EXERCISE 1 - BASIC IMAGE OPERATIONS

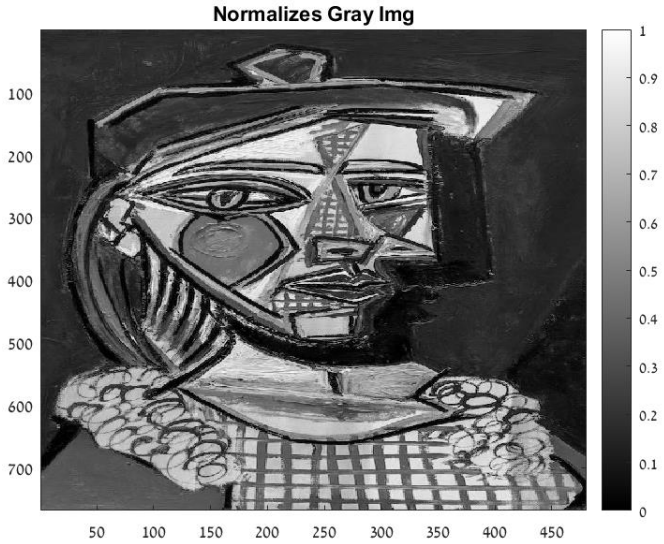
By:

Ayalla Reuven 314077033

Lior Avadyayev 206087611

שאלה 1:

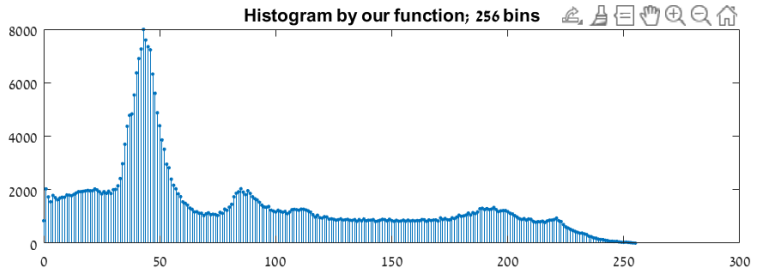
1.1 קראנו את התמונה במטלב, העברנו אותה לסקאלה של צבעים אפורים, ונירמלנו אותה ע"י הפונקציה 'dip_GN_imread' שהתבקשנו לכתוב. התמונה המנורמלת שהתקבלה:

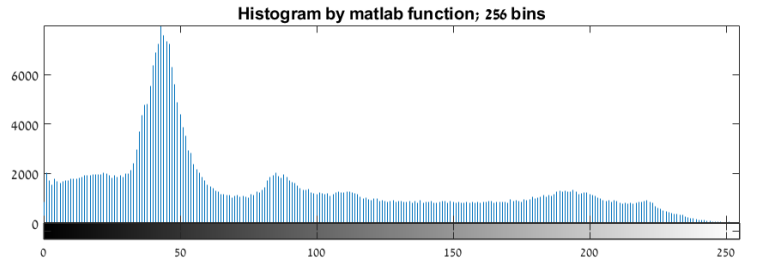


הפונקציה 'dip_GN_imread' מבצעת את הנרמול לפי:

$$\frac{Img - min(Img(:))}{max(Img(:)) - min(Img(:))}$$

1.2 התבקשנו לכתוב את הפונקציה 'dip_histogram' שמוצאת את ההיסטוגרמה של התמונה. עבור 256 בינים השווינו את התוצאות שהתקבלו מהפונקציה שלנו לעומת ההיסטוגרמה שהתקבלה ע"י שימוש בפונקציה 'imhist' של מאטלב. להלן התוצאות:



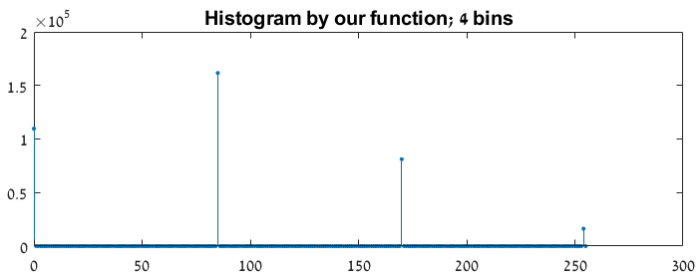


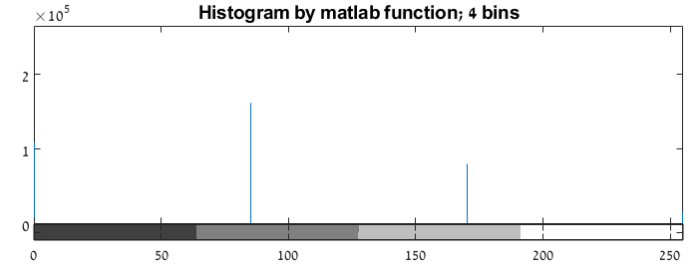
ניתן לראות שהגרפים שהתקבלו נראים זהים. בנוסף חיסרנו את הוקטורים של ההיסטוגרמות והתקבל וקטור של אפסים כך שהתוצאות אכן זהות.

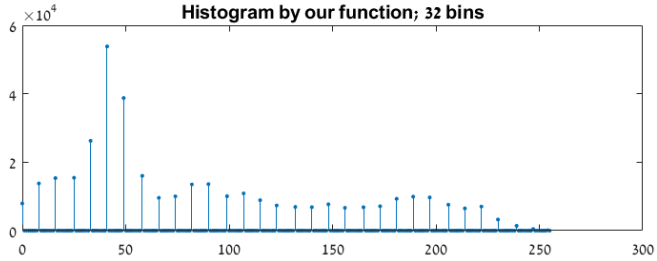
התבקשנו מתוך גרף ההיסטוגרמה שהתקבל להתאים את הגאוסיאנים בגרף לחלקים המתאימים בתמונות.

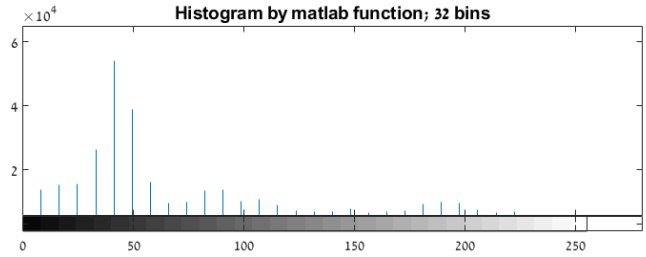
1. הגאוסיאן הראשון (משמאל לימין) והגדול ביותר נמצא בחלקים החשוכים יותר בתמונה. מהתבוננות בתמונה ניתן להבחין שהרקע הוא האיזור עם הכי הרבה פיקסלים באותו גוון כהה ולכן הגאוסיאן הראשון מתאים לרקע.
2. מתוך התבוננות במיקום בסקאלת הצבעים של הגאוסיאן השני נסיק שהוא מתאים ללחי, לשיער ולכתף של הדמות.
3. הגאוסיאן השני מתאים לאפורים הבהירים. לצוואר, לאף ולחולצה של הדמות.

