROP PROCEDURE [dbo].[PatientUpdateRow]
GO

-- =====
-- Entity Name: PatientUpdateRow
-- Create date: 4/11/2015 12:48:31 PM
-- Description: This SP updates Patient table rows.
-- =====

Create Procedure PatientUpdateRow
@PatientId numeric(18,0),
@ImagesDir nchar(200),
@PreProcessingStatus numeric(18,0),
@FcmClass numeric(18,0),
@FcmMembershipDegree real,
@RealClass numeric(18,0)
AS
BEGIN
UPDATE Patient
SET [PatientId] = @PatientId, [ImagesDir] = @ImagesDir,
    [PreProcessingStatus] = @PreProcessingStatus,
    [FcmClass] = @FcmClass,
    [FcmMembershipDegree] = @FcmMembershipDegree,
    [RealClass] = @RealClass
WHERE [PatientId] = @PatientId
END
GO

```

```

--
-- Dropping stored procedure PatientDeleteRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[PatientDeleteRow]') AND OBJECTPROPERTY(id, N'IsProcedure')
= 1)
    DROP PROCEDURE [dbo].[PatientDeleteRow]
GO

=====
-- Entity Name: PatientDeleteRow
-- Create date: 4/11/2015 12:48:31 PM
-- Description: This SP delete specify row from Patient table
-- =====

Create Procedure PatientDeleteRow
@PatientId numeric(18,0)
AS
BEGIN
DELETE Patient WHERE [PatientId] = @PatientId
END
GO

--
-- Dropping stored procedure SliceSelectAll :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[SliceSelectAll]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[SliceSelectAll]
GO

=====
-- Entity Name: SliceSelectAll
-- Create date: 4/11/2015 12:23:51 PM
-- Description: Select all rows form Slice
-- =====

Create Procedure SliceSelectAll
AS
BEGIN
SELECT * FROM Slice
END
GO

```

```

--
-- Dropping stored procedure SliceSelectRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[SliceSelectRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[SliceSelectRow]
GO
-- =====
-- Entity Name: SliceSelectRow
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP select a specify row from Slice
-- =====
Create Procedure SliceSelectRow
@PatientId numeric,
@SliceIndex numeric
AS
BEGIN
SELECT * FROM Slice WHERE [PatientId]=@PatientId and
[SliceIndex]=@SliceIndex
END
GO

--
-- Dropping stored procedure SliceInsertRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[SliceInsertRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[SliceInsertRow]
GO
-- =====
-- Entity Name: SliceInsertRow
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP Inserts value to Slice table
-- =====
Create Procedure SliceInsertRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0),
@FileName nchar(200),
@PreProcessingStatus numeric(18,0),
@FcmClass numeric(18,0),
@FcmMembershipDegree real
AS
BEGIN
INSERT INTO Slice
([PatientId], [SliceIndex], [FileName], [PreProcessingStatus],
[FcmClass], [FcmMembershipDegree])
VALUES
(@PatientId, @SliceIndex, @FileName, @PreProcessingStatus,
@FcmClass, @FcmMembershipDegree)
END
GO

```



```

--
-- Dropping stored procedure SliceUpdateRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[SliceUpdateRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[SliceUpdateRow]
GO
-- =====
-- Entity Name: SliceUpdateRow
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP updates Slice table rows.
-- =====
Create Procedure SliceUpdateRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0),
@FileName nchar(200),
@PreProcessingStatus numeric(18,0),
@FcmClass numeric(18,0),
@FcmMembershipDegree real
AS
BEGIN
UPDATE Slice SET [PatientId] = @PatientId,
[SliceIndex] = @SliceIndex,
[FileName] = @FileName,
[PreProcessingStatus] = @PreProcessingStatus,
[FcmClass] = @FcmClass,
[FcmMembershipDegree] = @FcmMembershipDegree
WHERE [PatientId] = @PatientId and [SliceIndex] = @SliceIndex
END
GO

--
-- Dropping stored procedure SliceDeleteRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[SliceDeleteRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[SliceDeleteRow]
GO
-- =====
-- Entity Name: SliceDeleteRow
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP delete specify row from Slice table
-- =====
Create Procedure SliceDeleteRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0)
AS
BEGIN
DELETE Slice WHERE [PatientId] = @PatientId and [SliceIndex] =
@SliceIndex
END
GO

```

```

-- =====
-- Entity Name: SliceSelectByPatient
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP select a specify row from Slice
-- =====
Create Procedure [dbo].[SliceSelectByPatient]
@PatientId numeric
AS
BEGIN
SELECT * FROM Slice WHERE [PatientId]=@PatientId
END
GO

