

אלגוריתמים ויישומים

בראייה ממוחשבת –

046746

**Homework 1 -
Features descriptors**

Ido Nutov – 305242968

Sarah Buzaglo - 931161798

Part 1 – Keypoint detector

שאלה 1.2 : Gaussian Pyramid

הפירמידה המתקבלת היא בעלת 3 מימדים :

- הראשון הוא מספר ה-levels, כלומר, מספר התמונות בפירמידה שהוא מספר רמות הטשטוש השונות.
- שני המימדים הבאים הם מימדי התמונה המקורית.



שאלה 1.3 : The DoG Pyramid

בנינו את הפונקציה createDoGPYramid על-ידי חישוב תמונות ההפרשים של שכבות הפירמידה GaussianPyramid שמתקבלת כקלט. לכן, המימד הראשון של DoGPYramid קטן ב-1 מזו של GaussianPyramid.



שאלה 1.4 : Edge suppression

עברנו בלולאה על כל שלב בפירמידה וחישבנו אופרטור sobel מסדר שני בכיוונים x, y, xy, לכל פיקסל בכל רמה חישבנו principal curvature משלו כאשר $trace = sobel_{xx} + sobel_{yy}$

$$det = sobel_{xx} \cdot sobel_{yy} - sobel_{xy}^2$$

ולאחר מכן את הערך של ה-principal curvature ratio לפי $\frac{trace^2}{det}$.

הוספנו ערך קטן ($1e-8$) למטריצה כדי לדאוג ליציבות נומרית ולהימנע מ חלוקה ב 0 .

שאלה 1.5: Detecting extrema

בנינו מסכה בינארית לספים שהיה עלינו לבדוק – ל-contrast, ל-principle curvature ratio. עוברים על השמונה השכנים הסמוכים בשכבה הנוכחית (neighbors in space) ועל שמונת השכנים בשכבה מעל ומתחת (neighbors in scale) בודקים עבור הנקודה הנוכחית האם זה local extrema ואם אכן זה כך מוסיפים אותה ל locsDoG .

שאלה 1.6: Putting it together

```
sigma0 = 1
k = np.sqrt(2)
levels = np.array([-1, 0, 1, 2, 3, 4])
th_contrast=0.03
th_r=12
```





בחרנו בתמונה זאת כיוון שיש לה מאפיינים גיאומטריים רבים (פינות שפות, מעגלים) ניתן לראות כי אכן האלגוריתם זיהה פינות ונקודות משמעותיות, אך בנוסף נראה שבסקאלות הנמוכות הוא זיהה גם שפות ו שפות שנהיו משוננות מתופעת ה aliasing (הקטנו את התמונה), בנוסף נשים לב שטקסט מזדהה בסקאלות הנמוכות (גם בתמונת השימורים וגם בתמונה הנוכחית). אחד השיפורים שהיינו מציעים הוא טשטוש לפני הקטנת התמונה כדי לצמצם את תופעת ה aliasing.

Part 2 – BRIEF descriptor

שאלה 2.1 : Creating a set of BRIEF tests

בנינו את ה-makeTestPattern פונקציה בעזרת שיטה הראשונה, כלומר בעזרת התפלגות אחידה.

שאלה 2.2 : Compute a BRIEF descriptor

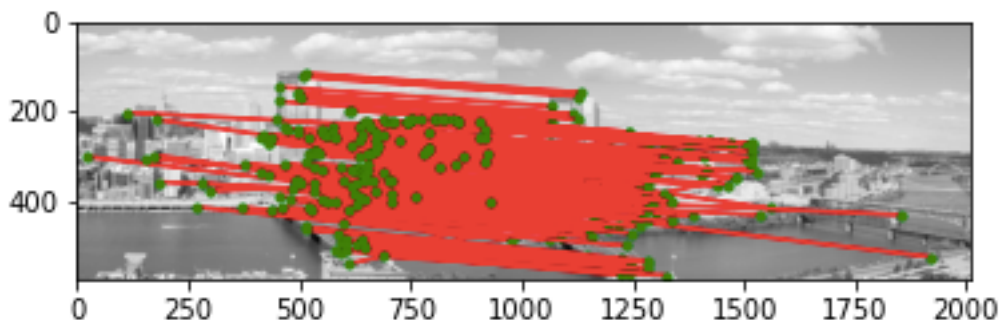
לכל נקודה שחישבנו ב locsDoG חישבנו פאטצ' בגודל $\text{patchWidth} \times \text{patchWidth}$ והשתמשנו ב testPattern (שנכנסת כ2 פרמטרים לפונקציה compareX ו compareY) כדי לקבל descriptor עבור הפאטצ'.

