ראיה ממוחשבת – דו"ח תרגיל בית 1:

מגישים:

איתי גיא, ת.ז. – 305104184 , אליאס ג'דעון, ת.ז. - 207755737

בתרגיל זה למדנו בצורה מעמיקה כיצד לעבוד עם python בתחום העיבוד תמונה ועם ספריית OpenCV בפרט. השתמשנו באלגוריתם grabcut הבינארי שנלמד בהרצאה והרחבנו אותו ל-4 סגמנטים באמצעות multivoting באופן הבא:

1. ביצוע grabcut במתכון של 1-against-all כאשר 1 מהסגמנטים המסומנים הוא ה-foreground והיתר הם background מהסוגים הבאים: הפיקסלים שסומנו איטראקטיבית הם בודאות background והיתר probable_background
   - תהליך זה מתבצע לכל סגמנט מסומן כ-7 פעמים עבור כל הזוגות ופעם נוספת למסכה המשלימה

אלגוריתם זה מתבצע בפונקציה: calc_grabcut_combinations

2. לאחר שיש לנו 4 מסכות ראשוניות של האזורים הודאיים שזיהינו לכל סגמנט נותר להכריע את השיוך לסגמנט של הפיקסלים בעלי קונפליקטים עם יתר המסכות. נפתור את הבעיה באמצעות multi-voting בני המסכות שהתקבלנו משלב 1

אלגוריתם זה מתבצע בפונקציה: calc_segmenting_isolation

3. הוספת פיקסלים שהוכרעו למסכות שהתקבלנו משלב 1 ,הרחבת כל מסכה ל-3 ערוצים עם הצבע המתאים ,הצגת התוצאות הנדרשות למסך ושמירתם בפורמט המתאים לתיקיית results

חוזקות:

- האזורים הכלליים של כל סגמנט מסומנים היטב
- כל אזור בתמונת הסגמנטים יקבל צבע כלשהו מבין הצבעים הנתונים (לרוב צבע הגיוני בעל קירוב טוב)
- מוצא תבניות לפעמים – כגון החביות ב-china.jpg – מונע סימונים נוספים של הסגמנט (יותר אוטומציה)

חולשות:

- מציאת תבניות של grubcut בינארי יכול להניב מציאת חפיפות בין סגמנטים כתלות במורכבות התמונה ולכן דורש הערכה לכל פיקסל בחפיפה כתלות בסימן של המשתמש. פתרון אפשרי – multivoting
- זמן ריצה גבוה

- הקוד תומך ב-2 עד 4 סגמנטים מסומנים
- תיעוד מלא למתכנת נמצא בתוך הקוד
- התוצאות ישמרו בתוך תיקיית results תחת השמות שנבחרו בראש הסקריפט של המטלה Multi_Segmentation.py
- ישנם דוגמאות הרצה בתוך תיקיית my_examples שניתן לראות גם בהמשך הדו"ח
- בכדי לקבל פלט נוסף לתוך my_examples בצורה של 4 תמונות רצופות שמציגות את התהליך ניתן להוריד את ההערה משורת הקוד save_seq_images בתחתית הפונקציה main_loop

צילומי מסך: