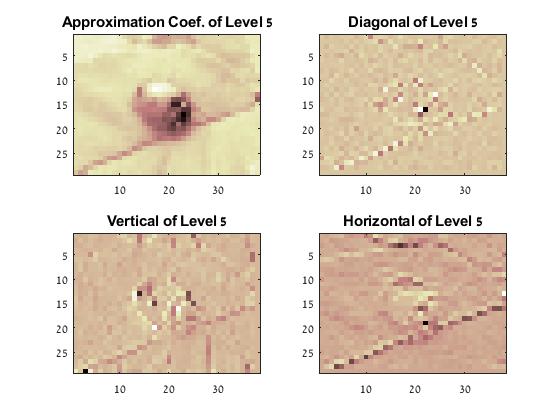
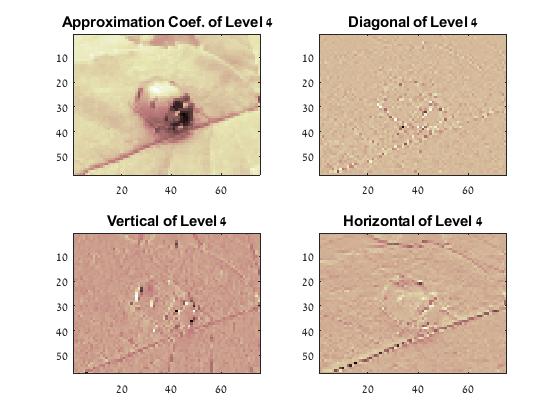
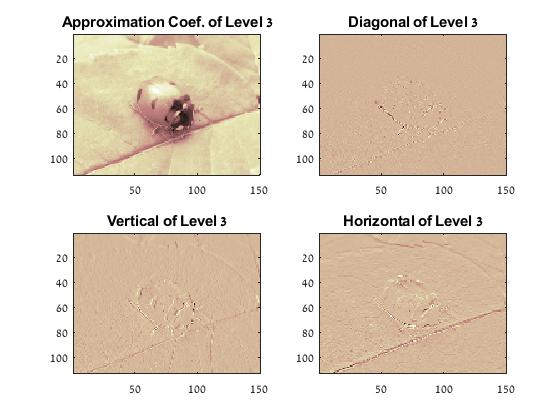
שאלה 2

חלק 1

A





E

אנחנו רואים שכל רמה הרמה של הwavlet מיותר גבוהה אז התמונה מפוקסלת יותר,

וזה גם מסתדר מבחינת גודל התמונה.

כיוון שבתמונה המקורית יש 900\*1200= 1080000 פיקסלים

ו ב LEVEL 3 יש 113\*150 = 16950 שזה 1.569% מגודל התמונה המקורית כלומר מהמידע שהיה בתמונה המקורית

ב LEVEL 4 יש 57\*75 = 4275 שזה 0.39% מהתמונה המקורית

ו LEVEL 5 יש 29\*38= 1102 שזה 0.1% מהתמונה המקורית

לכן התוצאות שלנו מתאימות לחישובים

חלק 2

א

התמונה המקורית



ב

התמונה עם רעש גאוסיאן



ג



אחרי הוספת הרעש היינו צריכים לנקות אותו לכן יצרנו threshold

THR = wthrmngr('dw2ddenoLVL','sqrtbal\_sn',C,S);

ובחרנו באופצית ה'dw2ddenoLVL' כי הוא 2-D **denoising** using level-dependent thresholds

הערה למאור: אין לי מושג מה זה 'sqrtbal\_sn' אבל זה עובד

אחרי זה קראנו לפונקציה wdencmp() עם הערכים

[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('gbl',img\_noise,'haar',3,THR(1,1),'h',1);

כאשר img\_noise זה התמונה שלנו 'haar' זה ה*wname* ו 3 זה הרמה.

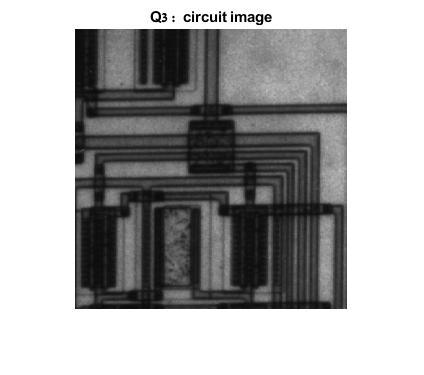
בנוסף הכנסנו את הTHR שלנו וקבענו את הסינון ל'h' סינון קשה

והצגנו את XC שהוא התמונה המסוננת

3.

