

## קבוצה 3 – AgriView

### מגישים:

203195706	צבי פוצ'נסקי
336382833	אלה קיטייבה
310514997	מיכאל אפונין
303071302	אלעד בן אברהם

- קישור לפרויקט ב GitHub - <https://github.com/Tzvi23/AgriView>

### • תיאור כיסוי דרישות:

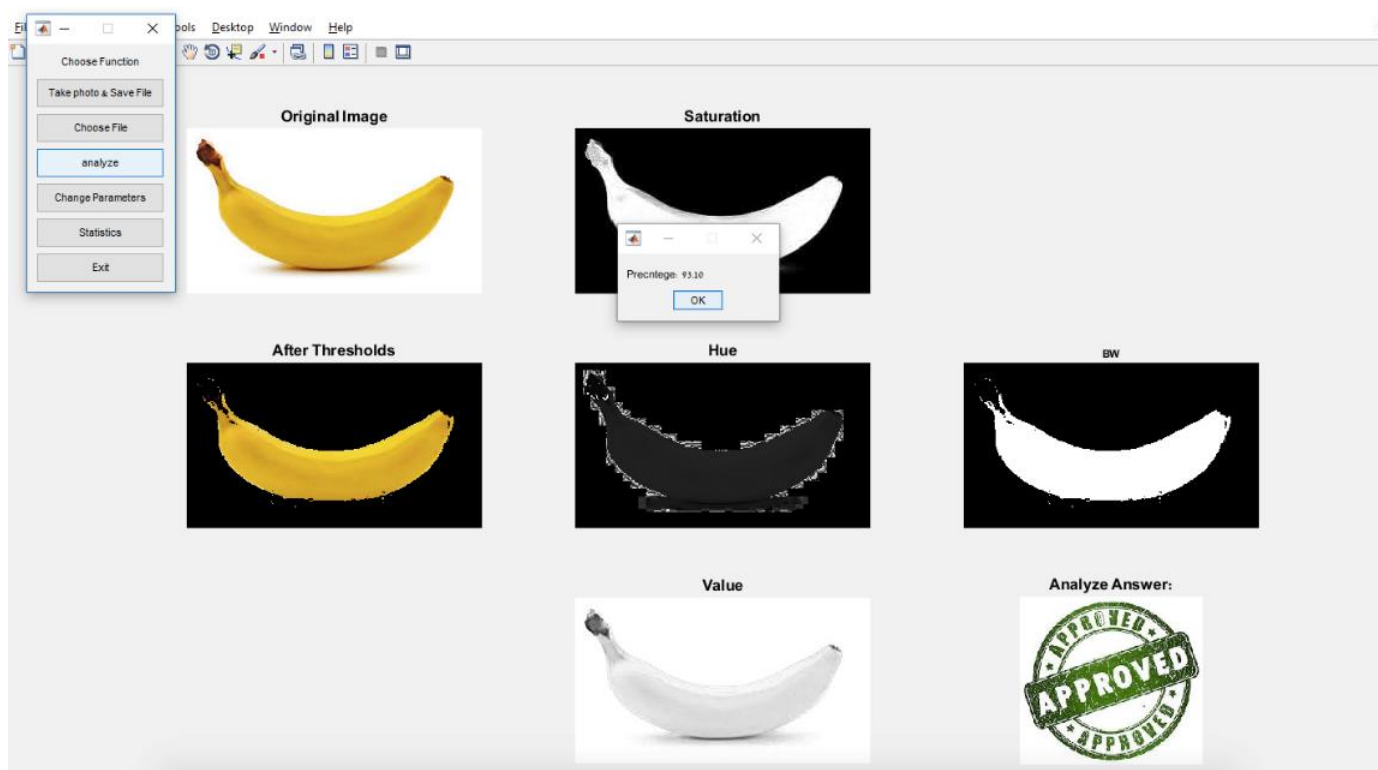
בהאקטון הקודם ציינו בקובץ את הדרישות העיסוקיות שעליהן אנו רוצים לענות בפיתוח המוצר ללקוח. ציינו את הדרישות שרצינו לענות עליהן:

