Image Processing Final Project

שם: מחמד חיו

תיאור פרויקט:

הפרויקט עוסק בשערוך מספר הרכסים מיתוך טביעת אצבע בעיבוד תמונה. המטרה היא להתמודד עם האתגרים הלא תמיד מדוייקים, כמו זיהוי הרכסים בתמונה שאינם בהירים במיוחד או תמונות שבהם ישנם רעשים. הפתרון יתבסס על חשיבה ממני, ניסיון ומחקרים בלי קוד כדי להשיג את העיבוד והזיהוי הטוב ביותר לכל התמונות ביחד.

קבצים וספריות:

- קובץ im2023sol.py הוא קובץ פייתון שכולל את הקוד הנחוץ לביצוע המשימה הנדרשת. הקובץ כלול פונקציות לאיתור החלון של 120 על 120 פיקסלים בו בטביעת אצבע נראית במטיבה ולספירת מספר הרכסים בחלון זה.
- input_images היא תיקייה שכוללת את כל תמונות הטביעות האצבעות שיש לבדוק בתרגיל. התמונות בפורמט PNG ובגודל 512*512 פיקסלים.
- out_images היא תיקייה שבה נשמרות את התמונות המעובדות שיוצאות מהקוד. כל תמונה כלולת ריבוע אדום בגודל 120*120 שמתאר את החלון בו מוצגת טביעת האצבע ואת מספר הרכסים באותו החלון. הפורמט של התמונות המעובדות ב-PNG.

ספריות דרושות :

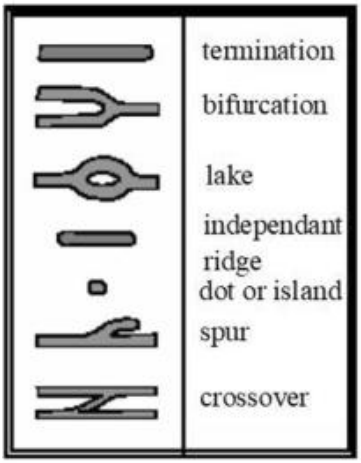
- scipy
- numpy
- os
- cv2
- scikit-image

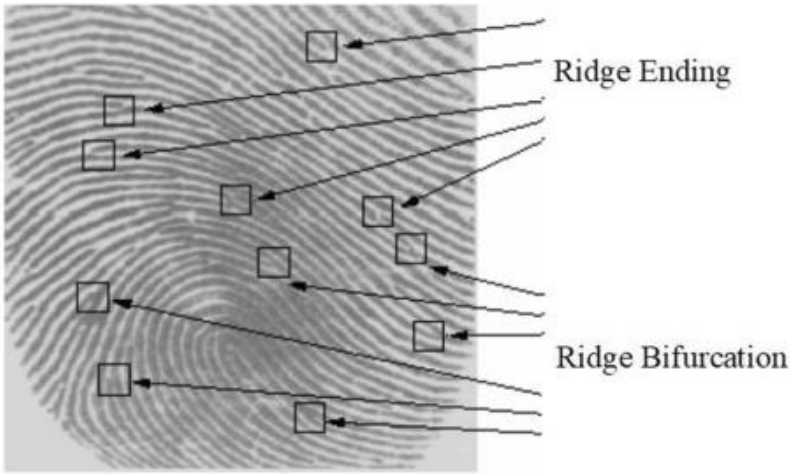
הוראות ריצת הקוד :

Python im2023sol.py

דרך החשיבה:

- קיבלתי 25 תמונות טביעות אצבע בגווני אפור בגודל (512*512).
- לא כל התמונות היו קלות לעבודה, והמשימה שלי הייתה למצוא את האזור הטוב ביותר של (120*120) פיקסלים לספירת רכסים.
- ההגדרה שלי לגבי החלון הנראה הכי טוב לספור, זה היה החלון שיש בו כמה שיותר שניות מבחינת צבע עם כמה שפחות מ- bifurcation ו-ending בשביל לספור כמה שיותר טוב בתוך החלון.