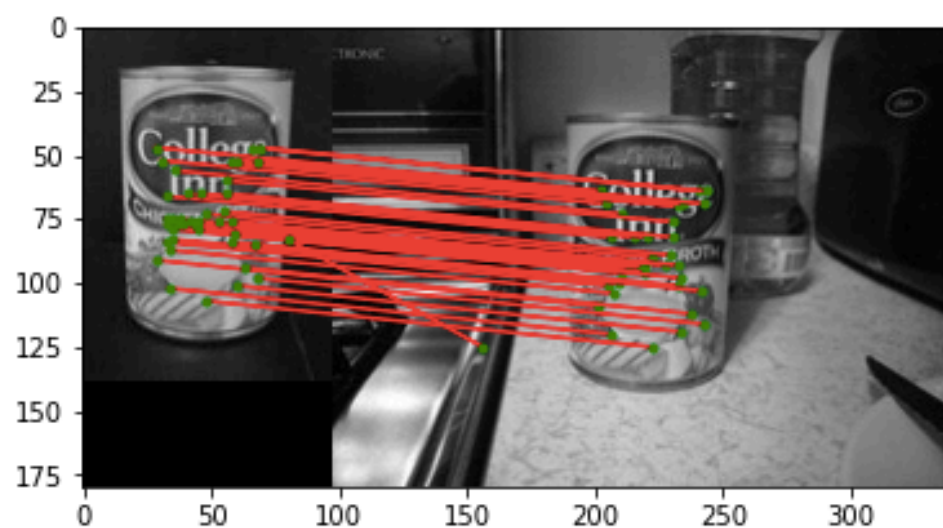
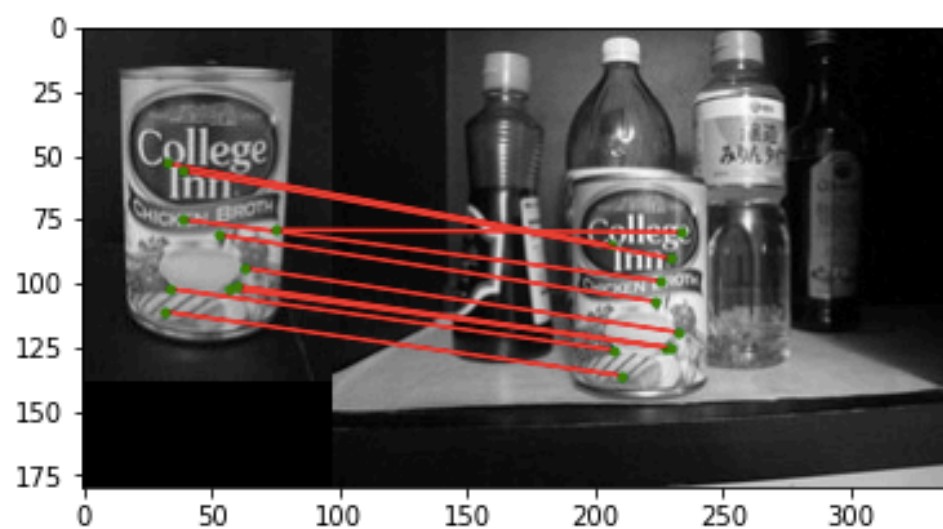
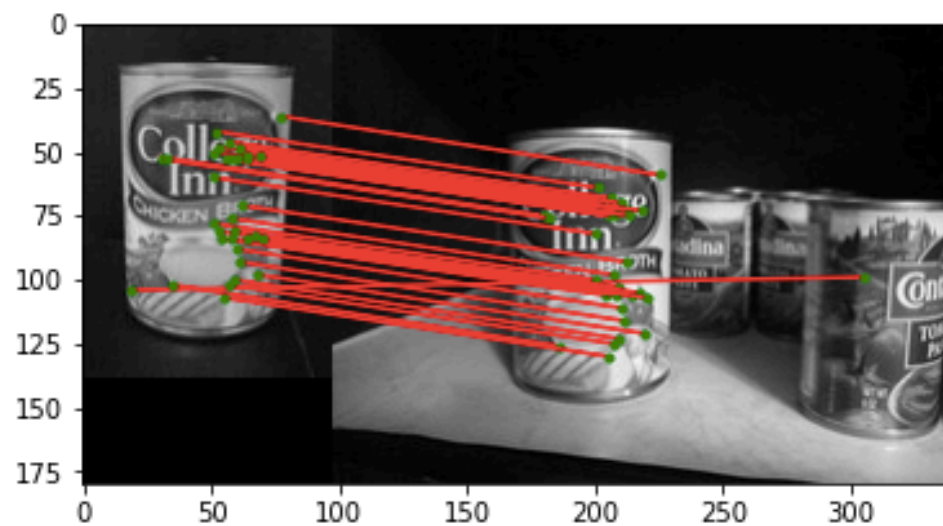
שאלה 2.3 : Putting it all together

