

תרגיל 2 – עיבוד תמונה ופעולות מורפולוגיות



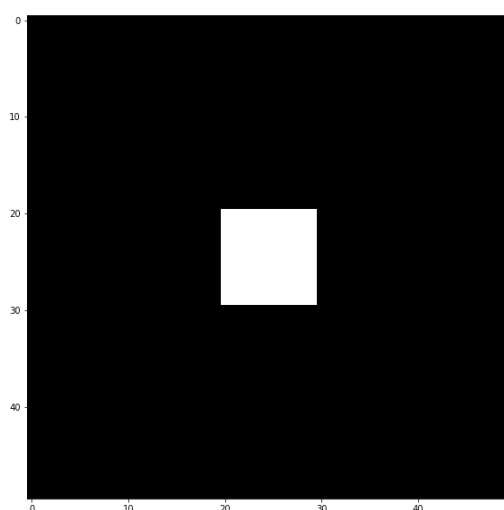
שם המגיש – עומר גבאי

ת.ז – 302390984

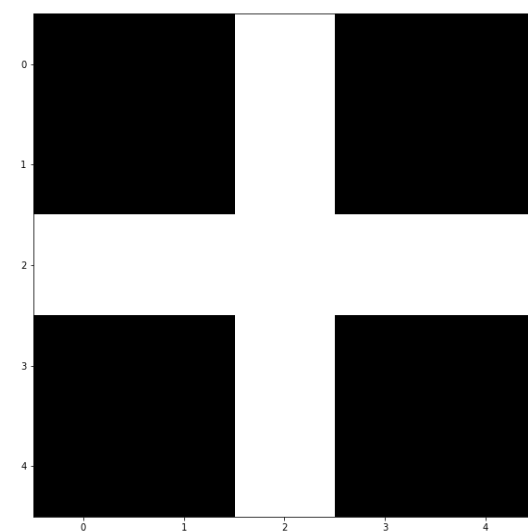
תמונה להמחשה איך המגיש נראה ←

שאלה 1 – מימוש פעולות Dilate and Erode

Original image



Structuring element for Erosion / Dilation (AKA kernel)



My Dilate Function

```
def my_dilate(img, kernel):  
    #TODO: build dilate function without cv2.dilate  
    img_new = cv2.filter2D(img, ddepth=-1, kernel=kernel)  
    T, img_new = cv2.threshold(img_new, 0, 1, cv2.THRESH_BINARY)  
  
    return img_new  
  
plt.figure(figsize=figsize)  
plt.imshow(my_dilate(img, kernel), cmap="gray")  
plt.show()
```

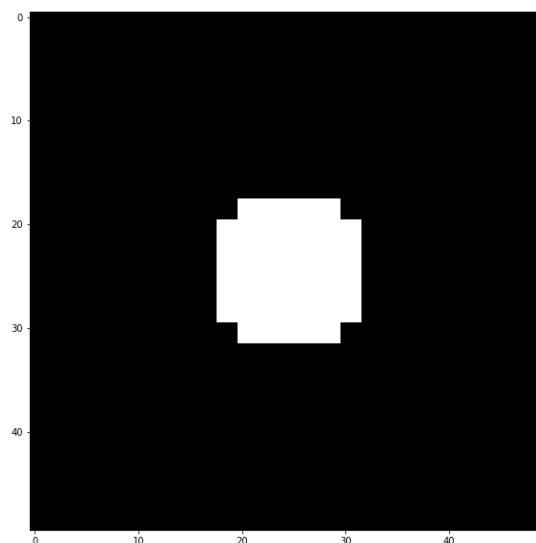
הסבר על הקוד של Dilation – הפונקציה מקבלת את התמונה המקורית והקרנל שיצרנו,

עוברת על התמונה ב cross correlation באמצעות הפונקציה filter2D. לאחר פעולה זו כל פיקסל יקבל את הערך של מספר נקודות ההשקה שהיו לסביבה שלו (הפיקסל עצמו ושכניו) עם ה kernel.

