

# **פרויקט מתקדם במדעי המחשב-**

## **22997**

**מערכת לסגמנטציה של חדר שמאל מתמונות CMRI**

**ע"י שימוש באלגוריתם Fuzzy C Means**

**תוך שיפור איכות הפתרון בעזרת MMAS**

**ואופטימיזציה בעזרת PCA**

**מגיש: אבני חן ת.ז. 307859983**

**הפרויקט הוכן בהנחייתה של ד"ר מיה הרמן**

**מאי 2017**

## תוכן עניינים

7	1. מבוא
7	1.1. רקע
9	1.2. הגדרת הבעיה
10	1.3. אלגוריתמים קיימים לביצוע סגמנטציה אוטומטית של חדר הלב השמאלי
11	1.4. רציונל גישת הפתרון
11	1.5. סקירה של הרעיון בפרויקט
12	2. רקע על האלגוריתמים שנערך בהם שימוש בפרויקט
12	2.1. אלגוריתם Fuzzy C Means ( FCM )
14	2.2. אלגוריתם Ant Colony
15	2.3. אלגוריתם Principle Components Analysis ( PCA )
16	2.4. אלגוריתם Run Based Two Scans Connected Components Labeling
20	3. הגדרת דרישות המערכת
20	3.1. דרישות פונקציונליות
20	3.2. דרישות לא פונקציונליות
21	4. בסיס הנתונים
21	4.1. תכן בסיס הנתונים
22	4.2. ארכיטקטורת בסיס הנתונים
23	5. ניתוח ועיצוב המערכת
23	5.1. תרשים זרימה
24	5.2. דיאגרמת תרחיש כלל המערכת
25	5.3. דיאגרמת תרחיש עיבוד מקדים
26	5.4. דיאגרמת תרחיש סגמנטציה
26	5.4.1. דיאגרמת תרחיש סגמנטציה – פירוק
27	5.4.2. דיאגרמת תרחיש סגמנטציה – סגמנטציה
27	5.4.3. דיאגרמת תרחיש סגמנטציה – הרכבה
28	5.5. דיאגרמת תרחיש בידוד חדר שמאל
28	5.6. דיאגרמת תרחיש ניתוח והצגת תוצאות הסגמנטציה
29	6. הפתרון המוצע בפרויקט
29	6.1. עיבוד ראשוני
31	6.2. שלב הסגמנטציה
36	6.3. דיסקציה של חדר שמאל

37	7. פיתוח המערכת
37	7.1. סביבת פיתוח
37	7.2. חבילות פיתוח
39	7.3. כלי פיתוח
39	7.3.1. כלי פיתוח – שלב העיבוד המקדים
39	7.3.2. כלי פיתוח – שלב הסגמנטציה
40	7.3.3. כלי פיתוח – שלב ניתוח והצגת תוצאות
41	7.4. קבצי סקריפטים במערכת
42	8. מבנה המערכת
43	9. ארכיטקטורת המערכת
43	9.1. דיאגרמת רצף כללית
44	9.2. דיאגרמת רצף עבור תהליך טעינה
45	9.3. דיאגרמת רצף עבור תהליך סגמנטציה
46	10. ממשק משתמש
47	11. בדיקות המערכת
47	11.1. בדיקות יחידה
47	11.1.1. בדיקת מימוש אלגוריתם PCA
50	11.1.2. בדיקת מימוש אלגוריתם MMAS
51	11.1.3. בדיקת מימוש אלגוריתם FCM
52	11.1.4. בדיקת מימוש אלגוריתם Run-Based two pass connected components labeling adapted to grayscale images
53	11.2. בדיקות אינטגרציה
54	11.3. מקרה בוחן
54	11.3.1. ביצוע תהליך עיבוד מקדים לתמונה
55	11.3.2. ביצוע תהליך סגמנטציה
55	11.3.3. ביצוע תהליך דיטקציה של חדר שמאל
55	11.3.4. מדדית איכות זיהוי חדר שמאל
56	12. תוצאות
56	12.1. מדדים לאיכות הסגמנטציה
57	12.2. פסאודוקוד לחישוב מדד איכות הסגמנטציה
59	12.3. בחינת מדדי איכות הסגמנטציה של האלגוריתמים השונים
60	12.4. ניתוח התוצאות

61	13. סיכום והצעות להמשך מחקר
62	14. מקורות

## נספחים

64	נספחים
65	נספח א' - הוראות התקנה
66	נספח ב' - הוראות הרצה
66	נספח ג' – מדדי איכות עבור האלגוריתם בפרויקט על המטופלים השונים

## טבלת איורים

- איור 1.1.1 הדגמה לסריקת הלב במכשיר ה-MRI בהתאם לקצב הלב
- איור 1.1.2 הצגת החתכים השונים של הלב בתמונה רדיולוגית
- איור 2.4.1  $N4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$
- איור 2.4.2  $N4-nbr(p) = \{(x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1)\}$
- איור 2.4.3 (a)  $ND(p) = \{(x-1, y-1), (x+1, y-1), (x-1, y+1), (x+1, y+1)\}$  (b)  $N8(p) = N4(p) \text{ or } ND(p)$
- איור 2.4.4 Eight-connected connectivity for the current run (a) Eight-connected pixel region for the current run; (b), (c) samples of runs eight-connected with the current run; (d) sample of no run eight-connected with the current run
- איור 2.4.5 Operations for label equivalence resolving; (a) before processing the current run; (b) after processing the current run
- איור 4.2.1 ארכיטקטורת בסיס הנתונים
- איור 5.1.1 תרשים זרימה מערכת
- איור 5.2.1 דיאגרמת תרחישים של כלל המערכת
- איור 5.3.1 דיאגרמת תרחישים עבור עיבוד מקדים
- איור 5.4.1 דיאגרמת תרחישים עבור סגמנטציה
- איור 5.4.1.1 דיאגרמת תרחישים עבור סגמנטציה שלב הפירוק
- איור 5.4.2.1 דיאגרמת תרחישים עבור סגמנטציה שלב הסגמנטציה
- איור 5.4.3.1 דיאגרמת תרחישים עבור סגמנטציה שלב ההרכבה
- איור 5.5.1 דיאגרמת תרחישים עבור הפרדת חדר שמאל
- איור 5.6.1 דיאגרמת תרחישים לבחינת והצגת התוצאות
- איור 6.1.1 קוד פייתון לקריאת תמונת DICOM
- איור 6.1.2 קוד פייתון לקריאת קובץ טקסט המתאר את סימון החדר השמאלי שבוצע על ידי מומחים באופן ידני
- איור 6.1.4 (a) Original image (b) Expert hand labeled contour (c) ROI image (d) ROI image contour
- איור 6.2.1 קוד פייתון לביצוע סגמנטציה שלב ראשון - קריאת כל התמונות בחתך
- איור 6.2.2 קוד פייתון לביצוע סגמנטציה שלב שני - חישוב פירוק PCA לתמונות בחתך
- איור 6.2.3 קוד פייתון לביצוע סגמנטציה שלב שלישי - סגמנטציה FCM + MMAS של הוקטורים העצמיים שחושבו בפירוק PCA
- איור 6.2.4 קוד פייתון לביצוע סגמנטציה שלב רביעי - סגמנטציה FCM + MMAS של הוקטורים הממוצע שחושבו בפירוק PCA
- איור 6.2.5 קוד פייתון לביצוע סגמנטציה שלב חמישי - לחשב את תמונת הסגמנטציה לכל תמונה בחתך
- איור 6.2.6 דוגמא לפירוק PCA לקבוצת תמונות לב שצולמו בבדיקה של מטופל בחתך short axis
- איור 6.2.7 דוגמא של סגמנטציה של חדר שמאל לקבוצת תמונות לב שצולמו בבדיקה של מטופל בחתך short axis ע"י שימוש באלגוריתם Fuzzy C Means תוך שיפור איכות הפתרון בעזרת MMAS ואופטימיזציה בעזרת PCA

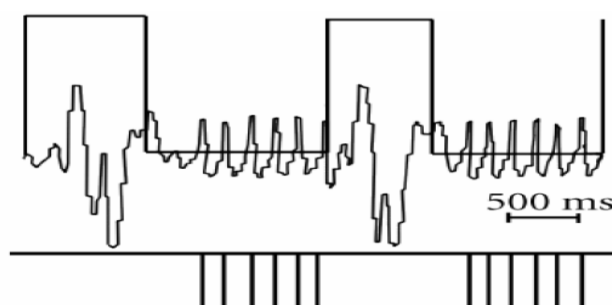
- איור 6.3.1 תמונת הרכיבים הקשירים שחושבו בעזרת אלגוריתם run based two pass connected components labeling algorithm adapted to grayscale images
- איור 8.1 תרשים מבנה המערכת
- איור 9.1.1 דיאגרמת רצף כללי
- איור 9.2.1 דיאגרמת רצף עבור תהליך טעינה
- איור 9.3.1 דיאגרמת רצף עבור תהליך סגמנטציה
- איור 10.1 ממשק משתמש לטעינת נתונים של מטופל חדש במערכת
- איור 10.2 ממשק משתמש לבחירת מטופל עבורו יוצגו תוצאות הסגמנטציה של החדר השמאלי
- איור 10.3 ממשק אינטרנטי לצפייה בתוצאות הסגמנטציה של החדר השמאלי
- איור 11.1.1.1 הזוויות השונות בהם מצולם נבדק yaleB11
- איור 11.1.1.2 תנאי תאורה בהם מצולם נבדק yaleB11
- איור 11.1.1.3 תמונות של חמשה עשר וקטורים עצמיים בעלי הערכים העצמיים הגבוהים ביותר שהתקבלו בפירוק PCA
- איור 11.1.1.4 מטריצת השונות המשותפת שהתקבלה בפירוק PCA
- איור 11.1.1.6 קוד לביצוע בדיקת מימוש אלגוריתם PCA
- איור 11.1.2.1 אזור חדר שמאל של אחד המטופלים ממאגר המידע
- איור 11.1.2.2 המחלקות שהתקבלו על ידי המרכזים שחושבו באלגוריתם MMAS
- איור 11.1.2.3 המחלקות שהתקבלו על ידי המרכזים באלגוריתם MMAS מאוחדים
- איור 11.1.3.1 תמונה אזור חדר שמאל של אחד המטופלים ממאגר המידע
- איור 11.1.3.2 תוצאות סגמנטציה שהתקבלה על ידי שימוש באלגוריתם FCM
- איור 11.1.3.3 תוצאות המרכזים שהתקבלו בסגמנטציה באלגוריתם FCM
- איור 11.1.4.1 תמונה צבעונית סירה במים
- איור 11.1.4.2 תמונת סירה במים בגווני אפור
- איור 11.1.4.3 תוצאות סגמנטציה שהתקבלה על ידי שימוש באלגוריתם FCM לתמונה סירה במים
- איור 11.1.4.4 רכיבי קשירות שהתקבלו עבור תמונה סירה במים בעזרת אלגוריתם run based two pass connected components labeling adapted to grayscale images
- איור 11.3.1.1 תמונות הלב ומתאר חדר שמאל שסומן על ידי הרדיולוג עבור המטופל שמורות בפורמט PNG
- איור 11.3.1.2 תמונות הלב ומתאר חדר שמאל אחרי חיתוך להצגה של אזור חדר שמאל
- איור 11.3.2.1 תמונות הלב ומתאר חדר שמאל אחרי סגמנטציה
- איור 11.3.3.1 תמונות הלב ומתאר חדר שמאל אחרי תהליך דיטקציה
- איור 12.2.1 קוד פייתון להערכת דיוק שיטת הסגמנטציה

# 1. מבוא

## 1.1. רקע

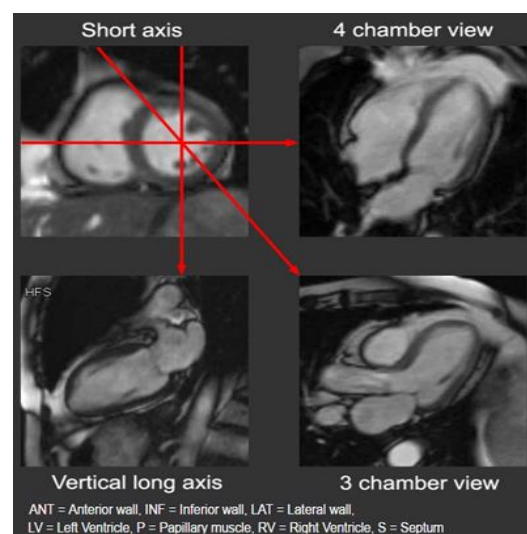
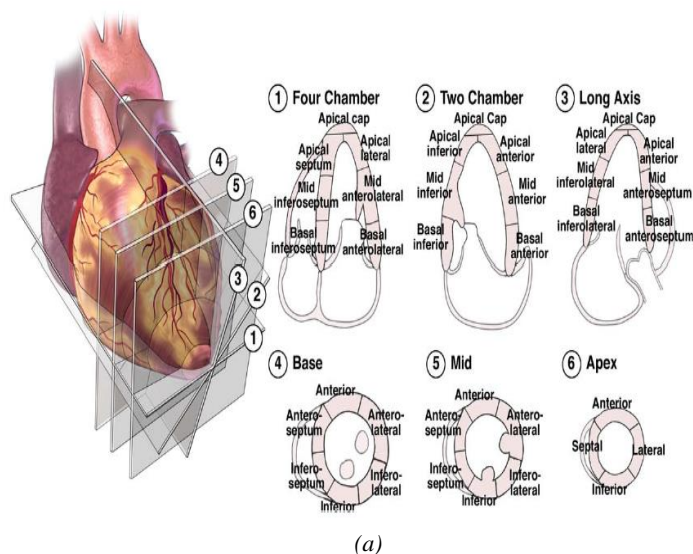
דימות תהודה מגנטית (Magnetic Resonance Imaging – MRI) מאפשרת סריקה לא פולשנית של איברים פנימיים בגוף לצורך אבחון רפואי. הסריקה נועדה לבחון איברים פנימיים בגוף ובכך לזהות מחלות ופגיעות שונות באיברים הפנימיים של המטופלים. היתרון בשימוש ב-MRI הוא שהמכשיר מאפשר סריקה ללא חשיפה מיננט. בטכניקה זאת משתמשים בשילוב של קרינת רדיו ושדה מגנטי חזק בכדי להשפיע על ספין האלקטרונים באטומי המימן שנמצאים במולקולות המים שבגוף. שינוי תכונות הספין תלוי ברקמה בה הוא נמצא, ובכך בעזרת שינוי הספין ניתן לקבל הפרדה בין הרקמות השונות.

בדיקת ה-MRI מתאימה למספר גדול של שימושים ביניהם: הערכת תפקוד הלב וחדר שמאל בפרט [27,28,29]. מערכת ה-MRI מותאמת לסריקה של הלב ודוגמת תמונות בהתאם לקצב הלב לדוגמא איור 1.1.1.



איור 1.1.1 הדגמה לסריקת הלב במכשיר ה-MRI בהתאם לקצב הלב

הצגת הלב נעשה בחתכים שונים כאשר כל חתך משמש לאבחנות שונות על תפקוד הלב דוגמאות לחתכים השונים מוצגים באיור 1.1.2. בפרויקט הנוכחי נתמקד בחתך short-axis cine MRI בלבד. חתך זה נרכש במאונך ללב ומציג פריסה של החדר השמאלי והימני. חתך זה משמש לחישובים שונים על תפוקת הלב והערכת מצבו.



איור 1.1.2 (a)(b) הצגת חתכים שונים של חדר שמאל של הלב בתמונת MRI

לאחר תהליך הסריקה הרדיולוג נדרש להעריך את תפוקת הלב השמאלית מתוך תמונות חתך short axis שהתקבלו. כדי לעשות זאת הרדיולוג מבצע תהליך סגמנטציה של חדר שמאל מתוך תמונת חלל הלב הכללית. הסגמנטציה מאפשרת לרדיולוג להעריך את כמות הדם שהצטברה בחלל הלב השמאלי בנקודת הזמן שבה התמונה נדגמה. נכון להיום תהליך הסגמנטציה נעשה באופן ידני ועל ידי כך שהרדיולוג מותח קווים בתמונה המתארים את החללים השונים.

תהליך מתיחת הקווים באופן ידני על ידי הרדיולוג הוא תהליך מתיש ואורך זמן רב. כדי לעמוד במגבלת זמן הרדיולוגים לא מבצעים סגמנטציה מלאה של כל התמונות בחדר רק באופן חלקי לתמונות מסוימות שהתקבלו בדגימה.

הרדיולוג מזהה מכלל התמונות רק את התמונות של הלב בשתי נקודות הזמן הבאות ורק להן מבצע סגמנטציה:

- סוף סיסטולה (פעולת התכווצות מדורי הלב).

- סוף דיאסטולה (תקופת המנוחה בין שתי התכווצויות הלב דבר המאפשר למדורי הלב להתמלא בדם).

נקודות זמן אלו מאפשרות לרדיולוג לקבל הערכה על תפקודו של הלב בצורה הטובה והמינימאלית ביותר. אופן בחירת התמונות מתבססת על כך שנפחו של חדר שמאל בסוף דיאסטולה הינו מקסימלי ופוחת בהדרגה לאחר התכווצות שריר הלב בשלב הסיסטולה. לפיכך סגמנטציה בשלבים האלו תתן לנו מושג על נפח הלב המינימאלי וכן המקסימאלי. וכך ניתן ללמוד רבות על תפקוד הלב.

פרמטר חשוב להערכת תפקוד הלב הינו הפרמטר שקובע את תפוקת הלב ונקרא **Ejection Fraction (EF)**. פרמטר זה מודד את אחוז התרוקנות הלב בכל פעימה. לצורך חישוב EF נעזרים בנפחים שחושבו בשלב הקודם המתארים את הנפח של חדר שמאל של הלב בסוף הסיסטולה והדיאסטולה. החישוב מתבצע ע"י חילוק מנת היחס בין נפח הלב המרוקן (נפח הלב בסוף הדיאסטולה **(EDV)**) פחות נפח הלב בשלב הסיסטולה **(ESV)**) בנפח הלב בסוף הדיאסטולה **EDV**.

$$EF = (EDV - ESV) / EDV \quad (Eq.1.1.1)$$



## 1.2. הגדרת הבעיה

החסרונות העיקריים בביצוע סגמנטציה ידנית להערכת תפקוד הלב הם: **תהליך איטי** המבזבז משאבי זמן של הצוות הרפואי וזמן יקר עבור החולה ומהווה משימה סזיפית ומעייפת לצוות הרפואי. במסגרת מגבלת הזמן נאלץ הרופא לנטר את תפקוד הלב באמצעות סגמנטציה לשתי תמונות מהחתך בלבד. דבר זה הוא חסרון משמעותי בעולם הרפואי היות ומדובר בחיי אדם ועיתוי ניתוח המידע לעיתים קריטי בהצלת חיים. לפיכך מערכת ממוחשבת שיכולה לתפקד 24/7 תסייע בהפחתת הזמנים.

**חוסר עקביות** הנובע בין היתר ממוערבות הגורם האנושי המפעיל שיקול דעת בבחירת התמונות וסימונם. השוני יכול לבוא לידי ביטוי הן בבחירות שונות של תמונות המציגים את סוף הסיסטולה והדיאסטולה והן בסימונים של החללים השונים. חוסר העקביות יכולה להיות בין רופאים וכן גם אותו רופא עשוי לספק תוצאה שונה על אותו קלט של חתך.

**שימוש חלקי במידע.** כמו שציינו הרדיולוג בבואו להעריך את תפקוד הלב משתמש בהערכה על פי חישוב נפחים בשתי תמונות מהחתך בלבד. פועל יוצא מכך, נעשה שימוש מצומצם מהמידע שהופק ממכשיר ה-MRI, דהיינו אם המכשיר מפיק 30 תמונות, נעשה שימוש רק ב-2 מהן. בחלופה האופטימלית יכולה להתקבל רמת דיוק טובה יותר באבחון אם ניתן היה להשתמש במספר רב יותר של תמונות שמופיעות בחתך.

**חוסר דיוק** הנובע ממספר גורמים וקשור לכל אחד מהחסרונות שציינו קודם. נכון להיום, לגורם האנושי חלק מכריע בקביעת תפקוד הלב. אולם אין להשען על כך מכיוון שכבני אדם אנו מושפעים מגורמים סביבתיים כגון: לחץ זמן, שעות עבודה ומנוחה. לעומת זאת, מחשב מסוגל לתפקד ברמת דיוק גבוהה לכל אורך היום.

**קושי בניתוח מידע רב שהולך וגדל במהלך השנים** כיום לדוגמא בדיקת סייטי בטן כוללת 1,500 תמונות, כאשר לפני 20 שנה כללה 100 תמונות. לעומת ההתקדמות הטכנולוגית לשמירת והצגה מידע רב על המטופל הטכנולוגיה איננה מחליפה את הגורם האנושי בביצוע ניתוח המידע ואז נותר הרדיולוג עם כמות רבה של מידע אך הוא מוגבל בזמן לפענוח התמונות.

**כמות מטופלים.** הבדיקות כיום הרבה יותר נגישות וזמינות לכלל האוכלוסייה. אם בעבר הייתה מגבלה להפנות מטופלים לבדיקת MRI כיום יש הנגשה לאוכלוסייה רחבה יותר. אי לכך, לרדיולוגים יש יותר מטופלים ויותר מידע הדורש פענוח.

**כוח אדם.** הכשרה ולימוד והתמקצעות בתחום הרדיולוגיה הוא תהליך הצורך זמן.

המסקנה הנובעת מכך היא שיש חשיבות בהפיכת תהליך הסגמנטציה לממוחשב ואוטומטי. דבר זה יאפשר אבחון מדויק ומהיר יותר של מטופלים עם בעיות רפואיות ובפרט בעלי בעיות לבביות.

### 1.3. אלגוריתמים קיימים לביצוע סגמנטציה אוטומטית של חדר הלב השמאלי

סגמנטציה בתחום ראייה ממוחשבת מתייחסת ליכולת לחלק את התמונה למחלקות המפרידות אובייקטים שונים בתמונה. בתחום עיבוד תמונה רפואית מבצעים סגמנטציה לשם איתור אזורים בעלי חשיבות רפואית והפרדתם מהרקע ואזורים אחרים כדוגמת סגמנטציה לחדר שמאל מתמונות MRI לבביות. לתחום זה חשיבות רבה הן בהצלת חיי אדם והן בחסכון במשאבי מערכת. חשיבות התחום בהצלת חיי אדם נובעת מכך שמחלות לבביות מהוות גורם התמותה העיקרי במדינות המפותחות. לפיכך, איתור כשלים במערכת הלב מאפשרים לרופא לפענח ולהחיש לחולה את הטיפול ולמנוע תמותה. חשיבות התחום בחסכון במשאבי מערכת נובעת מהשימוש הרב שנעשה בבדיקות אלו במהלך ביקורת תקופתית. בשל החשיבות נעשו מחקרים רבים במטרה להפוך את פיענוח הבדיקות הללו ליותר אוטומטיות וכך לסייע לרדיולוג [1, 2].