1. צמצום כח אדם (עסקית)
  - 1.1. פיתוח אלגוריתמים לבדיקה אוטומטית (פיתוח)
  2. הגדלת כמות הפירות הנבדקים לדקה (עסקית)
    - 2.1. ייעול אלגוריתם בדיקה (פיתוח)
    - 2.2. ייצור עמדה ייעודית בעלת רקע לבן לצורך סינון פשוט ומהיר יותר (פיתוח)
  3. הפקת דוחות סטטיסטיים (עסקית)
    - 3.1. שמירת נתונים מכל בדיקה לצורך הפקת דוחות (פיתוח)
    - 3.2. יצירת דוחות על פי פרמטרים ותבנית מוגדרים מראש (פיתוח)
  4. קיצור זמן הכשרת עובדים (עסקית)
    - 4.1. ממשק נוח וברור למשתמש (פיתוח)
  5. הקטנת עלות החומרה הדרושה למערכת (עסקית)
    - 5.1. סינון תמונה יעיל שלא יצרוך שימוש בעמדה ייעודית (פיתוח)

### 1. דרישה עסקית – צמצום בכח אדם.

על מנת לענות על דרישה עסקית זו נדרשנו קודם כל לענות על דרישת פיתוח שהוגדרה מראש "פיתוח אלגוריתמים לבדיקה אוטומטית". בפיתוח המוצר אכן נבנה אלגוריתם בדיקה וממומש על ידי לחצן ANALYZE במוצר. האלגוריתם עובד על פי שיטת פירוק תמונה על ידי המרת ההתמונה ל HSV(Hue,Saturation,Value) על ידי כך אנו יכולים בקלות להפריד את רקע התמונה ולקבל רק את שטח הפרי הנבדק (על ידי עמדה ייעודית שעליה נפרט בהמשך) וזהו השלב הראשון בתהליך. השלב השני הוא בחירה של המשתמש – סוג פרי הנבדק. לאחר מכן התהליך ממשיך לשלב ניתוח התמונה אחרי סינון הרקע וסינון הצבעים על פי פרמטרים מוגדרים מראש או על ידי טווחים שהמשתמש הזין קודם לכן. התהליך המרכזי מתבצע באופן הבא:

- נספרים כמות הפיקסלים הנמצאים בטווחים שהוזנו.
- יש לציין כי כמות הפיקסלים נספרים על בסיס צבעים של RGB וטווחי הצבעים מוגדרים בהתאם על ידי משתנים : Red Min/Max, Green Min/Max, Blue Min/Max.
- נלקחת התמונה אחרי ביצוע ההפרדה, מומרת לצבע אפור ולאחר מכן מומרת לבינארי כלומר מתקבל שטח הפרי כאשר כל האזור הצבוע מיוצג על ידי '1' ואז נספרים כמות הפיקסלים. התוצאה, כמות הפיקסלים בשטח התמונה הנבדקת.
- בשלב זה קיימים שני משתנים – מספר הפיקסלים המסוננים וסה"כ כמות הפיקסלים. נבצע חילוק בין מספר הפיקסלים המסוננים וסה"כ כמות הפיקסלים = החלק היחסי של הטווח הנבדק וזהו אחוז ההצלחה המגדיר את איכות הפרי. הוגדר כי מעל 75% הפרי תקין.

