



**Ben Gurion University**

# **Automated multi-choice test system**

Ilan Git (206904336)<sup>1</sup>  
Lee-or Alon (209329796)<sup>2</sup>

---

## **Work Report**

March 07 ,2020

---

<sup>1</sup>Student at Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

<sup>2</sup>Student at Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

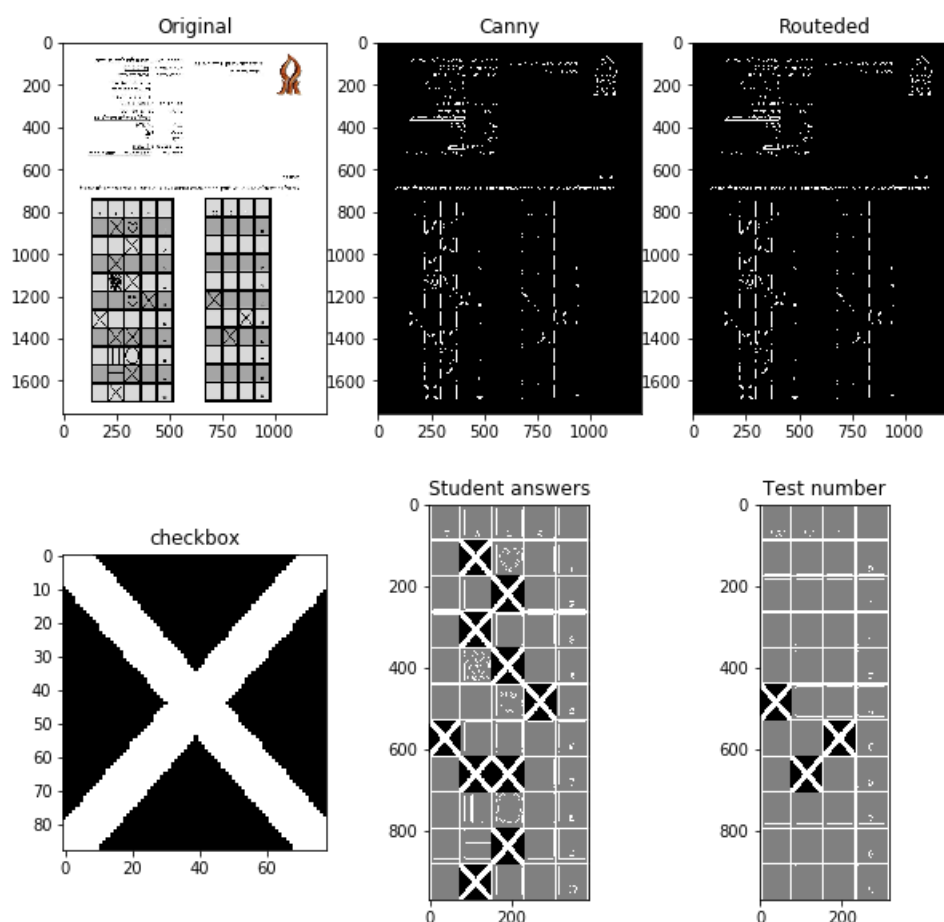
# תוכן עניינים

3	מבוא
4	אלגוריתם הראייה - שלבי התוכנית
6	דרישות התכנית
7	סיכום
8	סימוכין

## מבוא

כיום, באוניברסיטת בן גוריון מבחנים רבי-ברירה (מבחנים אמריקאים) נבדקים אוטומטית על-ידי חברי הסגל. בדיקה זו גוזלת זמן ומיותרת שכן כיום ישנם הכלים הדרושים להפיכת הבדיקה הנ"ל לאוטומטית. המערכת מושתתת על אלגוריתמים שונים הנלמדו בקורס ושילובם איפשר את הצלחת המערכת. בפרויקט זה נעשה שימוש באלגוריתמים לזיהוי צורות, סיבוב תמונה תוך שימוש בזיהוי אובייקטים, חילוף מידע מהתמונה ושמירתו בקובץ נפרד.

