

# 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 וראשון ונראה 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 הפריטים של התמונה (כמו שלא העמsty יותר מדי על הזיכרון, שהוא 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](#) 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 בזיכרון אך אני רציתי לעבוד על איטרטור ולא שהכל יהיה בזיכרון במקביל.

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

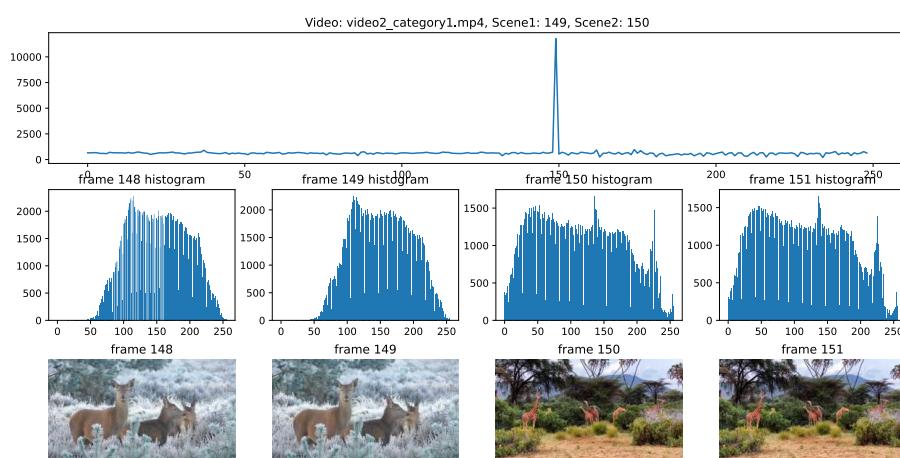
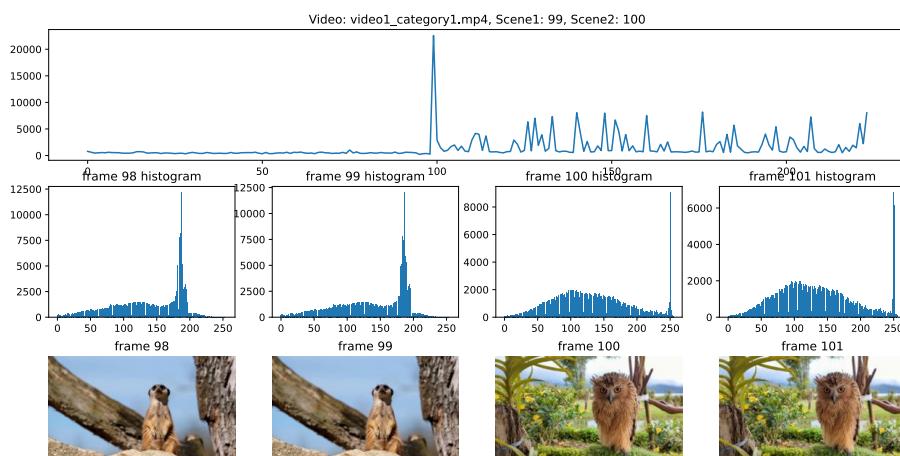
אחרי כתיבת הקוד באופן ישיר הקוד עבר אך לא החזיר את התשובה המצופה (הצגתי את שני הpriymois אחד ליד השני וראיתי שאין מעבר סצינה). לאחר כמה דקות הסתבר שימוש מה בדיעד בעותה השתמשתי בפונקציה `min` במקום `max` 😊 . חווין מזה כל הקוד עבר חלק ישר מהפעם הראשונה.

אגגר מעניין שהיה הוא למש piping ב-`python`, שהוא לא שפה טבאי לה זהה קונספט מעולם ה-FP... כתבתי מימוש עצמי שענה לי על הדרישות ועשה עבודה נחרצת.

הפרויקט נמצא [github](#) שלו.

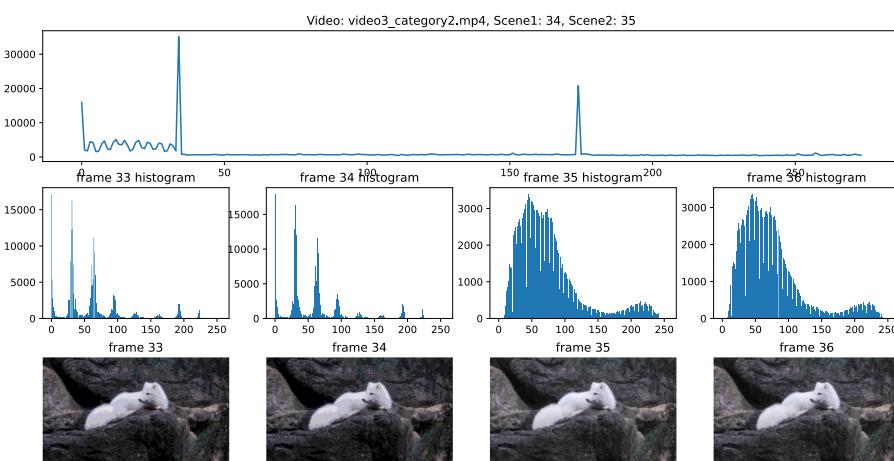
## 4. תוצאות קטgorיה 1

עבור כל וידאו: בכוורת שם הויידאו, והpriymois בהם היה החיתוך. מתחתיו גրף הנורמות של הpriymois ההיסטוגרמות לאורך הסרטון, מתחתיים שני הpriymois האחרונים בסצנה הראשונה ושני הראשונים בשנייה, עם ההistogramה המתאימה מעל כל priymim.

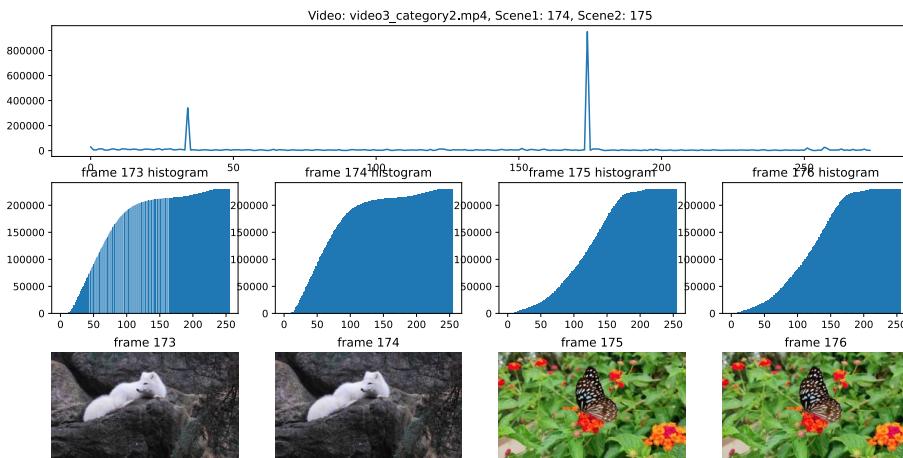


## 5. תוצאות קטgorיה 2

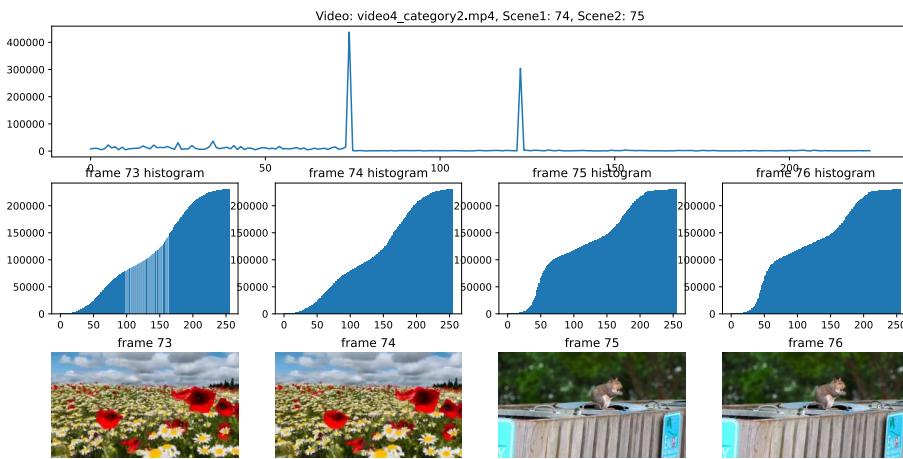
אכן לא חישוב של הסכומים המאצברים, הקוד לא מצא את הpriymois הנכון. לדוגמא, עבור וידאו 3 עם שימוש בקטgorיה 1:



מצער. אכן רואים בין שני הפרייםים שינוי ייחודי באור – רוב הצבע של הפורה השתנה מ-פרויים 34 לפרויים 35, מה שודאי עשה הבדל כה משמעותי בין ההיסטוגרמות, אך אף שהוא הבדל לא גדול כי הם רק זווים עמודות, אך ככל הפתוח הוא הבדל יותר גדול מהשינוי האמתי בין שתי הסצנות שם אותו ניתן לראות בגרף. אך עם שימוש בקטגוריה 2, שמשתמשה בסכומים מצטברים (גרפי ההיסטוגרמה עבור הסכומים המצטברים):



ניתן לראות שעדיין יש משמעות לפחות לפרויים 34, אך הוא נהייה פחות משמעותי מפרויים 174.



## 6. מסקנות

בתרגיל זה השתמשנו בהיסטוגרמה של תמונה בגווני אפור כדי להזוהה מעבר בין סצנות. רأינו כמה חישוב סכום מצטבר של היסטוגרמה הוא שימושי לביצועים של האפליקציה להזוהה נכוון ולא להיותמושפע משינוי אור יחסית מינורי.

בזמני הפנו כשבורת עלי מציגת 1 או גם כתבת קוד לשיפור תמונה ע"י שיוי היסטוגרמה. מגניב וrecht.  
כמו כן היה אתגר מהנה מהפן התכנית, ותרגיל מרענן לחזרה מהמילואים. תודה רבה !

Video: we\_will\_win\_together.mp4, Scene1: 30, Scene2: 31