ניתן לסווג עבודות אלו לארבע שיטות עיקריות:

- שיטות המבוססות על מאפייני תמונה כגון סיווג על פי גבולות (thresholding), איתור ספים (edge detection), מורפולוגיות (morphology), שיטות הקבצה (clustering) [3].
- שיטות המבוססות על הצגה פרמטרית של עקם ועדכון משוואת העקם כדי לחלץ את הצורה הרצויה בתמונה [4, 8, 9, 10].
- שיטות המבוססות על אימון נתונים על תמונות בהם התבצע תהליך הסגמנטציה ואיתור יכולת לביצוע תהליך כזה באופן עצמאי על קלטים שלא הופיע בסט האימון [5].
- שיטות המבוססות על שימוש באלגוריתמים מתחום תאוריית הגרפים. ממירים את התמונה לגרף על ידי כך שכל פיקסל בתמונה מתואר כצומת ובין פיקסלים סמוכים קיימת קשת בעלת ערך המבטא את מידת הדימיון של אותם פיקסלים [6].

כל אחת מהשיטות משתמשת בטכניקה שונה על מנת לנסות לפתרון את הבעיה. לכל שיטה יתרונות וחסרונות הנובעים בין היתר על הנחות שונות שעליהם מתבססת השיטה.

## 1.4. רציונל גישת הפתרון

בפרויקט זה ננסה להציג שיטה נוספת לפתרון הבעיה של סגמנטציה אוטומטית לחדר שמאל של הלב. השיטה שבחרנו לממש תבצע את הסגמנטציה בצורה שלדעתנו תהיה יותר תואמת לשוני בין המטופלים השונים ובסבירות גבוהה תפיק תוצאות טובות יותר משאר האלגוריתמים. להבדיל מהשיטות הקיימות השיטה המוצעת בפרויקט לא תסתמך על מאפיינים שונים כגון צורת הלב או גווניו אלא היא תתאים את עצמה לכל תמונה לבבית של מטופל ותבצע סגמנטציה עמומה (fuzzy) ובכך אולי תתן תוצאות טובות ומותאמות יותר של מצבים רפואיים שונים של הלב.

## 1.5. סקירה של הרעיון בפרויקט

בפרויקט זה מוצעת שיטה חדשה לסגמנטציה אוטומטית של חדר שמאל של הלב מתמונת MRI לבבי.

לצורך כך, פיתחנו מערכת המשלבת מספר אלגוריתמים מתחום עיבוד תמונה ראייה ממוחשבת ולמידה חישובית. השימוש בשיטות השונות ואופן שילובים בכדי לבצע סגמנטציה נעשה באופן יחודי וחדשני.

המטרה היא לבחון את האלגוריתמים האלו על הבעיה של סגמנטציה של חדר שמאל (LV) מתמונות MRI ולבדוק אם יש בהם ערך מוסף.

השיטה מתבססת על סגמנטציה באמצעות אלגוריתם **Fuzzy C Means** תוך שיפור:

- איכות הפתרון על ידי שימוש באלגוריתם **Min-Max Ant System (MMAS)**.
- זמני הריצה על ידי שימוש באלגוריתם **Principle Components Analysis (PCA)**.

הפרויקט מהווה פריצת דרך בשימוש באלגוריתמים אלו לצורך ביצוע סגמנטציה של MRI לבבי. הסיבה לכך היא שבמחקרים שנעשו עד היום השימוש בכל אחד מהאלגוריתמים האלו לפתרון הבעיה המתוארת הינו מועט ואפילו חלק מהאלגוריתמים אף לא נבחנו כלל. זאת ועוד אנחנו מתארים ובוחנים בפעם הראשונה יכולת של שילוב האלגוריתמים לצורך סגמנטציה של תמונות MRI לבבי. בנוסף, השיטה שבחרנו לממש מבצעת סגמנטציה בצורה שלדעתנו תואמת יותר לשוני בין המטופלים השונים ולמקרים הפתולוגיים השונים של הלב ובכך מאפשרת להפיק תוצאות טובות יותר מהאלגוריתמים הקיימים כיום. השיטה תבחן על מספר מקרי בוחן באופן השוואתי לשיטות המקובלות בספרות.

## 2. רקע על האלגוריתמים שנערך בהם שימוש בפרויקט

בפרק זה נציג אלגוריתמים שהשתמשנו בפרויקט לצורך סגמנטציה אוטומטית של חדר שמאל. יש לציין כי בפרויקט הנוכחי נעשה שימוש באלגוריתמים קיימים לצורך פתרון הבעיה. עם זאת אופן שילובם והתאמתם לבעיה הנוכחית מאפיינת את הייחודיות של הפתרון.

### 2.1 אלגוריתם Fuzzy C Means (FCM)

**אלגוריתם FCM** מנסה לפתור את בעיית החלוקה לאשכולות באופן עמום (fuzzy) [11-12].

בעיית החלוקה לאשכולות הינה בעייה שנחקרה בתחום כריית המידע ומתייחסת למשימה של חלוקת אובייקטים שונים לקבוצות. החלוקה נעשת כך שאובייקטים בכל קבוצה יהיו דומים ככל שניתן והאובייקטים בין המחלקות השונות יהיו בעלי הבדל גדול ככל שניתן.

בתחום עיבוד תמונה נעזרים בפתרונות לבעיית החלוקה לסגמנטציה של תמונות.

היתרון בשימוש באלגוריתם ה-FCM הוא שהאובייקטים יכולים להיות משוייכים ליותר ממחלקה אחת.

האלגוריתם פותר את בעיית החלוקה על ידי כך שהוא מנסה למזער ככל שניתן את ערך פונקציית המטרה המתוארת בנוסחא 2.1.3.

Step 1:

Randomly select  $c$  cluster centers.

Step 2:

Calculate the fuzzy membership grade function  $u_{ij}$  for every corresponding  $i$  and  $j$  (the membership of data  $x_i$  to cluster  $c_j$ ).

Using the formula:

$$(Eq.2.1.1) \quad u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

Step 3:

Compute the fuzzy centers  $c_j$  for every corresponding center  $j$

Using the formula:

$$(Eq.2.1.2) \quad c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m}$$

Step4:

If the coefficients' change between two iterations is no more than  $\epsilon$  (the given sensitivity threshold) then stop; otherwise return to step 2

$$(Eq.2.1.3) \quad J_m = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m \|x_i - c_j\|^2$$

**רמת הדיוק ומהירות הריצה של אלגוריתם FCM תלויים במרכזים שנבחרו בשלב הראשוני.**  
 לכן כדי לשפר את איכות האלגוריתם נדרשת בחירה מושכלת של מרכזים התחלתיים. אי לכך, נעזרים באלגוריתם שונים כדי לנסות לאתחל את המרכזים ההתחלתיים לערכים כך שיהיו המתאימים ביותר ויפיקו את תוצאות הסגמנטציה הטובות ביותר. לכן, במקום בחירה אקראית של מרכזים בפרויקט הזה נשתמש באלגוריתם FCM תוך שימוש באלגוריתם המדמה מושבה נמלים לבחירת המרכזים ההתחלתיים לשיפור התוצאות.

## 2.2. אלגוריתם Ant Colony

**אלגוריתם מושבות נמלים (Ant Colony)** נוצר בהשראת וחקי התנהגות הנמלים בטבע. אחת מהיכולות המפתיעות של נמלים היא לאתר מסלולים קצרים ביותר למזון. מחקרים ביולוגיים הראו כי הנמלים מתקשרות ביניהן באמצעות הפרשת תרכובת כימית לאורך מסלול צעידתם ובאמצעות כך יודעות לאבחן את המסלול הקצר ביותר. בתחום מדעי המחשב מנסים לחקות את התנהגות הנמלים בבואם לפתור או לתת קירוב מספיק טוב לבעיות שונות ממחלקת הסיבוכיות NP.

באותו אופן ניתן להשתמש בטכניקה זו כדי לפתור את בעיית הסגמנטציה של תמונה [13-16]. נשתמש באלגוריתם זה כדי ליצור מושבה של נמלים בה כל נמלה בוחרת לשייך פיקסל  $s$  בתמונה למחלקה  $c$  תוך שימוש בכללים הסתברותיים. האלגוריתם [13-16] מתבצע באופן הבא:

Create ant colony and let it search the image for a number of iterations:

- Generate a random value  $q$  from 0 to 1.
- If this value  $q \leq q_0$  where  $q_0$  is a constant representing the ratio of using history pheromone trails results over exploration of new solution then

- Cluster is selected according to  $\arg \max ( \tau(s, c) * (\eta(s, c)^\beta) )$ .

- Otherwise

- Select cluster using a probabilistic action selection rule,

$$\Pr(s, c) = \frac{\tau(s, c) * (\eta(s, c)^\beta)}{\sum_{u \in labels} \tau(s, u) * (\eta(s, u)^\beta)}.$$

$\beta$  is the weight of the importance of the pheromone trail and the heuristic function.

$\tau(s, c)$  is the amount of pheromone deposit by an ant when the pixel  $s$  is mapped into cluster  $c$ .

$\eta(s, c)$  is heuristic function which set to be the similarity and dissimilarity between grayscale value of the pixel to the mean grayscale level of the candidate cluster and other clusters.

- Once a solution is built it is evaluated and pheromone is deposit relative to the quality of the solution.

ants deposit pheromone on the arcs they visited using the formula

$$\tau(s, c) += \sum_{k=1}^{ants \ number} \Delta \tau(s, c)_k.$$

Where  $\Delta \tau(s, c)_k$  is the amount of pheromone ant  $k$  will add to the arc mapping pixel  $s$  to cluster  $c$ .

if arch  $(s, c)$  is in the path of ant  $k$  then

- $\Delta \tau(s, c)_k = \frac{1}{c_k}$  Where  $C_k$  is the total cost of the path solution

Otherwise 0.

החסרון של שימוש באלגוריתם סגמנטציה באמצעות מושבה של נמלים, הוא איטיות התהליך הנובע ממורכבות התהליך.

אחת השיטות שפותחו כדי לשפר את הביצועים הוא האלגוריתם **Max-Min Ant System** (MMAS). האלגוריתם פועל דומה לאלגוריתם Ant Colony למעט מספר ההבדלים. ההבדל הראשון הוא הגבלה על כמות התרכובת הכימית במסלולים להיות בטווח של  $[T_{min}, T_{max}]$ . ההבדל השני הוא שהמערכת מאתחלת את כל הצמתים במסלולים לערך תרכובת כימית מקסימאלית  $T_{max}$ . בעזרת השינויים האלו המערכת מציגה ומתייצבת על פתרונות טובים יותר במהירות גבוהה יותר.

## 2.3. אלגוריתם (PCA) Principal Components Analysis

השימוש באלגוריתם **Principal Components Analysis (PCA)** כשיטה סטטיסטית למציאת ייצוג ממימד נמוך למידע במימד גבוה ותוך פגיעה מינימאלית במידע. האלגוריתם מחשב טרנספורמציה לינארית אורתוגונלית אופטימלית להמרת סט דגימות לסט וקטורים לא תלויים הנקראים principal components (וקטורי הפירוק).

להלן תיאור שלבי החישוב:

א. מחשבים את מטריצת השונות המשותפת (covariance)

$$\text{Cov}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \quad (Eq.2.3.1) \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (Eq.2.3.2)$$

( $\mathbf{x}_i$  = הדגימה ה-i-ית)

- ב. מחשבים את הוקטורים העצמיים של מטריצת השונות המשותפת.
- ג. מחשבים את הערכים העצמיים המתאימים לכל וקטור עצמי של מטריצת השונות המשותפת.
- ד. ממיינים את הוקטורים העצמיים על פי ערכי הערכים העצמיים התואמים להם בסדר יורד.
- ה. להורדת מימד יש לוותר על הוקטורים העצמיים בעלי הערכים העצמיים הנמוכים ביותר.

בכדי לשחזר ולבדוק הטלה של מידע במרחב שנוצר עלינו לחשב

$$\text{PCA\_reconstruction} = \text{PCA\_score}[i] * \text{Eigenvector}[i] + \boldsymbol{\mu} \quad (Eq.2.3.3)$$

## 2.4. אלגוריתם Run-Based Two Pass Connected Components Labeling

אלגוריתמים למציאת רכיבים קשירים מקבלים כקלט תמונה ומנסים למצוא ולהפריד אובייקטים שונים בתמונה ולסמן תוויות שונות לרכיבים בעלי קשר שונה. כל הרכיבים שמופו לאותו אובייקט הם בצורה כלשהי מחוברים. כל אלגוריתם קובע את דרישת הסמיכות שיקבעו אם פיקסל מסויים יחשב בעל קשר לקבוצה מסומנת.

להלן דוגמאות לפונקציות היסודיות והבסיסיות למציאת קשר/שכנות:

- 4-adjacency: הפיקסלים  $p$  ו- $q$  הם 4-adjacency אם ערך גווני הצבע שלהם שייכי לאותה קבוצה  $V$  ובנוסף הפיקסלים נמצאים בסמיכות  $N_4$  המוצגת באיור 2.4.1.

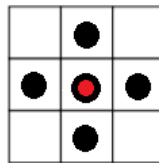


Figure 2.4.1:  $N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$

- 4 neighbor forward raster scan adjacency: הפיקסלים  $p$  ו- $q$  הם 4-nbr adjacency אם ערך גווני הצבע שלהם שייכי לאותה קבוצה  $V$  ובנוסף הפיקסלים נמצאים בסמיכות  $N_{4-nbr}$  המוצגת באיור 2.4.2.

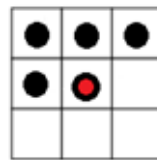
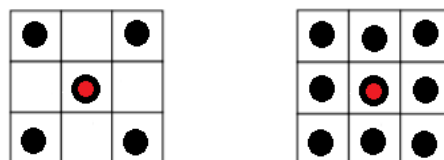


Figure 2.4.2:  $N_{4-nbr}(p) = \{(x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1)\}$

- 8-adjacency: הפיקסלים  $p$  ו- $q$  הם 8-adjacency אם ערך גווני הצבע שלהם שייכי לאותה קבוצה  $V$  ובנוסף הפיקסלים נמצאים בסמיכות  $N_8$  המוצגת באיור 2.4.3.



(a) (b)

Figure 2.4.3: (a)  $ND(p) = \{(x-1, y-1), (x+1, y-1), (x-1, y+1), (x+1, y+1)\}$  (b)  $N_8(p) = N_4(p) \cup ND(p)$

- m-adjacency: הפיקסלים  $p$  ו- $q$  הם 4-adjacency אם ערך גווני הצבע שלהם שייכי לאותה קבוצה  $V$  ובנוסף הפיקסלים נמצאים בסמיכות הבאה:  
 $((q \in N_4(p)) \text{ or } (q \in N_8(p) \text{ and not exists } t \text{ value from } V \text{ and } t \in N_4(p) \cap N_4(q)))$



ישנן מספר גישות למציאת רכיבים קשירים בתמונה שאותן ניתן לסווג לשלושה סוגי מחלקות עיקריות:

#### Raster-scan and label equivalence resolving algorithm:

התמונה נסרקת פיקסל אחר פיקסל לפי הסדר וכאשר נמצא פיקסל אשר לא שויך לו תוית אזי האלגוריתם מנסה לשייך לו תוית על ידי חיפוש פיקסלים סמוכים שלהם יש תוית. אם האלגוריתם לא מצא פיקסלים סמוכים שלהם יש תוית אזי האלגוריתם משייך את הפיקסל לתוית חדשה אותה הוא יוצר. בנוסף אם האלגוריתם מוצא שפיקסל מסוים משתייך ליותר מקבוצה אחת של תויות אזי משתמע שהקבוצות צריכות להתאחד לקבוצה אחת. את התויות של הקבוצות שצריכות להתאחד האלגוריתם שומר במערך זהויות.

#### Searching and label propagation:

התמונה נסרקת פיקסל אחר פיקסל לפי הסדר וכאשר נמצא פיקסל אשר לא שויך לו תוית אזי האלגוריתם משייך את הפיקסל לתוית חדשה אותה הוא יוצר. לאחר מכן האלגוריתם מנסה לשייך פיקסלים נוספים לתוית החדשה שנוצרה על ידי סריקת הסמיכויות של הפיקסלים עם התויות החדשה.

#### Contour tracing and label propagation:

האלגוריתם מחפש אחר גבולות של רכיבים שכבר אותרו. לגבולות של הרכיבים האלגוריתם נותן תוית חדשה. ובדומה לאלגוריתם הקודם מנסה לשייך פיקסלים נוספים לתוית החדשה.

בפרויקט זה נשתמש באלגוריתם המוצא קשירויות בתמונה על ידי מעבר כפול (Two pass). האלגוריתם תואר תחילה במאמר [18] ומשמש למציאת רכיבי קשירות בתמונה בינארית.

האלגוריתם [18] פועל בשני שלבים:

**בשלב הראשון** האלגוריתם מבצע מעבר על התמונה ומעניק סימונים ראשוניים לכל הפיקסלים בתמונה ושומר מטריצת זהויות באופן הבא:

Scan image by moving along a row for each pixel  $p(i)$

$i$  form 0 to  $N \times M$ , where  $N$  is the image rows  $M$  is image columns and  $p(y \times N + x)$

represents the pixel value at  $(x,y)$  location.

- If pixel value is  $V = V_o$  ( $V_o$  means object pixel contrary to  $V = V_B$  which represent background pixel) then
  - A new run  $r(s, e)$  is found the run data is recorded.
  - A run is a block of continuous object pixels in a row when  $r(s,c)$  mean a block of continuous object pixels starts from pixel  $p(s)$  and ends in pixel  $p(e)$ .
  - Detect adjacency area with the current run in the above row.

For example let  $r(s, e)$  be the current run so by saying eight connected adjacency connectivity we mean all runs in the row above the scan row such that one of their pixels occurs between  $p(s - N - 1)$  and  $p(e - N + 1)$  are connected to current run.

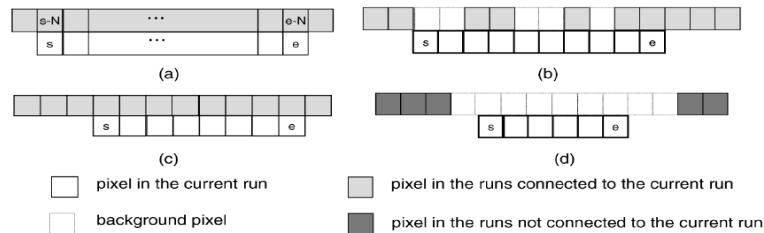


Figure 2.4.4: Eight-connected connectivity for the current run (a) Eight-connected pixel region for the current run; (b), (c) samples of runs eight-connected with the current run; (d) sample of no run eight-connected with the current run

- If there is no adjacency with the current run in the row above the scan row then it means the current run belongs to a new connected component that was not found so far
  - Assign all pixels in the current run with a new label.
- Else if there are some adjacency with the current run in the row above the scan row then
  - Assign all pixels in the current run with the same provisional label as the upper left-most run.
  - Record all the connected labels to the same class in the equivalence matrix.

**בשלב השני** מבצעים מעבר על הסימונים הראשוניים ובהסתמך על מטריצת הזהויות מחליפים כל סימון בסימון ייצוגי של הקבוצה לה הוא משתייך במטריצת הזהויות.

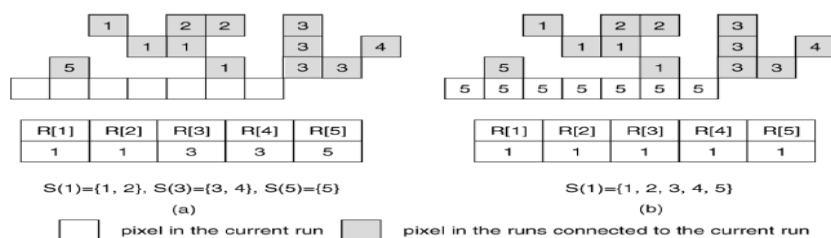


Figure 2.4.5: Operations for label equivalence resolving; (a) before processing the current run; (b) after processing the current run

במאמר [17] מוצע עדכון לאלגוריתם כך שיתמוך בתמונות בגווני אפור. באלגוריתם המתואר במאמר [17] ההפרדה לרכיבי קשירות מתבססת הן על קשירות הרכיבים והן על פרמטר המציין את ערך האפור של הפיקסלים השונים ברכיבי הקשירות. קיימת חשיבות רבה במעבר מאלגוריתם שמתבצע על תמונות בינאריות לאלגוריתם שפועל על תמונות בגווני אפור. האלגוריתם רובסטי יותר ומתאים למגוון רחב יותר של תמונות. בנוסף, במקרה שלנו אנחנו מתעדים לבצע מציאת רכיבי קשירות על תמונה שעברה סגמנטציה. מציאת רכיבי קשירות בתמונה בינארית מגבילה אותנו לקבל כקלט תמונה שעברה סגמנטציה לשתי מחלקות בלבד. עם זאת נרצה תמיכה גם בתנאי שנרצה לבצע סגמנטציה למספר מחלקות גדול משניים. במובן זה אלגוריתם שמבצע מציאת רכיבי קשירות על תמונות בגווני אפור מאפשר לנו תמיכה רחבה יותר בתמונות שעברו סגמנטציה עד 256 מחלקות.

### 3. הגדרת דרישות המערכת

#### 3.1. דרישות פונקציונליות

##### דרישה תפעולית (OR = Operational Requirements):

מטרת הפרויקט הינה פיתוח מערכת שתבצע סגמנטציה של חדר שמאל של הלב באופן אוטומטי מלא. המערכת תהיה אוטומטית באופן מלא כלומר לאחר קבלת הקלט שהינו תמונת MRI של הלב על המערכת להחזיר את התוצאות השונות הכוללת סגמנטציה של חדר שמאל של הלב בתמונה ללא התערבות נוספת של המשתמש.