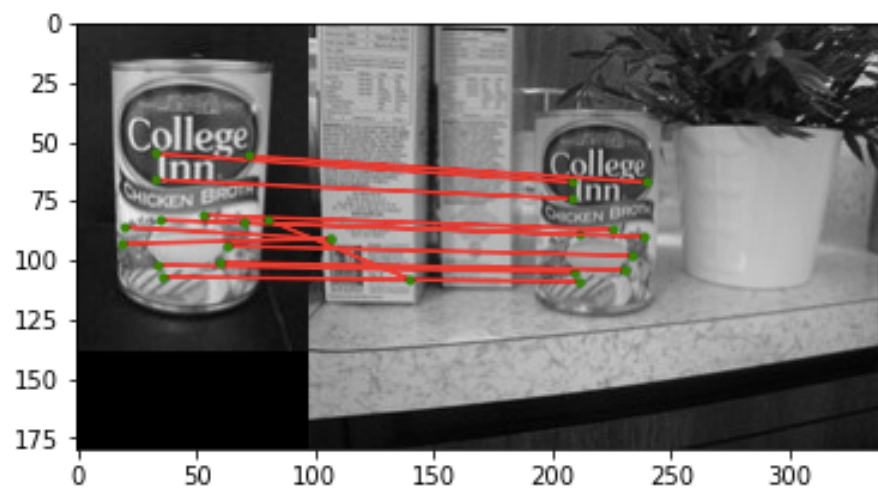
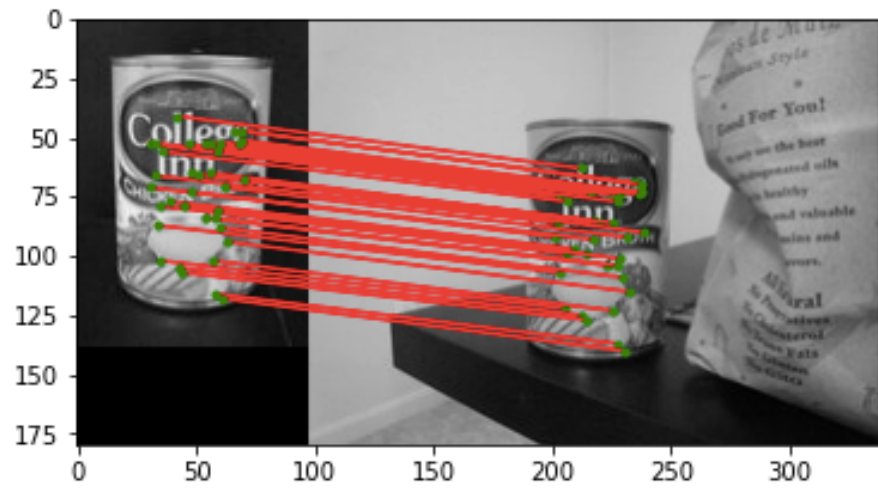
השתמשנו ב-DoGdetector כמו שראינו בחלק ראשון וטענו את ה-compareX ו-compareY. ואז קראנו לפונקציה computerBrief.