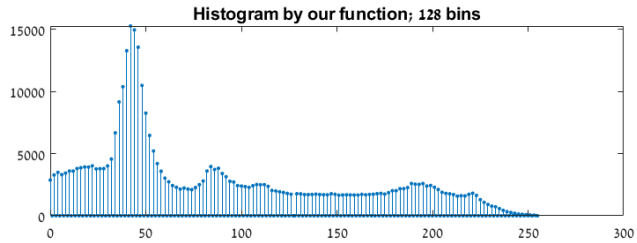
התבקשנו להציג את ההיסטוגרמות עבור 128,32,4 בינים שהתקבלו ע״י הפונקציה שכתבנו ולהשוות את התוצאות להיסטוגרמות שהתקבלו ע״י הפונקציה imhist של matlab. הגרפים שהתקבלו הם:

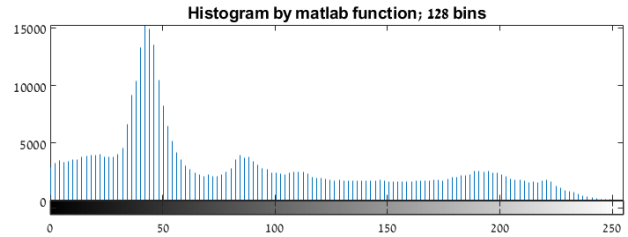

Histogram by our function; 4 bins
×10^5
2
1.5
1
0.5
0
0
50
100
150
200
250
300


Histogram by matlab function; 4 bins
×10^5
2
1
0
0
50
100
150
200
250


Histogram by our function; 32 bins
×10^4
6
4
2
0
0
50
100
150
200
250
300


Histogram by matlab function; 32 bins
×10^4
6
4
2
0
50
100
150
200
250


Histogram by our function; 128 bins
15000
10000
5000
0
0
50
100
150
200
250
300


Histogram by matlab function; 128 bins
15000
10000
5000
0
0
50
100
150
200
250

גם במקרים האלו קיבלנו וקטור אפסים כאשר השווינו בין ההיסטוגרמות. נשים לב שלמרות הירידה בכמות הbins ניתן להבחין בצורת הגאוסיאן המרכזי בכל ההיסטוגרמות.

1.3 התבקשנו לכתוב פונקציה שמבהירה את תמונה מנורמלת ע״י חיבור או הכפלה של פרמטרים. הפעלנו את הפונקציה עבור חיבור של [+0.4,+0.8] וכפל ב [+1.4,+1.6]. התוצאות:

ניתן לראות שכשהוספנו או הכפלנו בערכים קיבלנו תמונה בהירה יותר. נרמלנו את התמונה ע״י הנחתת כל הפיקסלים שגבוהים מ1 ל1 והעלאת הפיקסלים שקטנים מ0 ל0. בנוסף ניתן להבחין שע״י הוספה קיבלנו תמונה בהירה יותר מאשר עבור הכפלה.

1.4 מיפוי לינארי:

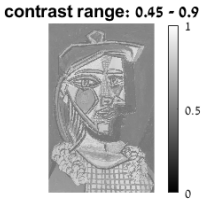
התבקשנו לכתוב פונקציה שמשנה את הניגודיות של התמונה ע״י שינוי התחום הדינאמי בצורה לינארית. הפעלנו את הפונקציה עבור התווכים הבאים: [1,0],[0.4,0.5],[0.45,0.9]. ניתן לראות את התמונות החדשות[ביחד עם ההיסטוגרמות המתאימות בעמוד הבא.

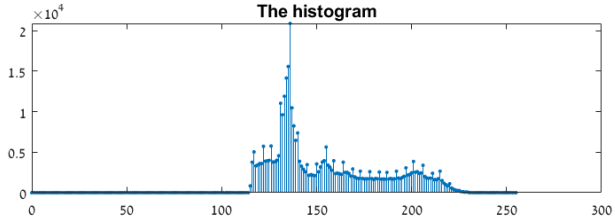
את המיפוי הלינארי ביצענו לפי:

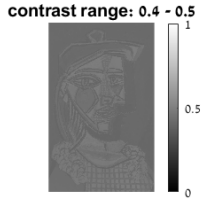
$$new\ image = \frac{high\ range}{low\ range} * img + low\ range$$

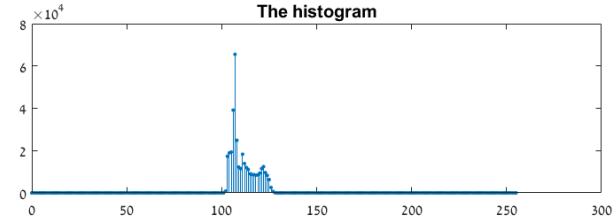
מכיוון שהתמונה מנורמלת, את ההיסטוגרמה ביצענו עבור $256 * img$, לכן הציר בסקאלה של 256.

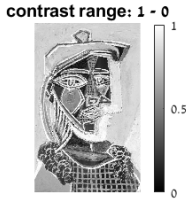
ניתן לראות שעבור כל אחד מהתווכים החדשים קיבלנו היסטוגרמה שמאוד דומה להיסטוגרמה של התמונה המקורית רק מכווץ, זה מכיוון שהמיפוי הוא לינארי. בנוסף בתמונה השניה הכי קשה לעין שלנו להבחין בפרטים, זה מכיוון שהתווך קטן פי 10. בנוסף בתמונה השלישית אפשר לראות שקיבלנו ניגודיות בצבעים (תמונה הופכית) וזו מכיוון שהפכנו את התווך.

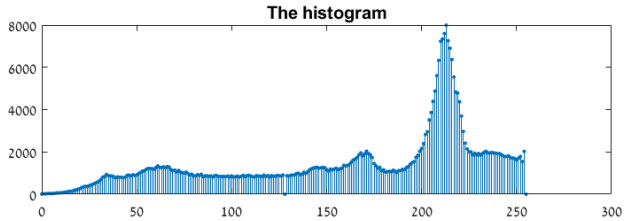












מיפוי לא לינארי :

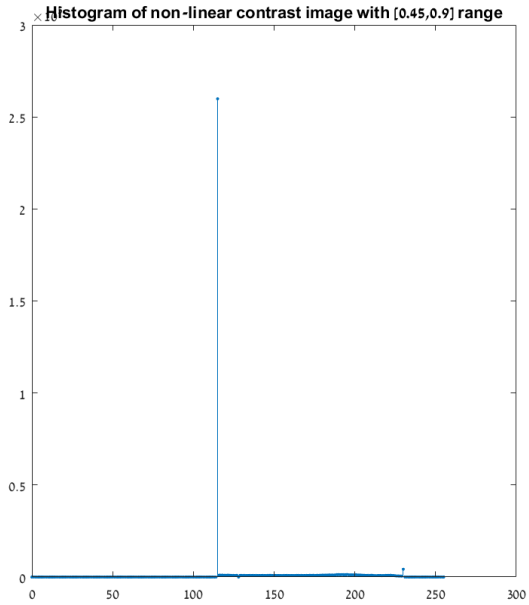
את המיפוי הלא לינארי של התווך ביצענו בצורה הבאה :