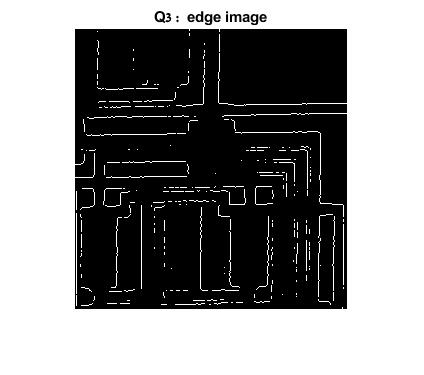
א.

התמונה המקורית



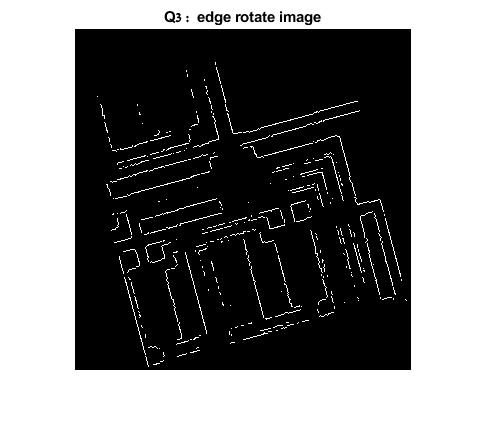
ב.1

אחרי העלאה מצאנו את הedge בעזרת פונקצית edge



ב.2

ואז סובבנו בעזרת פונקצית BW



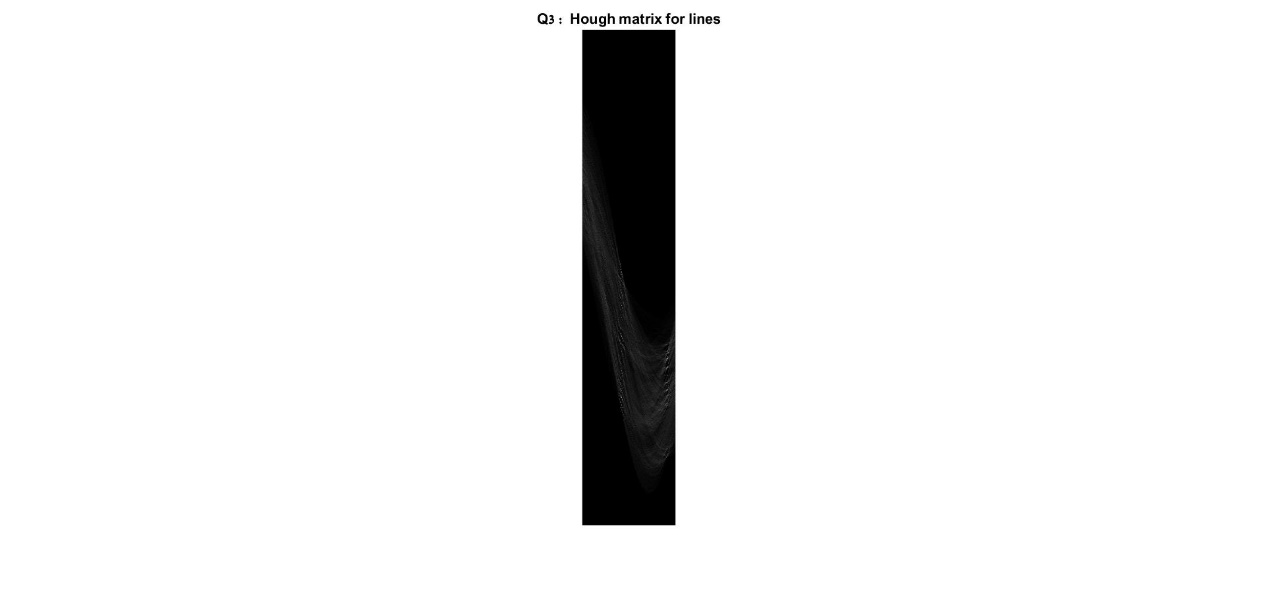
ג.

