

# עיבוד תמונה – תרגיל בית 1

אוריין חסידים | 31931579 | oryan.hassidim

## 1. מבוא

בתרגיל זה נישם שימוש בהיסטוגרמה של רמות בהירות של תמונות כפי שלמדו בהרצאה 1 כדי להזוהה מעבר בין סצנות.



frame 0

frame 1

frame 2

frame 3

frame 4

frame 5

באופן המתבקש מההרצאה הנ"ל, נשטמש בשתי גישות – נורמת ההפרש בין ההיסטוגרמות ונורמת ההפרש בין הסכומים המוצטברים של ההיסטוגרמות. כפי שהיה ניתן לנחש עוד לפני שראויים את הסרטונים – זהו הבדל שנדרש במשמעות בין שתי הקטגוריות...

## 2. אלגוריתם

א'. טעינה של הויידאו frame by frame

ב'. עברור כל פריים:

a. עיבוד לגוני אפור

b. חישוב ההיסטוגרמה

c. עברור קטgorיה 2 בלבד – חישוב הסכומים המוצטברים

ג'. חישוב נורמת ההפרש בין כל שני וקטורים עוקבים

ד'. מציאת האינדקס בעל ההפרש המקסימלי

## 3. פרטי מימוש

השתמשתי בספרייה [OpenCV](#) כדי לקרוא מקובץ הוידאו – שזו הייתה המלצת בראש וacen נראה שסיפקה לי את הפונקציונליות הנדרשת בקורס פשוט וקצר, ומוכן בספרייה [NumPy](#).

מהאפקט התכנוני, השתמשתי בכלים מעולם ה-Functional Programming כדי למש את רצף הפעולות שפורטו לעיל באלגוריתם Pipeline.

א'. טענתי כיitrטור את הפריים של התמונה (ככה שלא העמsti יותר מדי על הזיכרון, שהיה צריך להחזיק כל פעם פריים אחד בלבד בזיכרון ולא את כולם יחד)

ב'. את האיטרטור הנ"ל מיפוי (map) לפונקציית המרת לגוני אפור מ-BGR (הפורמט התמונה של OpenCV) שימפה פריים פרים. השתמשתי בפונקציה המובנית של [OpenCV](#).

ג'. מיפוי להיסטוגרמה באמצעות [NumPY](#).ד'. עברור קטgorיה 2 – מיפוי לפונקציית cumsum של [NumPY](#).

ה'. שימוש בפונקציה המובנית pairwise iterator מモודול [itertools](#) (מבנה, החל מגרסה 3.10) – כדי לקבל כל פעם זוג היסטוגרמות לפונקציה הבאה.

ו'. מיפוי כל זוג להפרש הנורמות. שוב [NumPY](#).

ד. מיציאת argmax – פונקציה שמיימשתי בעצמי. לא השתמשתי בפונקציה המובנית של NumPY כי היא עובדת על מערכיים שישובים בזכרון אך אני רציתי לעבוד על איטרטור ולא שהכל יהיה בזכרון במקביל.

ה. המרה מאינדקס בודד לתצורת tuple של האינדקס והאינדקס העוקב.