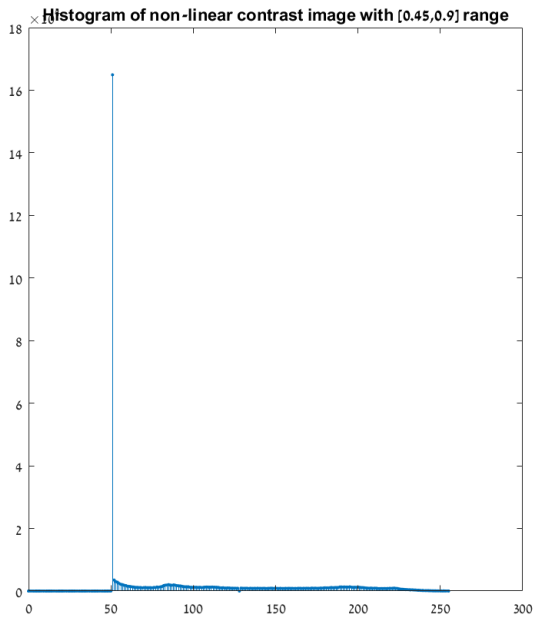
-פיקסל שערכו מעל ה$high\ range$ קיבל את הערך $high\ range$.

-פיקסל שערכו מתחת ל$low\ range$ קיבל את הערך $low\ range$.

ביצענו את המיפוי עבור התווכים [0.2,0.8],[0.45,0.9] וחישבנו את ההיסטוגרמות שלהם.

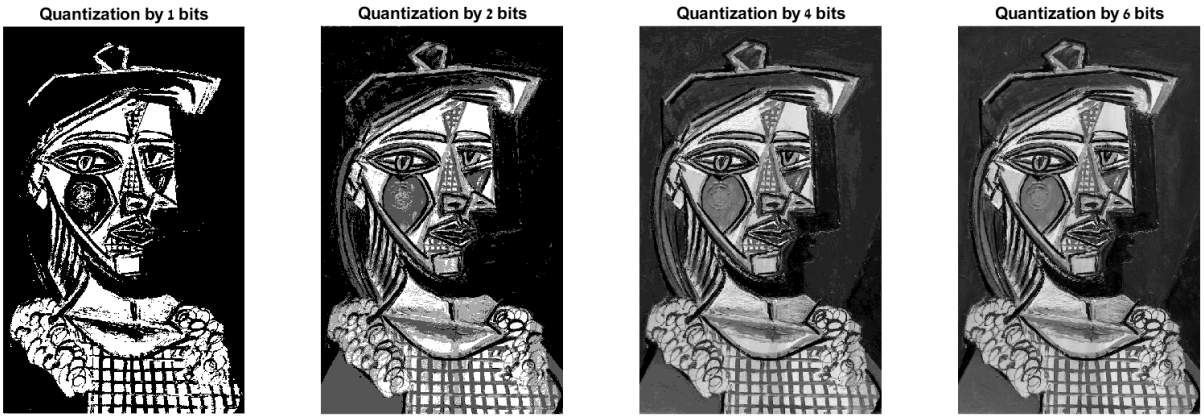
התוצאות :





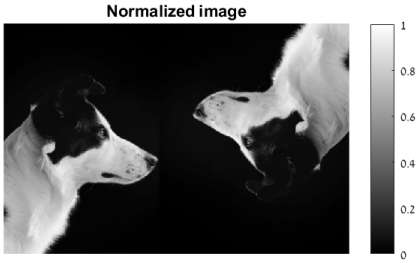
הסבר התוצאות: מתוך ההיסטוגרמות ניתן לראות שרוב הפיקסלים נמצאים בערך התחתון של התווך, זאת מכיוון שרוב הפיקסלים בתמונה המקורית מקבלים ערכים נמוכים, ולכן כאשר אנחנו מבצעים מיפוי לא לינארי, רוב הערכים יקבלו את הערך התחתון של התווך.

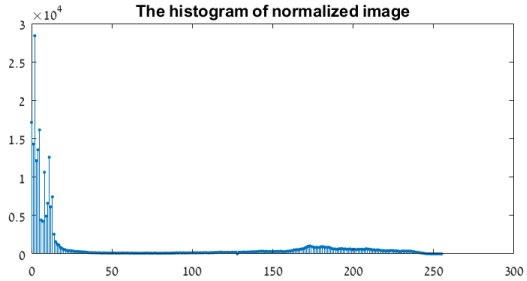
1.5 התבקשנו לבצע קוונטיזציה של התמונה המקורית עבור 6bit,4bit,2bit,1bit. התוצאות:

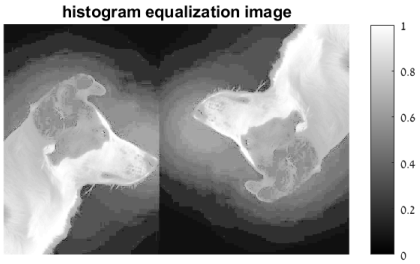


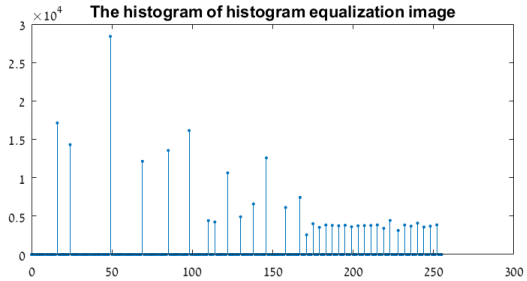
אפשר לראות שעבור הצגה של 1bit נקבל בתמונה את הערכים 0 או 1 בלבד.. וככל שנגדיל את מספר הביטים מספר הגוונים בתמונה ישתנה.

1.6 בסעיף זה התבקשנו לקרוא את התמונה של הכלב, להעביר אותה לסקאלה של צבעים אפורים ולנרמל את התמונה בין 0ל1. ע״י שימוש בפונקציית הנרמול שכתבנו בסעיף 1.1. לאחר מכן הפעלנו על התמונה את הפונקציה histeq של מאטלב ע״מ לקבל את histogram equalization של התמונה. הצגנו את שתי התמונות (הישנה והחדשה) יחד עם ההיסטוגרמות שלהן. התוצאות בעמוד הבא.