שאלה 2.4 : Check point : descriptor matching

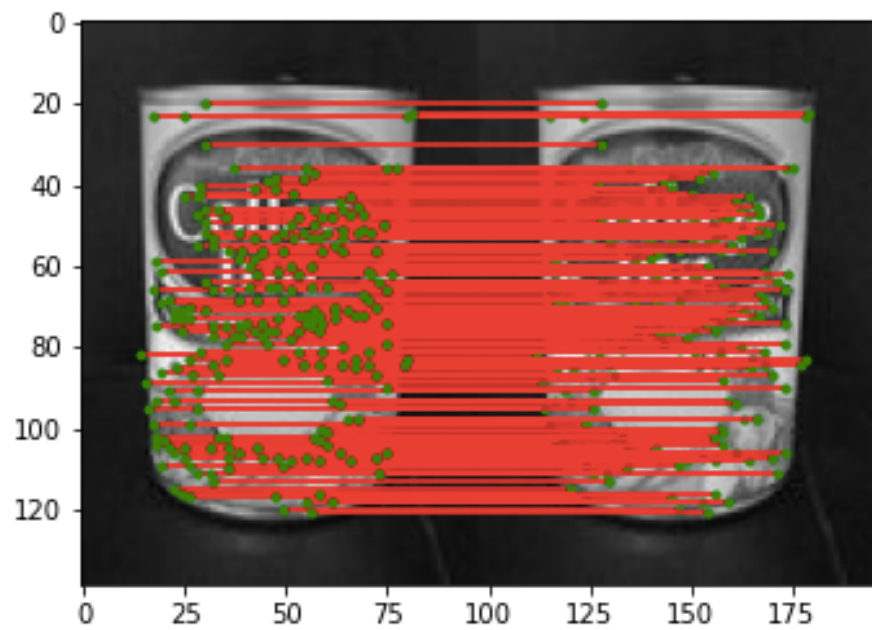
בדקנו את התמונה incline_L עם incline_R. אפשר לראות שאלגוריתם שלנו מצא נקודות מתאימות באופן די מדויק בין שתי תמונות.