##### דרישת מידע (DR = Data Requirements):

המערכת תקבל קלט נתונים הכוללים תיקייה המכילה תמונות MRI שהתקבלו בחדר סריקה short axis cine. לאחר עיבוד הנתונים והרצת אלגוריתם הסגמנטציה על המערכת להחזיר פלט המציג את תמונת חדר שמאל של הלב כפי שהמערכת הצליחה לאתר.

המערכת תתעד ותשמור תוצאות של השלבים השונים (1) שלבי עיבוד התמונה וניקויה מרעשים, (2) שלבי דיטקציה, (3) שלבי הסגמנטציה, (4) שלב בידוד חדר שמאל על סמך הרכיבים הקשירים בתמונה.

#### 3.2. דרישות לא פונקציונליות

##### דרישות ביצועים (PR = Performance Requirements):

- מהירה – על המערכת להפיק פתרונות לכל מטופל בזמן סביר שלא יעלה על דקה למטופל.
- יעילה – יישום האלגוריתמים בפרויקט תוך חסכון במשאבי זמן ומקום.

##### מאפייני איכות (QA = Quality Attributes):

- מדויקת – על המערכת להפיק תוצאות איכותיות.
- התוצאות ייבחנו בהשוואה לתוצאות ניתוח ידני שביצעו רדיולוגים מוסמכים.
- בנוסף נבצע השוואה של התוצאות שהתקבלו למול אלגוריתמים אוטומטים מתקדמים שפותחו לאותה מטרה.
- עמידות – מערכות רפאיות בנויות כך שהן עמידות בפני השוני בין מטופלים.
- כל מטופל וכל בדיקה תפיק סט ייחודי של קלט שהמערכת נתקלת בו בפעם הראשונה. על המערכת להיות עמידה בפני המקרים השונים. בשל חשיבות יכולת המענה במקרים השונים נבחן את המערכת הן במטופלים בעלי תפקוד לבבי תקין והן בקשת של מטופלים עם מחלות לב שונות.
- אמינות (reliability) – על המערכת לפעול ללא תקלות לאורך זמן.
- זמינות (availability) – על המערכת לתת שירות רציף.
- אחזקתיות (maintainability) – נבנה את המערכת כך שקטעי הקוד מוגדרים היטב. כל קטע בתוכנית מוגדר לתחום פעילותו כלומר בכל קטע תוגדר המטרה, הקלט, העיבוד הנדרש, והפלט הרצוי.
- שימושית (usability) – המערכת תאפשר לרדיולוג לבצע סגמנטציה אוטומטית תוך מתן דגש לדייקן וריזות ונוחות תפעול.

## 4. בסיס הנתונים

### 4.1. תכן בסיס הנתונים

בפרויקט נעשה שימוש במאגר מידע שנאסף מביה"ח קנדי Sunnybrook health science center [19].

בסיס נתונים התפרסם כחלק מאתגר MICCAI'09 ופתוח לשימושים אקדמיים.

בחרנו להשתמש בבסיס הנתונים הנתון ממספר סיבות עיקריות:

- מאגר הנתונים פתוח לשימוש לצרכים אקדמיים.
- המאגר מכיל תמונות MRI לבבי של מספר רב של מטופלים שונים ועם מגוון רחב של בעיות לבביות. לפיכך, נוכל לבדוק את איכות האלגוריתם שלנו על מקרים פתולוגיים שונים.
- המאגר מכיל בנוסף לתמונות ה-MRI של הלב של המטופלים השונים גם תוצאות סגמנטציה ידנית שבוצעו על ידי מומחים לחדר שמאל של הלב. לכן נוכל לבחון את האלגוריתם שלנו אל מול תוצאות סגמנטציה איכותיות ומדויקות שבוצעו על ידי רופאים וישמשו אותנו כ-ground truth (התוצאה רצויה).
- המאגר נמצא בשימוש רחב במאמרים אקדמיים המציגים אלגוריתמים לביצוע סגמנטציה של חדר לב שמאל. התוצאות של האלגוריתמים בכל מאמר נבחנים על בסיס הנתונים ומתוארים במאמר. כך ניתן לבצע ניתוח השוואתי של האלגוריתם שלנו מול אלגוריתמים אחרים.
- התמונות בבסיס הנתונים נשמרו בפורמט DICOM. פורמט DICOM הינו הפורמט המוכר שבו משתמשים לשמירה ופענוח תמונת MRI במכשירים ובמערכת הרפואית.

המאגר מכיל 45 רשומות של מטופלים כאשר לכל מטופל מצויינים הפרטים הבאים:

- נתונים אישיים של המטופל כגון גיל מין משקל וכו'.
- תמונות שבוצעו למטופל :
  - לכל מטופל בוצעו כ-10 תצלומי חתכי short axis cine.
  - כל חתך מכיל 20 תמונות של הלב שבוצעו באותה סריקה.
  - מכשיר ה-MRI בו השתמשו לסרוק את המטופלים כויל עם הפרמטרים הבאים:
    - Slice thickness of 8 mm, FOV = 320 mm, Acquisition matrix 256 × 256, 1.5T MR scanner (GE Healthcare).
- סגמנטציה המתארת את החדר השמאלי התבצעה על ידי רדיולוגים מומחים. הסגמנטציה התבצעה לצערנו רק לחלק מהתמונות כאלו שהרופא בחר לכל מטופל בכל סריקה כמייצגים את הלב בשלבים של סיסטולה ודיאסטולה. תוצאות מתאר הסגמנטציה נשמרו בקובץ text.
- אבחנה רפואית על מצבו הקרדיאלי של כל מטופל.
  - N (Normal) מצב תקין - 9 מקרים.
  - HYP (hypertrophic cardiomyopathy) היפרטרופיה – 12 מקרים.
  - HF-I (Heart failure due to ischemia) – 12 מקרים.
  - HF-NI (Heart failure without ischemia) – 12 מקרים.

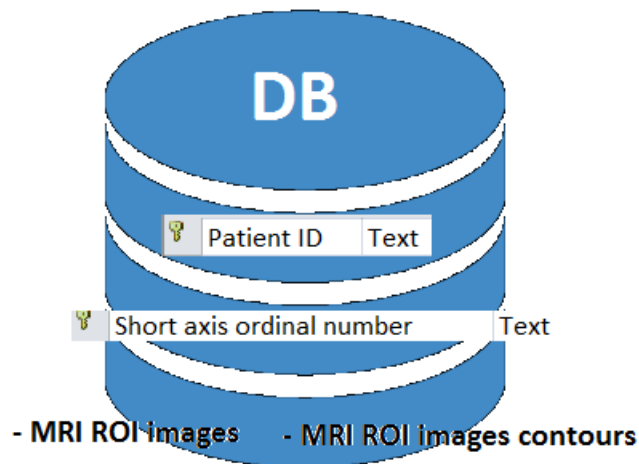
## 4.2. ארכיטקטורת בסיס הנתונים

עבור המידע שנאסף ונקרא ממאגר המידע יבוצע עיבוד ראשוני של הנתונים ושמירתם בבסיס נתונים חדש. בסיס הנתונים החדש יכיל:

- תמונת ה-MRI הלבבי לאחר שעברו עיבוד ראשוני ושמירה בפורמט מטיפוס PNG.
- תמונות הסגמנטציה שיופקו ממסמכי הטקסט שמכילים את המתאר הידני שהתבצעה על ידי רדיולוגים מומחים. תמונות הסגמנטציה ישמרו בפורמט תמונה מסוג PNG.

התמונות ישמרו במבנה קבצים תחת מפתח מספר המטופל ומספר החתך.

המאגר ישמר בארכיטקטורה הבאה:



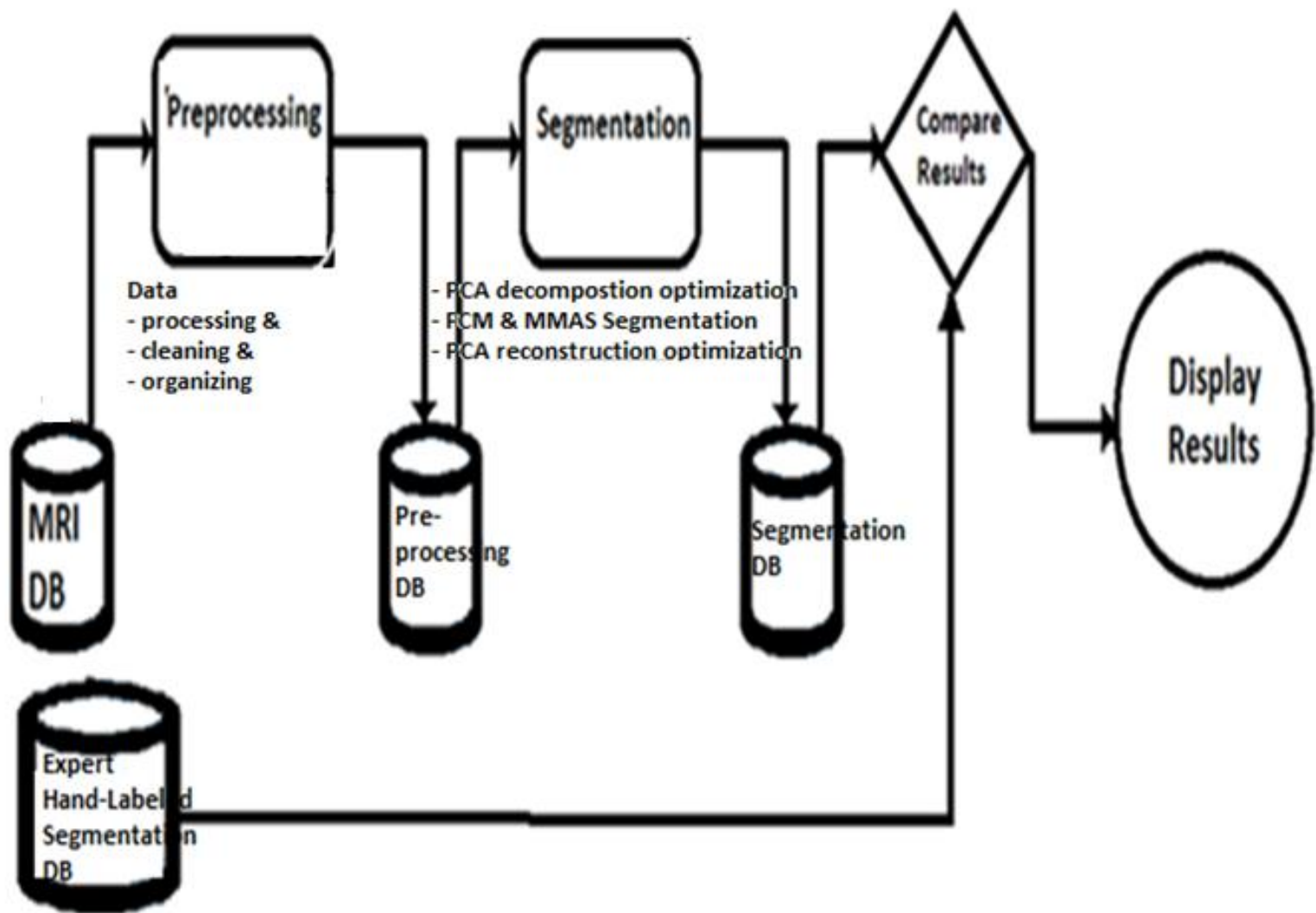
איור 4.2.1 ארכיטקטורת בסיס הנתונים בפרויקט

בפרויקט בוצע תהליך הסגמנטציה במספר שלבים. כחלק מפיתוח נכון נשמרו תוצאות כל אחד מהשלבים השונים במטרה לבחון כל אחד מהם בנפרד ולבחון את ההתקדמות בפתרון. לפיכך את התוצאות שהופקו בכל אחד מהשלבים נשמרו במאגר המידע.

## 5. ניתוח ועיצוב המערכת

תהליך סגמנציה והפרדה חדר שמאל מתמונת MRI בחתך short axis איננה משימה טריוויאלית. כדי לפשט את התהליך ניתן להפריד את פתרון הבעיה לתתי תהליכים. כל תהליך אחראי על משימה מוגדרת ומצומצמת ושילוב ההתהליכים יביא לפתרון הבעיה כולה. נציג תרשים זרימה המתאר את השלבים השונים הנדרשים לביצוע המשימה.

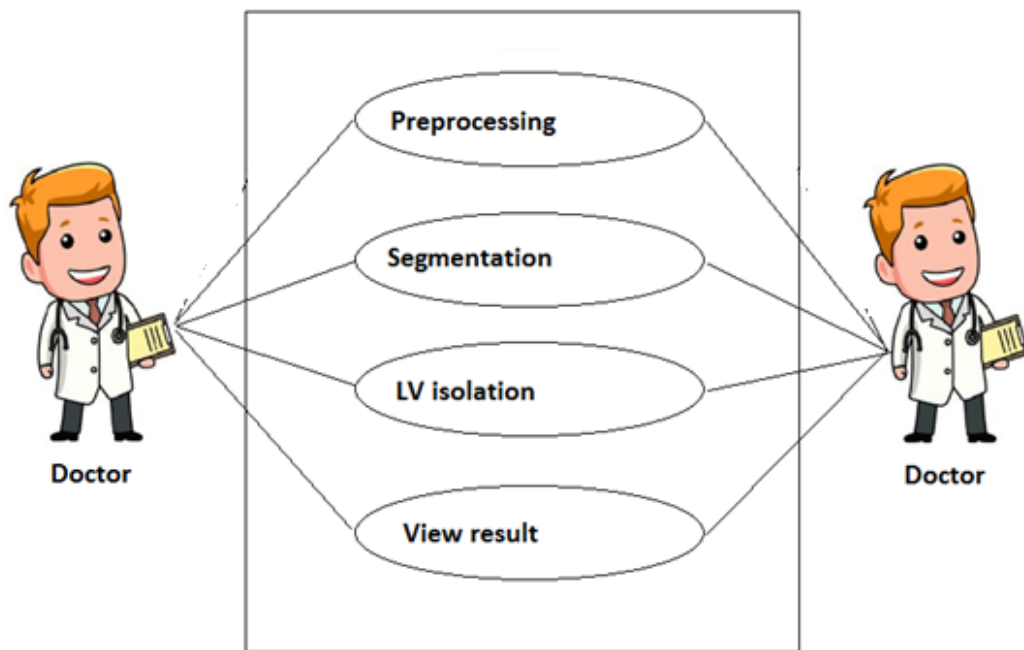
### 5.1. תרשים זרימה



איור 5.1.1 תרשים זרימה מערכת

## 5.2. דיאגרמת תרחישים כלל המערכת

נציג תרשים המתאר את התהליכים השונים במערכת.  
נשים לב שאחרי כל שלב נשמור את התוצאות שהתקבלו במאגר מידע ונעביר אותם כקלט לשלב הבא.



איור 5.2.1 דיאגרמת תרחישים של כלל המערכת

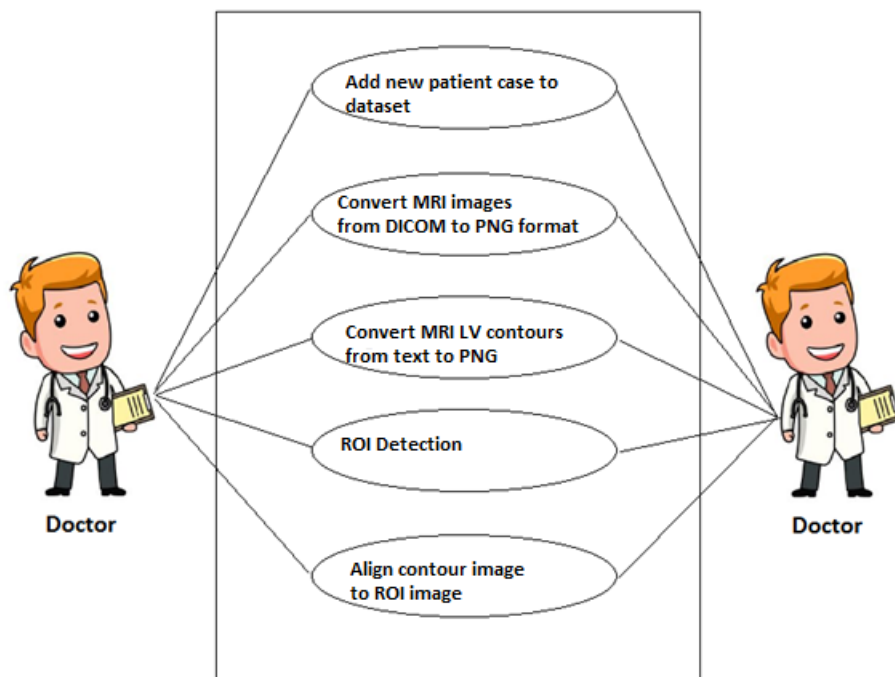


### 5.3. דיאגרמת תרחיש עיבוד מקדים

שלב העיבוד המקדים מטרתו להמיר את המידע שנאסף במאגרי מידע שונים לפורמט אחיד לבצע לו טיוב ראשוני ולשמור אותו במערכת בצורה מאורגנת.

שלב זה כולל בין היתר –

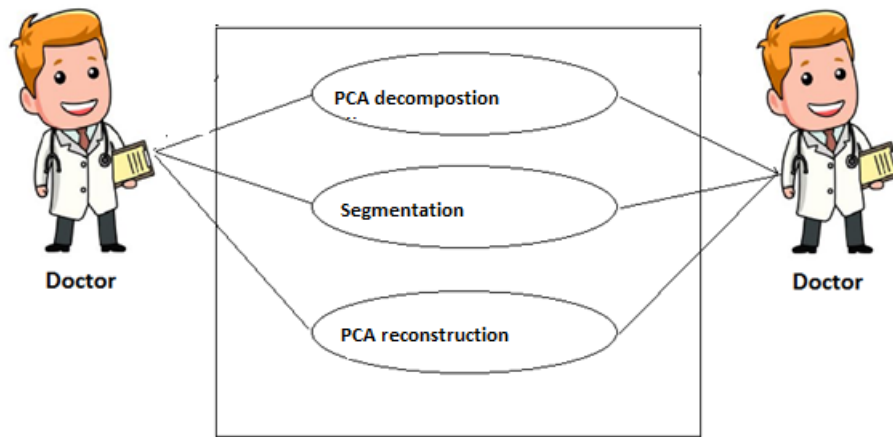
- יצירת מסד נתונים חדש שירכז את הנתונים השונים.
- המרה של תמונות ה-CMRI מפורמט DICOM לפורמט יותר מוכר לתמונות PNG.
- המרה של מתאר של חדר שמאל שתואר הרדיולוג לתמונות ה-CMRI ונשמר בקובץ טקסט.
- ההמרה תתבצע לתמונה בפורמט PNG בה הרקע מוצג בשחור וחלל הלב מסומן בלבן.
- מטרת הפרויקט היא לבצע סגמנטציה איכותית עבור החדר השמאלי של הלב, לכן נבצע תהליך זיקוק של אזור חדר שמאל מפרטים אחרים שמופיעים בתמונה.
- התאמת תמונות המתאר של חדר שמאל ששרטטו הרדיולוגים כך שיחפפו לחתך שבצענו לאזור חדר שמאל של הלב. התמונות יהיו זהים בגודלן וברזולוציה כך שנוכל להלביש את תמונת הסגמנטציה על תמונת ה-MRI ולזהות את האזור המסומן בתמונה.



איור 5.3.1 דיאגרמת תרחיש עיבוד מקדים

## 5.4. דיאגרמת תרחיש סגמנטציה

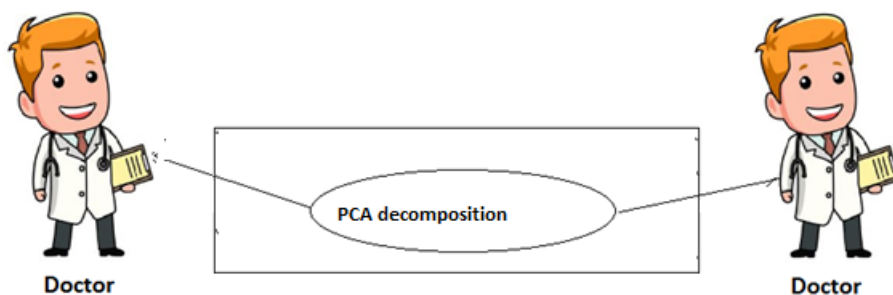
לביצוע סגמנטציה איכותית ויעילה נסתייע במספר אלגוריתמים ונפרק את התהליך למספר שלבים.



איור 5.4.1 דיאגרמת תרחיש עבור סגמנטציה

### 5.4.1. דיאגרמת תרחיש סגמנטציה – פירוק

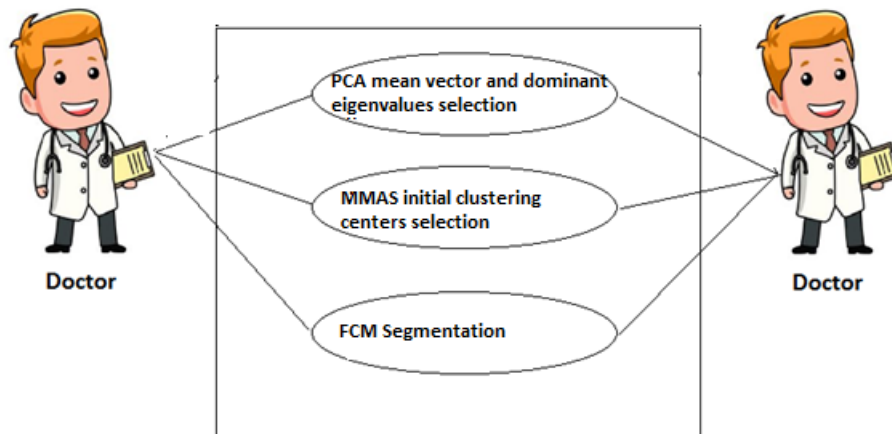
כל חתך המכיל כ-20 תמונות מפורק בעזרת אלגוריתם PCA בכדי למצוא הצגה ממימד נמוך יותר. מימד החדש שנבחר יפרס על ידי מספר וקטורים עצמיים בעלי הערכים העצמיים הדומיננטים של מטריצת השונות המשותפת והוקטור הממוצע. ובכך נאפשר הורדת מימד ושחזורו תוך פגיעה מינימאלית במידע. לאחר הפעלת אלגוריתם PCA ניתן לשחזר כל תמונה בחתך על ידי הצגתה בעזרת הוקטורים העצמיים ומטריצת השונות המשותפת.