בנינו את הפונקציה dip\_hough\_lines(BW) ,נזכיר כי הריצה בפונקציה זאת עוברת על כל הפיקסלים ועבור כל פיקסל שהוא edge .,כלומר במטריצת הBW הערך שווה 1, אנחנו בודקים בין זווית 90- ל90 ויוצרים r ועבור tata ו r מעלה ב1 את הערך בתא הנתון, כלומר

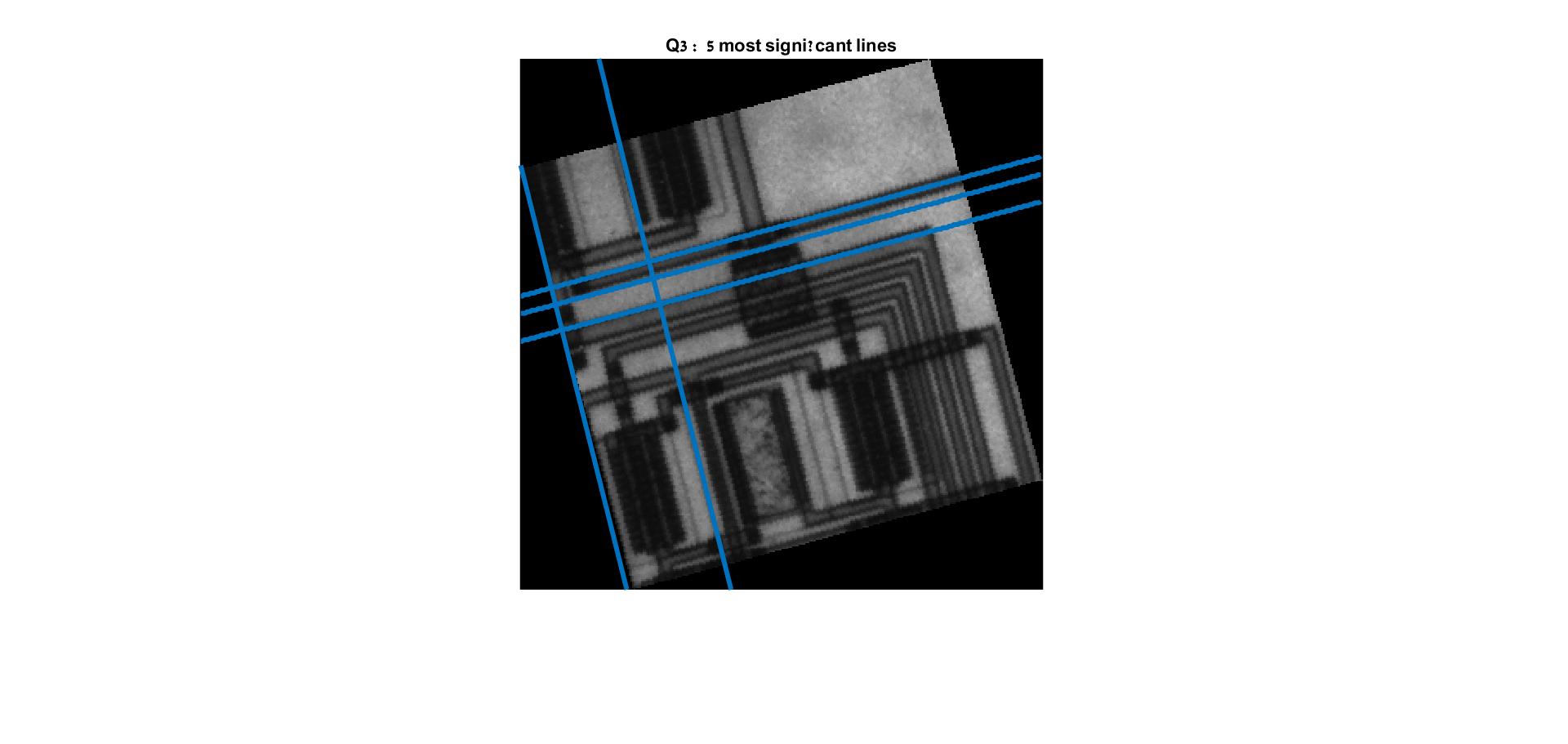
M(r,tata) = M(r,tata) + 1

את הקווים נזהה לפי הפיקים, כלומר איפה שיש פיק (ערך גבוה במטריצת M ) יעבור קו (לפי ה r ו tata )

ד.