ב dilation מספיק לנו של- kernel ולסביבה של הפיקסל יהיה פיקסל אחד משותף כי אנחנו רוצים להרחיב או לנפח את גבולות האובייקט. לכן לאחר filter2D אני הופך את התמונה חזרה לבינארית וכל הפיקסלים שקיבלו ערך גדול מ- 0 יצבעו עכשיו לבן.

התוצאה המתקבלת –

```
cv2.dilate & my_dilate are the same!
```



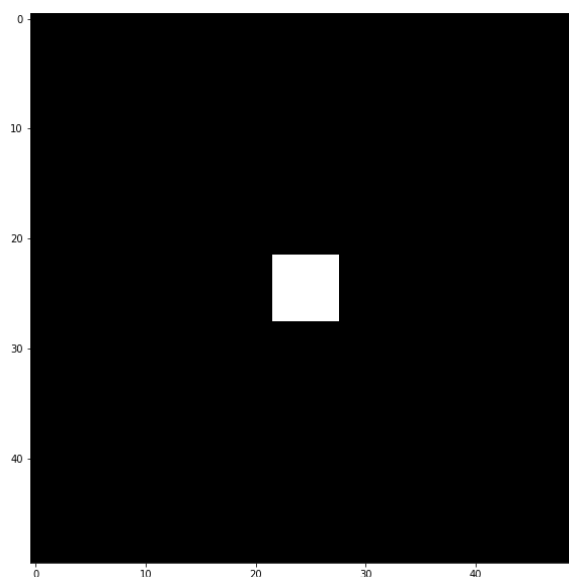
My Erode Function

```
def my_erode(img, kernel):  
    #TODO: build erode function without cv2.erode  
    img_new = cv2.filter2D(img, ddepth=-1, kernel=kernel)  
    _,img_new = cv2.threshold(img_new, 8 ,1, cv2.THRESH_BINARY  
)  
    return img_new  
  
plt.figure(figsize=figsize)  
plt.imshow(my_erode(img, kernel), cmap="gray")  
plt.show()
```

הסבר על הקוד של Erode – הפונקציה בדומה ל-dilate עושה cross correlation עם התמונה המקורית רק הפעם ה threshold ישתנה כי אנחנו רוצים להשאיר רק את הפיקסלים שמוכילים לחלוטין באובייקט שלנו משמע מספר הנקודות השקה שלהם עם הקרנל צריך להיות מקסימלי במקרה זה ערך זה הוא צריך להיות גדול מ – 8 ולכן זה מה שנתתי בפונקציה (ערך זה נקבע לפי מספר הפיקסלים הדולקים בקרנל שהוא בצורה של צלב במטריצה 5x5).

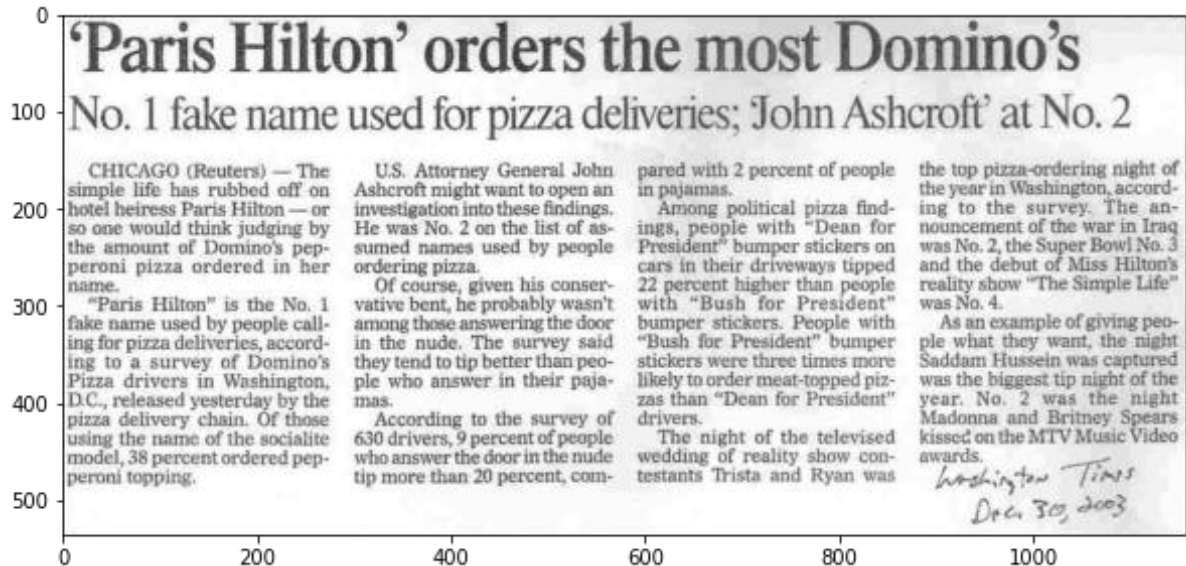
התוצאה המתקבלת - כמו שניתן לראות הריבוע קטן, כי רק הפיקסלים בריבוע הקטן היו ממש מוכילים באובייקט כשהקרנל עבר על הפיקסלים. אם היינו בוחרים קרנל גדול יותר הצורה היתה יכולה להיעלם לגמרי.