איור 5.4.1.1 דיאגרמת תרחיש עבור סגמנטציה שלב הפירוק

## 5.4.2. דיאגרמת תרחיש סגמנטציה – סגמנטציה

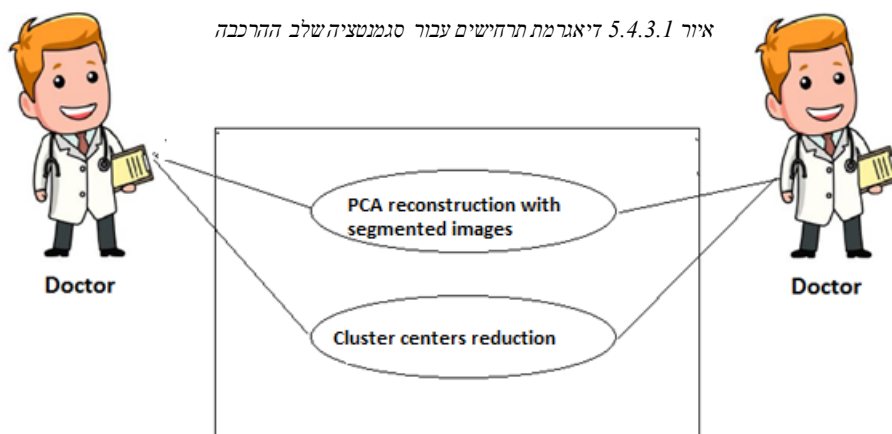
- שלב הסגמנטציה מבוצע לוקטורים העצמיים בעלי הערכים העצמיים הגבוהים ביותר והוקטור הממוצע שקבלנו אחרי הרצת אלגוריתם PCA. התוצאה שנקבל תאפשר לנו לחשב סגמנטציה על כל תמונות החתך מבלי שנדרש לעשות זאת על כל תמונה בנפרד.
- מציאת מרכזים לפני הרצת אלגוריתם FCM -
  - רמת הדיוק ומהירות הריצה של אלגוריתם FCM תלויים במרכזים הראשוניים שהוצבו באלגוריתם ולכן כדי לבחור את המרכזים ההתחלתיים נעזר באלגוריתם Min Max Ant System algorithm (MMAS).
- הרצת אלגוריתם FCM
  - סגמנטציה תעשה על ידי הרצת אלגוריתם FCM עם המרכזים שקבלנו מהרצת אלגוריתם MMAS.



איור 5.4.2.1 דיאגרמת תרחיש עבור סגמנטציה שלב הסגמנטציה

## 5.4.3. דיאגרמת תרחיש סגמנטציה - הרכבה

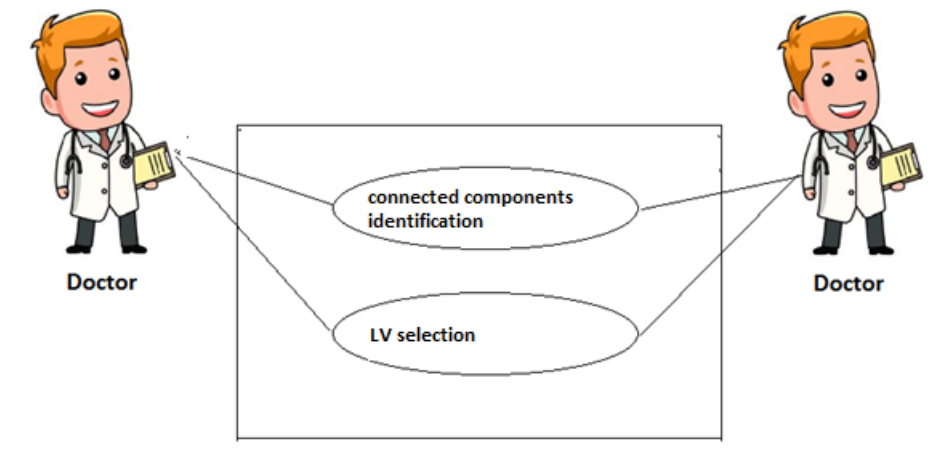
- האופן שבו נוכל לקבל את תמונת הסגמנטציה של כל תמונה בחתך דומה לאופן שבו משחזרים תמונה בחתך בעזרת הרכבת PCA. אולם להבדיל מההרכבה הרגילה אנחנו נשתמש בתמונות הרכיבים הראשיים והוקטור הממוצע לאחר שעברו סגמנטציה.
- תמונת הסגמנטציה של כל תמונה בחתך תתקבל עם מספר מחלקות גדול הרבה יותר ממספר המחלקות כמרכיבים הראשיים והוקטור הממוצע. לפיכך נדרש לצמצם את מספר המחלקות תיאור יותר מפורט כיצד יופיע בהמשך.



איור 5.4.3.1 דיאגרמת תרחיש עבור סגמנטציה שלב ההרכבה

## 5.5. דיאגרמת תרחיש בידוד חדר שמאל

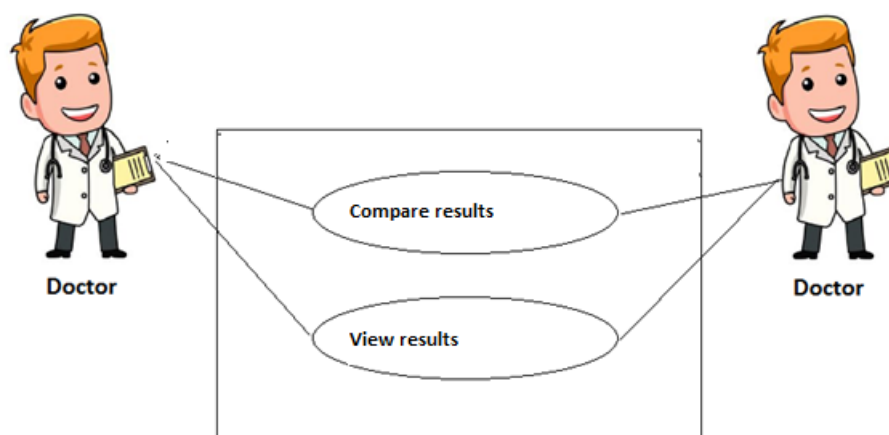
לאחר תהליך הסגמנטציה עלינו לבודד את החדר השמאלי. שלב הבידוד נעשה באמצעות אלגוריתם שמוצא את האובייקטים המקושרים בכל תמונה בגווני אפור. באופן זה נוכל לקבל את כל רכיבי הקשירות ולסמן את הרכיבים השייכים לחדר שמאל.



איור 5.5.1 דיאגרמת תרחיש עבור הפרדת חדר שמאל

## 5.6. דיאגרמת תרחיש ניתוח והצגת תוצאות הסגמנטציה

נבחן את איכות השיטה שהצענו בסימון חדר שמאל. ההשוואה תעשה אל מול הסגמנטציה הידנית של חדר שמאל בלב שבצעו רדיולוגים מוסמכים. בנוסף נשווה את תוצאות הסגמנטציה למול שיטות קיימות בספרות ונמדוד את איכות השיטה שלנו מולם בהתאם למידת הקרבה לסגמנטציה הידנית שבצעו הרדיולוגים. נציג בעזרת ממשק WEB בצורה ויזואלית את תוצאות הסגמנטציה של השיטה שלנו.



איור 5.6.1 דיאגרמת תרחיש לבחינת ההצגת התוצאות

## 6. הפתרון המוצע בפרויקט

### 6.1. עיבוד ראשוני

עיבוד מקדים נועד לסדר לארגן ולטייב את המידע לו נדרש בהמשך תהליך העבודה. בשלב זה נקרא את הנתונים שנאספנו ונמיר אותם לפורמט סטנדרטי אותו נשמור במסד נתונים בצורה מאורגנת. להלן שלבי העיבוד הראשוני שבצענו:

- **המרת תמונות מפורמט DICOM לפורמט PNG.** פורמט PNG הוא פורמט יותר נפוץ לשמירת תמונה. הסיבה העיקרית לעשות זאת היא שפונקציות ספרייה רבות בשפת Python שמבצעות עיבוד תמונה תומכות בפורמט זה להבדיל מפורמט DICOM.

```
f = dicom.read_file(full_path)
img = f.pixel_array.astype(np.int)
```

איור 6.1.1 קוד פייתון לקריאת תמונת DICOM

- **המרת קובץ הטקסט המתאר את הסגמנטציה שעשה הרדיולוג לחדר שמאל לכדי תמונה.** בתמונה הרקע יסומן בשחור וחלל הלב השמאלי יסומן בלבן.

```
ctrs = np.loadtxt(contour.ctr_path, delimiter=" ").astype(np.int32)
label = np.zeros_like(img, dtype="uint8")
cv2.fillPoly(label, [ctrs], 1)
```

איור 6.1.2 קוד פייתון לקריאת קובץ טקסט המתאר את סימון החדר השמאלי שבוצע על ידי מומחים באופן ידני

- **הפרדת אזור חדר שמאל מפרטים אחרים שהופיעו בתמונה המקורית וזאת היות ומטרת הפרויקט הייתה לבחון את יכולת הסגמנטציה עבור החדר השמאלי של הלב.** בחתך מופיעים איברים נוספים שאינם שייכים ללב לפיכך בצענו חיתוך מכל תמונה לאזור בגודל 70X70 שציג בצורה טובה ביותר את מתחם חדר הלב השמאלי. הזיקוק התבצע ע"י חיתוך תמונה בהסתמך על הידע המקדים על מיקום אזור חדר שמאל בתמונות. החיתוך בוצע כך שחדר שמאל יופיע במרכז התמונה.
- **הסגמנטציה הידינית שבוצעה על ידי הרופאים הותאמה לחיתוך שבוצע.** לצורך כך חתכנו את תמונת הסגמנטציה הידינית על פי החיתוך שבצענו בתמונה המתאימה. באופן הזה קבלנו תמונה מצומצמת יותר המתארת את אזור חדר שמאל ואת הסגמנטציה המתאימה לה.

```

def find_center(pil_image):
    """
    Find the center of expert hand labeled ROI contour image
    """
    if pil_image.mode == 'RGBA':
        r,g,b,a = pil_image.split()
        rgb_image = Image.merge('RGB', (r,g,b))
        inverted_image = PIL.ImageOps.invert(rgb_image)
        r2,g2,b2 = inverted_image.split()
        final_transparent_image = Image.merge('RGBA', (r2,g2,b2,a))
    else:
        inverted_image = PIL.ImageOps.invert(pil_image)

    im = inverted_image
    immat = im.load()
    (X, Y) = im.size
    m = np.zeros((X, Y))

    m = np.sum(np.asarray(im), -1)
    m[m==0] = 1.0
    m[m==765] = 0.0
    m = m / (np.sum(np.sum(m)) * 1.0)

    #for x in range(X):
    #    for y in range(Y):
    #        m[x, y] = (immat[(x, y)] != (255, 255, 255))
    #m = m / np.sum(np.sum(m))

    dx = np.sum(m, 0) # there is a 0 here instead of the 1
    dy = np.sum(m, 1) # as np.asarray switches the axes, because
                        # in matrices the vertical axis is the main
                        # one, while in images the horizontal one is
                        # the first

    # expected values
    cx = np.sum(dx * np.arange(X))
    cy = np.sum(dy * np.arange(Y))

    return cx, cy

```

איור 6.1.3 קוד פייתון למציאת מרכז החדר השמאלי

- שמירת התמונות במאגר מידע שיכיל תמונות לאחר עיבוד וניקוי מקדים.
- כעת יש לנו מאגר מידע המכיל את התמונות לאחר תהליך של עיבוד מקדים למאגר המידע שפורסם ב-MICCAI'09 עליו נוכל לבחון את תהליך הסגמנטציה שממשנו בפרויקט.

Figure 6.1.4: (a) Original image (b) Expert hand labeled contour (c) ROI image (d) ROI image contour



## 6.2. שלב הסגמנטציה

שלב זה כולל מספר תתי שלבים ומשלב מספר אלגוריתמים כדי ליצור פתרון איכותי לבעיית סגמנטציה חדר שמאל מתוך תמונת MRI לבבית. השיטה המוצעת מתבססת על סגמנטציה באמצעות אלגוריתם **Fuzzy C Means**.

באמצעות סגמנטציה FCM נוכל להפריד את התמונה למחלקות ובכך לאתר את כל האזורים השייכים לחדר שמאל. כדי לשפר את איכות הפתרון של אלגוריתם FCM השתמשנו באלגוריתם MMAS לחישוב ערכי המרכזים ההתחלתיים של FCM. בנוסף כדי לשפר את זמני הריצה של האלגוריתם הצגנו שימוש באלגוריתם **Principle Components Analysis (PCA)**. הרעיון נובע מכך שביצוע סגמנטציה על כל אחד מהתמונות בעזרת אלגוריתם MMAS ולאחריו FCM היא משימה שתעריך זמן רב. לכן השימוש ב-PCA יאפשר לנו לבצע את תהליך הסגמנטציה על כל תמונות החדר במקביל ולא על כל תמונה בנפרד. בחדר לבבי של מטופל יש כ-20 תמונות וכדי ליעל את התהליך נפרק את תמונות החדר לגורמים הראשיים שלהם (principle components) שיחושבו באמצעות אלגוריתם PCA ורק לאותם חלקים נבצע סגמנטציה. לבסוף נוכל לחשב את תמונת הסגמנטציה של כל תמונה על ידי תמונות המרכיבים הראשיים שעברו סגמנטציה ועל פי המשקלים המתאימים של כל תמונה. באמצעות תהליך הורדת המימד התאפשר לנו לבצע סגמנטציה מימד נמוך יותר לכל התמונות בחדר בבת אחת. אולם, נשים לב שאם כל וקטור עצמי והוקטור הממוצע חולקו ל- $c$  מחלקות אזי כאשר נבצע את תהליך השחזור לקבלת תמונה בחדר נקבל תמונה המחולקת לעד (מקסימום)  $(\text{principle\_componets\_number} + 1)^c$  מחלקות. לפיכך עלינו להוסיף שלב בסוף של רידוד כמות המחלקות על פי המרכזים שכבר חושבו.

להלן פירוט שלבי התהליך בקוד:

1. קריאת התמונות בחדר.

```
#####
# step1 : Read slice images
imgs, imgs_outfile = [], []
x_train_vec, x_train = [], []
for (dirpath, dirnames, filenames) in os.walk(slicedir):
    for filename in filenames:
        if (-1 != filename.find("overlay")):
            continue
        imgs.append(os.path.join(dirpath, filename))
        imgs_outfile.append( os.path.join( outputdir, filename ) )
for img_path in imgs:
    img = cv2.imread( img_path, cv2.IMREAD_GRAYSCALE )
    x_train.append(img)
    x_train_vec.append(img.ravel())
h, w = img.shape
x_train_vec = numpy.array( x_train_vec )
imgs_in_slice_cnt = x_train_vec.shape[0]
```

איור 6.2.1 קוד פייתון לביצוע סגמנטציה שלב ראשון - קריאת כל התמונות בחדר



## 2. הורדת מימד לתמונות בחתך בעזרת אלגוריתם PCA.

```
#####  
# step2 : Compute a PCA (eigenvalues) on the slice dataset  
  
n_components = 3  
  
print("Extracting the top %d eigenvectors from %d slices" % (n_components, imgs_in_slice_cnt))  
  
pca = PCA(n_components=n_components).fit(x_train_vec)  
  
eigenvectors = pca.components_.reshape((n_components, h, w))
```

איור 6.2.2 קוד פייתון לביצוע סגמנטציה שלב שני - חישוב פירוק PCA לתמונות בחתך

## 3. הפעלת אלגוריתם FCM לביצוע סגמנטציה לרכיבי הוקטור עצמיים הדומיננטים תוך שימוש באלגוריתם MMAS לחישוב ערכי המרכזים ההתחלתיים של אלגוריתם FCM.

```
#####  
# step3 : Calculate segmentation pictures for eigenvectors and means  
#         using Ant Colony algorithm  
  
tmp_component_file = os.path.join(settings.PROCESS_DB_BASE_DIR, 'comp.png')  
tmp_component_file2 = os.path.join(settings.PROCESS_DB_BASE_DIR, 'comp2.png')  
  
#C = []  
  
pca_components_segmented = numpy.copy(pca.components_)  
  
for i in range(len(pca_components_segmented)):  
    eigenvector = pca.components_[i,:]  
    eigenvector = eigenvector.reshape(h,w)  
    eigenvector = cv2.normalize(eigenvector,alpha=0,beta=255,norm_type=cv2.NORM_MINMAX)  
    eigenvector = eigenvector.astype(numpy.uint8)  
    cv2.imwrite(tmp_component_file, eigenvector)  
  
    cent = get_centroids_using_MMAS( tmp_component_file, K )  
  
    cluster = sifcm.FCM(tmp_component_file, len(cent), cent, tmp_component_file2, 0.05, -1)  
    cluster.form_clusters()
```

איור 6.2.3 קוד פייתון לביצוע סגמנטציה שלב שלישי - סגמנטציה MMAS + FCM של הוקטורים העצמיים שחושבו בפירוק PCA

## 4. הרצת אלגוריתם FCM לביצוע סגמנטציה לוקטור הממוצע תוך שימוש באלגוריתם MMAS לחישוב ערכי המרכזים ההתחלתיים של אלגוריתם FCM.

```
#####  
# step4 : Calculate segmentation pictures for means  
#         using Ant Colony algorithm  
  
cv2.imwrite(tmp_component_file, pca.mean_.reshape(h, w))  
c_mean = get_centroids_using_MMAS( tmp_component_file, K )  
c_mean = numpy.asarray(c_mean)  
cluster = sifcm.FCM(tmp_component_file, len(c_mean), c_mean, tmp_component_file2, 0.05, 10)  
cluster.form_clusters()
```

איור 6.2.4 קוד פייתון לביצוע סגמנטציה שלב רביעי - סגמנטציה MMAS + FCM של הוקטורים הממוצע שחושבו בפירוק PCA



5. הרכבה מחדש של כל תמונה בחתך לכדי תמונה שעברה סגמנטציה.  
 התמונה תחושב כפי שמחשבים את ההרכבה של רכיבים במרחב שנפרס על ידי הוקטורים הדומינאנטים  
 רק עם הוקטורים החדשים שעברו סגמנטציה.

$$\text{PCA\_Segmented\_reconstruction} = \text{PCA\_score}[i] * \text{Segmented\_Eigenvector}[i] + \text{Segmented\_}\mu$$

```
#####
# step5 : for each image calculate segmented image

pca_probability_list = pca.transform( x_train_vec )

img_num = 0

for prob in pca_probability_list:

    im = numpy.zeros(pca_mean_segmented.shape, dtype=numpy.float64)
    i = 0
    for c in prob:
        im = im + ( c * pca_components_segmented[i] )
        i += 1
    im += pca_mean_segmented

    im = (cv2.normalize(im,alpha=0,beta=255,norm_type=cv2.NORM_MINMAX)).flatten()
```

איור 6.2.5 קוד פייתון לביצוע סגמנטציה שלב חמישי - לחשב את תמונת הסגמנטציה לכל תמונה בחתך

6. רידוד לכמות המחלקות על פי המרכזים שחושבו.

7. שמירת התוצאות במאגר מידע.

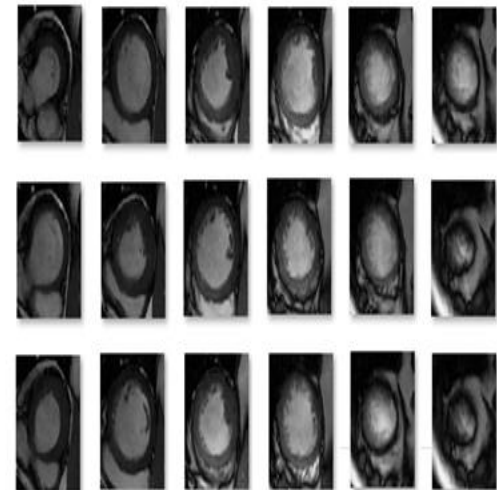
## דוגמא:

באיור 6.2.6 ניתן לראות הורדת מימד של 18 תמונות חתך שצולמו למטופל לשלושה וקטורים עצמיים. בחתך ישנם 18 תמונות של הלב שנלקחו לאותו המטופל בזמנים סמוכים לאורך פעימות הלב ומתארות את צורת הלב בזמנים השונים. ניתן לראות כיצד מפעילים לתמונות בחתך את אלגוריתם PCA ובכך מאפשרים הצגת התמונות בחתך באמצעות מימד נמוך יותר תוך פגיעה מינימאלית באיכות התמונים. את התמונות בחתך במימד נמוך יותר ניתן לראות בצד ימין של המשוואה באיור 6.2.6. המימד הנמוך נובע מכיוון שאת אותם תמונות במרחב הצלחנו לפרוס בעזרת שלושה וקטורים עצמיים של מטריצת השונות המשותפת ומטריצת הממוצע. כל שנדרשנו לעשות היה לקחת את מטריצת המשקלים לכפול אותה בוקטורים העצמיים ולהוסיף את מטריצת הממוצע. באיור 6.2.6 ניתן לראות כיצד מבוצע החישוב.

### PCA SCORES

[ -1209.98332565	-272.72843626	1707.71671798]
[ -343.40532619	-565.41800025	1092.67052876]
[ -806.60721465	-224.06839366	1351.28386157]
[ -339.63237093	-963.19752477	246.69385325]
[ -348.86248594	-383.16628248	1237.43579537]
[ -139.70334801	-1112.15929565	-101.76983743]
[ 434.35071079	-420.41672674	498.75568888]
[ 808.77921195	-1365.49679645	-654.96675946]
[ 1055.13041444	-234.65973182	-530.91536935]
[ 2058.79301341	-1218.6641192	-1215.38914629]
[ 1156.31584693	533.17739741	-293.07161527]
[ 1561.40102379	123.79266002	-581.47768243]
[ 1090.81234473	876.87006064	366.20309608]
[ 970.8429497	918.23716381	205.78060404]
[ -175.59486541	2386.5004392	-352.70674193]
[ -216.07819206	2252.20103927	-109.63186759]
[ -2703.25636731	176.88293553	-1393.59197537]
[ -2853.30201957	-507.68638858	-1473.01915079]

$$\begin{matrix} \text{eigenvector 0} \\ \text{eigenvector 1} \\ \text{eigenvector 2} \end{matrix} \times \begin{matrix} \text{mean vector} \end{matrix} =$$



איור 6.2.6 דוגמא לפירוק PCA לקבוצת תמונות לב שצולמו בבדיקה של מטופל בחתך short axis

באיור 6.2.7 ניתן לראות כיצד חישובנו את תוצאות הסגמנטציה של 18 תמונות החתך.