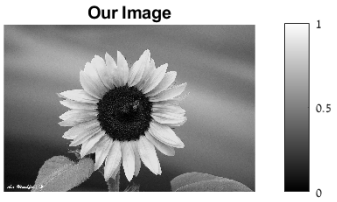


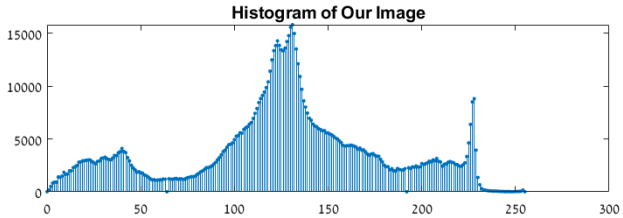


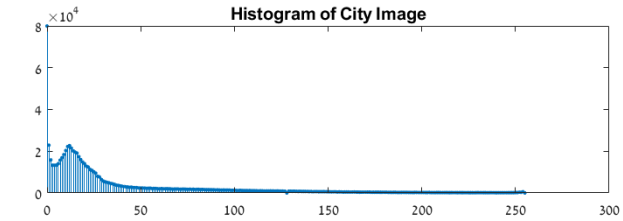
נשאלנו מדוע histogram equalization לא משפר את התמונה. הסיבה היא שהרקע מכסה את רוב התמונה, ולכן פיקסלים רבים מקבלים את הערך 0 מה שמשפיע על המיפוי מחדש של הפיקסלים : הפונקציה מבצעת ממוצע על ערך הפיקסלים בכל איזור, מכיוון שבתמונה הזאת קיימת ניגודיות גדולה בן הרקע לאובייקטים נקבל מיפוי של הפיקסלים הכהים השייכים לרקע לפיקסלים בהירים יותר וזה מה שגורם לירידה באיכות התמונה.

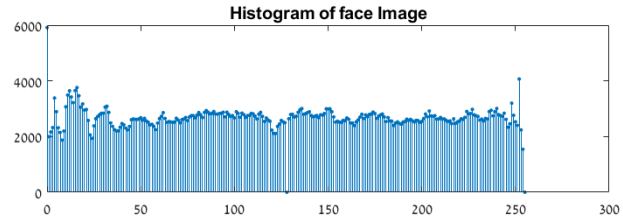
הפיתרון שלנו : נרצה ליצור רקע אחיד, לכן ניקח את כל הערכים הנמוכים וננחית אותם ל0, ועל התמונה החדשה נבצע את הפונקציה histeq של מאטלב, בדרך הזאת כאשר הפונקציה תבצע ממוצע על הפיקסלים של הרקע היא לא תשנה את ערכם וכך נפריד בין הרקע לאובייקטים.

1.7 בחרנו תמונה מהמחשב, העברנו אותה לסקאלה של אפורים ונרמלנו אותה בין 0 ל1 בעזרת הפונקציה שכתבנו בסעיף 1. ביצענו זאת גם עבור התמונה של העיר והתמונה של הפנים.
בנוסף חישבנו את ההיסטוגרמות של התמונות. התוצאות :

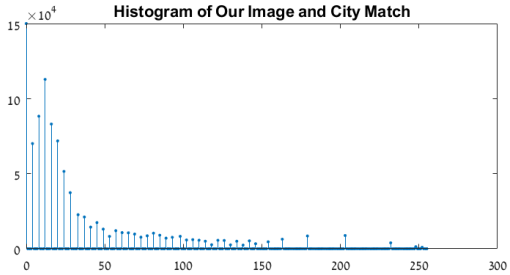


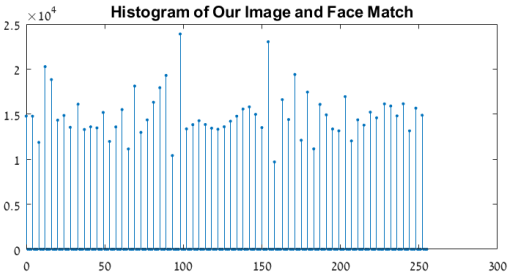






כעת ביצענו התאמה בין הסקאלה של העיר והפנים לסקאלה של התמונה שלנו ע״י שימוש בפונקציה imhistmatch של מטלב. התמונות החדשות ביחד עם ההיסטוגרמות שלהן :





התמונה הראשונה היא תוצאת של התאמה של התמונה שלנו ביחד עם התמונה של העיר. ניתן לראות שההיסטוגרמה של התמונה החדשה כמעט וזהה להיסטוגרמה של התמונה של העיר. במקרה הזה האיכות של התמונה נפגע. זאת מכיוון שההיסטוגרמה של התמונה שלנו מפוזרת לכל אורך הסקאלה לעומת התמונה של העיר שרובה מורכבת מפיקסלים נמוכים, לכן כאשר מיפינו מחדש קיבלנו תמונה בעלת גוונים כהים יותר ופחות ניגודיות בין הפיקסלים השונים.

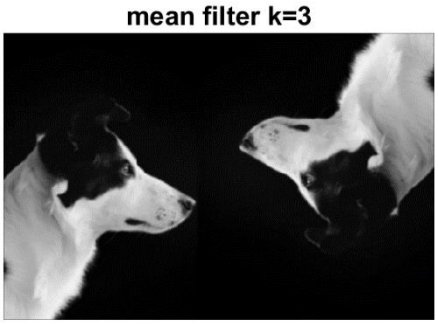
התמונה השניה היא התאמה בין התמונה שלנו לתמונה של הפנים וגם במקרה הזה קיבלנו היסטוגרמה דומה להיסטוגרמה של התמונה של הפנים. במקרה הזה מכיוון שבהיסטוגרמה

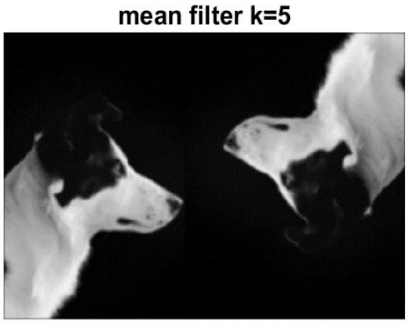
של התמונה שלנו יש יותר פיקסלים כהים מאשר בהיסטוגרמה של תמונת הפנים קיבלנו ירידה באיכות התמונה אך עדיין קיבלנו תמונה טובה יותר מאשר התמונה הראשונה.

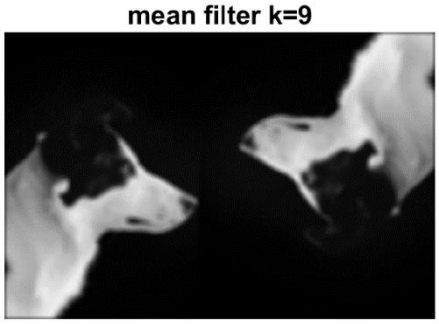
שאלה 2:

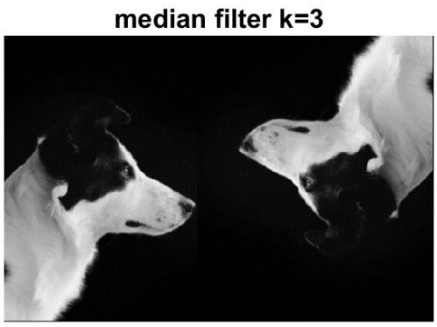
2.1. התבקשנו לקרוא לתמונת הכלב להעביר אותה לסקאלה של אפורים ולנרמל אותה. התמונה שהתקבלה:

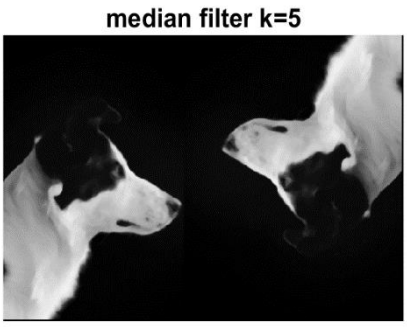
2.2. יצרנו שני מסננים, מסנן אחד שמחשב את הממוצע סביב פיקסל בתמונה, ומסנן נוסף שמחשב את החציון סביב הפיקסל. את mean filter יצרנו על ידי קונבולוציה עם מטריצה k*k, ואת median filter על ידי חלוקה של התמונה למטריצות k*k מסביב לכל פיקסל וחישוב החציון. מכיוון שאורך מוצא הקונבולציה עבור שורה ועמודה הוא n+k-1 גזרנו (k-1)/2 קצוות מארבעת הכיוונים מכיוון שבערכים אלה חישוב הממוצע והחציון כוללים את גבולות שלא קיימים. קיבלנו את התוצאות הבאות:

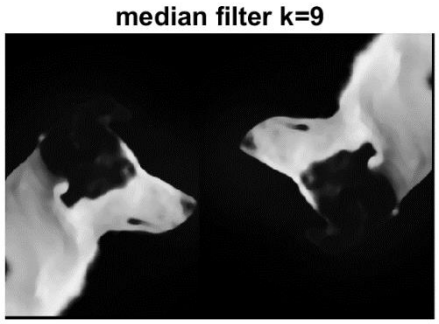










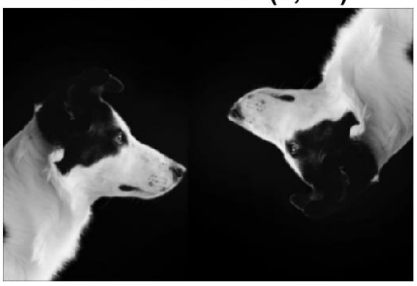


מה שמשותף לשני המסננים זה שככל שהגדלנו את k כך התמונה נהייתה יותר מטושטשת. תוצאה זו הגיונית מכיוון שככל שהערך של k גדל, חישוב הממוצע והחציון נפרס על פני יותר איברים מה שמגדיל את השונות של ערכי המוצא. אנו רואים שהתמונה שעברה במסנן mean יותר מטושטשת, ניתן להסביר זאת כך שבקצוות מתקיימים שינויים חדים בערכי הפיקסלים מה שיוצר שינוי חד גם בממוצע, לעומת זאת החציון מגיב בצורה יותר מתונה לשינויים אלה. במישור התדר שינויים חדים הם בעלי תדר גבוה, מכיוון מסנן הממוצע מגיב לא טוב לשינויים אלה נסיק שהוא סוג של LPF.

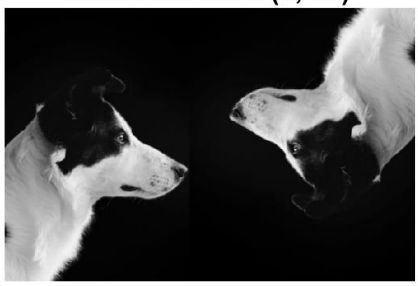
2.3. יצרנו מטריצת k*k שערכיה מדמים התפלגות גאוסית דו ממדית ונבצע קונבולוציה עם התמונה שלנו.

נגזור את הקצוות בצורה זהה לצורה שבה גזרנו בסעיפים הקודמים. התוצאות:

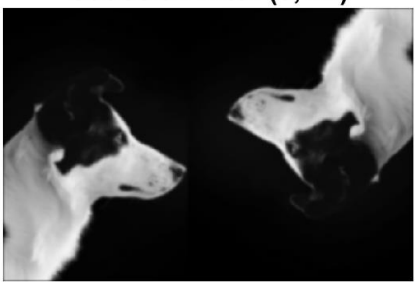
Gaussian filter (3,1.7)



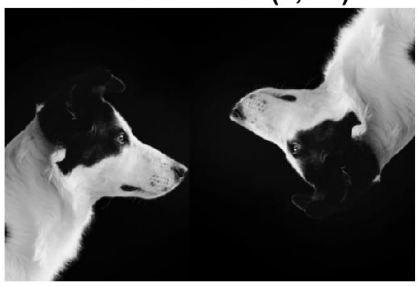
Gaussian filter (3,0.2)



Gaussian filter (9,1.7)



Gaussian filter (9,0.2)



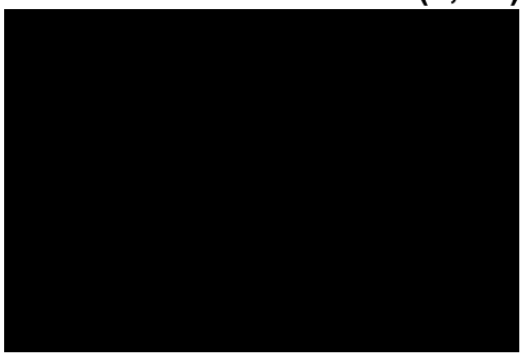
ניתן לראות שככל שהשונות של המסנן גדלה, התמונה נהיית יותר מטושטשת, זאת מכיוון שמסנן גאוסי הוא LPF, וככל שהשונות גדלה אז רוחב הסרט שלו קטן ומגיב פחות טוב לשינויים חדים בעלי ערכי תדר גבוהים.

נמשיך ונבצע החסרה בין התמונה לבין מוצאי המסנן. התוצאות בעמוד הבא:

subtract Gaussian filter (3,0.2)

subtract Gaussian filter (3,1.7)

subtract Gaussian filter (9,0.2)

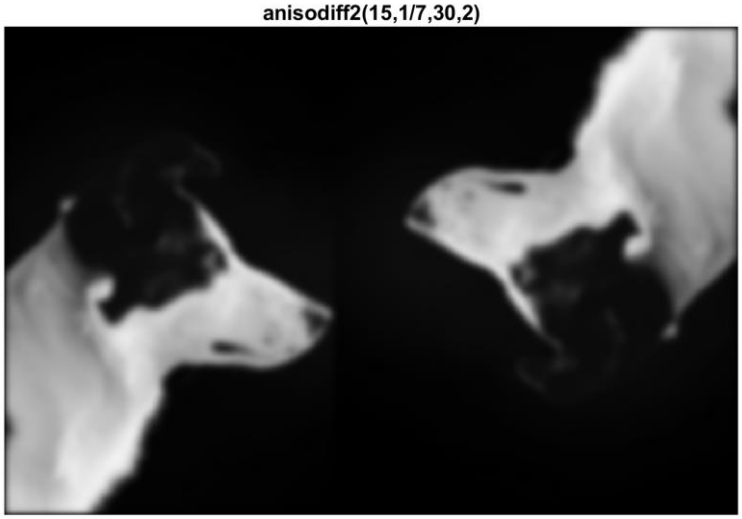


subtract Gaussian filter (9,1.7)

ניתן לראות שכאשר השונות היא 0.2 (שונות קטנה יחסית), המסנן מכיל את הקצוות אך לעומת זאת כאשר השונות היא 1.7 ניתן לראות שאחרי פעולת ההחסרה רק הקצוות נשארים (כלומר התדרים הגבוהים), תוצאה הגיונית בגלל היותו של המסנן LPF.

2.4. השתמשנו בפונקציה המצורפת והצבנו את הערכים שבדוגמה בתיאור הפונקציה. קיבלנו את המוצא הבא:

anisodiff2(15,1/7,30,2)

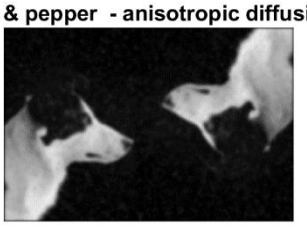


קיבלנו תמונה מטושטשת, לכן המסנן הוא LPF. המסנן משמש לניקוי רעשים לכן נסיק שהוא מנקה את הרעשים על ידי ניקוי התדרים הגבוהים.

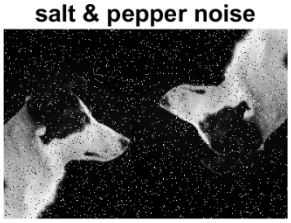
2.5. התבקשנו ליצור שלוש תמונות שונות באמצעות רעשים מסוגים שונים ולהעביר אותן דרך המסננים השונים.

תוצאות עבור תמונה מורעשת בsalt&pepper:

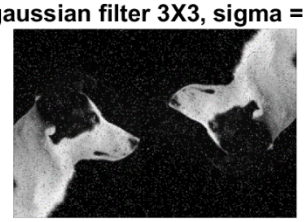
salt & pepper - anisotropic diffusion



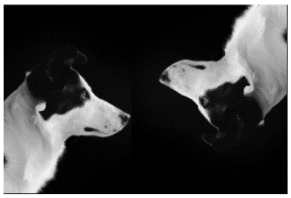
salt & pepper noise



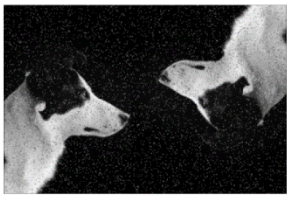
gaussian filter 3X3, sigma = 1



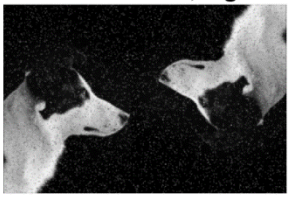
Median filter 3X3



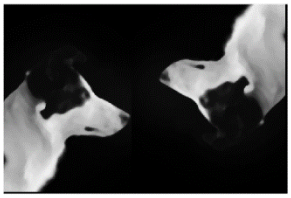
Mean filter 3X3



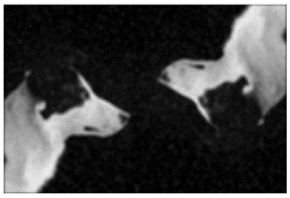
gaussian filter 9X9, sigma = 1



Median filter 9X9



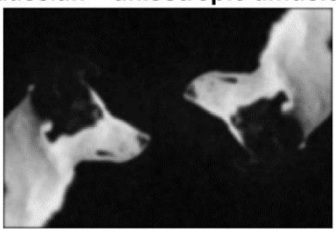
Mean filter 9X9



- **Median filter -** מסנן בצורה מעולה את התמונה עבור k=3, וקשה לראות הבדל גלוי לעין בין הפלט שלו לבין התמונה המקורית ולעומת זאת מורח את התמונה בk=9.
- **Gaussian filter -** מקטין בצורה משמעותית את הרעשים אך לא מנקה אותם לגמרי, ויוצר פלט מורעש פחות ולא מרוח.
- **Mean filter -** גם הוא מנקה את הרעש משמעותית אך לא לגמרי, ככל שk גדל הרעש נהיה פחות בולט אך התמונה נמרחת.
- **Anisotropic filter -** מוציא פלט עם רעש זניח אך התמונה מטושטשת.

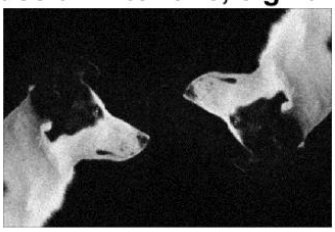
תוצאות עבור תמונה מורעשת ברעש גאוסיוני:

gaussian - anisotropic diffusion

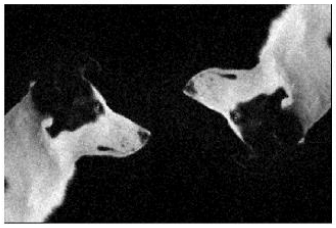


gaussian noise

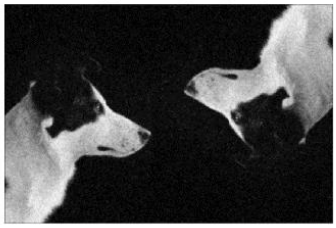
gaussian filter 3X3, sigma = 1



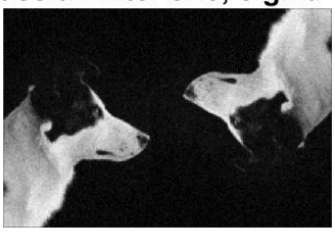
Median filter 3X3



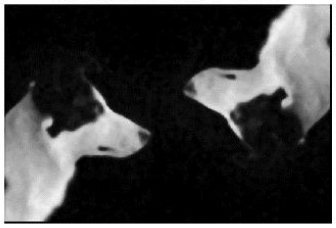
Mean filter 3X3

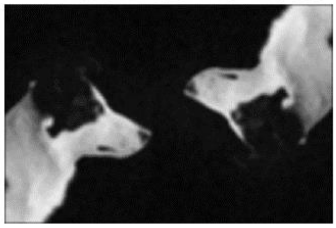


gaussian filter 9X9, sigma = 1



Median filter 9X9

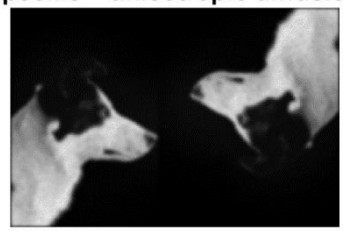




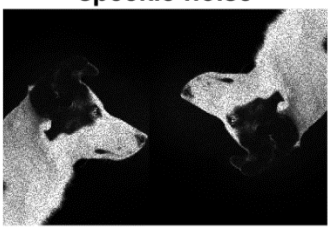
- **Median filter -** מוריד משמעותית את הרעש ומורח את התמונה ככל שk גדל.
- **Gaussian filter -** מקטין בצורה משמעותית את הרעשים ויוצר פלט מורעש פחות ולא מרוח.
- **Mean filter** – מקטין את הרעש משמעותית אך מורח את התמונה ככל שk גדל אפילו יותר ממסנן החציון.
- **Anisotropic filter** - מוציא פלט עם רעש זניח אך התמונה מטושטשת.