ה.



על ידי חישוב ההפוך לr ול tata נוכל לשחזר את הקווים, במקרה שלנו את 5 הקווים המשמעותיים ביותר.

מאור: את הנוסחא אפשר למצוא בעבודה, אני פשוט עשיתי את ההופכי שלה

ו.

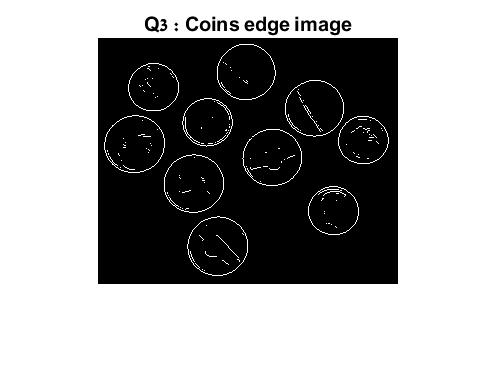
1.

התמונה של המטבעות



2.

תמונה אחרי גילוי הקצוות



3. מציאת המעגלים בעזרת הפונקציה שבנינו

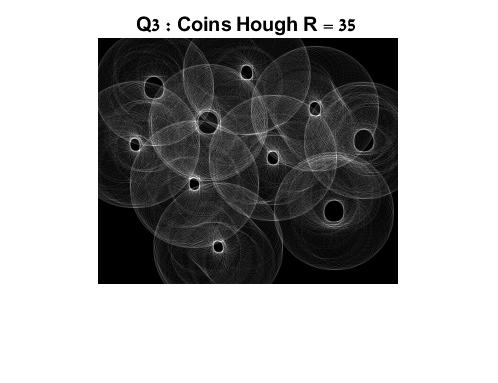
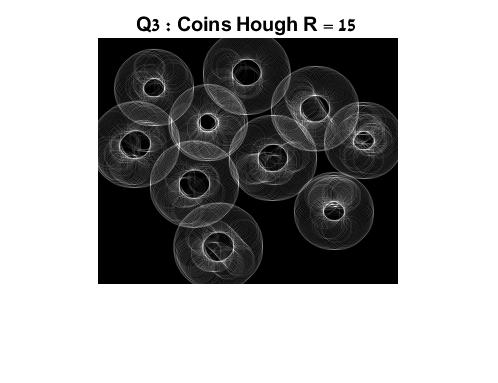
dip\_hough\_circles

כמו במקרה הקודם, בנינו פונקציה שמחפשת את הפקסלים ששווים 1 (במטריצת הBW ) .

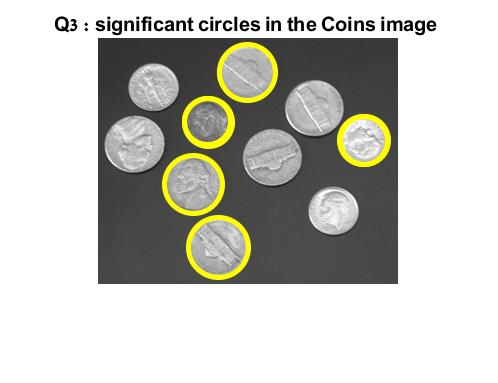
רק שכאן זה מעגלים , לכן חישבנו בעזרת פיתוח ממשואת המעגל

ולכן התוצאה היא מטריצה 3 ממדית (a, b ,r )

כאן יש שתי תמונות (מתוך 21 אפשריות ) כאשר ה 15=R וכאשר 35 = R את ההבדל אפשר לראות בגודל המעגלים שנוצרים.



4.



בעזרת האלגוריתם

Houghpeaks

ל D3 שסופק לנו בעבודה, מצאנו את 5 העיגולים המשמעותיים ביותר

ועל ידי פונקצית

insertShape

הכנסנו אותם לתמונה.

במקרה הזה למצוא את המעגלים היה פשוט יותר כיוון והאלגוריתם משתמש בR ו בtata כמשתנים , כלומר ההתמרה מוצאת לנו את מרכזי המעגלים.