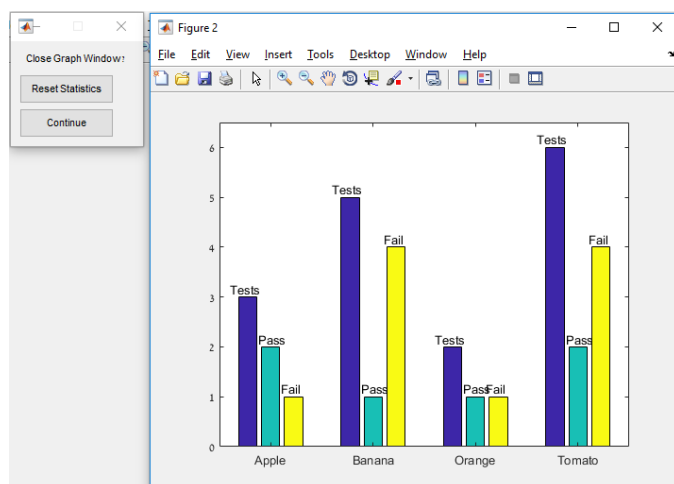


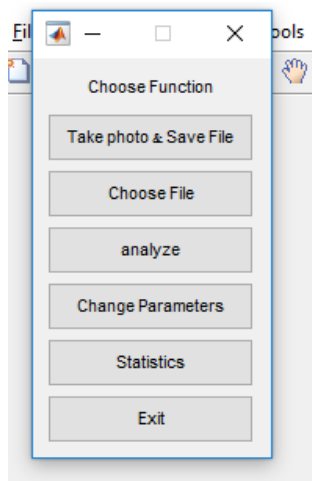
## 2. דרישה עסקית - הגדלת כמות הפירות הנבדקים לדקה

על מנת לענות הדרישה העסקית הוגדרו 2 דרישות פיתוח נוספות: ייעול אלגוריתם הבדיקה, ייצור עמדה ייעודית לבדיקה. ייעול אלגוריתם הבדיקה – נבנה אלגוריתם בדיקה בסיסי המבצע את המשימה. האלגוריתם אכן בר ייעול וניתן להוסיף עוד פרמטרי בדיקה והקטנת טווחי הבדיקה לדיוק מירבי וזהו יעד עתידי. ייצור עמדה ייעודית לבדיקה – הוגדרה בנייה של עמדה ייעודית פשוטה הכוללת תאורה בסיסית ורקע לבן מלא על מנת שהמוצר יעבוד בצורה יעילה ותקינה בהפרדת הפרי מהרקע של התמונה. העמדה הייעודית תכלול כמו כן את המצלמה אשר מצלמת את הפרי, לכן איכות המצלמה גם משפיע על איכות תמונה ואיכות הבדיקה.

## 3. דרישה עסקית - הפקדת דוחות סטטיסטיים

דרישות פיתוח שהוגדרה על מנת לענות דרישה עסקית זו : שמירת נתונים מכל בדיקה, יצירת דוחות על פי פרמטרים ותבנית מוגדרים מראש. שמירת נתונים מכל בדיקה - לאחר כל בדיקה נשמר במאגר מידע (הממומש על ידי טבלת אקסל) הפרמטרים הבאים: כמות הבדיקות המבוצעות לפרי, בדיקה עברה, בדיקה נכשלה. יצירת דוחות על פי פרמטרים ותבנית מוגדרים מראש – התבנית שנבחרה הינו גרף המוצג למשתמש כאשר הגרף מוגדר שציר ה X שמות הפרי ועל ציר ה Y הנתונים בהתאם.





4. דרישה עסקית - קיצור זמן הכשרת עובדים  
 על מנת לקצר את ההכשרה של העובדים נדרש לבנות ממשק נוח ברור וקל למשתמש (דרישת פיתוח).  
 המוצר עונה על כך על ידי הצגת תפריטים קצרים וברורים למשתמש המנחים אותו להמשך בדיקת התהליך.  
 כל תהליך בדיקה הינו קצר ופרוצדורלי כך שקל להסביר ולהדגים למשתמש.

5. דרישה עסקית - הקטנת עלות החומרה הדרושה למערכת  
 על מנת להקטין עלות החומרה נוכל לצמצם עלויות בתחום המצלמה והעמדה ייעודית.