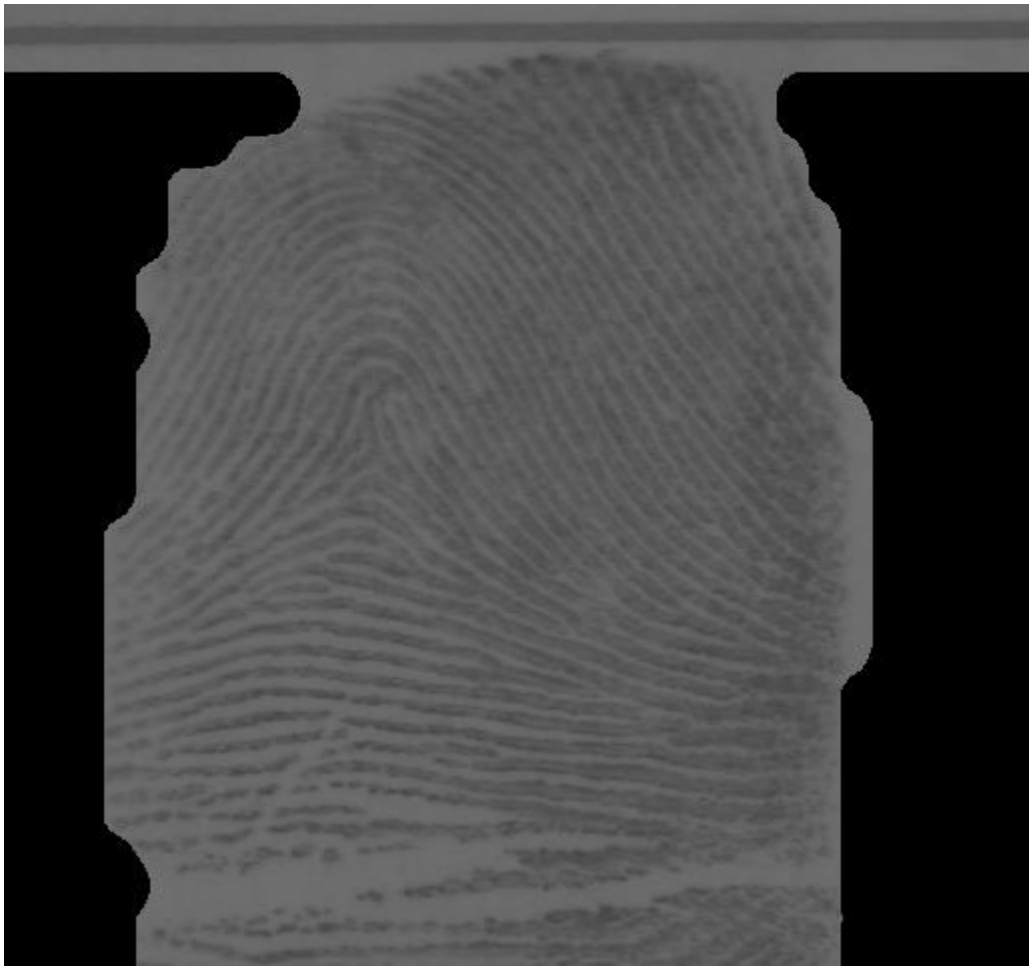
- ובשביל לספור כמה שאפשר נכון השתמשתי ב- Gabor filter המאפשר להראות את התמונה כמה שאשפר טוב כדי להתחיל לספור רכסים, וכל הדרך מוסברת בחלק Function algorithm Flow

Function algorithm Flow:

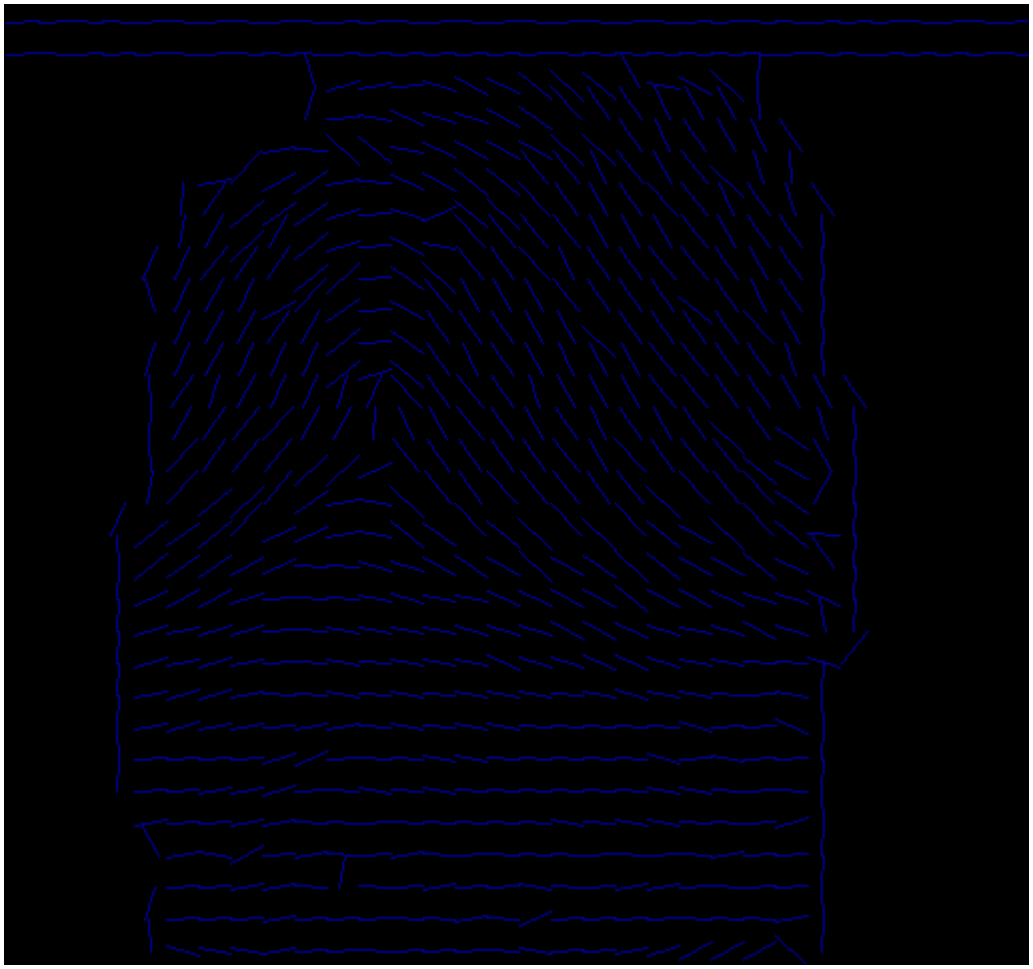
1. normalize image: פונקציה שמקבלת תמונת קלט ועושה עליה נורמליזציה. מטרת הנורמליזציה היא להתאים את נתוני התמונה כך שיהיו להם ממוצע יעוד ושונות מוגדרים, מה שיכול לעזור לשפר את איכות התמונה ולהקל על העבודה איתה.

2. segment image: הפונקציה מבצעת חלוקה של תמונה לפי קבוצות בלוקים ויצירת מסכה בינארית על פי השטח הקריטי של התקן של סטיית תקן בכל בלוק. המסכה הבינארית נוצרת על ידי חיתוך השטח הזה עם ערך קבוע המקבל את ערכו מהפרמטר threshold. הפונקציה מבצעת גם פעולות מורפולוגיות על המסכה כדי להסיר רעשים ולייצר תמונה חלוקה סופית. הפלט מחזיר תמונה חתימת האצבע שהופעל לעליה המסכה, תמונה מנורמלת, ומסכה בינארית.

