**דו"ח מיני פרויקט בעיבוד תמונה:**

**מגישים:**

חנן בוסקילה, ת"ז: 208375543

ניב רביב, ת"ז: 312414295

יניב קרול, ת"ז: 208386219

הפרויקט שעשינו הוא בנושא של דחיסה וצמצום של קטעי וידיאו.

הפרויקט מקבל כקלט מהמשתמש סרטון, בחירה של פונקציית מרחק (מתוך סט מוגדר של פונקציות שמוגדרות במודול מיוחד בספריית הקוד, עליהן נרחיב בהמשך) ומספר רמות במבנה הנתונים שנרחיב עליו בהמשך. בשלב הזה התוכנה מכילה מבנה נתונים דמוי עץ אשר מרמה לרמה מצטמצמים הפריימים הכי פחות משמעותיים. לבסוף המשתמש יכול לבחור כמה פריימים הוא רוצה שהיה בסרטון , המערכת תמצא ב o(1) במבנה הנתונים את הרמה הכי קרובה לבקשה שלו ותחזיר את הסרטון לאחר צמצום.

**שיטות בדיקת המרחק**

1. Euclidean – פונקציה המודדת מרחק אוקלידי בין כל שני פריימים ומורידה את הפריימים בעלי המרחק הקטן ביותר.
2. Motion vector- מציאת הmotion vector בין כל שני פריימים, זהו בעצם וקטור דו ממדי שבודק מרחק בין כל קואורדינטה בפריימים (כלומר לאיפה כל חלק בתמונה זז בין הפריימים) ולפי זה מוצאת את הפריימים הכי פחות משמעותיים ומורידה אותם. כלומר בשיטה זו אנו בוחנים בין כל 2 פריימים סמוכים לאן התמונה זזה , ומורידים את אלו שהתזוזה בהם הייתה קטנה יותר.   
   כמובן ששימוש בשיטה זו הרבה יותר איטי מהשיטה הראשונה

ניתן להגדיר פונקציות מרחק נוספות על ידי הוספת פונקציות מהצורה def func(f1, f2) למודול “distfunctions.py”

**מבנה הנתונים לשמירת העדכונים:**

מבנה הנתונים לשמירה על העדכונים נבנה כך:

אנו מקבלים מהמשתמש כקלט כמה רמות אנחנו רוצים במבנה הנתונים , בכל רמה אנו מחשבים מרחק סף שכל פריים שקטן ממנו נוריד.  
כל רמה לאחר צמצום הפריימים המתואר למטה אנו שומרים מצביעים לפריימים הנותרים .  
בשיטה זו אנו מייצירם מבנה זיכרון אשר נותן לנו גישה מהירה לבניית הסרטון המבוקש ללא עלות רבה של זיכרון ושל חישוב נוסף בעת הבקשה.

**צמצום הפריימים:**

צמצום הפריימים מתבסס על מבנה נתונים של ערימת מינימום אשר כל אחד מהיחידות בערימה מצביעה על ערימה מקושרת אשר מייצגת לנו את הסרטון שאנו רוצים לצמצמם. הערימה בנויה על בסיס פונקציית המרחק.  
הצמצום מתבצע כך שאנו שולפים מערימה את הראש (האיבר עם המרחק המינימלי) ואנחנו מעדכנים את החולייה שלפניו ברשימה המקושרת כך שתצביע על החולייה שהוא הצביע

**לשקול להוסיף סרטוט**

**כיצד להריץ את התוכנית:**

1. להריץ את תוכנית הפייתון.
2. להזין מיקום של סרטון
3. להזין את מספר הרמות שרוצים במבנה הנתונים
4. להזין את המספר של פונקציית המרחק שבחרנו
5. להזין את השם המבוקש לסרטון התוצאה, או q בשביל לצאת
6. להזין את מספר הפריימים שרוצים (מספר הפריימים יבחר לפי מספר הפריימים הקרוב ביותר שזמין בעץ הפריימים)

דוגמת הרצה:

$ python main.py

Please enter video path:

sample\_videos/Light nature(5 second)-(144p).mp4

Please enter wanted number of levels for the frames tree: 4

Please enter the number of the wanted distance function:

1 - euclidean

2 - motion\_vector

1

Loading video...

Building frames tree...

Available frame levels: [139, 116, 56, 31]

Please enter the name for the output video, or 'q' to quit:

my\_video

Please enter the number of wanted frames:

72

Created video my\_video.mp4 with 56 frames

Please enter the name for the output video, or 'q' to quit:

q

$