בתחום הצילום - המצלמה המתממשקת למוצר משפיעה על איכות התמונה המתקבלת ובצורה ישירה משפיעה על תוצאת הבדיקה.  
 עמדה ייעודית - בתצורה הקיימת של המוצר נדרשת עמדה ייעודית בעלת רקע לבן המאפשרת סינון פשוט ומהיר של הפרי מהרקע של התמונה.  
 על ידי שיפור אלגוריתם הבדיקה יתאפשר בעתיד לבצע סינון רקע תוכנתי ובכך לחסוך את הרכבת העמדה הייעודית (שמלכתחילה העלות שלה נמוכה לכן נדרש גם לבדוק עלות/תועלת).

#### • בדיקות יחידה

בוצעו מספר בדיקות יחידה לפונקציות בודדות:

- קליטת תמונה למשתנה.
- קליטה מטריצה מקובץ Excel.
- לטעון מטריצה לתוך קובץ Excel.
- בדיקת ערך מוחזר מפונקצייה האחראית על הצגת תוצאה.

## • שאלון אינטגרציה

1. הבטחת איכות בפיתוח תוכנה מתבטאת בדאגה להבטיח שרמת הדרישות של איכות התוכנה תושג במוצר.  
בהבטחת איכות תוכנה נתמקד ב-3 תחומים חשובים:
  - ברמה הארגונית, הבטחת איכות דואגת להקמת מסגרת המיועדת לתהליך הארגוני ולתקנים אשר יובילו בסוף לתוכנה באיכות גבוהה.
  - ברמת הפרויקט, הבטחת איכות כוללת השמת תהליכי איכות מסוימים ובדיקה שהתהליכים המתוכננים האלו מיושמים כהלכה.
  - בנוסף, ברמת הפרויקט, הבטחת האיכות דואגת להקמת תכנית איכות עבור הפרויקט. התוכנית צריכה לקבוע את מטרות האיכות עבור הפרויקט ולהגדיר מהם התהליכים והסטנדרטים שנשתמש בהם.

בין תהליכי הבטחת האיכות:

- תכנון של תהליכי בקרת איכות
  - בדיקת התאמת התוכנה לדרישות
  - עבודה לפי סטנדרטים
  - סקירות של התוכנה
  - בקרה הדדית של איכות התוכנה בזמן הפיתוח
  - בדיקת מדדים לאיכות התוכנה
- בפרויקט שלנו יישמנו את התהליכים הבאים:
- התכנון של תהליכי בקרת האיכות, כאשר בתחילתו קבענו כיצד נעקוב אחרי פיתוח הקוד כך שיהיה בעל מינימום תקלות.
  - בדיקת התאמת התוכנה לדרישות, כאשר עבדנו בצמוד למסמכי התכנון ועקבנו אחרי תהליך בניית התוכנה כך שתהיה מתאימה לדרישות העסקיות שקבענו.
  - סקירות של התוכנה, כאשר עברנו על התוכנה בכל שלב כדי למצוא תקלות אפשריות ולפתור אותן בזמן אמת.

## 2. שני מדדים לאיכות התוכנה:

- כמות שורות הקוד.  
זהו אחד המדדים הראשונים שהתפתחו לצורך בקרת איכות על הקוד. המדד מתחלק לשני סוגים – שורות קוד פיזיות (כל פעם שהמתכנת יורד שורה בקוד) ושורות קוד לוגיות. למדד של שורות הקוד הפיזיות יש מספר יתרונות, ביניהם קלות המדידה והיכולת לבצע בקרה אוטומטית אחרי כמות שורות הקוד, אך גם מספר חסרונות כמו האפשרות להימצאות של הערות וירידת שורות מיותרת שמאריכים את הקוד והימצאות של "קוד מת" (קוד שמתבצע אך אין שימוש בתוצאה שלו) שפוגעים באמינות המדד.
- ניקוד הפונקציונליות של הקוד.  
מדד זה בודק את הקלטים, הפלטים והקבצים שמשותפים בתקשורת בין האלמנטים של הקוד. השיטה מפרטת את יחידות הקוד שמתקשרות ביניהן ומנקד את הקוד ביחס למדדים מוגדרים של יעילות הקוד לפי סטנדרטים ידועים. בין יתרונות המדד הזה נמצאים יכולת ביצוע המדידה ללא תלות בשפת התכנות, היכולת למדוד פעולות כמו דוקומנטציה, שאינן קשורות לכתיבת הקוד עצמו והאפשרות להערכת עלות התוכנה. החסרונות של השיטה כוללים דרישה לניסיון רב בשימוש בשיטה זו לצורך ביצועה היעיל, אי-וודאות לגבי דיוק המדידה בקטע קוד נתון ועלות גבוהה (הן מבחינת זמן והן מבחינה כספית) של ביצוע הבדיקה.