תמונה חתימת האצבע שהופעל לעליה המסכה :

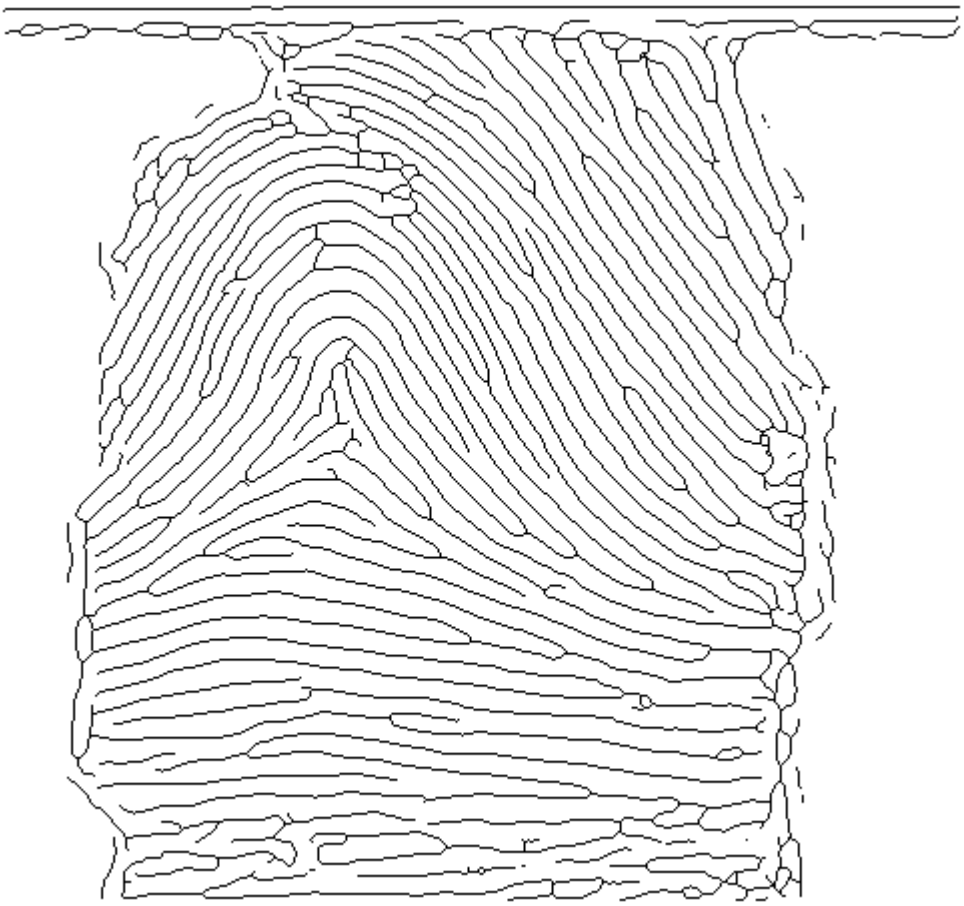


3. calculate angles: הפונקציה מקבלת תמונה, גודל חלון ופרמטר smooth שהוא בוליאני (כן או לא לבצע סינון ממוצע על התמונה לפני העיבוד). היא מחשבת את זוויות השיפוע עבור כל חלון בתמונה. לכל חלון, הפונקציה מחשבת את הגרדיאנט עם פעולת סובל ובעזרתו מחשבת את זוויות השיפוע באמצעות חישוב זווית השיפוע עם שימוש בנוסחת טנגנס. הפונקציה מחזירה מערך numpy דו-ממדי המכיל את זוויות השיפוע עבור כל חלון בתמונה.
התמונה להבהרה :



4. calculate frequency: הפונקציה מקבלת תמונה, זווית ופרמטרים נוספים, ומחשבת את הבלוק התדרי של התמונה. היא מבצעת כמה שלבים כגון סיבוב התמונה לפי הזווית הנתונה, חישוב הסכום של עמודות התמונה, הפעלת פילטר כדי להפחית את הרעש וחישוב התדר המקסימלי של התמונה ובודקת האם הוא נמצא בטווח התדרים הנתונים. הפונקציה מחזירה את הבלוק התדרי.
5. Estimate ridge frequency: הפונקציה משמשת לחישוב התדר של שולי האצבעות בתמונות מצולמות. הפונקציה מקבלת תמונה, מסיכה, זווית אוריינטציה, גודל בלוק, גודל נוגד וטווח אפשרי של אורך גל. הפונקציה מחלקת את התמונה לבלוקים, מחשבת את התדר של כל בלוק, ומחזירה את המדיאן (median) של התדרים הללו. הפונקציה מאפשרת למשתמש לזהות את התדר של השוליים בתמונה ולהשתמש בזה בהמשך לצורכי עיבוד נוספים של התמונה.
6. gabor ridge filter: הפונקציה מיישמת פילטר Gabor על תמונה דו-ממדית על מנת לזהות רכסים גבעוליות על התמונה. הפילטר מתאר בצורה מדוייקת את המבנה המרכזי של הרכסים הגבעולית ומאפשר לחשוף את כיוונה ותדירותה. הפונקציה מקבלת כקלט את התמונה, כיוון הרכסים, התדירות של הרכסים ופרמטרים נוספים ומחזירה את התמונה המסוננת.

7. Skeletonize: הפונקציה מבצעת skeletonization על תמונת גווני אפור ומחזירה תמונה skeletonized חדשה.



8. minutiae at: הפונקציה מזהה את סוג ה-minutiae במיקום הנתון בתמונה בינארית על פי הערכים של פיקסלים השכנים שלו, ומחזירה את סוג ה- minutiae: "bifurcation", "ending" או "none".
9. calculate minutes: הפונקציה מקבלת תמונה של טביעת אצבע בגוון אפור ומחשבת את מספר minutiae בתמונה. זאת נעשה על ידי יצירת מערך דו-ממדי חדש באותו גודל התמונה, המערך מאותחל באפסים והיתרון בשימוש בו שהיה לכל פיקסל משקל של minutiae . הפונקציה משתמשת בקרנל לסינון תמונה, ובפונקציה נוספת שקוראת minutiae_at המזהה האם הפיקסל הזה בהתמונה הנוכחית הוא סיומת (ending) או ביפורקציה (bifurcation). בכל פעם שפיקסל בתמונה הוא סיומת (ending), הפונקציה מוסיפה למספר minutiae באותו מקום 2.5, וכאשר הנקודה היא ביפורקציה (bifurcation) , הפונקציה מוסיפה למספר minutiae באותו מקום 1.0. ואת מספר minutiae בפיקסל הזה מוצב בתוך ה-מערך הדו-ממדי ובסוף הפונקציה מחזירה את המערך הזה.
10. Count lines: הפונקציה מקבלת תמונה ומחזירה את מספר הקווים המתארים את הקו הראשי והקו המשני בתמונה, וגם את שם הקו שמופיע יותר פעמים מביניהם. הפונקציה פועלת על ידי החישוב של מספר הקווים בציר הראשי ובציר המשני של התמונה. עם התמונה הזאת, הפונקציה מבצעת חישוב של מספר הקווים המתארים את הקו הראשי והקו המשני על ידי חיפוש פיקסלים של לבן ושל שחור על הציר הראשי והציר המשני. היא מחזירה את מספר הקווים ואת השם של הציר שיש בו מספר גדול יותר של קווים.
11. Draw diagonal: הפונקציה מקבלת כקלט תמונה, נקודות התחלה וסיום של קו, מספר אלכסוני, גודל בלוק, מיקום טקסט, צבע טקסט ומציירת על התמונה קו עם מספר האלכסון ומילת "ridges" מתחת לקו.
12. Calculate singularities: הפונקציה מגלה את המספר הקלטות הקטנות ביותר של בעיות מינוטיה בבלוק הכי קטן המכוסה במסיכה על פי גודל הבלוק שהועבר כפרמטר לפונקציה. היא מחשבת את מספר האלכסונים ומציירת אותו על עותק של התמונה המקורית המתקבלת כקלט. הפונקציה מקבלת מספר פרמטרים עם תמונת טביעת האצבעות הדקה (thin_image), תמונה המכילה את מספר בעיות המינוטיה לכל פיקסל בתמונה הדקה (minutia_problems_image), גודל הבלוק, מסכה

ותמונת הקלט המקורית. הפונקציה מחזירה תמונת תוצאה, מספר אלכסונים ומספר הבעיות המינוטיה בבלוק הטוב ביותר.