--
-- Dropping stored procedure BlockSelectAll :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[BlockSelectAll]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
DROP PROCEDURE [dbo].[BlockSelectAll]
GO

-- =====
-- Entity Name: BlockSelectAll
-- Create date: 4/10/2015 2:48:48 PM
-- Description: Select all rows form Block
-- =====
Create Procedure BlockSelectAll
AS
BEGIN
SELECT * FROM Block
END
GO

--
-- Dropping stored procedure BlockSelectRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[BlockSelectRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
DROP PROCEDURE [dbo].[BlockSelectRow]
GO

-- =====
-- Entity Name: BlockSelectRow
-- Create date: 4/10/2015 2:48:48 PM
-- Description: This SP select a specify row from Block
-- =====
Create Procedure BlockSelectRow
@PatientId numeric,
@SliceIndex numeric,
@BlockIndex numeric
AS
BEGIN
SELECT * FROM Block WHERE [PatientId]=@PatientId and
[SliceIndex]=@SliceIndex and [BlockIndex]=@BlockIndex
END
GO

```

```

--
-- Dropping stored procedure BlockInsertRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[BlockInsertRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[BlockInsertRow]

GO

-- =====
-- Entity Name: BlockInsertRow
-- Create date: 4/10/2015 2:48:48 PM
-- Description: This SP Inserts value to Block table
-- =====

Create Procedure BlockInsertRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0),
@BlockIndex numeric(18,0),
@VerticalWavelet real,
@HorizontalWavelet real,
@DiagonalWavelet real,
@Mean real,
@Variance real,
@Contrast real,
@FcmClass numeric(18,0),
@FcmMembershipDegree real,
@IsNormalized bit
AS
BEGIN
    INSERT INTO Block
        ([PatientId], [SliceIndex], [BlockIndex], [VerticalWavelet],
        [HorizontalWavelet], [DiagonalWavelet], [Mean], [Variance], [Contrast],
        [FcmClass], [FcmMembershipDegree], [IsNormalized])
        VALUES
            (@PatientId, @SliceIndex, @BlockIndex, @VerticalWavelet,
            @HorizontalWavelet, @DiagonalWavelet, @Mean, @Variance, @Contrast,
            @FcmClass, @FcmMembershipDegree, @IsNormalized)
END
GO

```

```

--
-- Dropping stored procedure BlockUpdateRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[BlockUpdateRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[BlockUpdateRow]
GO
-- =====
-- Entity Name: BlockUpdateRow
-- Create date: 4/10/2015 2:48:48 PM
-- Description: This SP updates Block table rows.
-- =====
Create Procedure BlockUpdateRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0),
@BlockIndex numeric(18,0),
@VerticalWavelet real,
@HorizontalWavelet real,
@DiagonalWavelet real,
@Mean real,
@Variance real,
@Contrast real,
@FcmClass numeric(18,0),
@FcmMembershipDegree real,
@IsNormalized bit
AS
BEGIN

UPDATE Block
SET [PatientId] = @PatientId,
    [SliceIndex] = @SliceIndex,
    [BlockIndex] = @BlockIndex,
    [VerticalWavelet] = @VerticalWavelet,
    [HorizontalWavelet] = @HorizontalWavelet,
    [DiagonalWavelet] = @DiagonalWavelet,
    [Mean] = @Mean,
    [Variance] = @Variance,
    [Contrast] = @Contrast,
    [FcmClass] = @FcmClass,
    [FcmMembershipDegree] = @FcmMembershipDegree,
    [IsNormalized] = @IsNormalized
WHERE [PatientId] = @PatientId and [SliceIndex] = @SliceIndex and
[BlockIndex] = @BlockIndex

END

GO

```

```

--
-- Dropping stored procedure BlockDeleteRow :
--
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[BlockDeleteRow]') AND OBJECTPROPERTY(id, N'IsProcedure') =
1)
    DROP PROCEDURE [dbo].[BlockDeleteRow]
GO
=====
-- Entity Name: BlockDeleteRow
-- Create date: 4/10/2015 2:48:48 PM
-- Description: This SP delete specify row from Block table
-- =====

Create Procedure BlockDeleteRow
@PatientId numeric(18,0),
@SliceIndex numeric(18,0),
@BlockIndex numeric(18,0)
AS
BEGIN
DELETE Block
WHERE [PatientId] = @PatientId and
      [SliceIndex] = @SliceIndex and
      [BlockIndex] = @BlockIndex
END
GO

-- =====
-- Entity Name: BlockSelectByPatientAndSlice
-- Create date: 4/11/2015 12:23:51 PM
-- Description: This SP select a specify row from Slice
-- =====