נלקח מהמאמר:

Survey on Impact of Software Metrics on Software Quality, Mrinal Singh Rawat *et al.*, IJASCA vol.3, No.1, 2012

3. מערכות סוציוטכניות הן מערכות מורכבות שמערבות את האדם או הארגון באופן הפעולה שלהן. מערכות כאלה משתנות לפי דרישות הארגון, מותאמות לשינויים תוך-ארגוניים ולרוב בעלות מאפיינים כמו אמינות, אבטחה ושמישות גבוהים הנובעים מצורכי המשתמשים בה. ההבדל בין מערכות כאלה למערכות מבוססות מחשב, לפי מודל השכבות, הוא המחסור של 3 השכבות העליונות של STS במבנה של מע' מבוססות מחשב. הבסיס – ציוד/מערכת, מערכת הפעלה, תקשורת וניהול מידע, מערכת התוכנות – זהה בין שני הסוגים, אך במערכת מבוססת מחשב אין את שכבות התהליכים העסקיים, הארגון והחברה שמשפיעים על התנהגות ומבנה המערכת. דוגמה למערכת מבוססת מחשב: מערכת לניטור טמפרטורה בחדר שרתים. דוגמה למערכת סוציוטכנית: מערכת לניהול בקשות סטודנט ומענה של הבקשות האלה.

4. מקרה א': המשתמש לא בחר את הפרי הנכון והאפיון התבצע לפי קריטריונים שגויים. הדו"ח יתעדכן עם תוצאות שגיאות של בדיקה. דרך התמודדות – בנייה עתידית של מערכת זיהוי פירות אוטומטית שתצמצם את מרווח הטעות. מקרה ב': המשתמש לא הזין מאפייני בדיקה אישיים ובדק את הפרי לפי מאפיינים חסרים אלה. דרך התמודדות – הגדרת ערכים אישיים דיפולטיביים אם לא הוגדרו ערכים אישיים אחרים והצגת הודעת שגיאה אם נערכה בדיקה לפי ערכים לא מוגדרים כאלה. מקרה ג': המשתמש מחק את כל המידע הסטטיסטי מהמערכת בלי שהתכוון כי הוא הצליח להגיע אליהם. דרך התמודדות – הגדרת סיסמא לכל חלק בתוכנה שניתן לשנות או למחוק בו הגדרות אישיות.

5. תכונה לא דטרמיניסטית היא תכונה שתלויה במשתמש, כלומר שהפלט תלוי לא רק בתהליכים הפנים-מערכתיים אלא גם בפעולות המשתמש. בפרויקט שלנו הדוגמה לכך היא תהליך ניתוח הפרי עצמו. גם אם הצילום, העלאת התמונה, הניתוח לפי נתונים של פרי מסוים ואיסוף המידע הסטטיסטי מתבצעים בצורה נכונה, המשתמש יכול לבחור פרי לא נכון ואז, למשל, עגבנייה שצולמה תהיה מאופיינת לפי פרמטרי בשלות של בננה – דבר אשר יוביל לתוצאת בדיקה שאינה נכונה.