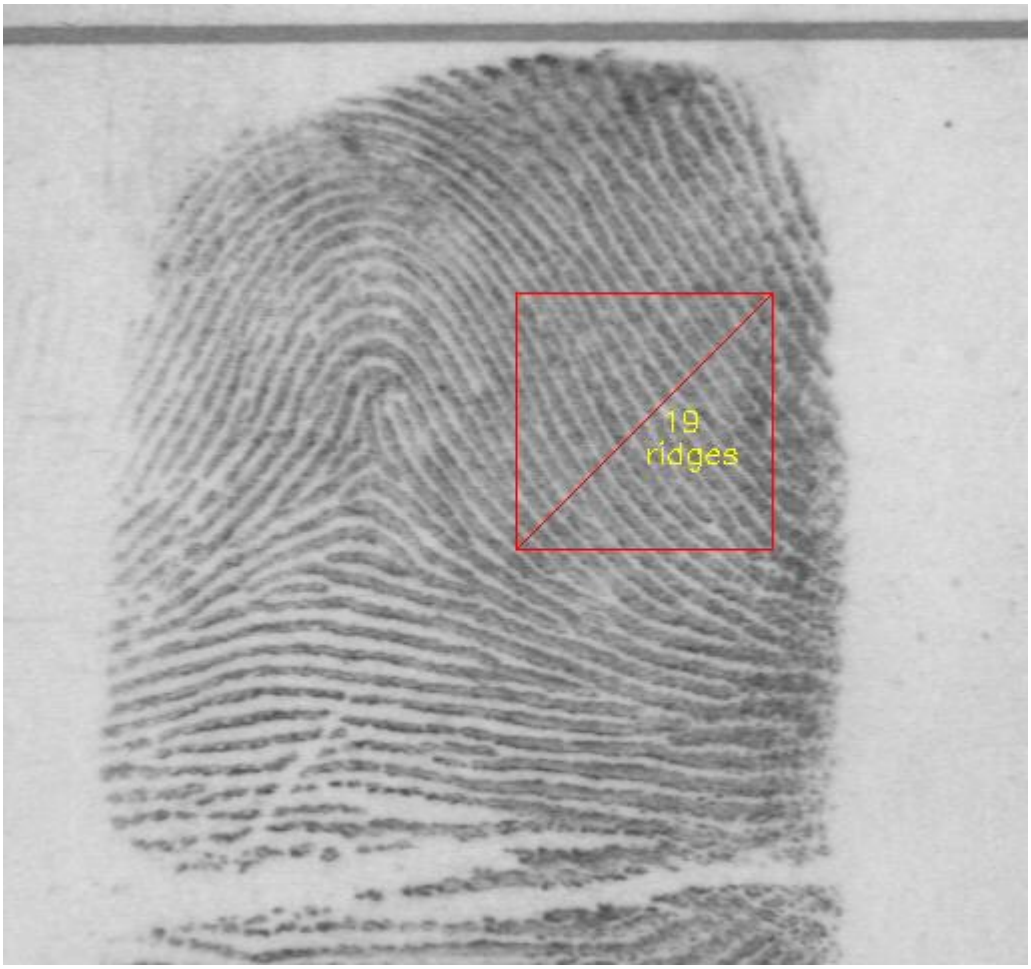
13. Gender fingerprint pipeline: הפונקציה מקבלת את התמונה המקורית ואת גודל החלון ואת ערך threshold_level והיא מעבדת תמונת טביעת אצבע כדי לאתר את הבלוק שנראה הכי טוב לספירת מספר רכסים. הפונקציה קוראת לפונקציות הבאות לפי הסדר הבא:

- normalized_img
- segment_image
- calculate_angles
- estimate_ridge_frequency
- gabor_ridge_filter
- skeletonize
- calculate_minutes
- calculate_singularities

לבסוף, הפונקציה מחזירה את: התמונה המקורית שמתואר בתוכה את החלון עם מספר הרכסים כתוב בתוך החלון , מספר הרכסים בחלון, וסכום מספר הבעיות של minutiae בשביל לבדוק את התוצאה עם ערכי threshold אחרים.