התוצאות עבור תמונה מורעשת ברעש בspeckle:

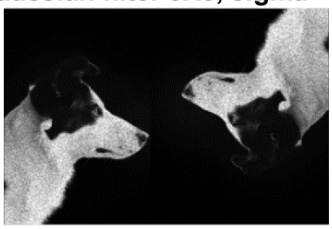
speckle - anisotropic diffusion



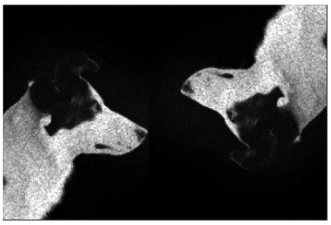
speckle noise



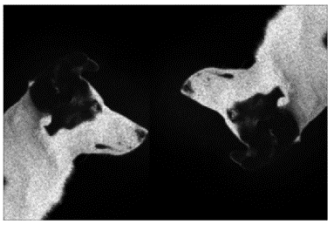
gaussian filter 3X3, sigma = 1



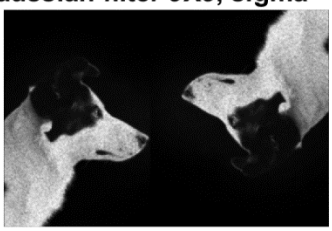
Median filter 3X3



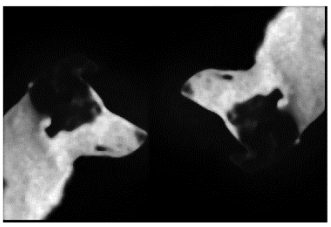
Mean filter 3X3



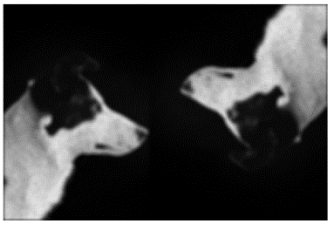
gaussian filter 9X9, sigma = 1



Median filter 9X9



Mean filter 9X9

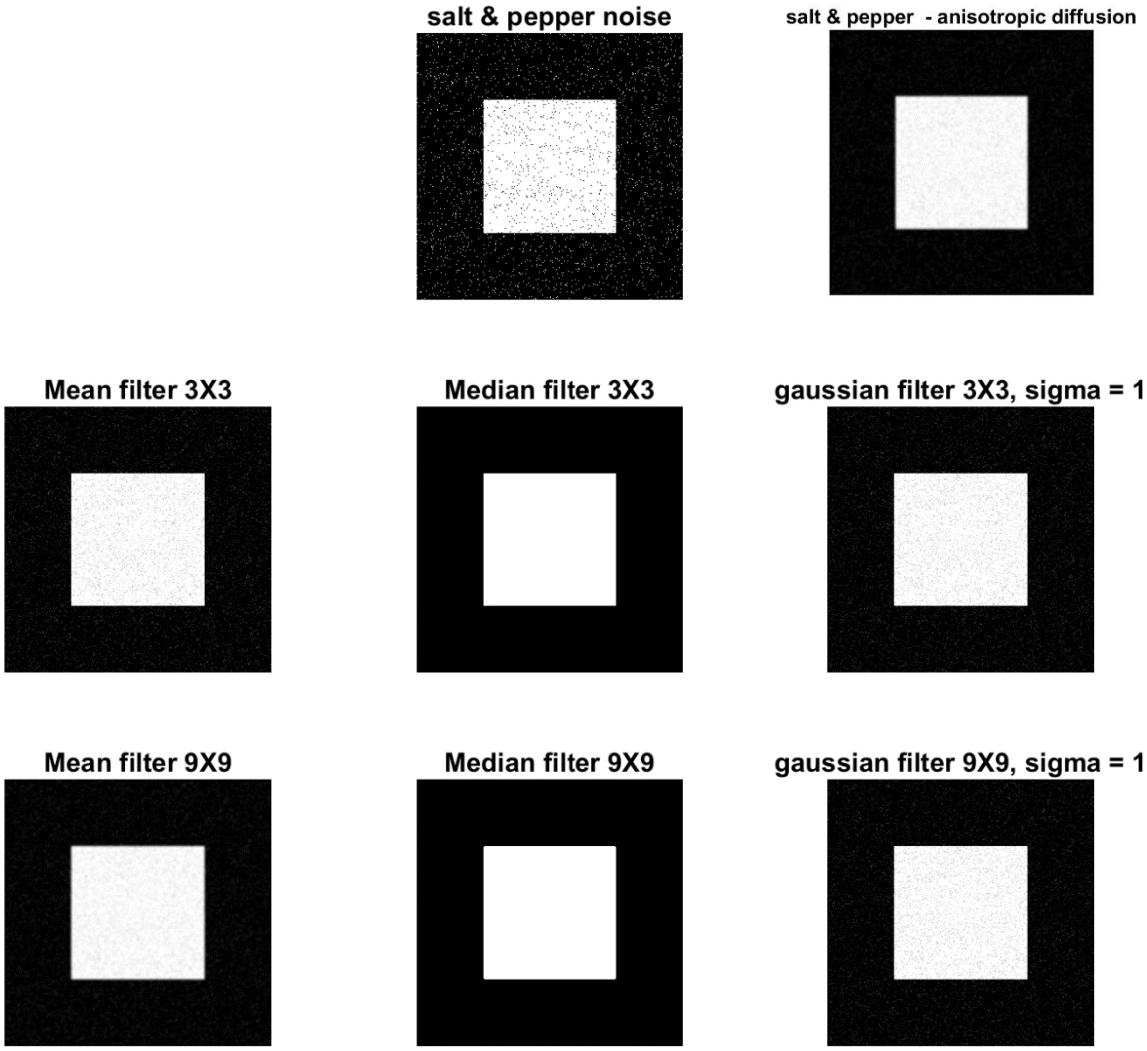


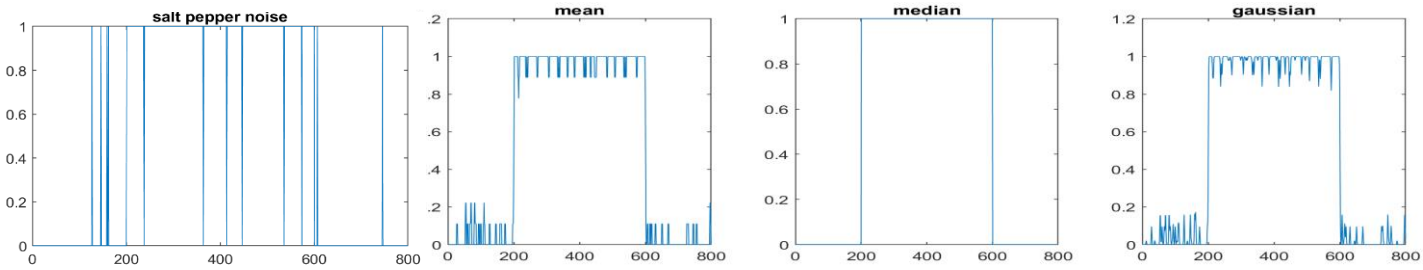
- **Median filter** – הפלט לא טוב לרעש מסוג זה, כמעט ולא מבחינים בהבדל בין התמונה המורעשת לתמונה בפלט. עבור k=9, ניתן להבחין גם ברעש וגם במריחה.
- **Gaussian filter** – מקטין משמעותית את הרעש, אך לא לגמרי.
- **Mean filter** – עבור k=3 מנקה את הרעש בצורה מאוד דומה למסנן הגאוסי, אך מורח את התמונה ככל שk גדל אפילו יותר ממסנן החציון.
- **Anisotropic filter** - מוציא פלט עם רעש זניח אך התמונה מטושטשת.