תוצאות הסגמנטציה עבור 18 תמונות החתך המופיעות בצד ימין של המשוואה באיור 6.2.7.

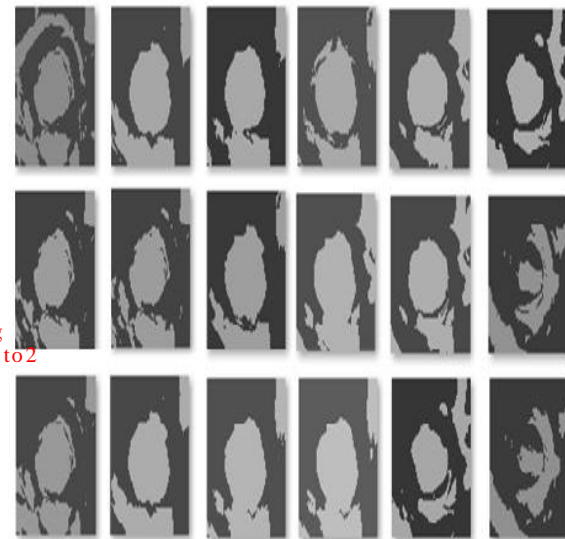
נסביר כיצד התקבלה התוצאה הזאת (חישוב המופיע בצד שמאל של המשוואה):

- בצענו סגמנטציה לתמונות הוקטורים העצמיים והצבנו אותם במשוואה במקום תמונות הוקטורים העצמיים המקוריים. הסגמנטציה התבצעה באמצעות אלגוריתם FCM ו-MMAS.
- בצענו סגמנטציה לתמונת הוקטור הממוצע והצבנו אותו במשוואה במקום תמונת הוקטור הממוצע המקורי. הסגמנטציה התבצעה באמצעות אלגוריתם FCM ו-MMAS.
- נשמור על טבלת המשקלים כפי שהתקבלה במשוואה המקורית של הורדת המימד.
- נבצע את החישוב המתואר כדי לקבל התמונות.
- נבצע צמצום למספר המחלקות שהתקבלו בתמונות אחרי החישוב. הצמצום יתבצע בהתאם למרכזים שכבר חושבו והערכה לפיהם של המרכזים החדשים.

#### PCA SCORES

[ -1209.98332565	-272.72843626	1707.71671798]
[ -343.40532619	-565.41800025	1092.67052876]
[ -806.60721465	-224.06839366	1351.28386157]
[ -339.63237093	-963.19752477	246.69385325]
[ -348.86248594	-383.16628248	1237.43579537]
[ -139.70334801	-1112.15929565	-101.76983743]
[ 434.35071079	-420.41672674	498.75568888]
[ 808.77921195	-1365.49679645	-654.96675946]
[ 1055.13041444	-234.65973182	-530.91536935]
[ 2058.79301341	-1218.6641192	-1215.38914629]
[ 1156.31584693	533.17739741	-293.07161527]
[ 1561.40102379	123.79266002	-581.47768243]
[ 1090.81234473	876.87006064	366.20309608]
[ 970.8429497	918.23716381	205.78060404]
[ -175.59486541	2386.5004392	-352.70674193]
[ -216.07819206	2252.20103927	-109.63186759]
[ -2703.25636731	176.88293553	-1393.59197537]
[ -2853.30201957	-507.68638858	-1473.01915079]

$$\begin{matrix} \text{eigenvector 0 segmented} \\ \text{eigenvector 1 segmented} \\ \text{eigenvector 2 segmented} \end{matrix} \times \begin{matrix} \text{mean vector segmented} \end{matrix} = \begin{matrix} \text{After reducing} \\ \text{cluster centers to 2} \end{matrix}$$

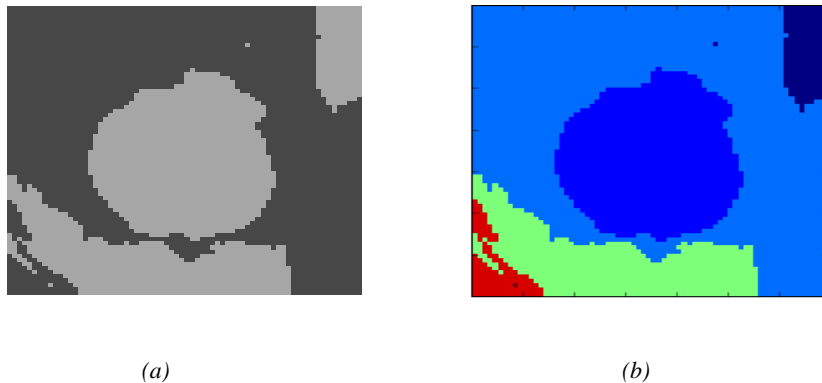


איור 6.2.7 דוגמא של סגמנטציה של חדר שמאל לקבוצת תמונות לב שצולמו בבדיקה של מטופל בחתך short axis ע"י שימוש באלגוריתם Fuzzy C Means תוך שיפור איכות הפתרון בעזרת MMAS ואופטימיזציה בעזרת PCA

### 6.3. דיטקציה של חדר שמאל

לאחר תהליך הסגמנטציה נפנה לשלב בידוד החדר השמאלי מכלל האזור. עלינו לקבוע לאחר תהליך הסגמנטציה מהו האזור המתאר את החדר השמאלי בתמונה ולבודד אותו משאר החלקים. לשם כך השתמשנו באלגוריתם **run-based two pass connected components labeling algorithm adapted to grayscale images**. אלגוריתם זה מוצא רכיבי קשירות בתמונה בגוויי אפור על ידי מעבר כפול (Two pass). באמצעות איתור הרכיבים הקשורים בתמונה נוכל לזהות את רכיב החדר השמאלי בלב ולבודד אותו. באלגוריתם המתואר [17,18] ההפרדה לרכיבי קשירות מתבססת הן על קשירות הרכיבים והן על פרמטר המציין את ערך האפור של הפיקסלים השונים ברכיבי הקשירות. ישנה חשיבות רבה בתמיכה של האלגוריתם בתמונות בגוויי אפור. הסיבה היא שכך אנחנו לא מגבילים את אלגוריתם הסגמנטציה לשתי מחלקות בלבד. לכן אם בעתיד נרצה לבצע סגמנטציה למספר מחלקות גדול משתיים נוכל לעשות זאת. יש לציין שאנחנו רואים בזה חשיבות רבה מכיוון שבמקרים בהם יש מחלות לבביות סגמנטציה למספר מחלקות גדול יותר יוכל אולי לאפשר לנו לזהות את המחלה ואולי להפיק תוצאות טובות יותר בזיהו החדר השמאלי.

לאחר ההפרדה לרכיבי קשירות השונים נוכל לזהות ולסמן רק את הרכיבים השייכים לחדר שמאל. דוגמא להפרדה ניתן לראות באיור 6.3.1.



איור 6.3.1 (a) תמונה חלל אזור חדר שמאל (b) תמונת הרכיבים הקשירים שחושבו בעזרת אלגוריתם *run based two pass connected components labeling algorithm adapted to grayscale images* לתמונה חלל אזור חדר שמאל

בסיום שלב הדיטקציה נקבל את חדר שמאל בלב כפי שזוהה זאת האלגוריתם שלנו מבודד מכל הרעשים והרכיבים השונים בתמונה. את התוצאות לאחר הרצת אלגוריתם הדיטקציה נשמור במאגר המידע.

## 7. פיתוח המערכת

### 7.1. סביבת פיתוח

הפרויקט מומש בעיקרו בשפת התוכנה Python.

השתמשנו בסביבת הפיתוח Anaconda ו-PyCharm המאפשרות יכולת דיבאג והוספת חבילות פיתוח בקלות. פייתון היא שפת תוכנה דינאמית המאפשרת כתיבת סקריפטים שונים. היתרון הגדול הוא ששפה זאת היא שפה מתפתחת ומומשו בה חבילות למגוון תחומים כגון: עיבוד תמונה, ראייה ממוחשבת, כריית מידע, למידה ממוחשבת, וקריאת מידע רפואי. החבילות מגדירות ממשק נוח וממשות אלגוריתמים נפוצים בתחומים אלו.

### 7.2. חבילות פיתוח

לצורך התאמת ויצירת המערכת בפרויקט היה עלינו להשתמש באלגוריתמים ובחבילות השונות על מנת לפתח את רעיון הפרויקט. החבילות השונות בהחלט אפשרו פיתוח ובחינת התוצאות בצורה מהירה יותר. אולם אינן מהוות תחליף למימוש האלגוריתם בו השתמשנו. כדי לממש את רעיון הפרויקט היה עלינו לבצע התאמות והתממשקויות בין האלגוריתמים השונים.

#### חבילות פיתוח בשפת Python שעשייתי בהן שימוש בפרויקט הנוכחי:

##### • חבילות פייתון לתחום הרפואי

- Pydicom – חבילה המאפשרת קריאת תמונות MRI שנשמרו בפורמט DICOM. פורמט DICOM הוא הפורמט המוכר בו נשמרות תמונות MRI. פורמט זה מאגד נתונים שונים כגון הנתונים הרפואיים של המטופל, נתוני הכיול של מכשיר ה-MRI שבאמצעותו בוצעה הסריקה והתמונה עצמה. החבילה מאפשרת לנו לפרסר ולגשת כל אחד מהמרכיבים בצורה נוחה.

##### • חבילות פייתון לתחום עיבוד תמונה

- Numpy – חבילה מתמטית המאפשרת עבודה עם מערכים בצורה יעילה ונוחה. כאמור את התמונות ניתן לשמור כמערכים נומריים ולהפעיל עליהם פעולות שונות.
- OpenCV, PIL – חבילות הממשות אלגוריתמים בתחומי עיבוד תמונה וראייה ממוחשבת. קריאת תמונה עבודה הכולל שינוי גודלה חיתוך וטרנספורמציות שונות ניתן לעשות תוך שימוש בחבילות האלו.

##### • חבילות פייתון לתחום ראייה ממוחשבת

- scikit-learn – מממשת אלגוריתמים מתחומי ראייה ממוחשבת כדוגמת אלגוריתם PCA.

- **חבילות פייתון לתחום כריית מידע ולמידה חישובית**

- Scipy – חבילה המממשת פונקציות מתמטיות שימושיות בינהן אלגוריתמי הקבצה לתחומי כריית המידע.

- **חבילות פייתון לתחום הצגה גרפית**

- Matplotlib – הצגה גרפית של הנתונים.

- **חבילת פייתון לפיתוח ממשק משתמש**

- TKinter – מאפשר יצירת ממשק גרפי בשפת פייתון בצורה נוחה.

### **חבילת פיתוח MMAS -**

נעזרנו במימוש קיים לאלגוריתם ACO + MMAS [30] שנכתב בשפת Java. האלגוריתם הוכן ונבדק עבור סגמנטציה של תמונת MRI של המוח.

### **חבילת פיתוח Run-Based Two Pass Connected Components Labeling algorithm adapted to grayscale images -**

נעזרנו במימוש קיים לביצוע בחינת רכיבי קשירות [31] שנכתב בשפת פייתון. האלגוריתם הוכן ונבדק עבור קשירות של רכיבים בתמונת שונות לאו דווקא בתחום הרפואי.

### **חבילת פיתוח FCM**

נעזרנו במימוש קיים עבור אלגוריתם FCM [33] בשפת פייתון. האלגוריתם הוכן ונבדק עבור סגמנטציה של תמונות MRI של המוח. בנוסף האלגוריתם המתואר בכלל מימש אלגוריתם הקבצה מסוג SIFCM שתואר במאמר [34]. עם זאת רוב החלקים של אלגוריתם FCM מומשו בקוד ולכן לאחר מספר עדכונים דאגנו שהקוד יאפשר גם הרצה של אלגוריתם FCM.

## 7.3. כלי פיתוח

### 7.3.1. כלי פיתוח – שלב העיבוד המקדים

- השתמשו בחבילה Pydicom לקרוא את תמונת ה-MRI שנשמרה בפורמט DICOM והמרתם לפורמט PNG.
- השתמשו בחבילה numpy לקרוא את התוויות שמסמנות את הסגמנטציה הידנית שבצע הרופא למערך. והשתמשו בחבילה opencv ליצור מהמערך את הפוליגון שמתאר את הסגמנטציה.
- המשך עיבוד וזיקוק הנתונים נעשה כבר על תמונות בפורמט PNG.
- חיתוך התמונות לאזורים נעשה תוך שימוש בחבילה PIL.

### 7.3.2. כלי פיתוח – שלב הסגמנטציה

שלב זה כולל מספר תתי שלבים ומשלב מספר אלגוריתמים כדי ליצור פתרון איכותי ויעיל יותר.

- שלב הפירוק –
  - בשלב זה נעזרנו באלגוריתם PCA הממומש בחבילה sklearn בכדי למצוא הצגה ממימד נמוך יותר. בחרנו את הוקטורים עצמיים בעלי הערכים העצמיים הדומיננטים של מטריצת השונות המשותפת והוקטור הממוצע ועליהם מבצעים את הסגמנטציה.
  - שלב הסגמנטציה –
    - מציאת מרכזים לפני הרצת אלגוריתם FCM -
      - כדי לבחור את המרכזים ההתחלתיים בצורה מושכלת הרצנו את אלגוריתם Min Max Ant System algorithm (MMAS). כמו שצינו נעזרנו במימוש האלגוריתם שנכתב ב-Java. היה עלינו להתממשק אל החבילה ולקרוא את הפלט בצורה יעילה.
    - הרצת אלגוריתם FCM
      - נריץ את אלגוריתם FCM אחרי שבודדנו אותו מהחבילה שמימשה את האלגוריתם SIFCM. את המרכזים ההתחלתיים נקבע להיות מרכזים שקבלנו לאחר הרצת אלגוריתם MMAS.
  - שלב ההרכבה –
    - נרכיב לכל תמונה בכתך את תמונת הסגמנטציה עבודה. לשם כך נעזר בחישובים שבצענו בשלב הורדת המימד בעזרת ה-PCA והתוצאה שקבלנו לאחר שלב הסגמנטציה.
    - דיטקציה של החדר השמאלי –
      - לאחר תהליך הסגמנטציה בודדנו את החדר השמאלי מכלל האזור. עשינו זאת באמצעות הרצת אלגוריתם run-based two pass connected components labeling algorithm adapted to grayscale images.

### 7.3.3. כלי פיתוח – שלב ניתוח והצגת התוצאות

שלב עיבוד והצגת התוצאות:

חשבנו את איכות השיטה שהצענו בסימון חדר שמאל והשוונו את התוצאות אל מול שיטות קיימות. הצגת הנתונים גם בצורה גרפית תוך שימוש בחבילות `sklearn`, `matplotlib`. כדי לבחון את התוצאות של אלגוריתמים אחרים נעזרנו במאגר מידע שהתפרסם [38] ומכיל את תמונות הסגמנטציה שיצרו אלגוריתמים שונים לזיהוי חדר השמאלי של הלב. התוצאות של האלגוריתמים האלו נשמרו בקבצי `dat` מקובצים לפיכך

- פתחנו בעזרת חבילת `gzip`.
- פרסרנו את הנתונים בעזרת חבילת `numpy`.
- ושמרנו את התמונות כתמונות PNG בעזרת חבילת `opencv`.



## MICCAI Database - DICOM images

### *Step0 image preprocessing -*

Convert images to PNG format.

Create contour images from expert hand labeled text contour description.



## MICCAI Database - PNG images

### *step1 ROI detection -*

Crop each image to 70x70 region of interest image,

Crop ground truth segmented contour as well.



## ROI Database

### *step2 segmentation -*

For each short axis view that contains about 30 images compute principle components decomposition. Use FCM segmentation with MMAS initial center enhancer to segment the principle components and mean vector. Reconstruct all images in the short axis view to segmented images.



## Segmented Database

### *step3 construt Left Ventricle overlays -*

Isolate Left Ventricle from the segmented image using run-based two pass connected components labeling algorithm.



## Left Ventricle Overlays Prediction Databse

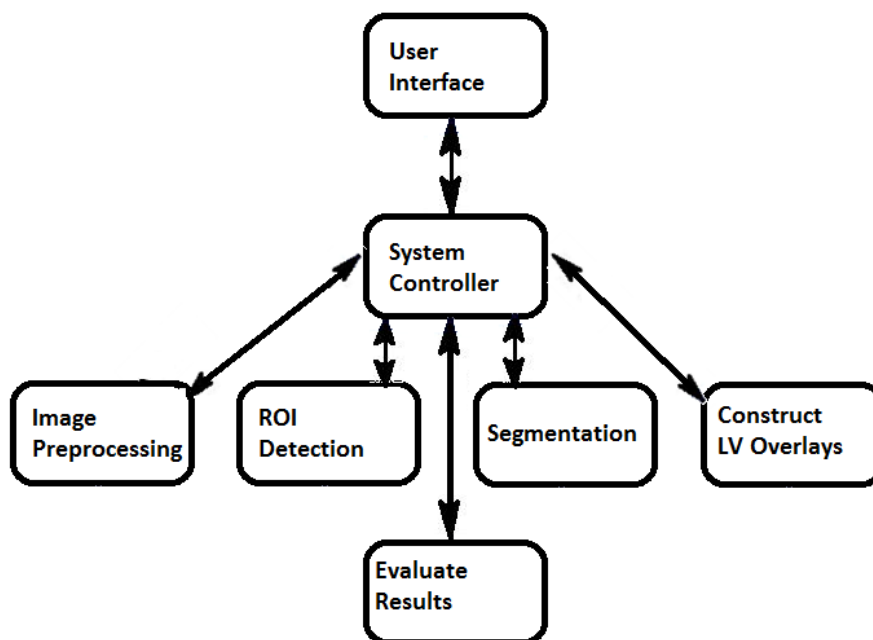
### *step4 evalute results -*

Evalute and compare the performance and accuracy of the proposed approach to other segmentation algorithms. Use the expert hand label contour as the ground truth segmentation.

## 8. מבנה המערכת

המערכת נבנתה בתצורת שליטה מרכזית.

במערכת קיימת תוכנה מרכזית המפעילה את הממשק הגרפי והיא אחראית להפעיל ולנהל את שאר פעילויות המערכת. בקשות מוזנות למערכת באמצעות הממשק הגרפי. המשתמש יכול להזין בקשה לעיבוד תמונות חתך MRI לבבי. לאחר בקשה כזאת המערכת דואגת לקרוא לתהליכים הרלוונטים שיריצו את שלבי האלגוריתם באופן סדרתי. המערכת אחראית לבצע את התהליכים של שמירת נתונים, עיבוד מקדים, סגמנטציה, איתור חדר שמאל, שליפת ובחינת התוצאות. התוצאות ישמרו במערכת בצורה מסודרת ומאורגנת כך שיהיה תיעוד לתוצאות של השלבים השונים. התוצאות יוצגו למשתמש אם תתקבל בקשה להצגה. אופן ההצגה יעשה בממשק אינטרנטי נוח לתפעול ולצפייה.



איור 8.1 תרשים מבנה המערכת

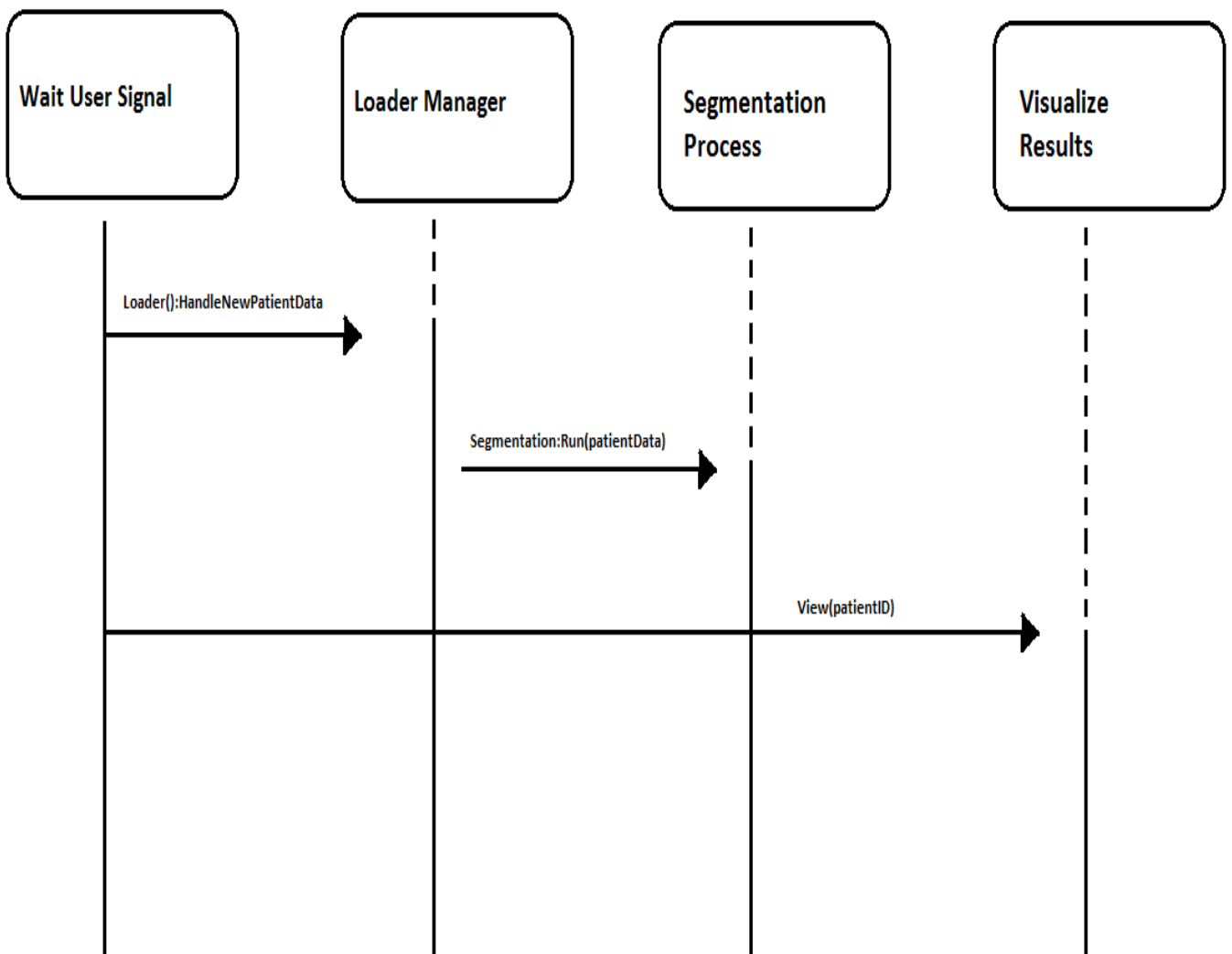
באיור 8.1 במרכז התמונה ניתן לראות את ממשק השליטה המסומן בשם system controller. בנוסף ניתן להבחין כי ממשק זה שולט ומנהל את שאר הפרוצדורות במערכת ולכן כל הממשקים האחרים מתקשרים ומנוהלים דרכו. בין היתר ממשק השליטה אחראי להפעיל את התהליך של הממשק הגרפי User Interface המוצג באיור 8.1 ודרכו המשתמש מעביר בקשות למערכת. כל בקשה מועברת לתהליך השליטה המרכזית שמפענח את הבקשה ודואג ליישם אותה תוך הפעלת הפרוצדורות הרלוונטיות וניטור פעולתיהן.