אני בודק 5 ערכי threshold שונים : threshold levels = [0.45, 0.35, 0.3, 0.25, 0.2]
ומוצא את הטוב בניהם לפי החלון בעל מספר מינימלי של minutiae.

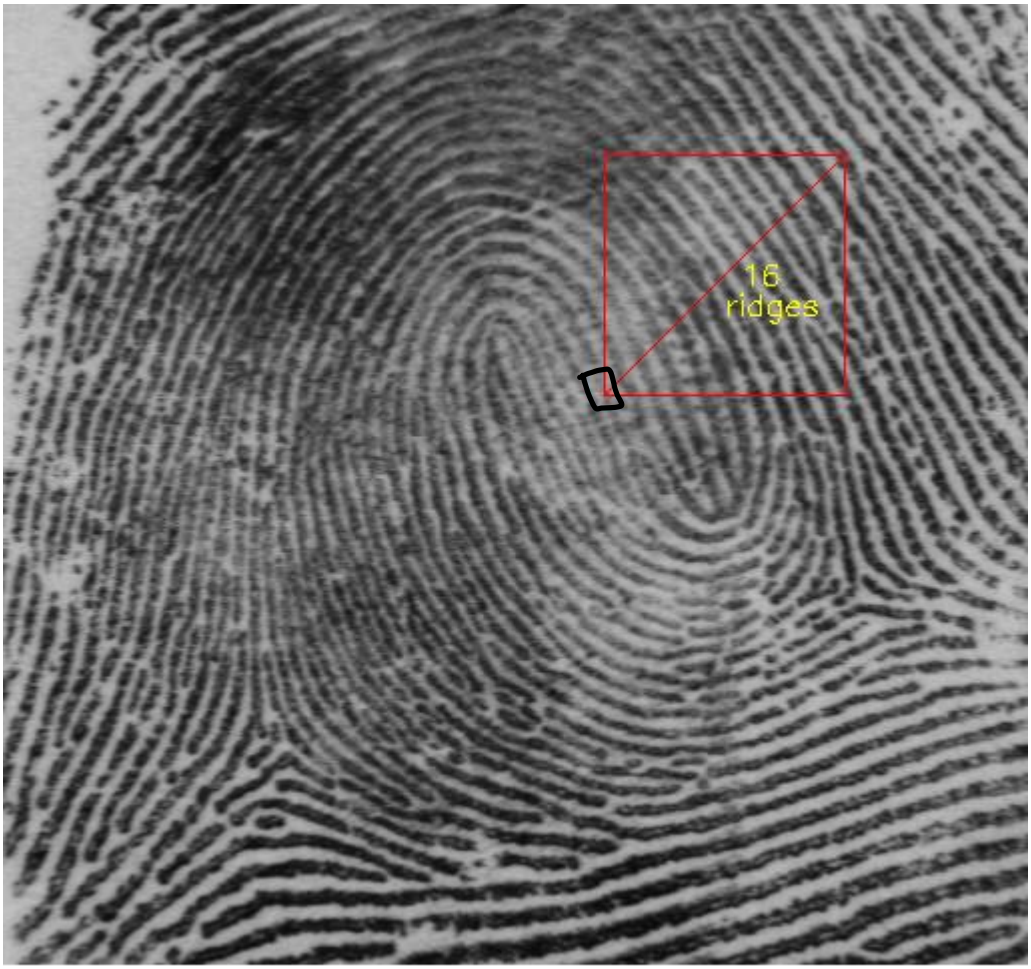
התוצאה הסופית :



הערה סופית :