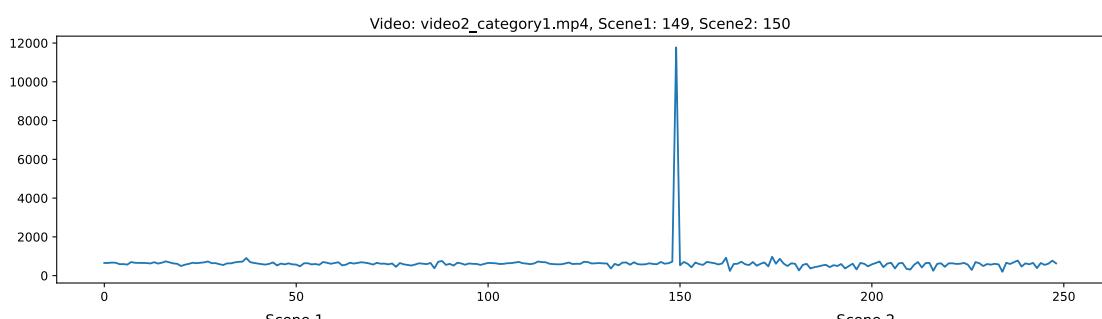
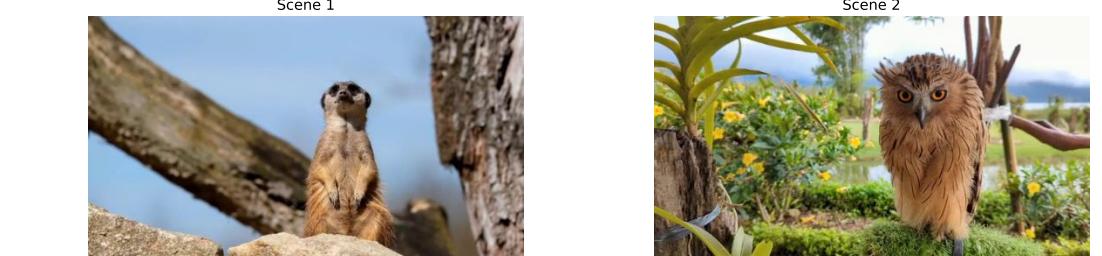
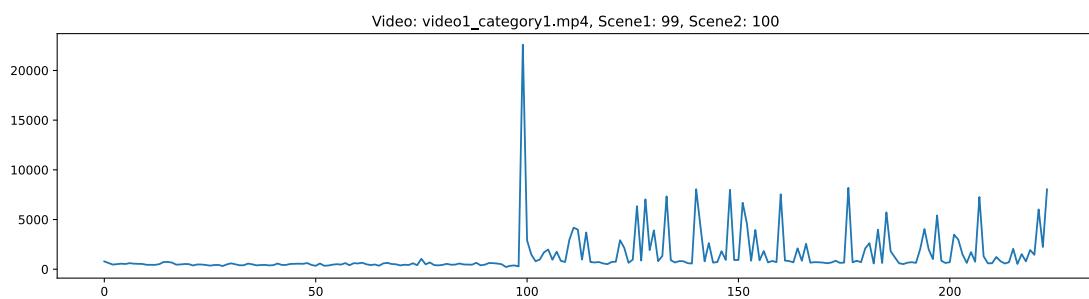
אחרי כתיבה של הקוד באופן ישיר הקוד עבר אך לא החזיר את התשובה המצופה (הצגתי את שני הpriymים אחד ליד השני וראיתי שאין מעבר סכינה). לאחר כמה דקות הסתבר שימוש מה בדיעבד בטעות השימוש בפונקציה `min` במקום `max` 😞 . חווין מזה כל הקוד עבר חלך ישר מהפעם הראשונה.

אתגר מעניין שהיה הוא למשר piped python-ב-piping, שהוא לא שפה טבعت לה וזה קונספט מעולם ה-FP... כתבתי מימוש עצמי שענה לי על הדרישות ועשה עבודה נחרצת.

הפרויקט נמצא ב-[github](#) שלו.

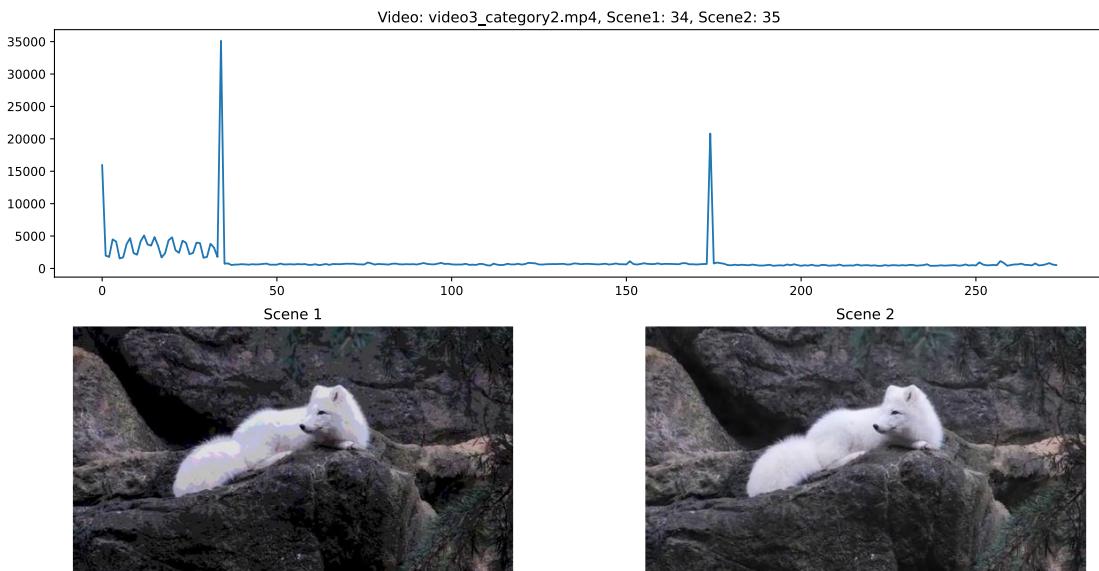
## 4. תוצאות קטגוריה 1

עבור כל וידאו: בគורת שם הוידאו, והpriymים בהם היה החיתוך. מתחתיה גורף הנורמות של הפרשי ההיסטוגרמות, ובתחתית הpriymים עצם.

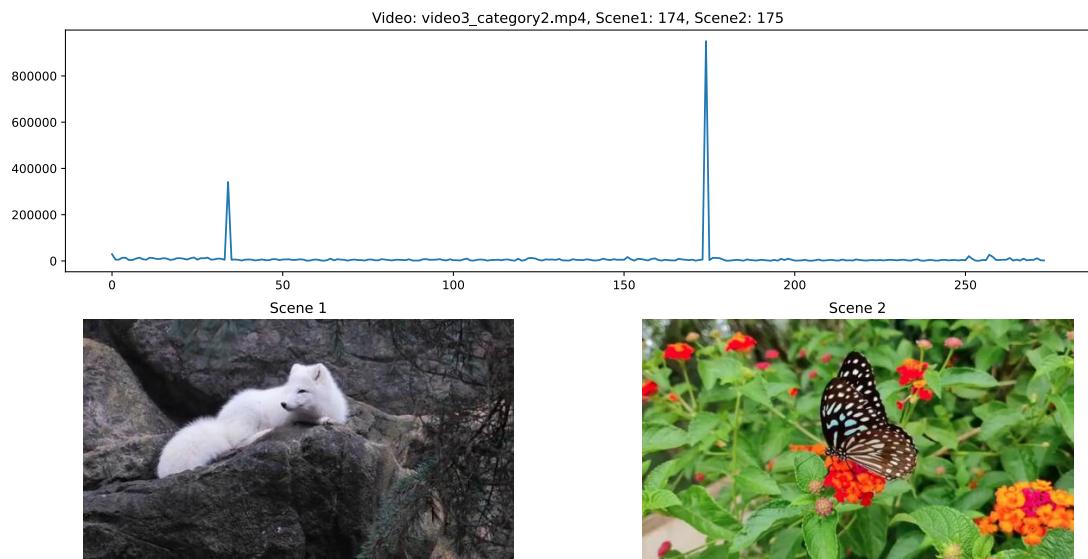


## 5. תוצאות קטגוריה 2

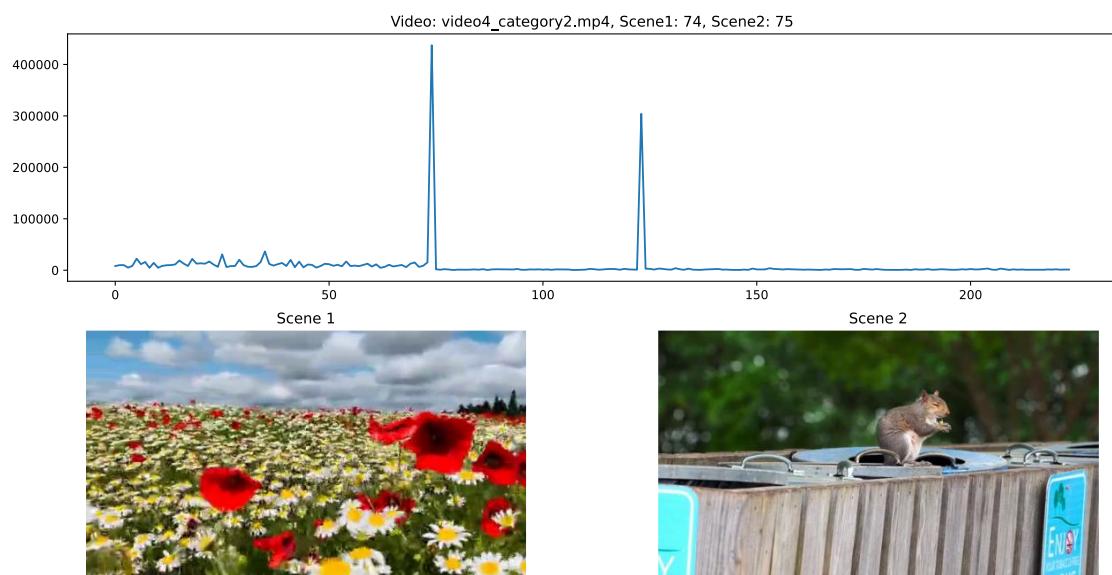
אכן לא חישוב של הסכומים המצביעים, הקוד לא מצא את הפריים הנכון. לדוגמה, עברו וידאו 3 עם שימוש בקטgorיה 1:



אכן רואים בין שני הפריים שניי יחסית קיצוני באור – רוב הצבע של הפרווה השתנה מ-פריים 34 לפריים 35, מה שהוא הבדל כה משמעותי בין ההיסטוגרמות, אע"פ שהוא הבדל לא גדול כי הם רק זווית עמודות, אך ככל הפתוח הוא הבדל יותר גדול מהשינוי האמתי בין שתי הסצנות שגם אותו ניתן לראות בגרף. אך עם שימוש בקטgorיה 2, שמשתמשה בסכומים מצביעים:



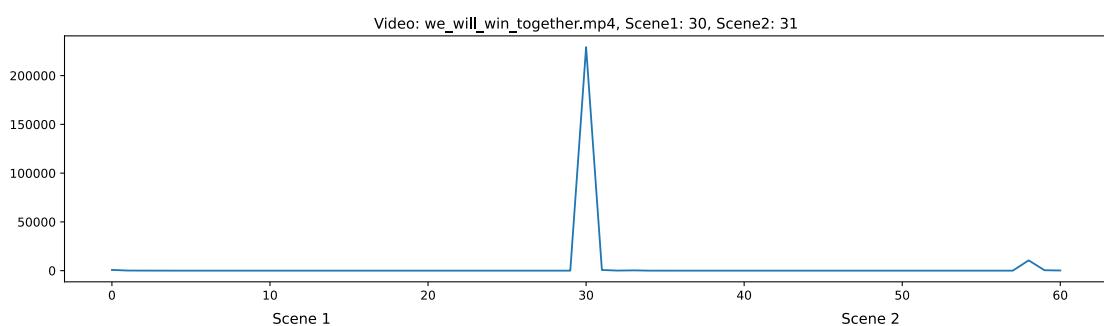
ניתן לראות שעדין יש משמעות לפריים 34, אך הוא נהייה פחות משמעותי מפריים 174.



## 6. מסקנות

בתרגיל זה השתמשנו בהיסטוגרמה של תמונה בגוני אפור כדי להזוהה מעבר בין סצנות. רأינו כמה חשוב סכום מצטבר של היסטוגרמה הוא שימושי לביצועים של האפליקציה להזוהה נכון ולא להיות מושפע משינוי או יחסית מינורי.

בזמן הפנוי כשעbaraתי על מצגת 1 אז גם כתבתי קוד לשיפור תמונה ע"י שיווי היסטוגרמה. מגניב וזכה.  
כמו כן היה אתגר מהפן התוכנות, ותרגיל מרענן לחזרה מהמילואים. תודה רבה !



כבר

יחד