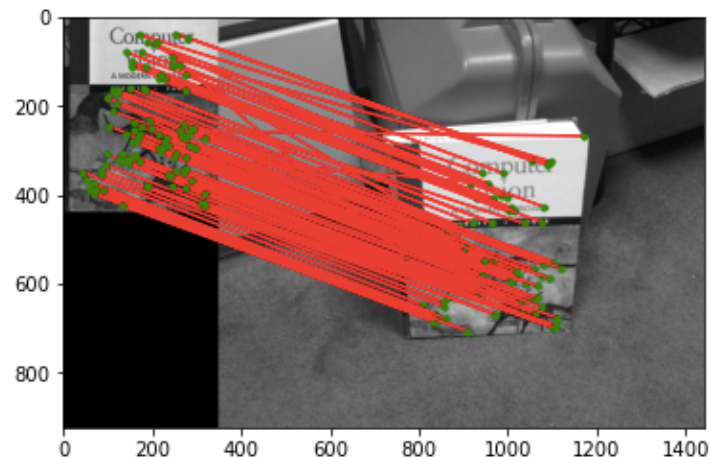
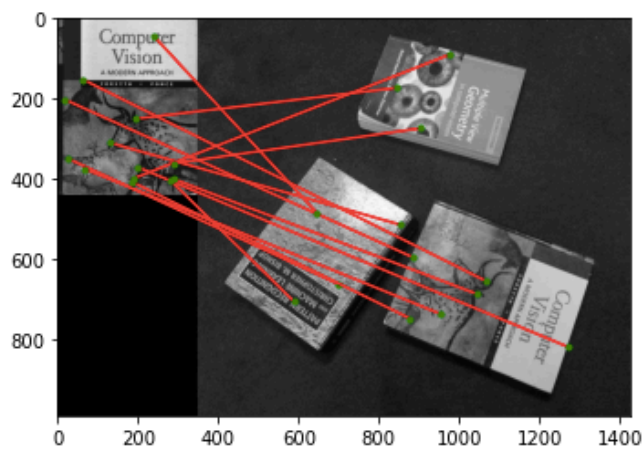
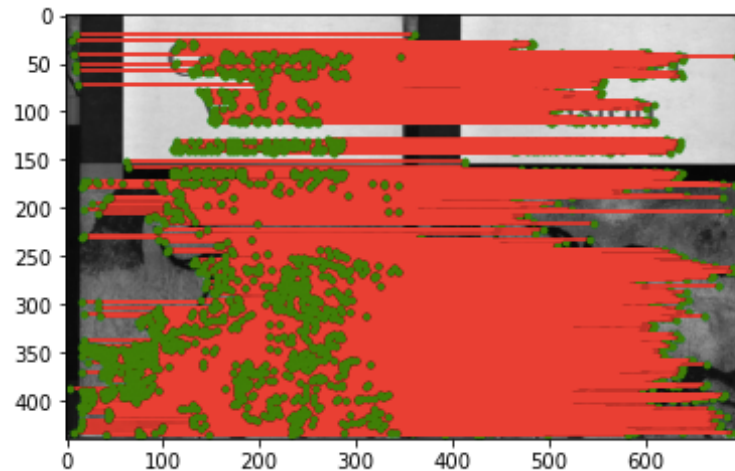
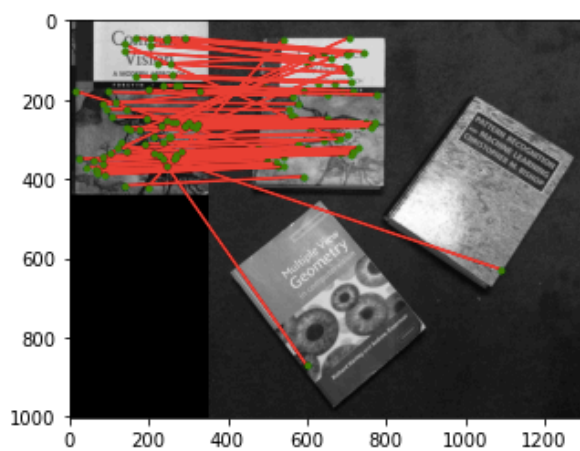
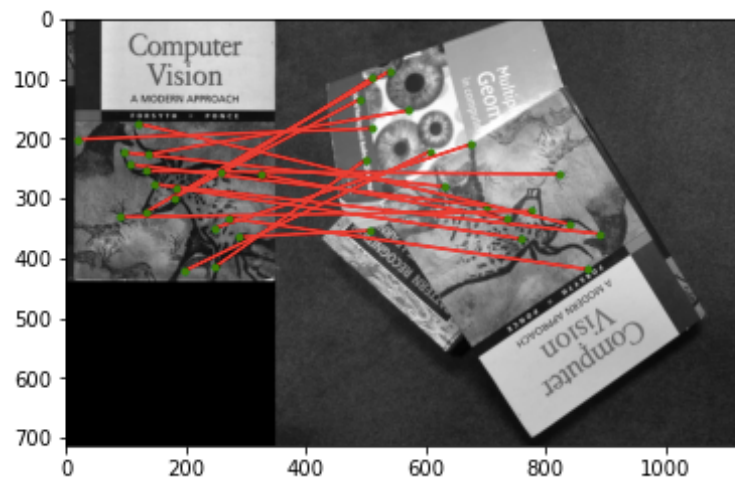
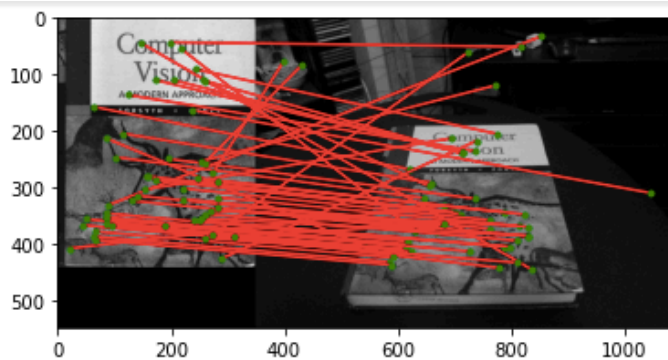






בדיקת תקינות : ננסה לקשר בין תמונה לעצמה כדי לבדוק שאלגוריתם שלנו עובד כנדרש.