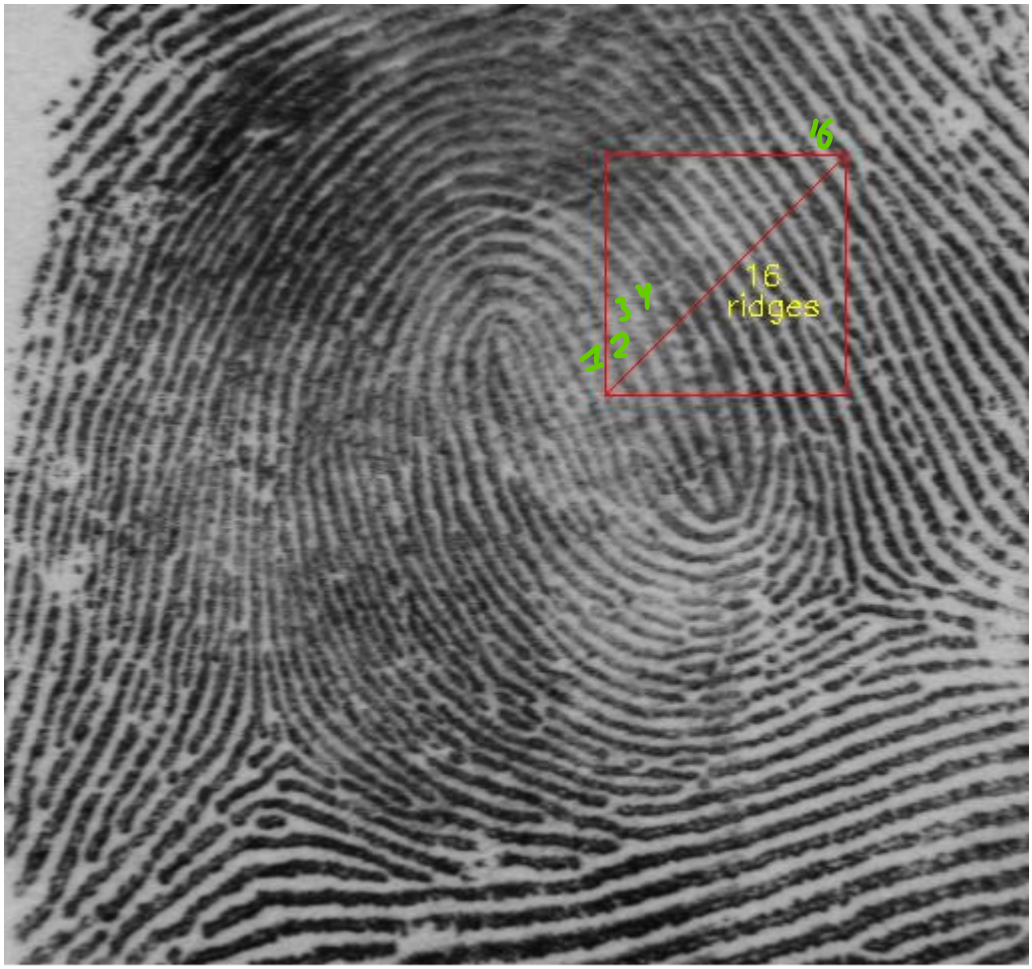
אני האמת התלבטתי האם לספור את הרכסים שנוגעים בקצוות או לא ,

ובסוף בחרתי לקחת אותם למשל התמונה הבאה :



אפשר לראות שהמרובע השחור שצרפתי נמצא רכס שנוגע בסוף האלכסון, אבל אני סופר על התמונה שנעשה עליה skeletonize ולפי זה בחרתי האם רכס כזה נספר או לא, בתמונה שצרפתי הוא כן נכנס לתוך החלון של התמונה שנעשה עליה skeletonize ולכן הוא כן נספר

ואז :



מקורות :

1) https://www.youtube.com/watch?v=kfGfHHXvYiA&ab_channel=DakshinaRanjanKisku
2) https://pdfs.semanticscholar.org/6e86/1d0b58bdf7e2e2bb0ecbf274cee6974fe13f.pdf
3) https://airccj.org/CSCP/vol7/csit76809.pdf
4) https://pdfs.semanticscholar.org/ca0d/a7c552877e30e1c5d87dfcfb8b5972b0acd9.pdf