cv2.erode & my_erode are the same!



שאלה 2 – סימון מילים בטקסט מתוך עיתון

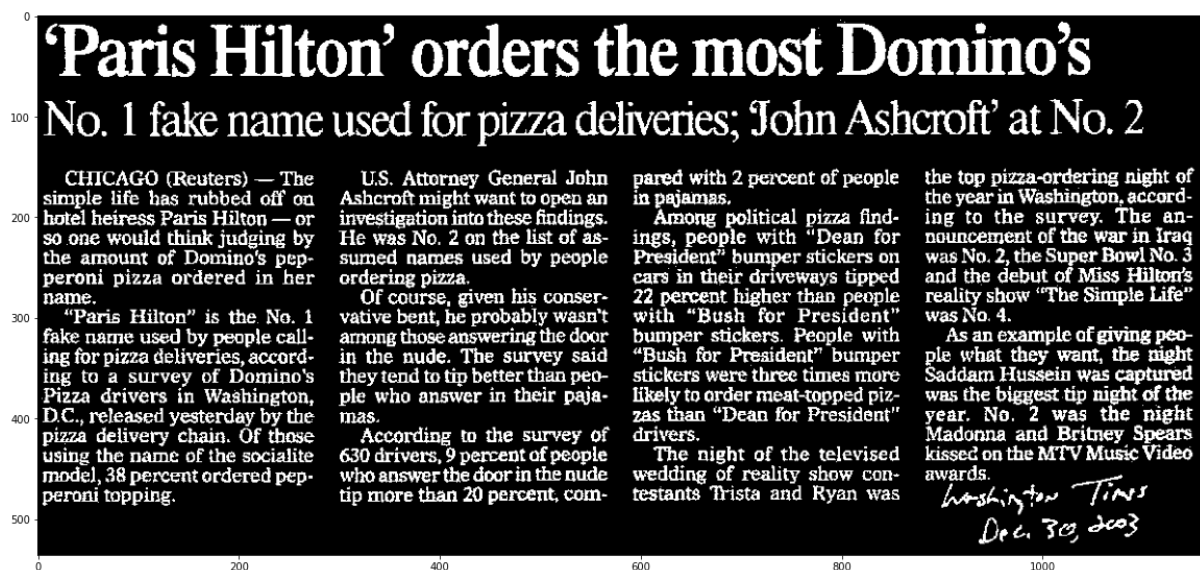
התמונה המקורית -



הפיכת התמונה לבינארית באמצעות cv2.threshold

```
# TODO: let's start with turning the image to a binary one
_, im_th = cv2.threshold(im_gray, 180, 1, cv2.THRESH_BINARY_INV)
plt.figure(figsize=(20, 20))
plt.imshow(im_th, cmap="gray")
plt.show()
```

הפיקסלים הלבנים יותר שעוצמתן עולה על 180 יהפכו לשחורים לגמרי 0, בעוד שהפיקסלים של הטקסט יקבלו ערך 1 וכך הפכתי את התמונה לבינארית.