הסטודנטים מסמנים בדפי התשובות את התשובה שלדעתם נכונה באמצעות X במשבצת המתאימה. המערכת מנתחת את תשובות הסטודנט, נותנת ציון על סמך תשובות אלה, מעבירה את תשובות הסטודנט ואת ציונו לקובץ אקסל ולבסוף מייצרת היסטוגרמת ציונים של כלל הסטודנטים הנבחנים. חבר הסגל מצידו נדרש רק לספק למערכת את התשובות הנכונות עבור המבחן. במהלך הפרויקט הייתה התנסות מרובה בבניית מערכת ראייה וחיבור של האלגוריתמים השונים לצורך מערכת אחת תוך יישום הידע הנרכש מהקורס.



**תמונה 1 - דוגמא לתהליך המערכת (משמאל ולמעלה):** 1. טופס הבחינה המקורי. 2. תמונת גובלות  
 3. סיבוב הטופס. 4. kernel למציאת תשובות הסטודנטים. 5. זיהוי פתרונות וסימון המערכת  
 6. זיהוי מספר בחינה של הסטודנט

# אלגוריתם הראייה - שלבי התוכנית

## 1. סיבוב הדף

יתכנו מספר מקרים בהם קלט במערכת יקבל דף "שצולם עקום", למשל עקב הדפסה לא תקינה, קימוט הדף, סריקה לא ישרה של טופס וכו'.

לא ניתן להשתמש באלגוריתמים למציאת קווים ישרים בלבד, מכיוון שאז נאסור בטפסי הבחינה שימוש בקווים למעט הטבלאות. שורות מלל בתחילת טופס הבחינה יכולות להיחשב לקווים ישרים. לכן בתחילת הריצה, התוכנית מנסה לאתר את זוויות סיבוב טופס הבחינה (בשימוש באלגוריתם Hough transform) וזאת על ידי קונבולוציה של מלבנים מסובבים מול תמונה גבולות (באמצעות Canny edge detector). התוכנית בוחרת כל פעם זוויות לבדיקה באופן איטרטיבי בין 0 ל- $\pi$  (סיבוב של 180 מעלות של מלבן יחזיר לצורתו המקורית).

התוכנית יוצרת Kernel של מלבן מסובב ובודקת את מידת התאמה לפי נרמול התוצאות להיקף המלבן. במידה ונמצאה התאמה של שני המלבנים באחוז מספק (אחוז נמוך יכול להעיד על מציאה לא תקינה או שהטבלה "נחתכה בצילום" וחלק מהפתרונות "הוסרו") נוכל לדעת כיצד לסובב את התמונה (מפני שמלבן הינו סימטרי לסיבוב של 180 מעלות, שני מלבנים (אחד ליד השני) יתן לנו את המידע האם התמונה הפוכה או מסובבת או גם וגם). לאחר הסיבוב התוכנית תוכל להוציא במייד את מיקומן של הטבלאות ולהמשיך לשלבים הבאים.

## 2. חלוקת הטבלאות וזיהוי סימונים

לאחר שהתוכנית מצאה את מיקומן של הטבלאות, המערכת תחלץ כל טבלה ותחלק אותה לאוסף משבצות (תאים). מכיוון שהטבלאות יכולות להיקלט בעובי גבולות שונים, החלוקה לא תהיה מדויקת בתוך כל טבלה, אך עדיין ניתן לחשב בצורה מדויקת מה סימן הסטודנט בכל משבצת. על כל משבצת נבצע קונבולוציה עם Kernel בעל ערכים חיוביים ושיליים בצורת איקס. כל הערכים על גבי האיקס יהיו חיוביים בעוד הרקע יהיה בעל ערכים שליליים. בצורה זו, כל סימון שאינו דומה לצורה של איקס או "מחיקת סימון" לא יזכו לערכים גבוהים וכך התוכנית תדע היכן הנבחן סימן את תשובתו והיכן ביצע סימונים לא רלוונטים או לא חוקיים.