Create Procedure [dbo].[BlockSelectByPatientAndSlice]
@PatientId numeric,
@SliceIndex numeric
AS
BEGIN
SELECT * FROM Block
WHERE [PatientId]=@PatientId and [SliceIndex]=@SliceIndex
                                                    END
GO

```

11.3 נטפח ג: קוד

דיסק עם קוד מקור של הפרויקט מצורף לספר פרויקט.

11.4 נספח ד: הוראות התקנה

התקנת לקוח

1. התקנת Node.js:
 - a. הורדת Windows installer מהאתר של Node.js: <http://nodejs.org>
 - b. הרצת ה-installer שהורד בשלב הקודם. בזמן הרצת ההתקנה יש לקבל את כל ברירות המחדל ולהסכים ל-license agreement.
 - c. בצע restart למחשב.
 - d. בדיקה ש-Node.js הותקן באופן תקין:
פתח command line prompt והרץ את הפקודות הבאות:
run node -v
run npm -v
run node hello.js
אם הפקודות לא מוכרות חזור לשלב התקנת Node.js.
2. העתק את הקוד של צד הלקוח מתוך הדיסקים מתוך הספרייה web.
3. התקנת ספריות שהפרויקט תלוי בהן:
 - a. פתח command line prompt והרץ את הפקודה הבאה:
run npm install

התקנת שרת

1. התקנת Node.js: אם מריצים שרת ולקוח באותו מחשב אז כבר ביצענו את השלב הזה. אם מריצים במחשבים שונים יש לבצע שלב זה במחשב השרת בדיוק כפי שבוצע במחשב הלקוח.
2. התקנת http-server:
 - a. פתח command line prompt והרץ את הפקודה הבאה:
npm install http-server -g
3. התקנת מסד הנתונים:
 - a. התקן Ms Sql Server
 - b. העתק את מסד הנתונים MriServer.ldf, MriServer.mdf למחשב השרת.
 - c. פתח SQL Server Management Studio ובדוק שמסד הנתונים מכיל נתונים.
4. העתק את הקוד של צד השרת מתוך הדיסקים מתוך הספרייה
Software\BrainMriServer

11.5 נספח ה: הוראות הרצה

הרצת לקוח

1. פתח command line prompt והרץ את הפקודות הבאות:
 - a. `cd "c:\Anat\Project\web"` (כאן יש לכתוב את ה-path של ספריית הלקוח במחשב הלקוח כפי שהועתקה בשלב 2 של התקנת הלקוח).
 - b. `npm start`
2. פתח חלון chrome menu->new incognito window
3. בשורת location bar כתוב את הכתובת של הלקוח:
 - a. `http://localhost:8000`

הרצת שרת

1. פתח command line prompt והרץ את הפקודה הבאה:
 - `http-server -p 8081`
 2. הרץ את:
`C:\Anat\Project\Software\BrainMriServer\BrainMriServerConsole\
bin\release\BrainMriServerConsole.exe`
- (כאן יש לכתוב את ה-path של ספריית השרת במחשב השרת כפי שהועתקה בשלב 4 של התקנת השרת).

- [1] 3D Slicer Wiki pages. [Online]. https://www.slicer.org/slicerWiki/index.php/Main_Page
- [2] Jose Alan, S Ravi, and M Sambath, "Brain Tumor Segmentation Using K-Means Clustering And Fuzzy C-Means Algorithms And Its Area Calculations," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 3, pp. 3496-3501, 2014.
- [3] American Brain Tumor Association. [Online]. <http://www.abta.org/brain-tumor-information/types-of-tumors/glioblastoma.html>
- [4] K M Bataineh, M Naji, and M Saqer, "A Comparison Study between Various Fuzzy Clustering Algorithms," *JJMIE*, vol. 5, pp. 335-343, 2011.
- [5] S Zulaikha Beevi, M Mohammed Sathik, and K SenthamaraiKannan, "A Robust Fuzzy Clustering Technique with Spatial Neighborhood Information for Effective Medical Image Segmentation," *International Journal of Computer Science and Information Security*, vol. 7, no. 3, pp. 132-138, 2010.
- [6] James C Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [7] Dibya Jyoti Bora and Anil Kumar Gupta, "Impact of Exponent Parameter Value for the Partition Matrix on the Performance of Fuzzy C Means Algorithm," *International Journal of Scientific Research in Computer Science Applications and Management Studies*, vol. 3, no. 3, May 2014.
- [8] Cancer Imaging Archive. [Online]. <https://public.cancerimagingarchive.net>
- [9] Cancer Imaging Archive - About. [Online]. <http://www.cancerimagingarchive.net/about-the-cancer-imaging-archive-tcia/>
- [10] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *Computerized Medical Imaging and Graphics*, vol. 30, pp. 9-15, January 2006.