## 9. ארכיטקטורת המערכת

### 9.1. דיאגרמת רצף כללית

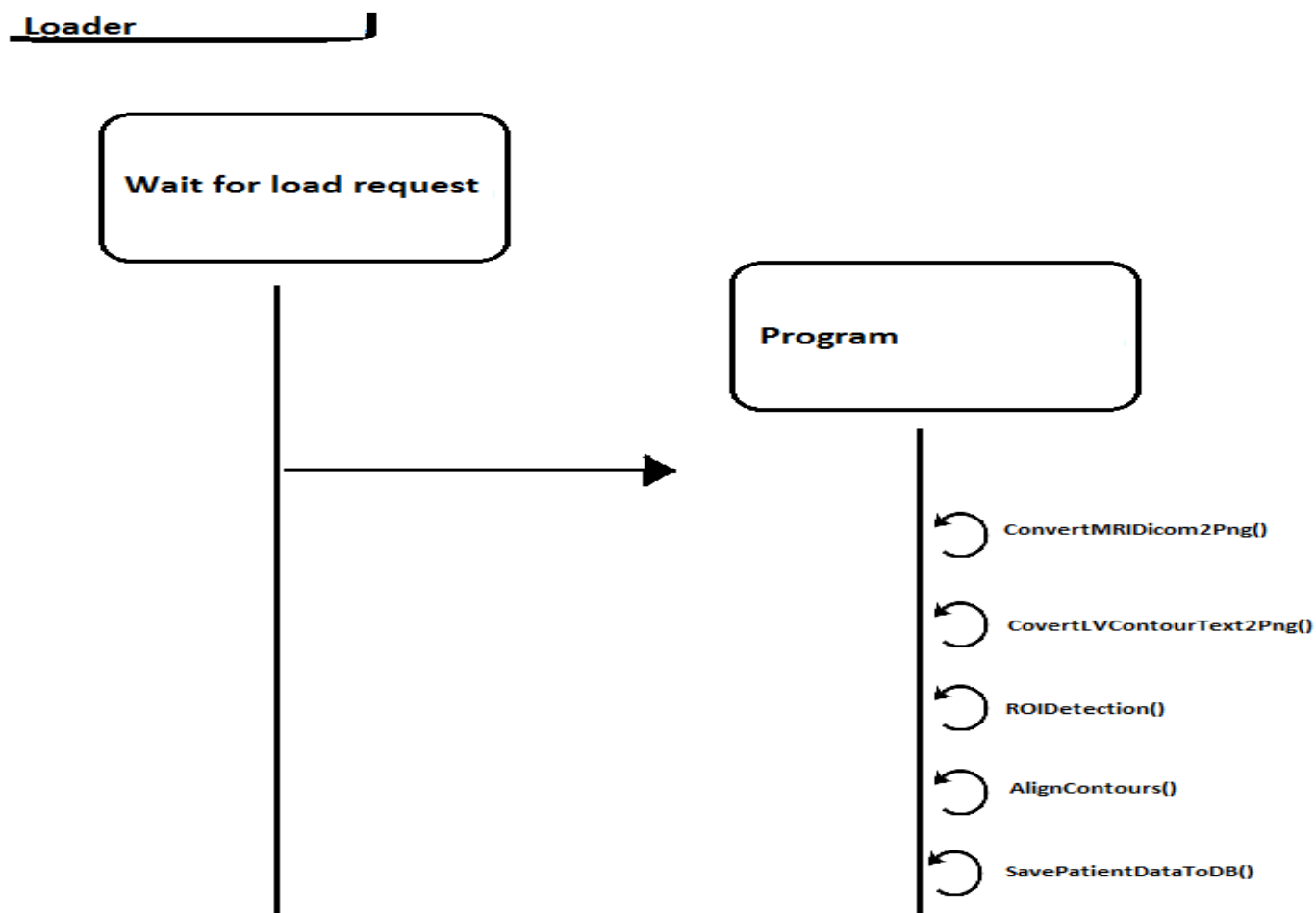
המערכת הינה מערכת אינטרקטיבית המצפה לאירוע מהמשתמש כדי להתחיל לפעול.

#### Interactive UI System



איור 9.1.1 דיאגרמת רצף כללי

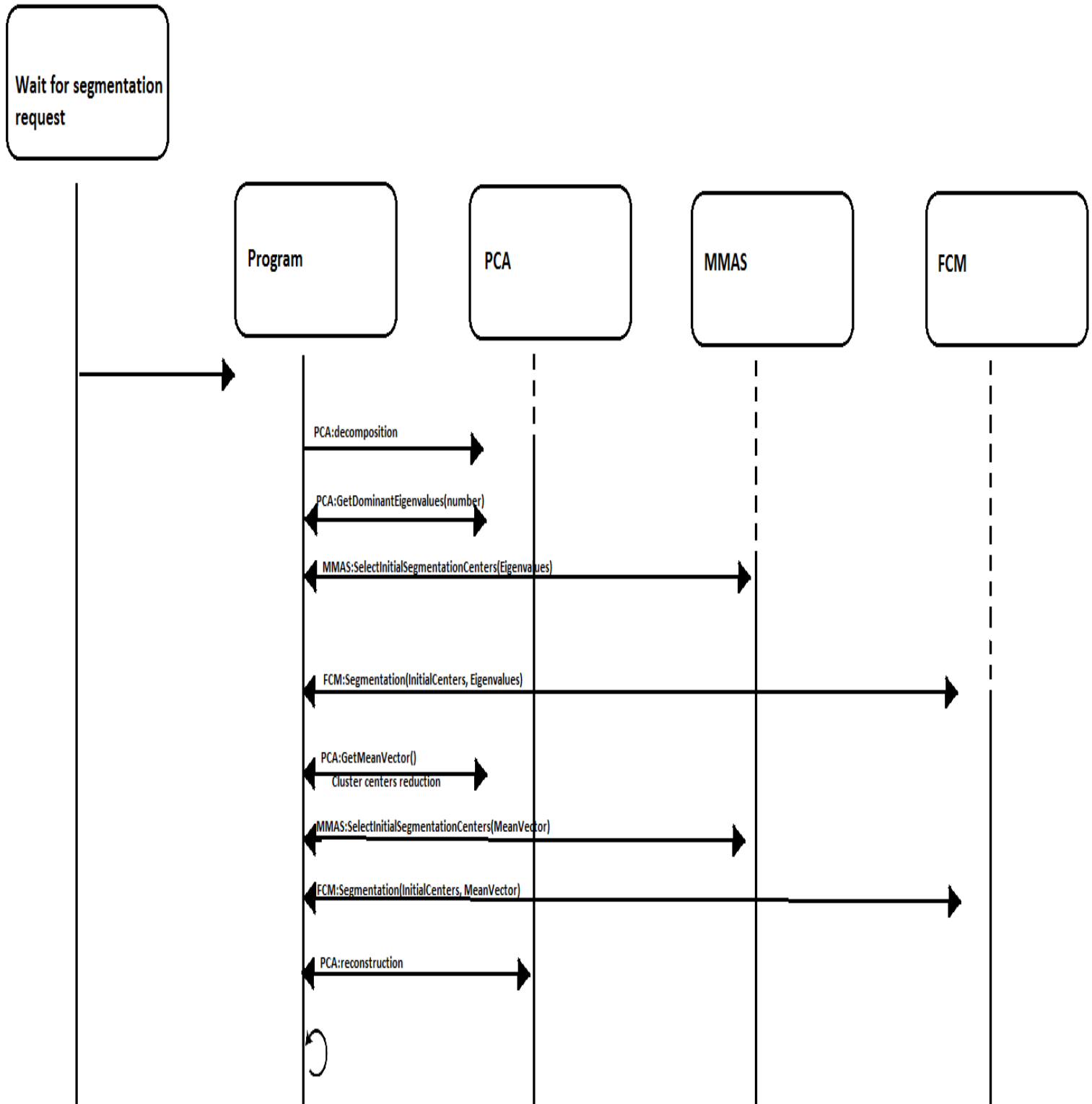
## 9.2. דיאגרמת רצף עבור תהליך טעינה



איור 9.2.1 דיאגרמת רצף עבור תהליך טעינה

### 9.3. דיאגרמת רצף עבור תהליך סגמנטציה

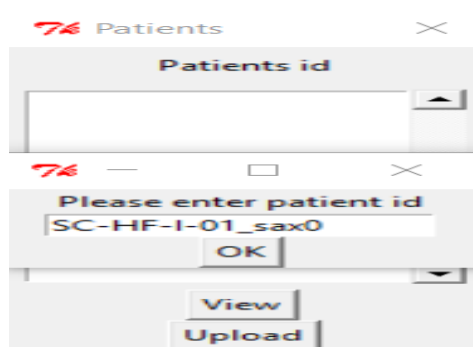
#### Segmentation Process



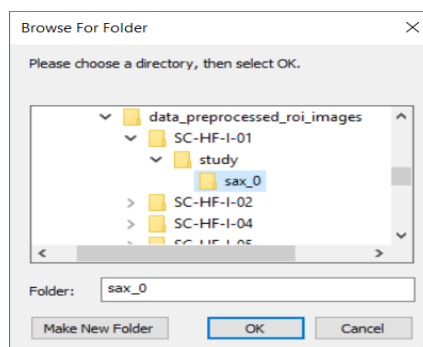
איור 9.3.1 דיאגרמת רצף עבור תהליך סגמנטציה

## 10. ממשק משתמש

יצרנו ממשק משתמש גרפי אינטרקטיבי למשתמש שיאפשר הפעלה של האלגוריתם בפרויקט והצגה חזותית של התוצאות בצורה נוחה וברורה. כדי להתחיל בתהליך הסגמנטציה על המשתמש לטעון תיקייה המכילה את תמונת החתך. בנוסף המשתמש ידרש להקיש מזהה יחודי עבור המטופל ומספר החתך. לאחר הזנת הקלטים המערכת תתחיל בפענוח ותפעיל את תהליך הסגמנטציה על החתך שנטען. המערכת תשמור את התוצאות באופן מסודר במאגר המידע לפי מאפייני הזיהוי שננטענו למערכת. גישה לתוצאות וצפייה בהן תעשה על פי אותם מאפייני זיהוי ייחודיים.



(a)



(b)

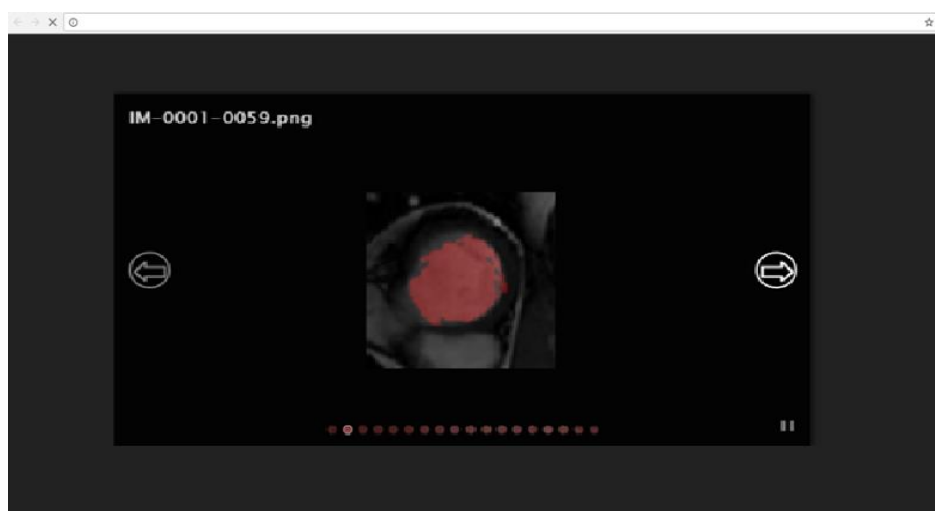
איור 10.1 (a) ממשק משתמש לטעינת נתונים של מטופל חדש במערכת (b)

בכדי לבחון את התוצאות יש ללחוץ על המקש view כפי שמוצג באיור 10.2.



איור 10.2 ממשק משתמש לבחירת מטופל עבורו יוצגו תוצאות הסגמנטציה של החדר השמאלי

לאחר הלחיצה על המקש יפתח ממשק web המציג את כל התמונות בחתך ואת הסגמנטציה עבור כל אחת מהתמונות דוגמא באיור 10.3. ההצגה תופיע כסרט נע דינאמי של תמונות החתך המתקדמות לאורך הזמן ובנוסף ניתן לגלול ולעבור ביניהם.



איור 10.3 ממשק אינטרנטי לצפייה בתוצאות הסגמנטציה של החדר השמאלי

## בדיקות המערכת

### 10.1. בדיקות יחידה

בדיקת היחידה כללה בחינת כל אחד מהאלגוריתמים הבאים, PCA, MMAS, FCM, Run-base two pass, connected components labeling adapted to grayscale images שפיתחנו והשתמשנו בפרויקט. כל אחד מהאלגוריתמים האלו נבחן בנפרד והבדיקה שנעשתה לו היא בדיקת נכונות. האלגוריתמים המתוארים משמשים ככלים בעיבוד התמונה הרפואית של הלב עם זאת הם בנויים כך שניתן לבחון את מהימנות האלגוריתמים על תמונות שונות לו דווקא על תמונות הלב. לפי התוצאות נוכל לבחון את האלגוריתמים השונים ולראות האם האלגוריתמים מבצעים את הנדרש מהם. הקלטים יבחרו כך שיהיה ניתן לבדוק בקלות את ביצועי האלגוריתמים.

#### 10.1.1. בדיקת מימוש אלגוריתם PCA

מטרת הבדיקה הייתה לבחון את יכולת ההטלה והשיחזור של אלגוריתם PCA בו אנחנו משתמשים. לשם כך השתמשנו במאגר המידע <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html> מאגר מידע זה מכיל 16128 תמונות של 28 נבדקים.

כל תמונה במאגר המידע היא בגודל של 640 רוחב על 480 אורך שמורה בפורמט PGM8 bit grayscale. התמונות צולמו מ-9 זוויות שונות ועם 64 תנאי תאורה שונים ביניהם



איור 11.1.1.2 תנאי תאורה בהם מצולם נבדק yaleB11

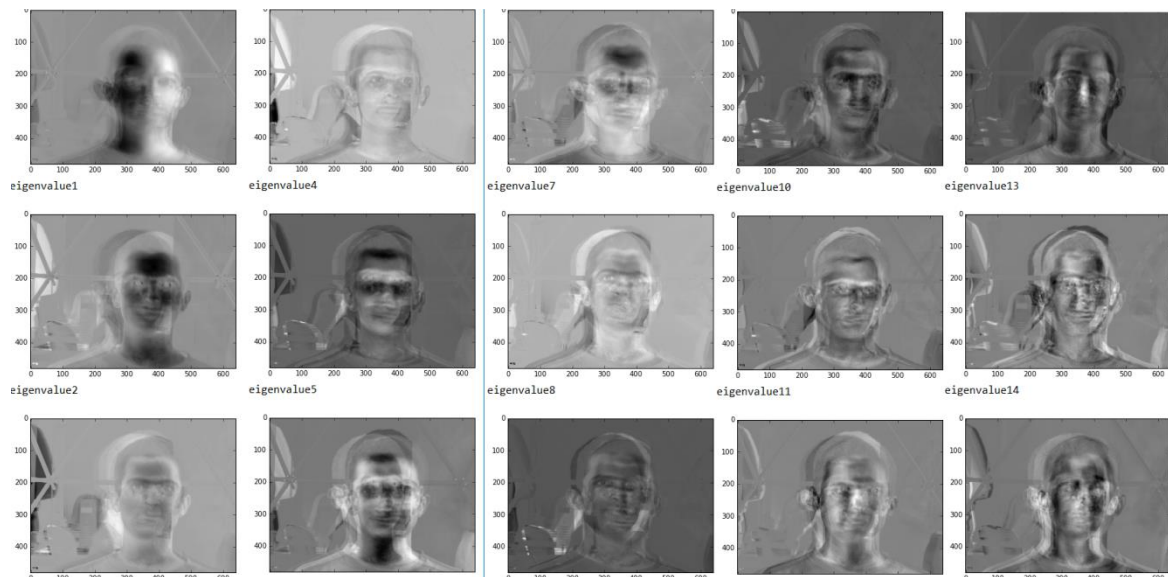
איור 11.1.1.1 הזוויות השונות בהם מצולם נבדק yaleB11

בחרנו לבדוק את ההטלה והשיחזור של האלגוריתם PCA על התמונות שנלקחו בזוויות שונות ותנאי תאורה שונים של נבדק 1 השמור בתיקייה yaleB11.

הרצנו את אלגוריתם PCA כקלט העברנו את כל התמונות של הנבדק (סה"כ 576 תמונות).

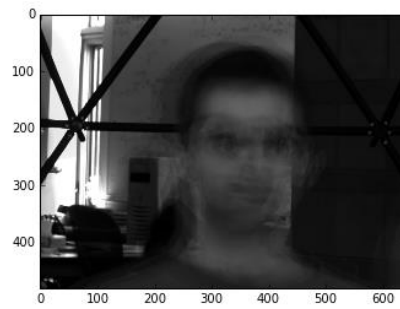
האלגוריתם התבקש לבצע הורדת מימד של מאגר התמונות ל-15 וקטורים עצמיים בעלי הערכים העצמיים הגבוהים ביותר. לאחר מכן בצענו שחזור של כל התמונות במאגר בהסתמך על הוקטורים העצמיים טבלת הפירוק ומטריצת השונות המשותפת.

### תמונות 15 הוקטורים העצמיים עם הערכים העצמיים הגבוהים ביותר



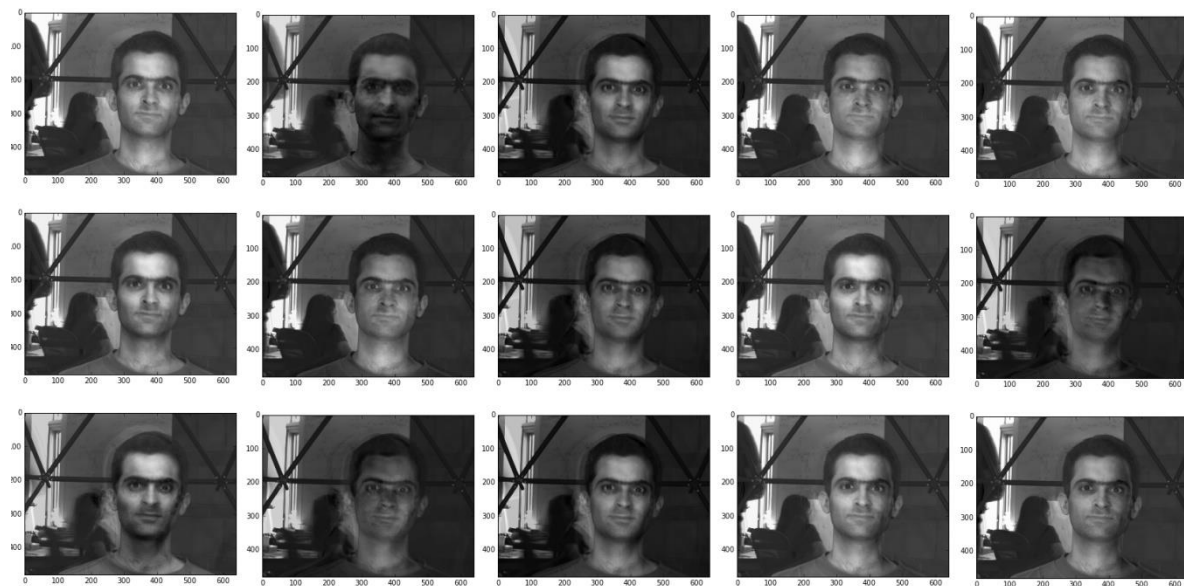
איור 11.1.1.3 תמונות של חמשה עשר וקטורים עצמיים בעלי הערכים העצמיים הגבוהים ביותר שהתקבלו בפירוק PCA