לסיכום אפשר לראות שאלגוריתם שלנו מקשר כמעט את כל הנקודות בצורה נכונה כאשר הזווית של העצם מספיק קרובה בין שתי התמונות. לעומת זאת עבור זוויות שונות נקבל תוצאות עבור רק כמה נקודות אשר יקושרו בצורה נכונה אם בכלל.

BRIEF properties : 2.5 שאלה

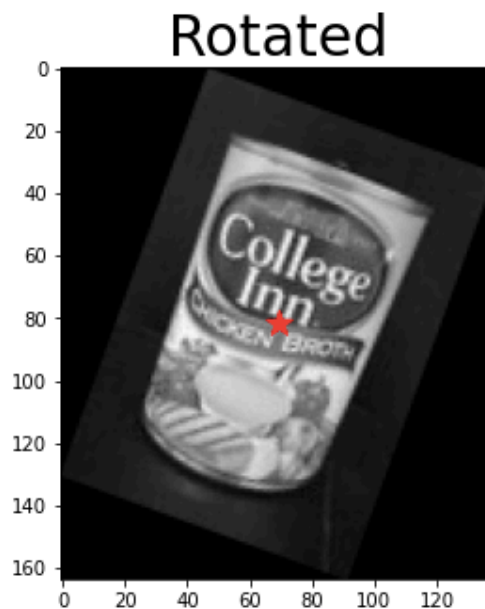
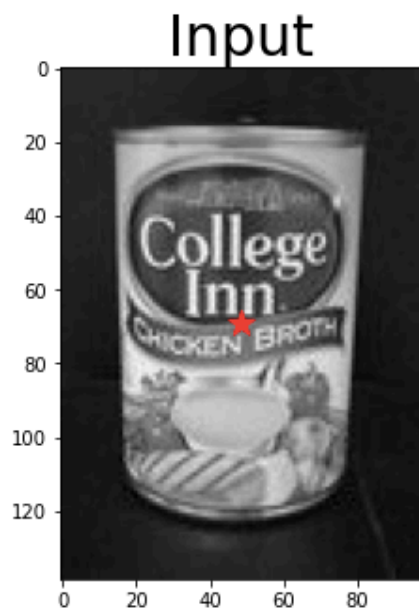
- Is BRIEF invariant to illumination changes?

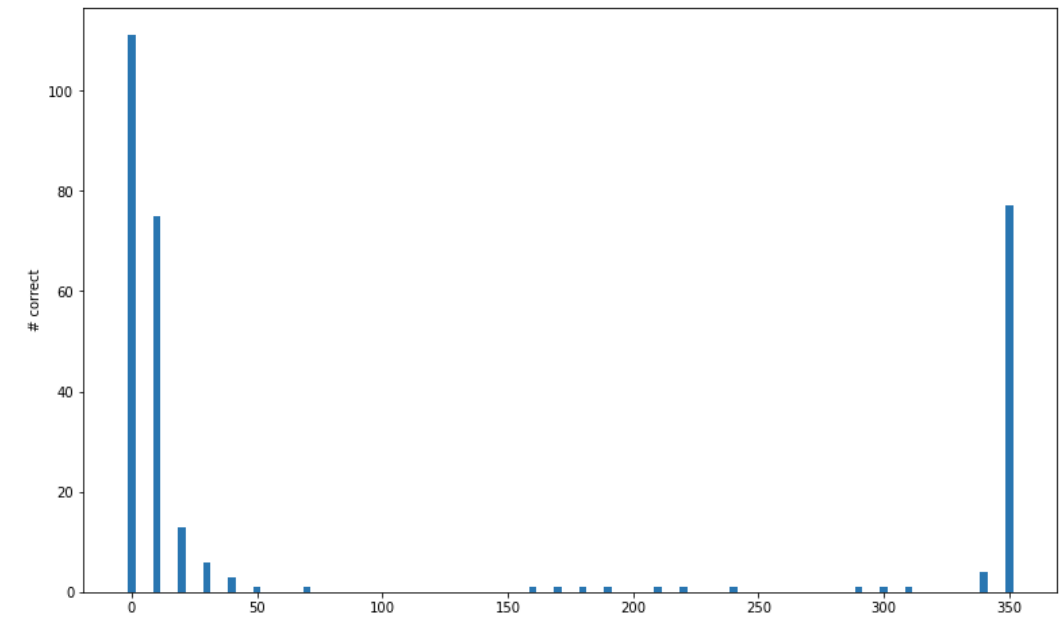
Brief is invariant to illumination changes because the DOG pyramid contains mostly edges. Brief is also invariant to illumination changes due to the fact that it is a comparison between values of the pixels in the same picture and not between two different pictures. So as long as the pixel ratio is maintained a good match will be found. In the case of two identical images except for their illumination, the difference is merely a constant change, so the relationships between the different pixels in the same image are preserved and the vectors representing each point of interest in the images are very similar.

