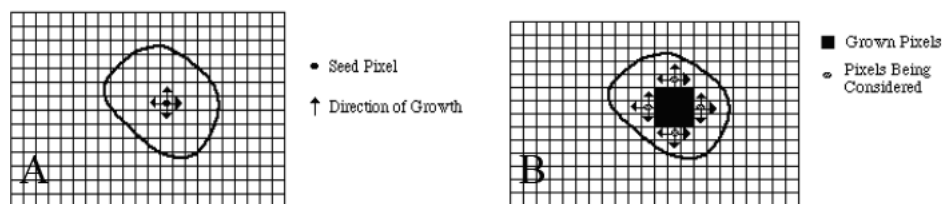


## בדיקת ביצוע סגמנט ע"י אלגוריתם Region growing

### רקע

בניסוי הקודם ביצענו סגמנט על 2000 תמונות מן המאגר של ISIC (מופיעות בקישור [הזה](#)) באמצעות Otsu's thresholding . ע"פ מדד Dice הגעת לרמת דיוק של כ-56%. מתוך 2000 התמונות, ישנה קבוצה מייצגת של 150 תמונות (הניתנת להורדה בקישור [הזה](#)) ובדיקתן קיבלנו 44% דיוק ע"פ מדד Dice.

בעקבות התוצאות הנ"ל ובעצת אסף המנחה אותנו, הוחלט לעשות שימוש באלגוריתם Region growing לביצוע הסמנט. לפי אלגוריתם זה, יש לחלק את התמונה לקבוצת התעניינות אחת אשר תכיל פיקסלים רלוונטיים ולזנוח את כל שאר הפיקסלים. לצורך כך, יש לקבוע נקודת התחלה (Seed point) ולבחון את כל שאר הפיקסלים לפי קריטריונים מוגדרים וברורים מראש תוך התייחסות לייחודיות ה-seed. כאשר פיקסל מסוים עומד בקריטריונים, הוא מתווסף לקבוצת ההתעניינות (שהיא למעשה תוצאת החישוב) ומכאן שם האלגוריתם growing (לאט לאט גדל ומתווסף לקבוצה). דוגמא לאופן ריצת האלגוריתם:



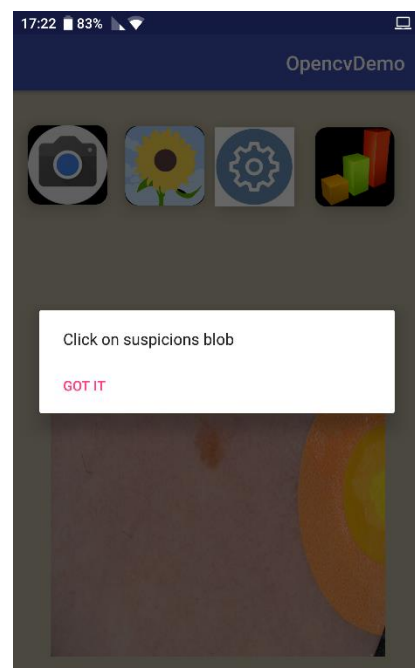
<http://www.scrigroup.com/limba/engleza/109/Deep-Brain-Stimulation24766.php>

בתמונה A ניתן לראות seed point ובאמצעותה מתבצעת בדיקה של שכניה לפי קריטריון מסוים (לרוב ייבדק דמיון בין הפיקסלים השכנים מבחינת צבעם עד לסף מסוים).

בתמונה B ניתן לראות את קבוצת הפיקסלים שעמדה בתנאים והיא צבועה בשחור וכעת מתבצעת בדיקה זהה לכל שכניהם וכך הלאה.

באפליקציה שלנו, מימשנו את אלגוריתם region growing באופן שבו אנו מבקשים מן המשתמש לטעון\לצלם תמונה ומיד לאחר מכן ללחוץ על הנקודה החשודה ומיקום הלחיצה יהווה קורדינטת התחלה בעלת ערכי X,Y וזו למעשה תהיה ה-seed point.

בתמונה שמימין, ניתן לראות שהמשתמש טען תמונה לתצוגה הראשית, וכעת הוא מתבקש ללחוץ על הנקודה החשודה לצורך מציאת seed point. הודעה דומה תוצג על המסך לצורך מציאת צבע עורו של המשתמש.