### תמונת מטריצת השונות המשותפת



איור 11.1.1.4 מטריצת השונות המשותפת שהתקבלה בפירוק PCA

### דוגמא של 15 התמונות הראשונות במאגר משוחזרות (הבחירה שרירותית יכלנו לבחור לשחזר כל תמונה במאגר):



איור 11.1.1.5 חמש עשרה התמונות הראשונות במאגר משוחזרות



```

# -*- coding: utf-8 -*-
"""
@author: ChenA
"""

import sys, os, shutil, subprocess, time, re
import cv2, numpy, glob
import matplotlib.pyplot as plt
from scipy.cluster.vq import *
from sklearn.decomposition import PCA

def read_pgm(filepath, byteorder='>'):
    """
    Return image data from a raw PGM file as numpy array.

    Format specification: http://netpbm.sourceforge.net/doc/pgm.html
    """
    with open(filepath, 'rb') as f:
        buffer = f.read()
        f.close()
    try:
        header, width, height, maxval = re.search(
            b"(\^P5\s(?:\s*#\s*[\r\n])*\s*"
            b"(\d+)\s(?:\s*#\s*[\r\n])*\s*"
            b"(\d+)\s(?:\s*#\s*[\r\n])*\s*"
            b"(\d+)\s(?:\s*#\s*[\r\n]\s*)")", buffer).groups()
    except AttributeError:
        raise ValueError("Not a raw PGM file: '%s'" % filepath)
    return numpy.frombuffer(buffer,
                           dtype='u1' if int(maxval) < 256 else byteorder+'u2',
                           count=int(width)*int(height),
                           offset=len(header)
                           ).reshape((int(height), int(width)))

def is_ambient_image(filename):
    if (-1 == filename.find("Ambient")):
        return False
    return True

if __name__ == "__main__":
    slicedir = "D:\\ChenA\\faceDB\\ExtendedYaleB"

    # =====
    # step1 : Read slice images
    imgs = []
    x_train_vec, x_train = [], []

    # for (dirpath, dirnames, filenames) in os.walk():
    #     for filename in filenames:
    #         imgs.append(os.path.join(dirpath, filename))

    for root, dirs, files in os.walk(slicedir, topdown=False):
        for name in files:
            if ((-1 == root.find("yaleB11"))):
                # and (-1 == root.find("yaleB12"))):
                continue
            if (name.rfind(".") == ".pgm"):
                if (is_ambient_image(name)):
                    continue
                else:
                    imgs.append(os.path.join(root, name))

    for img_path in imgs:
        img = read_pgm(img_path, byteorder='<')
        x_train.append(img)
        x_train_vec.append(img.ravel())

    h, w = img.shape
    x_train_vec = numpy.array(x_train_vec)
    imgs_in_slice_cnt = x_train_vec.shape[0]
    # =====
    # step2 : Compute a PCA (eigenvalues) on the slice dataset
    n_components = 15
    print("Extracting the top %d eigenvectors from %d slices" % (n_components, 1
    pca = PCA(n_components=n_components).fit(x_train_vec)
    eigenvectors = pca.components_.reshape((n_components, h, w))
    pca_components_s = numpy.copy(pca.components_)

    for i in range(len(pca_components_s)):
        eigenvector = pca.components_[i, :]
        eigenvector = eigenvector.reshape(h, w)
        eigenvector = cv2.normalize(eigenvector, alpha=0, beta=255, norm_type=cv2.N
        eigenvector = eigenvector.astype(numpy.uint8)
        #cv2.imshow(eigenvector)
        print "eigenvalue" + str(i)
        plt.imshow(eigenvector, cmap='gray')
        plt.show()

    pca_mean_s = pca.mean_ #img.ravel()

    nimg = numpy.asarray(pca_mean_s, dtype=numpy.float64)
    plt.imshow((nimg.reshape(h, w)), cmap='gray')
    plt.show()

    pca_probability_list = pca.transform(x_train_vec)

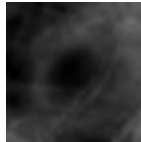
    img_num = 0
    for prob in pca_probability_list:
        im = numpy.zeros(pca_mean_s.shape, dtype=numpy.float64)
        i = 0
        for c in prob:
            im = im + (c * pca_components_s[i])
            i += 1
        im += pca_mean_s

        im = (cv2.normalize(im, alpha=0, beta=255, norm_type=cv2.NORM_MINMAX)).flat
        nimg = numpy.asarray(im, dtype=numpy.float64)
        #cv2.imshow(nimg.reshape(h, w))
        plt.imshow((nimg.reshape(h, w)), cmap='gray')
        plt.show()

```

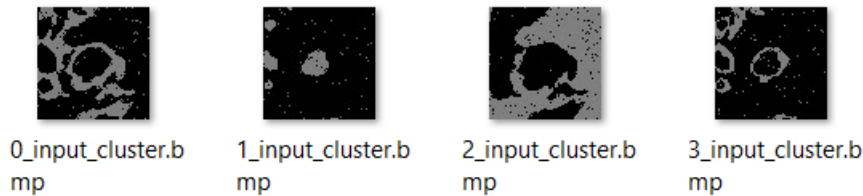
## 10.1.2. בדיקת מימוש אלגוריתם MMAS

מטרת הבדיקה הייתה לבחון את ביצועיו של אלגוריתם MMAS לבחירת מרכזים. כדי לעשות זאת הרצנו את האלגוריתם MMAS ועל פי המרכזים שקבלנו בפלט חלקנו את הפיקסלים בתמונה למחלקות. הרצנו את האלגוריתם על תמונה אזור חדר שמאל ממאגר המידע בה החלוקה למחלקות הינה קשה וזאת בשל המחלה בה סובל המטופל הגורמת לכך שהניגודים בין המחלקות השונות ממושטש. התמונה ממאגר המידע:



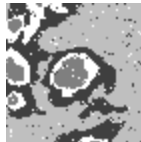
איור 11.1.2.1 אזור חדר שמאל של אחד המטופלים ממאגר המידע

הרצת אלגוריתם MMAS ויצירת 4 מרכזים שלפיהם נבצע הפרדה ל-4 מחלקות:



איור 11.1.2.2 המחלקות שהתקבלו על ידי המרכזים שחושבו באלגוריתם MMAS

תוצאה:

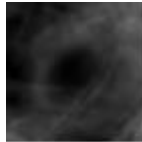


איור 11.1.2.3 המחלקות שהתקבלו על ידי המרכזים באלגוריתם MMAS מאוחדים

תוצאות ההרצה של האלגוריתם מראות שנבחרו מרכזים המחלקים את התמונה למחלקות בצורה טובה. מהמחלקות השונות ניתן להבחין בפרטים שונים המופיעים בתמונה ואותם פרטים מופרדים בצורה טובה. בנוסף ניתן לראות כי מהתמונה של המחלקות השונות ניתן לבודד את חדר שמאל.

### 10.1.3. בדיקת מימוש אלגוריתם FCM

מטרת הבדיקה הייתה לבחון את ביצועיו של אלגוריתם FCM לביצוע סגמנטציה. הרצנו את האלגוריתם על תמונה אזור חדר שמאל ממאגר המידע בה החלוקה למחלקות הינה קשה וזאת בשל המחלה בה סובל המטופל הגורמת לכך שהניגודים בין המחלקות השונות ממושטש. מספר המחלקות שחלקנו את התמונה היה 4 והמרכזים ההתחלתיים שניתנו היו  $[0,0,0,0]$ . התמונה ממאגר המידע:



איור 11.1.3.1 תמונה אזור חדר שמאל של אחד המטופלים ממאגר המידע

תוצאה:



איור 11.1.3.2 תוצאות סגמנטציה שהתקבלה על ידי שימוש באלגוריתם FCM

המרכזים שנקבעו על ידי האלגוריתם קבע הם:

```
In [5]: cluster.get_centroids()
Out[5]:
array([[ 40.76559333],
       [ 58.64732874],
       [ 16.7226048 ],
       [ 83.01176868]])
```

איור 11.1.3.3 תוצאות המרכזים שהתקבלו בסגמנטציה באלגוריתם FCM

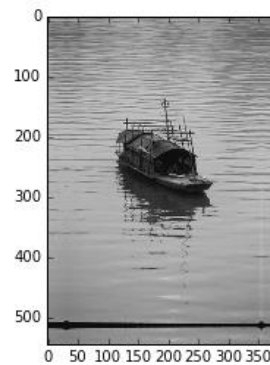
מתוצאות ההרצה קבלנו שהאלגוריתם מבצע סגמנטציה לתמונה למחלקות בצורה טובה מאוד. מהמחלקות השונות ניתן להבחין בפרטים שונים המופיעים בתמונה ואותם פרטים מופרדים בצורה טובה. בנוסף ניתן לראות כי מהתמונה של המחלקות השונות ניתן לבודד את חדר שמאל. לפיכך האלגוריתם הצליח למצוא מרכזים כך שיאפשרו הפרדה בין האובייקטים השונים בתמונה בצורה טובה.

## 10.1.4. בדיקת מימוש אלגוריתם Run-based two pass connected components labeling adapted to grayscale images

מטרת הבדיקה לבחון את איכות האלגוריתם במציאת רכיבים קשירים בתמונה. לצורך כך לקחנו תמונה של סירה במים והמרנו אותה לגווני אפור:

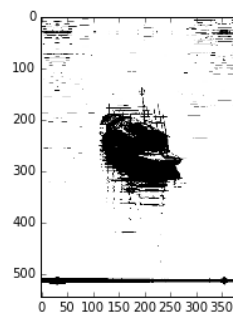


איור 11.1.4.1 תמונה צבעונית סירה במים



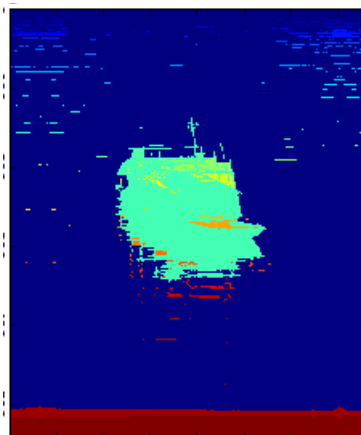
איור 11.1.4.2 תמונת סירה במים בגווני אפור

לאחר מכן הפרדנו את הפיקסלים בתמונה ל-2 מחלקות בעזרת אלגוריתם FCM



איור 11.1.4.3 תוצאות סגמנטציה שהתקבלה על ידי שימוש באלגוריתם FCM לתמונה סירה במים

ולבסוף הרצנו את האלגוריתם למציאת רכיבי קשירות והצגנו את התוצאה כאשר כל רכיב סומן בצבע שונה



איור 11.1.4.4 רכיבי קשירות שהתקבלו עבור תמונה סירה במים  
בעזרת אלגוריתם run based two pass connected components labeling adapted to grayscale images

תוצאות ההרצה מראות כי האלגוריתם מבצע מציאת רכיבי קשירות לתמונה בצורה טובה מאוד. האלגוריתם הצליח להפריד בין רכיבי הקשירות השונים בתמונה והצליח להפריד את הספינה מהמים בצורה מדויקת.

## 10.2. בדיקות אינטגרציה

בבדיקות האינטגרציה המטרה היא לבדוק את התנהגות המערכת כולה. נבנה בסיס נתונים מתוך הקלט ונריץ את תהליך הסגמנטציה ונבחן את איכות התוצאות.

קלט: צור מאגר המכיל את תמונות ה-MRI של נבדקים שאובחנו כבעלי מצב קרדיאלי תקין.

תוצאות צפויות לאחר ביצוע הבדיקה:

נבחן את תוצאות הסגמנטציה שבוצע בשיטה שפותחה בפרויקט זה על כל אחד מהמטופלים והערך את איכות התוצאות על פי המדדים ACC, PSNR, BDP. תיאור יותר מפורט על אופן חישוב המדדים יופיע בהמשך.

תוצאות:

PatientID	Gender	Age	PSNR	ACC(%)	BDP(%)
SC-N-2	Male	63	4.64	64.03	95.70
SC-N-3	Female	53	14.11	95.45	73.63
SC-N-5	Female	77	11.80	87.60	80.61
SC-N-6	Male	70	13.28	94.91	72.90
SC-N-7	Male	23	5.89	70.93	78.54
SC-N-9	Male	51	10.54	88.89	74.11
SC-N-10	Male	61	13.08	94.52	65.92
SC-N-11	Male	60	6.49	76.56	90.19
SC-N-40	Female	88	12.19	93.34	63.02

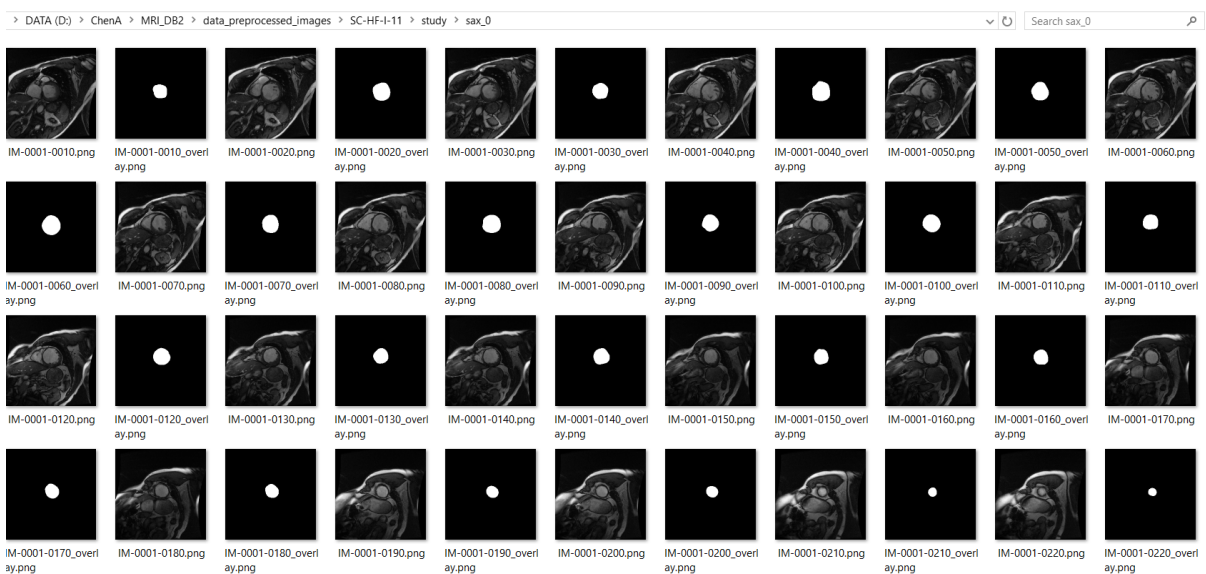
ניתן לראות מהתוצאות כי הצלחנו לשלב את המרכיבים השונים לכדי מערכת שלמה ומתפקדת. מבדיקות האינטגרציה עולה כי המערכת פועלת על תמונות שונות ממאגר המידע ובנוסף מפיקה תוצאות איכותיות.

## 10.3. מקרה בוחן

נבחנו את תוצאות הרצת האלגוריתמים השונים על חתך של מטופל לדוגמא. התוצאות שנציג הם של מטופל SC-HF-I-11 המופיע במאגר מידע. נריץ את התהליכים השונים ונבדוק את איכות התוצאות.

### 10.3.1. ביצוע תהליך עיבוד מקדים לתמונה

בשלב הראשון של העיבוד המקדים מתבצע המרת ההנתונים. את התמונות הלבביות שנשמרו בפורמט DICOM ממירים לפורמט PNG. ומקבצי הטקסט המכילים את המתאר שסמנו הרדיולוגים לחדר שמאל נבנה תמונות המתארות את הסגמנטציה.



איור 11.3.1.1 תמונות הלב ומתאר חדר שמאל שסומן על ידי הרדיולוג עבור המטופל שמורות בפורמט PNG

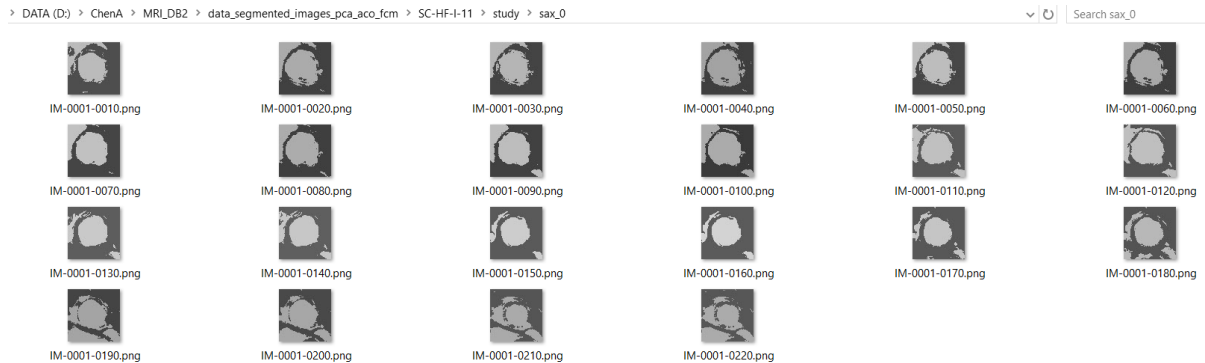
בשלב השני של העיבוד המקדים מתבצע תהליך זיקוק של אזור חדר שמאל מפרטים אחרים שמופיעים בתמונה.



איור 11.3.1.2 תמונות הלב ומתאר חדר שמאל אחרי חיתוך להצגה של אזור חדר שמאל

## 10.3.2. ביצוע תהליך סגמנטציה

נבחן את תוצאות תהליך הסגמנטציה.



איור 11.3.2.1 תמונות הלב ומתאר חדר שמאל אחרי סגמנטציה

## 10.3.3. ביצוע תהליך דיטקציה של חדר שמאל

נבחן את תהליך הדיטקציה שמטרתו לזהות ולבודד את חדר שמאל.



איור 11.3.3.1 תמונות הלב ומתאר חדר שמאל אחרי דיטקציה

## 10.3.4. איכות זיהוי חדר שמאל

נבחן את איכות תוצאות הסגמנטציה וזיהוי חדר שמאל בעזרת המדדים ACC, PSNR, BDP.

תיאור יותר מפורט על אופן חישוב המדדים יופיע בהמשך.

PatientID	Gender	Age	PSNR	ACC(%)	BDP(%)
SC-HF-I-11	Male	55	11.27	91.56	83.40

## 11. תוצאות

### 11.1. מדדים לאיכות הסגמנטציה

בשלב האחרון נרצה לכמת את טיב התוצאות בכדי לבדוק את איכות הסגמנטציה של האלגוריתם שלנו. בעבודה זה השתמשנו בשלושה מדדי איכות לסגמנטציה המחושבים באופן הבא:

$$(Eq.12.1.1) \quad BDP = \frac{TP}{TP+FN}$$

$$(Eq.12.1.2) \quad ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$(Eq.12.1.3) \quad MSE = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I_1(i,j) - I_2(i,j)\|^2}{m * n} \quad (Eq.12.1.4) \quad PSNR = 20 * \log_{10} \frac{1.0}{\sqrt{MSE}}$$

TP: True Positive, number of correct matches.

FN: False Negative, matches that were not correctly detected.

FP: False Positive, proposed matches that are incorrect.

TN: True Negative, non-matches that were correctly rejected.

BDP: Building Detection Percentage, fraction of the correctly classified pixels.

ACC: Accuracy, the percentage of true clustered elements from all elements.

MSE: Mean Squared Error, the average of the squared errors between ground truth to segmented image

PSNR: Peak Signal to Noise Ratio, ratio between the maximum possible power of a signal and the power of distorting noise that affects the quality of its representation.

Execution Time: the time period required to reach a result.

ככל שאיכות הסגמנטציה טובה יותר כך ערכי מדדי האיכות יהיו גבוהים יותר. לפיכך נשאף לערכים הגבוהים במדד האיכות BDP, ACC, PSNR ולזמן ריצה נמוך.



## 11.2. פסאודוקוד לחישוב מדד איכות הסגמנטציה

האלגוריתמים השונים יפיקו כפלט תמונה המתארת את חדר שמאל שאיתרו ובודדו מתמונת ה-CMRI. את התמונה נשווה מול תמונת הסגמנטציה הידנית שבוצעה על ידי רופאים מומחים בתחום הרדיולוגיה. ככל שהאלגוריתם ישיג תוצאה קרובה לסגמנטציה הידנית כך נסיק שהסגמנטציה שהושגה בפרויקט יותר איכותית. ההשוואה תבוצע באופן הבא:

- לכל תמונה בחתך בצע
  - הפעל את האלגוריתם שלנו למציאת חדר שמאל – תמונה א.
  - יבא את הסגמנטציה שבוצעה על ידי הרדיולוג – תמונה ב.
  - השווה בין שתי התמונות - התמונות ממופות כך שבכל תמונה פיקסל השייך לחדר שמאל מסומן בלבן אחרת מסומן בשחור.
  - חשב מדדי איכות לתיאמות הפיקסלים השונים בין התמונות.
- עבור כל פיקסל (I,J) בתמונה א
  - אם הפיקסל מסומן בלבן (זוהה על ידי האלגוריתם שלנו כשייך לחדר שמאל)
    - אם הפיקסל התואם (I,J) בתמונה ב גם מסומן בלבן (זוהה על ידי הרדיולוג כשייך לחדר שמאל) אזי TP.
    - אם הפיקסל התואם (I,J) בתמונה ב מסומן בשחור (זוהה על ידי הרדיולוג כלא שייך לחדר שמאל) אזי FP.
  - אם הפיקסל מסומן בשחור (זוהה על ידי האלגוריתם שלנו כלא שייך לחדר שמאל)
    - אם הפיקסל התואם (I,J) בתמונה ב גם מסומן בשחור (זוהה על ידי הרדיולוג כלא שייך לחדר שמאל) אזי TN.
    - אם הפיקסל התואם (I,J) בתמונה ב מסומן בלבן (זוהה על ידי הרדיולוג כשייך לחדר שמאל) אזי FN.
- חשב את מיצוע מדדי איכות הסגמנטציה עבור כל התמונות בחתך.
- חשב את מיצוע מדדי איכות הסגמנטציה עבור כל המטופלים.

```

def compare_segmented_images(ground_truth_img_dir, segmented_img_dir, show=False):

    ground_truth_img = cv2.imread( ground_truth_img_dir, cv2.IMREAD_GRAYSCALE )

    segmented_img = cv2.imread( segmented_img_dir, cv2.IMREAD_GRAYSCALE )

    #segmented_img = cv2.absdiff(255, segmented_img)

    ground_truth_img_binary = ground_truth_img / 255
    segmented_img_binary = segmented_img / 255

    ground_truth_vec = (numpy.asarray(ground_truth_img_binary, dtype=numpy.uint8)).flatten()
    segmented_vec = (numpy.asarray(segmented_img_binary, dtype=numpy.uint8)).flatten()

    P = len(ground_truth_img_binary[ground_truth_img_binary == 1])
    N = len(ground_truth_img_binary[ground_truth_img_binary == 0])
    TP = numpy.sum( ground_truth_vec & segmented_vec )
    TN = numpy.sum( binary_narray_not(ground_truth_vec) & binary_narray_not(segmented_vec) )
    FP = numpy.sum( (ground_truth_vec ^ segmented_vec) & segmented_vec )
    FN = numpy.sum( (ground_truth_vec ^ segmented_vec) & binary_narray_not(segmented_vec) )

    pixels_cnt = float(ground_truth_img_binary.shape[0] * ground_truth_img_binary.shape[1])
    MSE = (1.0 * numpy.sum((ground_truth_img_binary.astype("float") - segmented_img_binary.astype("float")) ** 2) ) / pixels_cnt

    if(show):
        show_overlay(ground_truth_img)
        show_overlay(segmented_img)
        roc_curve_2(ground_truth_vec, segmented_vec)

    return (P, N, TP, TN, FP, FN, MSE)

d = [('TPR', numpy.float), ('FPR', numpy.float), ('ACC', numpy.float), ('PPV', numpy.float),\
      ('BDF', numpy.float), ('BF', numpy.float), ('PSNR', numpy.float)]

results = numpy.asarray([], dtype=d)

pids = glob.glob(base_prediction_dir + "\\*")

for pid_dir in pids:

    #results = numpy.asarray([], dtype=d)

    #print "*****" + str(pid_dir) + "*****"

    for (dirpath, dirnames, filenames) in os.walk(pid_dir):

        for filename in filenames:

            img_dir = os.path.join(dirpath, filename)

            overlay_img_dir = os.path.join( settings.BASE_PREPROCESSED_ROI_IMAGES_DIR,\
                                             dirpath[len(prefix)-1:],\
                                             filename[:filename.rfind(".")] + "_overlay" + filename[filename.rfind("."):])

            P, N, TP, TN, FP, FN, MSE = compare_segmented_images(overlay_img_dir, img_dir, False)

            item_result = numpy.asarray([ (( 1.0 * TP / P ),\
                                           ( 1.0 * FP / N ),\
                                           ((1.0 * TP + TN) / (TP + TN + FP + FN)),\
                                           ( 1.0 * TP / (TP + FP) ),\
                                           ( 1.0 * TP / (TP + FN) ),\
                                           ( 1.0 * FP / (TP + 1) ),\
                                           ( 20.0 * math.log((1.0/math.sqrt(MSE)) ,10) ) ), dtype=d)

            results = numpy.append(results, item_result)

    print numpy.mean(results['ACC'])
    print numpy.mean(results['BDF'])
    print numpy.mean(results['PSNR'])