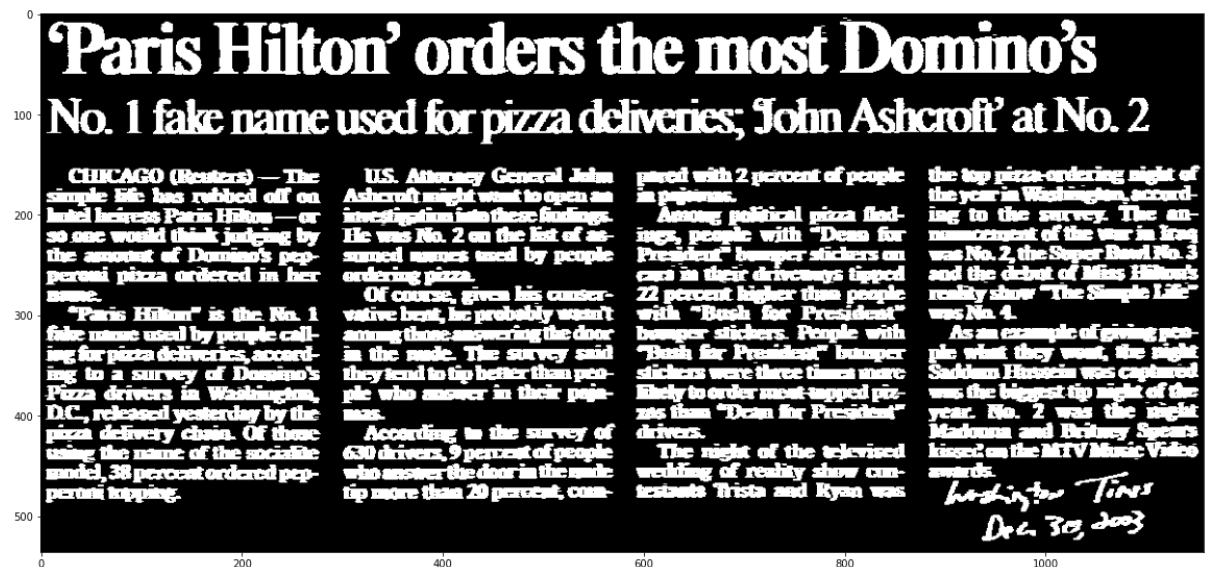
ננפח את המילים בטקסט באמצעות הפעולה המורפולוגית dilation.

נגדיר קרנל שהוא וקטור שורה בגודל 1×2 כך שלאחר הפעלת הפונקציה של הניפוח המילים ימרחו לצדדים כך שאותיות סמוכות יתחברו. מספר איטרציות של dilation הוא 3.

```
# TODO: next, merge all pixels of the same word together
kernel = np.ones(shape=(1,2))
dilated_im = cv2.dilate(im_th, kernel=kernel, iterations=3)

plt.figure(figsize=(20,20))
plt.imshow(dilated_im, cmap="gray")
plt.show()
```

טקסט לאחר ניפוח –



ניתן לראות שכל המילים בטקסט התעבו והרבה אותיות התחברו – יותר מודגש בטקסט הקטן מאשר בכותרת.

סעיף אחרון באמצעות פעולות מורפולוגיות נרצה לסמן רק את הכותרת של הכתבה.
על מנת לבצע זאת נעשה Erode על הטקסט הקטן שנמצא מתחת לכותרת.
הפיסקאות מתחילות בערך מ $x=150$ עד סוף התמונה וזה הקטע שעליו נבצע erode.

```
# Erode paragraphs then we will be left with only the headline
_, im_eroded_paragraphs =
    cv2.threshold(im_gray, 180, 1, cv2.THRESH_BINARY_INV)

kernel = np.ones(shape=(3,3)) # kernel size for the erosion

im_eroded_paragraphs[150:, :] = cv2.erode(im_eroded_paragraphs
[150:,:], kernel, iterations=3)
```

התוצאה המתקבלת – נמחק הטקסט מתחת לכותרת (כל הפיקסלים מתחת לכותרת התאפסו).



התוצאה המתקבלת לאחר חיפוש של המילים – רק המילים המרכיבים את הכותרת מזוהות