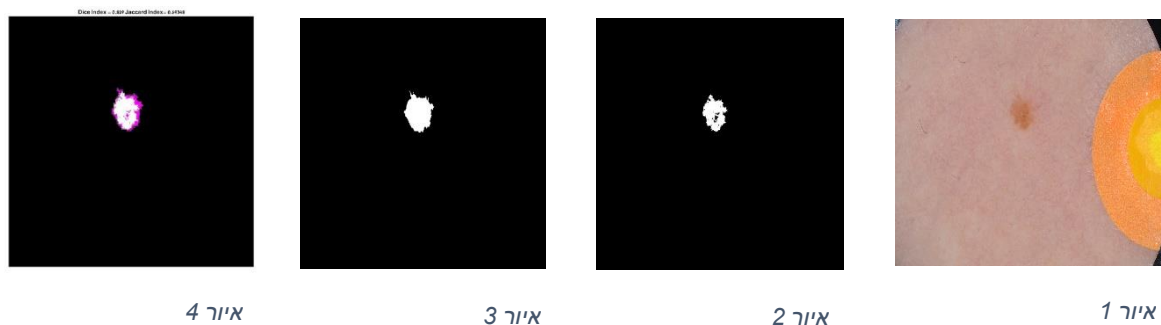


מנקודה זו מפיקים ערכי RGB ממוצעים ביחס לseed ולשמונת שכניה. בשלב הבא, המשתמש ילחץ על אזור אחר בתמונה המייצג באופן כללי את צבע עורו ובתהליך זהה אנו מפיקים ערכי RGB ממוצעים לצבע העור. מתוך שתי הנקודות הנ"ל ובנוסף ערכי ה-RGB המופקים, אנו יכולים לבצע סגמנט ע"פ האלגוריתם הנידון. דוגמא לסגמנט שבוצע על סמך אלגוריתם region growing:



## ניתוח התוצאות

- מכיוון שעבור כל תמונה יש להקיש על הנקודה החשודה ועל עור המשתמש (שהנקודה החשודה אינה נמצאת בו) באופן ידני ואינטראקטיבי, לא ניתן היה להריץ סקריפט שינתח את התמונות באופן אוטומטי דרך האפליקציה. לכן, השתמשנו במאגר של 150 התמונות מתוך האתר שבאמור, מהוות כלי טוב לבחינה ובדיקה של אפקטיביות הסגמנט שמופק.
- בדומה לניסוי הקודם השתמשנו בתוכנת מטלב לצורך השוואה בין שתי תמונות ולאחר מכן קבלת מדד Jaccard ו-Dice. אל תוכנת מטלב, טענו באופן אוטומטי את הסגמנט מהאפליקציה ואת התמונה שניתנה לנו בחלק ממאגר התמונות של ISIC.



איור 1 הוא התמונה המקורית שאותה יש לבחון.

איור 2 הוא התמונה שהופקה ע"י האפליקציה שלנו עם הקוד לאלגוריתם region growing.

איור 3 הוא תמונת הסגמנט המדוייקת אשר נמצאת במאגר התמונות של ISIC.

איור 4 היא התמונה שהופקה ע"י Matlab. הנקודות בסגול הינן נקודות אשר קיימות באיור 3 אך אינן קיימות באיור 2.

קשה להבחין בכיתוב של איור 4, אך הנתונים שכתובים למעלה הם:

$$\text{Dice Index} = 0.819 \quad \text{Jaccard Index} = 0.69348$$

מנתונים אלה, ישנה התאמה בין הסגמנטים ברמה של כ-82% (לפי דייס), ו-69% (לפי ג'אקארד).  
 לצורך השוואה, בניסוי הקודם עם ביצוע סגמנט ע"י Otsu's thresholding, לא הצלחנו לבצע  
 סגמנט בכלל כפי שניתן לראות להלן:



### ריכוז התוצאות

מבדיקה של 150 תמונות, הגענו לרמת דיוק בממוצע של:

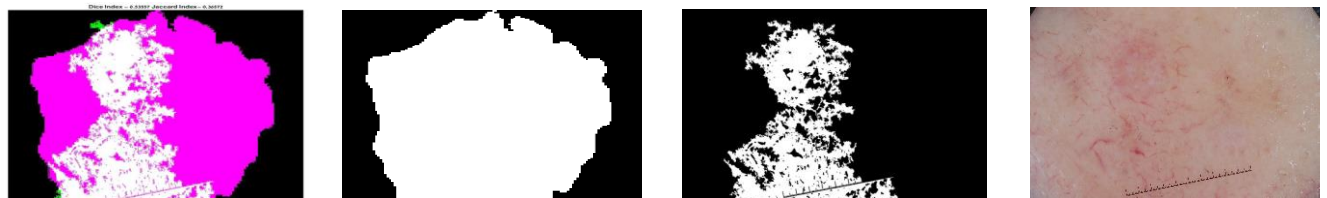
Dice: 0.653

Jaccard: 0.535

- ניתן לראות שיפור של כ-20% דיוק במדד Dice.
- בבדיקה הקודמת של ה-2000 תמונות, רמת הדיוק בממוצע לפי דייס הייתה כ-56%. יש להניח שגם כאן, אילו היו נבחנות 2000 תמונות, רמת הדיוק הייתה גדלה גם באופן דומה לכ-20%.
- כל הנתונים הופקו למסמך excel ומופיעים במאגר Github של הפרויקט, ויופיעו בספר פרויקט הגמר.

### נקודות חשובות

- בעת דגימת ה-seed point תיתכן קליטת פיקסל גבולי (שנמצא בשולי הנקודה החשודה) עקב דגימה עם אצבע המשתמש שאינו יכול לדייק עד רמת הפיקסל בעת הלחיצה.
- בחלק מן התמונות שבדקנו, התקשינו לאתר ולהבין מהו הכתם הנבדק. למשל בתמונה הבאה:



הבדיקה הנ"ל הניבה 53% דיוק במדד דייס.

- אופן מימוש האלגוריתם מתייחס לדגימה של seed point יחידה, ולכן כתמים לא מחוברים בעלי מספר מוקדים זהו באופן חלקי.