- [11] Designed Database of MR Brain Images of Healthy Volunteers. [Online]. <http://www.insight-journal.org/midas/community/view/21>
- [12] Moumen T El-Melegy and Mokhtar M Hashim, "Tumor segmentation in brain MRI using a fuzzy approach with class center priors," *EURASIP Journal on Image and Video Processing*, pp. 1-14, 2014.

- [13] J Han, M Kamber, and J Pei, *Data Mining Concepts and Techniques*, 3rd ed.: Morgan Kaufmann, 2011.
- [14] S Javeed Hussain, T Satya Savithri, and P.V Sree Devi, "Segmentation of Tissues in Brain MRI Images using Dynamic Neuro-Fuzzy Technique," *International Journal of Soft Computing and Engineering (IJSCE)*, pp. 416-423, January 2012.
- [15] Imaging Methods Assessment and Reporting. [Online]. <http://www.insight-journal.org/midas/community/view/15>
- [16] S R Kannan, S Ramathilagam, R Pandiyarajan, and A Sathya, "Fuzzy clustering Approach in segmentation of T1-T2 brain MRI," *ACEEE International Journal on Signal and Image Processing*, vol. 1, no. 2, pp. 43-47, 2010.
- [17] Dae-Won Kim, Kwang H Lee, and Doheon Lee, "On cluster validity index for estimation of the optimal number of fuzzy clusters," *Elsevier*, vol. 37, pp. 2009-2025, 2004.
- [18] Tao Li, Sheng Ma, and Mitsunori Ogihara, "Wavelet Methods in Data Mining," in *Data Mining and Knowledge Discovery Handbook*, 2nd ed., Oded Maimon and Lior Rokach, Eds.: Springer, 2010, ch. 27.
- [19] Albert A Michelson, *Studies in optics*. New York: Dover Publications, 1995.
- [20] Microsoft SQL Server 2005 Express Edition. [Online]. <http://www.microsoft.com/en-us/download/details.aspx?id=21844>
- [21] MIPAV - Medical Image Processing, Analysis, and Visualization. [Online]. <https://mipav.cit.nih.gov/>
- [22] Nevin A Mohamed, M N Ahmed, and A Farag, "Modified fuzzy c-mean in medical image segmentation," *Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3429-3432, 1999.
- [23] Malay K Pakhira, Sanghamitra Bandyopadhyay, and Ujjwal Maulik, "Validity index for crisp and fuzzy clusters," *Elsevier*, vol. 37, pp. 487-501, 2004.
- [24] Mohammad Rawashdeh and Anca Ralescu, "Fuzzy Cluster Validity with Generalized Silhouettes".
- [25] Shan Shen, William Sandham, Malcolm Granat, and Annette Sterr, "MRI Fuzzy Segmentation of Brain Tissue Using Neighborhood Attraction With Neural-Network Optimization," *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, vol. 9, no. 3, pp. 459-467, 2005.

- [26] Horng-Lin Shieh, "A Hybrid Fuzzy Clustering Method with a Robust Validity Index," *International Journal of Fuzzy Systems*, vol. 16, pp. 39-45, 2014.
- [27] M Y Siyal and Lin Yu, "An intelligent modified fuzzy c-means based algorithm for bias estimation and segmentation of brain MRI," *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2052-2062, 2005.
- [28] The Cancer Genome Atlas Glioblastoma Multiforme. [Online]. <https://wiki.cancerimagingarchive.net/display/Public/TCGA-GBM>
- [29] P Vasuda and S Satheesh, "Improved Fuzzy C-Means Algorithm for MR Brain Image Segmentation," *International Journal on Computer Science and Engineering*, vol. 2, no. 5, pp. 1713-1715, 2010.
- [30] Visual Studio 2008. [Online]. <http://www.microsoft.com/en-us/download/details.aspx?id=7873>
- [31] Weina Wang and Yunjie Zhang, "On fuzzy cluster validity indices," *Elsevier*, pp. 2095-2117, 2007.
- [32] Cai Weiling, Chen Songcan, and Zhang Daoqiang, "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation," *Pattern Recognition*, vol. 40, no. 3, pp. 825-838, 2007.
- [33] Kuo-Lung Wu and Miin-Shen Yang, "A cluster validity index for fuzzy clustering," *Elsevier*, vol. 26, pp. 1275-1291, 2004.
- [34] Y Yong , Z Chongxun, and L Pan , "A Novel Fuzzy C-Means Clustering Algorithm for Image Thresholding," *MEASUREMENT SCIENCE REVIEW*, vol. 4, pp. 11-19, 2004.