

## פרויקט מסכם בקורס ראייה ממוחשבת

סטודנטים יקרים,

- מסמך זה מכיל את הדרישות לפרויקט המסכם.
- יש לבצע את הפרויקט בשלוש או רביעיות, לא תאושר הגשה ביחידים.
- כל סטודנט/ית יעלו קובץ txt בלבד לתיבת ההגשה האישית במודל.
- הקובץ יכול הכולל קישור לתיקיה משותפת לבחירתם (Google Drive/OneDrive/GitHub) עם קוד המקור, כל התלויות הדרושות להרצה תקינה של הפרויקט וסרטון קצר אשר מראה ריצה תקינה של התוכנית.
- שם הקובץ יכול את שם הסטודנט/ית המגישים ולאחר מכן ת.ז של השותפים מופרדים באמצעות פסיקים.
- תאריך אחרון להגשה: **18.8.22**.
- במסגרת החומר שנלמד לאורך הסמסטר, כתבנו בין היתר שני פרויקטים אשר יסייעו לכם במילוי הדרישות ויחסכו זמן רב:  
<https://drive.google.com/drive/folders/1KiLAB7Hr7a3BUUpUN8Ctcl70T-oU9jHh4?usp=sharing>
- אני מאפשר לכם לקבוע את ה-API ברכיבים החדשים שתוסיפו (בפרט חתימות מתודות, שמות ממשקים/מחלקות חדשות).
- יש לתעד את הקוד באנגלית בלבד באופן תמציתי כולל פרמטרים.
- התמונות והסרטונים בהם תשתמשו בפרויקט נתונים לבחירתכם.

### **תמצית:**

עליכם לממש את המחלקה DigitalImaging אשר מספקת שירותים מגוונים בתחום הראייה הממוחשבת, בפרט עיבוד תמונות, ניתוח והצגת נתונים.

הפרויקט יעשה שימוש באלגוריתמיקה מתקדמת ובספריות צד-שלישי כפי שביצענו לאורך הסמסטר. אתם רשאים להיעזר בקוד משני הפרויקטים שכתבנו בכיתה, כולל שינוי והרחבה. שימו לב, יש להגדיר את המחלקה כ-**dataclass** ולהצהיר על data-members אשר ישמשו אתכם לפעילות התקינה. עבור אתחול של טיפוסים מורכבים יש להשתמש ב-defaultfactory כפי שלמדנו בכיתה.

להלן המתודות שיש לממש:

#### **convert\_to\_gs**

המתודה תקבל מחרוזת המציינת את הנתבי לקובץ תמונה (רצוי בפורמט jpg/jpeg), תייצר באמצעות הפונקציונאליות של הספרייה The Python Imaging Library (PIL) אובייקט תמונה ב-grey scale. יש להדפיס את ה-mode של התמונה ולהחזיר את אובייקט התמונה שנוצר.

#### **color\_at**

המתודה תקבל Numpy Array אשר מייצג תמונה, מספר שורה ומספר עמודה. במידה ודגל ה-writeable הינו True, יש להחזיר tuple שמכיל את ערכי הצבעים על-פי פורמט RGB בקואורדינטה הנתונה בקלט.

#### **reduce\_to**

המתודה תקבל מחרוזת המציינת את הנתבי לקובץ תמונה (רצוי בפורמט jpg/jpeg) ואות (גדולה או קטנה) אחת המייצגת אחד מערוצי הצבע RGB. במידה והאות תקינה, המתודה תחזיר את אובייקט תמונה בצבע הנבחר (חשבו מה קורה לשני הערוצים האחרים).

#### **make\_collage**

המתודה תקבל רשימה המכילה אובייקטים של תמונות שנוצרו באמצעות הפונקציונאליות של PIL. יש להחזיר Numpy array של שרשרת התמונות באופן **אנכי** כך ש-3 התמונות הראשונות מוצגות בערוץ אדום בלבד, 3 הבאות בערוץ ירוק ו-3 התמונות לאחר מכן בצבע כחול. התבנית תחזור על עצמה בהתאם למספר התמונות הנתון.

#### **shapes\_dict**

המתודה תקבל רשימה של אובייקטים של תמונות שנוצרו באמצעות הפונקציונאליות של PIL.

יש למפות כל תמונה ל-tuple שמייצג את ממדיה ולהחזיר את מילון ממיון לפי מספר השורות.

### **detect\_obj**

המתודה תקבל מחרוזת המציינת את הנתיב לקובץ תמונה (רצוי בפורמט jpg/jpeg) ומחרוזת נוספת המייצגת אזור בגוף שצריך לזהות בתמונה: Face או Eyes (ניתן לקבל אותיות גדולות/קטנות/שילוב).  
כפי שלמדנו, באלגוריתם Viola-Jones ניתן לבצע זיהוי של אלמנטים על-פי Haar Features.  
יש להשתמש במודול cv2 (ספריית OpenCV) תוך שימוש ב-classifiers המאומנים הבאים:

- frontalface\_default

- haarcascade\_eye.xml

ה-classifiers זמינים בכתובת הבאה: <https://github.com/opencv/opencv/tree/master/data/haarcascades>  
המתודה תחזיר העתק של התמונה עם מלבנים ירוקים/שחורים סביב האזור הרצוי שזוהה בתמונה.  
במידה ולא קיים, יש להדפיס הודעה שלא נמצא אזור מתאים בתמונה הנתונה.

### **detect\_obj\_adv**

בדומה למתודה הקודמת, יש לקבל כקלט מחרוזת המציינת את הנתיב לקובץ תמונה (רצוי בפורמט jpg/jpeg) ושני ערכים מסוג Boolean: הראשון עבור face והשני עבור eyes.  
יש להחזיר העתק של התמונה עם מלבנים ירוקים/שחורים סביב האזורים הרצויים.

### **detect\_face\_in\_vid**

עליכם לקבל נתיב לקובץ וידאו ובאמצעות המתודה VideoCapture לעבור על ה-frames שנוצרו מהסרטון.  
על המתודה להפעיל את ה-classifier הרלוונטי ולהציג את הפרצופים שזוהו באמצעות המתודה imshow (מלבנים ירוקים/שחורים).