- Assuming we want to boost the invariance of BRIEF to scale, which part of the SIFT algorithm should we include?

Assuming we want to boost the invariance of BRIEF to scale, we should include the part in the SIFT algorithm in which the Laplacian allows us to find keypoints in picture with different scales. So that we can identify keypoints in different sizes of the image.

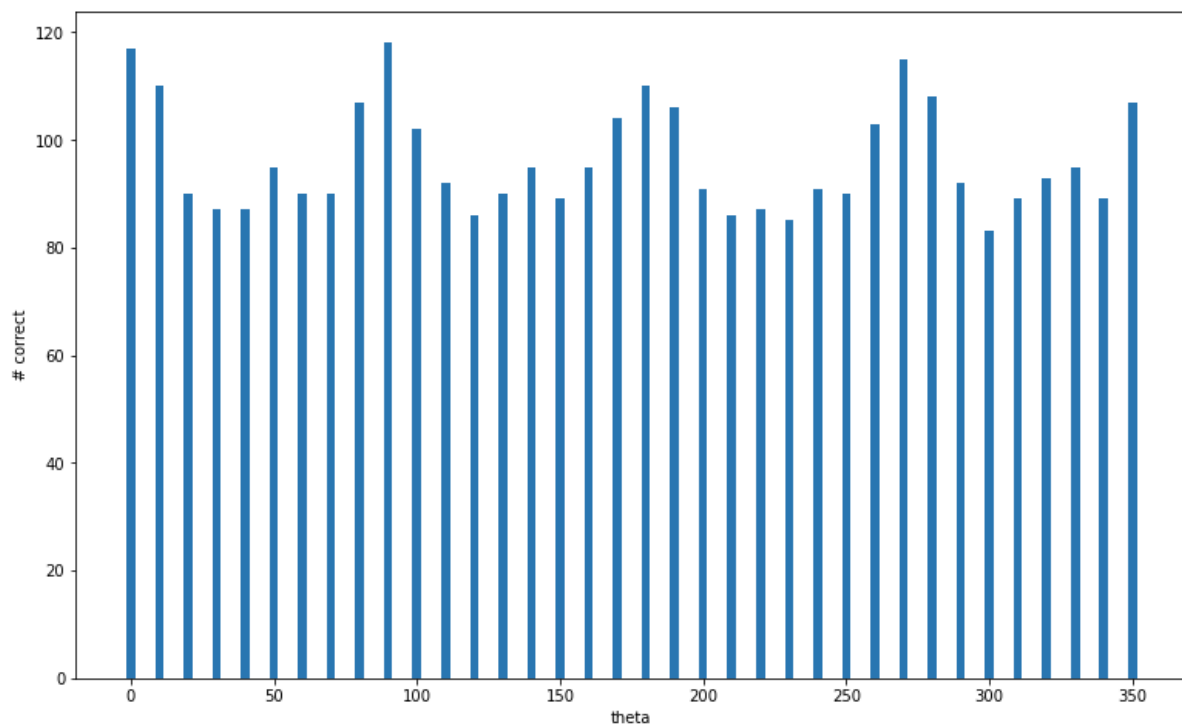
`Text(0.5, 1.0, 'Rotated')`





We can observe a correct identification when the inclination/angles are more or less conserved (0, 10 deg) or inversed (350 deg= -10 deg)

שאלה 2.6 : Oriented fast and rotated BRIEF (ORB)



- Which components in ORB are borrowed from SIFT?

ORB takes from the SIFT the invariance to scale, by using Gaussian pyramids taken on different scales, using pyramids. SIFT uses Gaussian pyramid while ORB uses pyramid of the image at different resolution.