```

### 11.3. בחינת מדדי איכות הסגמנטציה של האלגוריתמים השונים

- בסיס הנתונים שהשתמשו בפרויקט זה הוא בסיס הנתונים שפורסם באתגר MICCAI'09.
  - האלגוריתמים עליהם בחנו והשוונו את איכות התוצאות היו:
    - האלגוריתם שתארנו בפרויקט עם הפרמטרים הבאים:
      - Number of clusters to FCM and MMAS = 4; Number of ants = 5;
      - Number of iteration = 3; Cost function values A=5000 and B=10;
      - Update function weight = 0.6; Reduced Cluster Number = 2;
    - אלגוריתם סגמנטציה k-means פשוט כאשר k=2.
    - אלגוריתמי סגמנטציה ייחודיים שהוצעו במאמרים [20-24] ומיועדים לחדר השמאלי.
- כל אחד מהאלגוריתמים משתמש בטכניקת סגמנטציה שונה ואוטומיזציה אחרת חלקם אוטומטים באופן מוחלט וחלקם סמי-אוטומטים כלומר דורשים התערבות כלשהי של המשתמש.
- התמונות המתארות את הסגמנטציה של האלגוריתמים על בסיס הנתונים MICCAI'09 נשמרו ופתוחים לטובת הציבור [25,26]. הסימונים<sup>1</sup> M4-M8 כפי שהוצגו במאמר [26] לכל אחד מהאלגוריתמים הם בהתאמה השיטות [20-24].
- לכל אחד מהאלגוריתם חישבנו את מדדי האיכות.
  - נציג את התוצאות השונות ונבצע השוואה ביניהם.

בחינת מדדי האיכות של התוצאות עבור האלגוריתמים השונים:

-	P.Time (min)	PSNR (dB)	ACC (%)	BDP (%)
proposed approach	50	10.43	87.84	73.00
K-Means	18	11.13	90.35	64.57
M4	-	12.17	87.47	56.65
M5	-	9.34	85.02	53.19
M6	-	9.53	85.14	56.29
M7	-	10.83	86.62	60.43
M8	-	11.96	86.85	47.70

<sup>1</sup> M4 - Semiautomated cardiac segmentation on cine magnetic resonance images using GVF-Snake deformable models.

M5 - A dynamic elastic model for segmentation and tracking of the heart in MR image sequences.

M6 - Segmentation of 4D cardiac MRI: Automated method based on spatio-temporal watershed cuts.

M7 - LV ejection fraction calculation from automatically selected and processed diastolic and systolic frames in short axis cine-MRI.

M8 - Fully automated segmentation of the left ventricle applied to cine mr images: description and results on a database of 45 subjects.

## 11.4. ניתוח התוצאות

בפרויקט פיתחנו אלגוריתם חדש לביצוע סגמנטציה של חדר שמאל של הלב מתנות MRI לבבי. בוצע ניתוח השוואתי של האלגוריתם שהצענו אל מול אלגוריתמים מוקדים שהתפרסמו ומציעים פתרון לאותה בעיה. הבחינה נעשתה מול מטופלים שונים עם מגוון שונה של מקרי לב פתולוגיים וזאת כדי לבחון את עמידות האלגוריתמים השונים מול התרחישים השונים. מניתוח התוצאות ניתן לראות כי האלגוריתם שהצענו בפרויקט מספק תוצאות טובות איכותיות וכמותיות בהשוואה אל מול אלגוריתמים שפורסמו בספרות. נשים לב שהאלגוריתם שלנו מספק תוצאות דומות בהסתמך על המדדים ACC ו-PSNR. עם זאת ניתן להבחין כי לפי מדד האיכות BDP האלגוריתם שלנו מבצע סגמנטציה איכותית יותר. מדד BDP מספק הערכה לסיווג נכונה של פיקסלים כלומר הפיקסלים שהאלגוריתם שלנו זהה כמשתייכים לחדר השמאלי והם אכן כאלו. האלגוריתם שלנו מספק תוצאו טובות הרבה יותר משאר האלגוריתמים האחרים בקביעת מהם הפיקסלים ששייכים לחדר השמאלי. בנוסף לפי צפייה בשאר המדדים ניתן להבין כי הסיווג של פיקסל ככזה ששייך לחדר השמאלי איננו בא על חשבון זיהוי לא נכון של פיקסלים ככאלו המשתייכים לחדר השמאלי. לפיכך ניתן להסיק שהאלגוריתם שלנו קובע בצורה מדויקת יותר את מתאר חדר שמאל משאר האלגוריתמים. מהשוואה שנעשתה בין האלגוריתמים השונים לפי זמני ריצה ניתן ללמוד מספר דברים. תחילה שזמן הסגמנטציה הממוצעת למטופל באלגוריתם שלנו היא בערך דקה למול אלגוריתמים אחרים בהם זמן הריצה שלהם קצר הרבה יותר. אומנם ניסינו לשפר ואכן יש שיפור משמעותי בזמני הריצה לעומת אלגוריתמי מושבות נמלים אחרים אולם עדיין האלגוריתם שלנו בזבזני מבחינת זמן ריצה ויש לראות האם וכיצד ניתן לשפר זאת בהמשך.

## 12. סיכום והצעות להמשך מחקר

בפרויקט זה הצגנו אלגוריתם חדש לביצוע סגמנטציה של חדר שמאל (LV) מתמונות MRI לבביות. לביצוע הסגמנטציה השתמשנו באלגוריתם FCM תוך שיפורו בעזרת בחירת מרכזים התחלתיים בעזרת אלגוריתם MMAS. היינו הראשונים לבחון את אלגוריתם הסגמנטציה המתואר על תמונות MRI של הלב וזאת במטרה לבצע איתור איכותי ומדויק של חדר שמאל. בנוסף הראנו בפעם הראשונה כיצד ניתן להשתמש באלגוריתם PCA לביצוע אופטימיזציה לתהליך הסגמנטציה. אחרי בחינה של איכות התוצאות ראינו את ביצועיו הטובים של האלגוריתם. זאת ועוד ניתן לראות כי האלגוריתם מתמודד בצורה טובה עם מטופלים שונים בעלי מחלות לבביות מגוונות. ניתן לראות מההשוואה אל מול אלגוריתמי אחרים שהאלגוריתם שלנו מהווה התקדמות משמעותית.

לאור התוצאות החיוביות נבחן בהמשך כיצד ניתן לשפר אלגוריתם זה.

בין הרעיונות לשיפור והתקדמות ניתן למנות:

- שיפור האלגוריתם לדיסקציה של חלל חדר שמאל לאחר שלב הסגמנטציה.
- חישוב תפוקת ומצב הלב תוך שימוש במידע שהפקנו הכולל את הסגמנטציה של כל התמונות בחתך חלל הלב.
- לבחון האם ניתן לשפר את זמני הריצה של האלגוריתם.

1. Petitjean Caroline and Dacher Jean-Nicolas. *"A review of segmentation methods in short axis cardiac MR images"*, Medical Image Analysis, Elsevier, 2010.
2. Kang Dongwoo, Woo Jonghye, Slomka Piotr.J, Dey Damini, Germano Guido and Jay-Kuo C. *"Heart chambers and whole heart segmentation techniques: review"*, SPIE Journal of Electronic Imaging, 2012.
3. Weng J., Singh A. and Chiu MY. *"Learning-based ventricle detection from cardiac MR and CT images"* IEEE Transactions on Medical Imaging 1997.
4. Marie-Pierre Jolly *"Automatic segmentation of the left ventricle in cardiac MR and CT images"*, International Journal of Computer Vision, 2006.
5. Lorenzo-Valdes M., Sanchez-Ortiz G., Elkington A., Mohiaddin R. and Rueckert D. *"Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm"*, Medical Image Analysis, 2004.
6. Kedenburg G., Cocosco C., Kothe U., Niessen W., Vonken E., and Viergever M. *"Automatic cardiac MRI myocardium segmentation using graphcut"*, SPIE 6144, Medical Imaging 2006.
7. Xulei Yang, Si Yong Yeo, Calvin Lim, Yi Su, Min Wan, Liang Zhong and RuSan Tan. *"A Framework for Auto-segmentation of Left Ventricle from Magnetic Resonance Images"*, APCOM & ISCM December 2013 Singapore.
8. Cootes, T, Cooper, D, Taylor, C.Graham. *"Active shape models - their training and application"*. Computer Vision and Image Understanding 61 (1), 38–59. 1995
9. Nikos Pargios etc. *"Active Shape Models & Segmentation of the Left Ventricle in Echocardiography"*. 2004.
10. H.C Van Assen etc. *"3D Active Shape Model Matching for Left Ventricle Segmentation in Cardiac CT"*, 2002.
11. Ganesh S.Raghtate and Suresh S.Salankar. *"Brain Tumor Segmentation Using Fuzzy C Means with Ant Colony Optimization Algorithm"*, Trends in technology and science 2015.
12. Abd-El-Ouahab Boudra *"Automated detection of the left ventricular region in magnetic resonance images by Fuzzy C-Means model"*, International Journal of Cardiac Imaging 1997.

13. C.Immaculate Mary, Dr. S.V. Kasmir Raja. *"Improved Fuzzy C-Means Clusters with Ant Colony optimization "*. International Journal of Computer Science & Emerging Technologies Volume 1, Issue 4, December 2010.
14. Yuanjing Feng and Zhejin Wang. *"Ant Colony Optimization for Image Segmentation"*, Ant Colony Optimization – Methods and Application, 2011.
15. Alice R. Malisia and Hamid R. Tizhoosh. *"Applying ant colony optimization to binary thresholding"*, IEEE 2006.
16. Salima Ouadfel and Mohamed Batouche *"Unsupervised Image Segmentation Using a Colony of Cooperating Ants"*, International Workshop on Biologically Motivated Computer Vision, 2002.
17. Roshan Dharshana Yapa and Koichi Harada. *"Connected Component Labeling Algorithms for Gray-Scale Images and Evaluation of Performance using Digital Mammograms"*, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008.
18. Lifeng He, Yuyan Chao, and Kenji Suzuki. *"A Run-Based Two-Scan Labeling Algorithm"*, IEEE Transactions on Image Processing, VOL. 17, No. 5, May 2008
19. MICCAI Workshop – Cardiac MR Left Ventricle Segmentation Challenge, 2009, [http://smial.sri.utoronto.ca/LV\\_Challenge/Home.html](http://smial.sri.utoronto.ca/LV_Challenge/Home.html)
20. Constantinides C., Chenoune Y., Kachenoura N., Roullot E., Mousseaux E., Herment A., et al., *"Semiautomated cardiac segmentation on cine magnetic resonance images using GVF-Snake deformable models"* , The MIDAS Journal—Cardiac MR Left Ventricle Segmentation Challenge, 2009. Available: <http://hdl.handle.net/10380/3108>.
21. Schaerer J., Casta C., Pousin J., and Clarysse P., *"A dynamic elastic model for segmentation and tracking of the heart in MR image sequences"* ,Medical Image Analysis, vol. 14, pp. 738–749, Dec 2010.
22. Cousty J., Najman L., Couprie M., Clement-Guinaudeau S., Goissen T., and Garot J., *"Segmentation of 4D cardiac MRI: Automated method based on spatio-temporal watershed cuts"*, Image Vision Computing, vol. 28, pp. 1229–1243, Aug 2010.
23. Lalande A., Salvé N., Comte A., Jaulent M. C., Legrand L., Walker P. M., et al., *"Left ventricular ejection fraction calculation from automatically selected and processed diastolic and systolic frames in short axis cine-MRI"* , Journal of Cardiovascular Magnetic Resonance, vol. 6, pp. 817–827, 2004.
24. C. Constantinides, E. Roullot, M. Lefort, and F. Frouin, *"Fully automated segmentation of the left ventricle applied to cine mr images: description and results*

- on a database of 45 subjects”, IEEE Engineering in Medicine and Biology Society(EMBC) , vol. 2012, pp. 3207–3210, 2012.
25. Lebenberg J, Lalande A, Clarysse P, Buvat I, Casta C, Cochet A, et al. *"Improved Estimation of Cardiac Function Parameters Using a Combination of Independent Automated Segmentation Results in Cardiovascular Magnetic Resonance Imaging"*, PLoS ONE, 2015.
  26. Eight segmentations on MICCAI 2009 LV Grand Challenge, 2015,  
<https://github.com/frederiquefrouin/Medieval>
  27. J. A. C. Lima and M. Y. Desai, “*Cardiovascular magnetic resonance imaging: current and emerging applications,*” Journal of the American College of Cardiology, vol. 44, no. 6, pp. 1164-71, September 2004.
  28. C. B. Higgins and H. Sakuma, “*Heart disease: Functional evaluation with mr imaging,*” Radiology, vol. 199, pp. 307-315, May 1996.
  29. G. Constantine, K. Shan, S. D. Flamm, and M. U. Sivananthan, “*Role of mri in clinical cardiology,*” The Lancet, vol. 363, pp. 2162-71, June 2004.
  30. <https://github.com/cptanalatriste/aco-image-segmentation>
  31. <https://github.com/spwhitt/cclabel>
  32. [https://en.wikipedia.org/wiki/Connected-component\\_labeling](https://en.wikipedia.org/wiki/Connected-component_labeling)
  33. <https://github.com/ab93/SIFCM>
  34. B. K. Tripathy, Avik Basu, Sahil Govel, *"Image Segmentation using spatial intuitionistic fuzzy C means clustering"* IEEE December 2014.



## נספח א' - הוראות התקנה

התחל התקנת סביבת הפיתוח:

הורד והתקן את סביבת הפיתוח Anaconda עבור פייתון 2.7 מהאתר:

<https://www.continuum.io/downloads>

הורד והתקן את סביבת הפיתוח PyCharm מהאתר:

<https://www.jetbrains.com/pycharm/>

סביבה זאת מאפשרת לפתח ולדבאג את הפרויקט יותר בקלות ( יש לציין כי לשם הרצה בלבד ההתקנה איננה חובה). הגדר שסביבת הפיתוח PyCharm תצביע ותשתמש בחבילות שיוורדו ב-Anaconda. לחץ על PyCharm's Preferences → Python Interpreters → Project Interpreter לחץ על סימן + והוסף את Anaconda לרשימה.

התקן את כלל החבילות שהתמשנו בפרויקט ומצויינות בפרק תיעוד החבילות שהשתמשנו בפרויקט. שים לב, רוב החבילות כבר מותקנות ולכן אין צורך להתקין אותם מחדש (החבילות שיש להוסיף הם pydicom,opencv,numpy). ובכל זאת כדי לבדוק אם חבילה נמצאת בתוכנית יש לפתוח את סביבת העבודה Spyder המותקנת כחלק מסביבת Anaconda ולכתוב שם import ושם החבילה, אם החבילה לא מותקנת תופיע הערה.

סביבת Anaconda מאפשרת התקנת חבילות חדשות בצורה פשוטה ונוחה למשתמש.

ההתקנת חבילה חדשה נעשת באופן הבא:

- יש לפתוח command prompt (ניתן לעשות זאת על ידי הזנת הערך cmd בחיפוש תוכניות של windows).
- יש לגשת לתיקייה בה נשמרה סביבת העבודה של Anaconda (אם לא שינית את ההגדרות תוכל למצוא אותה בתיקיית המשתמש C:\Users\\_\_\_\_\Anaconda2 ושם יש לנווט לתיקייה Scripts).

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ChenA>cd C:\Users\ChenA\Anaconda2
C:\Users\ChenA\Anaconda2>cd Scripts
C:\Users\ChenA\Anaconda2\Scripts>
```

- בתיקייה הזאת יש את הסקריפט conda שבאמצעותו ניתן להתקין את החבילות. הרץ את הסקריפט conda עם הדגל -install ושם החבילה ועבור החבילה opencv הקש conda install -c menpo opencv.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ChenA>cd C:\Users\ChenA\Anaconda2
C:\Users\ChenA\Anaconda2>cd Scripts
C:\Users\ChenA\Anaconda2\Scripts>conda --version
conda 4.3.16

C:\Users\ChenA\Anaconda2\Scripts>conda --install
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: too few arguments
```

הוראות התקנה לכלי הפיתוח NetBeans לפיתוח קוד JAVA:

- הורד JDK (Java SE Development Kit) כלים לפיתוח ודיבאג (יישומי Java) מאתר חברת .oracle
- הוסף את הנתיב JDK\bin למשתני הסביבה (Environment variables) במחשבך.
- בדוק שההתקנה עברה על ידי פתיחת command prompt והקלדת הפקודה java –version הפלט צריך להיות (למעט מספר הגרסא שיכול להשתנות בהתאם להתקנה):

```
Command Prompt
C:\Users\ChenA>java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b15, mixed mode)
```

- הורד והרץ את ההתקנה של כלי הפיתוח NetBeans מהאתר הרשמי בכתובת:

<https://netbeans.org/>

## נספח ב' - הוראות הרצה

- נשתמש בסביבת Spyder שמותקנת כחלק מסביבת Anaconda כדי להריץ את הסקריפטים השונים במערכת.
- יש לפתוח בסביבת Spyder את הסקריפט gallery.py ולהריץ אותו.
- הסקריפט gallery.py דואג –
- להריץ את ממשק המשתמש.
  - לנהל את ריצת האלגוריתמים השונים במערכת על הקלט.
  - לשמור את תוצאות הסגמנטציה בבסיס נתונים.
  - להציג בצורה נוחה למשתמש בעזרת ממשק WEB את התוצאות של הפעלת אלגוריתם הסגמנטציה.

## נספח ג' – מדדי איכות עבור האלגוריתם בפרויקט על המטופלים השונים

PatientID	Gender	Age	PSNR	ACC(%)	BDP(%)
SC-HF-I-1	Male	53	12.35	93.17	79.71
SC-HF-I-2	Male	48	8.31	84.02	88.06
SC-HF-I-4	Male	79	10.24	89.27	72.78
SC-HF-I-5	Male	45	10.90	90.88	76.06
SC-HF-I-6	Male	60	11.16	91.77	84.30
SC-HF-I-7	Male	74	10.25	89.34	55.71
SC-HF-I-8	Male	46	7.97	82.55	75.97
SC-HF-I-9	Male	57	9.29	86.02	75.99
SC-HF-I-10	Male	69	9.23	87.26	76.78
SC-HF-I-11	Male	55	11.27	91.56	83.40
SC-HF-I-12	Female	71	13.11	94.76	70.33
SC-HF-I-40	Male	77	11.09	89.81	67.96
SC-HF-NI-3	Female	81	8.70	85.52	69.87
SC-HF-NI-4	Male	77	9.56	86.42	72.62
SC-HF-NI-7	Male	81	10.26	90.12	67.80
SC-HF-NI-11	Male	71	9.17	86.15	67.64
SC-HF-NI-12	Female	63	8.01	81.86	79.01
SC-HF-NI-13	Female	68	11.43	91.68	72.99
SC-HF-NI-14	Male	82	11.43	92.10	62.37
SC-HF-NI-15	Male	27	8.45	81.68	74.97
SC-HF-NI-31	Male	52	8.07	81.94	70.66
SC-HF-NI-33	Male	47	9.22	85.59	68.22
SC-HF-NI-34	Male	52	11.87	91.52	82.16
SC-HF-NI-36	Female	77	10.69	90.78	81.46
SC-HYP-1	Male	83	12.98	92.23	72.29
SC-HYP-3	Male	42	12.62	93.99	54.49
SC-HYP-6	Female	46	13.47	95.04	79.44
SC-HYP-7	Male	61	12.00	93.26	57.72
SC-HYP-8	Male	62	9.57	86.48	64.39
SC-HYP-9	Female	48	10.00	82.32	78.45
SC-HYP-10	Male	47	9.67	85.37	79.48
SC-HYP-11	Female	53	13.50	95.17	64.85
SC-HYP-12	Female	76	12.03	91.14	52.53
SC-HYP-37	Female	68	14.87	95.74	63.91
SC-HYP-38	Male	38	10.26	81.86	70.95
SC-HYP-40	Male	68	12.44	92.43	51.39
SC-N-2	Male	63	4.64	64.03	95.70
SC-N-3	Female	53	14.11	95.45	73.63
SC-N-5	Female	77	11.80	87.60	80.61
SC-N-6	Male	70	13.28	94.91	72.90
SC-N-7	Male	23	5.89	70.93	78.54
SC-N-9	Male	51	10.54	88.89	74.11
SC-N-10	Male	61	13.08	94.52	65.92
SC-N-11	Male	60	6.49	76.56	90.19
SC-N-40	Female	88	12.19	93.34	63.02