במידה והסטודנט סימן לא לפי הכללים שהוגדרו מראש, התוכנית תשמור במערך פתרונות של הנבחן הערת מערכת ותמשיך לבדוק את שאר השאלות.

## 3. סיכום הנתונים לאקסל

עבור כל ריצה של טופס הבחינה המערכת מסכמת את אוסף הפתרונות של הסטודנט ושומרת את המידע אודות מספר הנבחן, אוסף הפתרונות שלו וציונו הסופי לקובץ אקסל. בנוסף התוכנית פולטת יחד עם טבלת התשובות של כל הסטודנטים את טבלת פילוג ציונים לשימוש המרצה.

# דרישות התכנית

על מנת להריץ את תוכנית זו נדרשים מספר פרמטרים שעל המשתמש לספק:

1. קובץ PDF של טפסי התשובות
2. רשימת תשובות נכונות וערכי הניקוד
3. גודל הטבלאות (בפיקסלים) - לצורך זיהוי מיקומן בטופס הבחינה
4. מספר שורות ועמודות - על מנת לדעת כיצד לקרוא את הטבלאות בבחינה
5. אחוז הצלחה - עד כמה המשתמש מוכן לקבל צילום מבחן שדף התשובות נחתך חלקית (אופציונלי)

יתר על כן, התקיימו מספר הנחות למערכת:

1. הסריקות מבוצעות על ידי סורק מסוים ולכן כל הסריקות יצאו תמיד באותו Scaling ולכן המשתמש לא צריך להגדיר זאת מחדש בכל ריצה למעט התקנה ראשונית
2. סטודנט רשאי להתחרט על סימון תשובתו על ידי השחרת המשבצת המתאימה (לא חייב בצורה מלאה) וסימון תשובה אחרת - אך לא ניתן "לבטל" את מחיקת התשובה לאחר השחרתה (במידה ומותר להשתמש בטיפקס ניתן לצייר על הריבוע המושחר איקס לבן והמערכת תקלוט זאת)
3. כל סימון שאינו בצורת איקס - המערכת לא תקלוט ולא תתייחס
4. קיימת תשובה נכונה אחת בלבד עבור כל שאלה

## סיכום

בפרויקט זה בנינו מערכת אוטומטית לבדיקת מבחנים רבי-ברירה. במערכת נעשה שימוש נרחב באלגוריתמים שנלמדו בקורס כמו גם שימוש באלגוריתמים נוספים. המערכת חוסכת זמן יקר בבדיקת מבחנים רבי-ברירה תוך הנחות הגיוניות הנעשות כבר היום במבחנים אמריקאים, למשל סימון התשובות הנכונות ב-X וקיומה של תשובה נכונה אחת עבור כל שאלה. תחילה המערכת מזהה את זווית סיבוב התמונה ואת מיקום הטבלאות. לאחר מכן המערכת מזהה את תשובות הסטודנט ומחלצת אותן מהטבלה. לבסוף המערכת מנקדת את תשובות הסטודנט ואף יוצרת היסטוגרמת ציונים של כלל הנבחנים תוך שמירת נתונים אלה בקובץ. אין ספק כי הטמעת המערכת בבדיקת מבחנים אמריקאים תחסוך זמן יקר בבדיקת המבחנים מחד, ומאידך תאפשר לסטודנטים לדעת את ציונם בקורס בזמן מהיר יותר.

## סימוכין

1. Canny, John. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986): 679-698.
2. Illingworth, John, and Josef Kittler. "A survey of the Hough transform." *Computer vision, graphics, and image processing* 44.1 (1988): 87-116.
3. XlsxWriter - Python module for creating Excel XLSX files  
<https://xlsxwriter.readthedocs.io/>
4. Pdf2image - Python module that wraps the pdftoppm utility to convert PDF to PIL Image object  
<https://github.com/Belval/pdf2image>
5. Wix.com - cloud-based web development services  
[www.wix.com](http://www.wix.com)