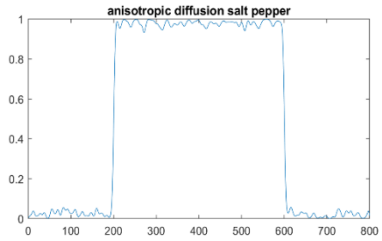
לקחנו את תמונת הריבוע, הרעשנו אותה באותם סוגי רעשים והעברנו את התמונה המורעשת באותם סוגי מסננים בצורה זהה לסעיפים הקודמים.

בנוסף נעביר קו משמאל לימין במרכז הריבוע (שורה 400) שחוצה את 800 הפיקסלים בשורה 400 ונמדוד את ערכי אותם הפיקסלים. ככל שהגרף יהיה יותר קרוב לפונקציית חלון בין 200 ל600 כך נסיק שהרעש קטן יותר.

נציג את תמונת הריבוע עבור רעש **salt & pepper**:







Median filter – בדומה לתמונה עם הכלב גם כאן ניתן להבחין שמסנן זה עושה עבודה נהדרת בסינון הרעשים עבור רעש זה. קיבלנו פונקציית חלון מושלמת למראית העין, ונראה שהריבוע שלנו לבן ונקי מרעשים.

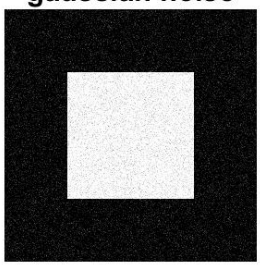
גם שאר המסננים מראים תוצאות זהות לתוצאות עם התמונה של הכלב.

ניתן לראות שהגרף של התמונה המורעשת הוא גרף מבולגן שלא מזכיר פונקציית חלון כלל וכלל, ולעומת זאת כל המסננים מוציאים פלט עם גרף של פונקציית חלון מורעשת.

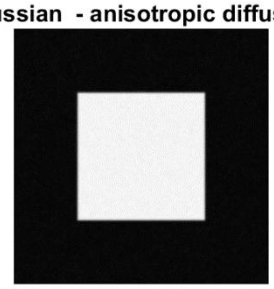
ניתן לראות בגרף של **Anisotropic filter** שהקפיצות של הרעש הן קפיצות עדינות ולא חדות כמו אצל השאר, נסיק שמסנן זה מעדן את הקפיצות של הרעש מה שמסביר את המריחה שהוא יוצר.

נציג את תמונת הריבוע עבור רעש **גאוסי:**

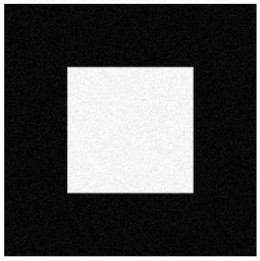
gaussian noise



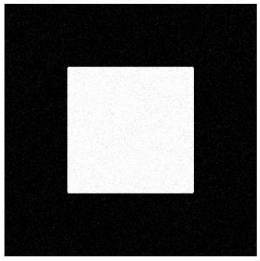
gaussian - anisotropic diffusion



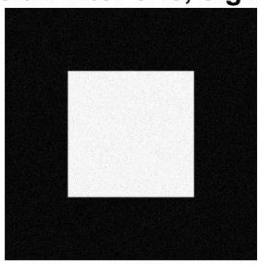
Mean filter 3X3



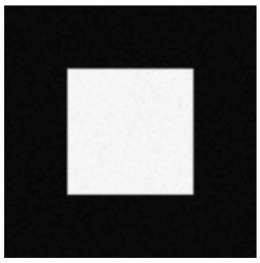
Median filter 3X3



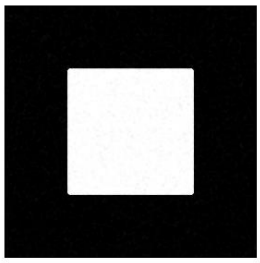
gaussian filter 3X3, sigma = 1



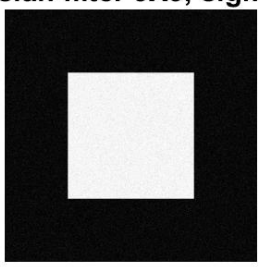
Mean filter 9X9

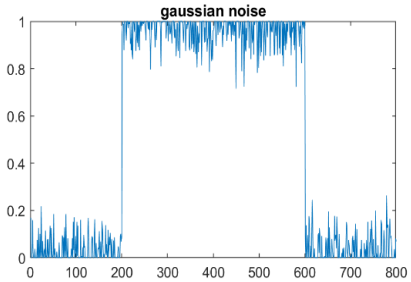


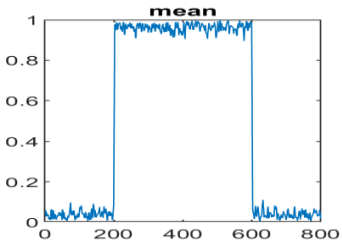
Median filter 9X9

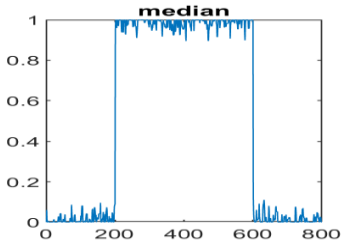


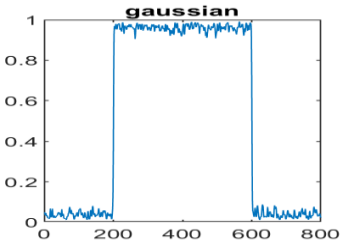
gaussian filter 9X9, sigma = 1

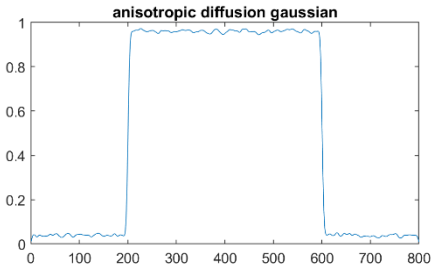








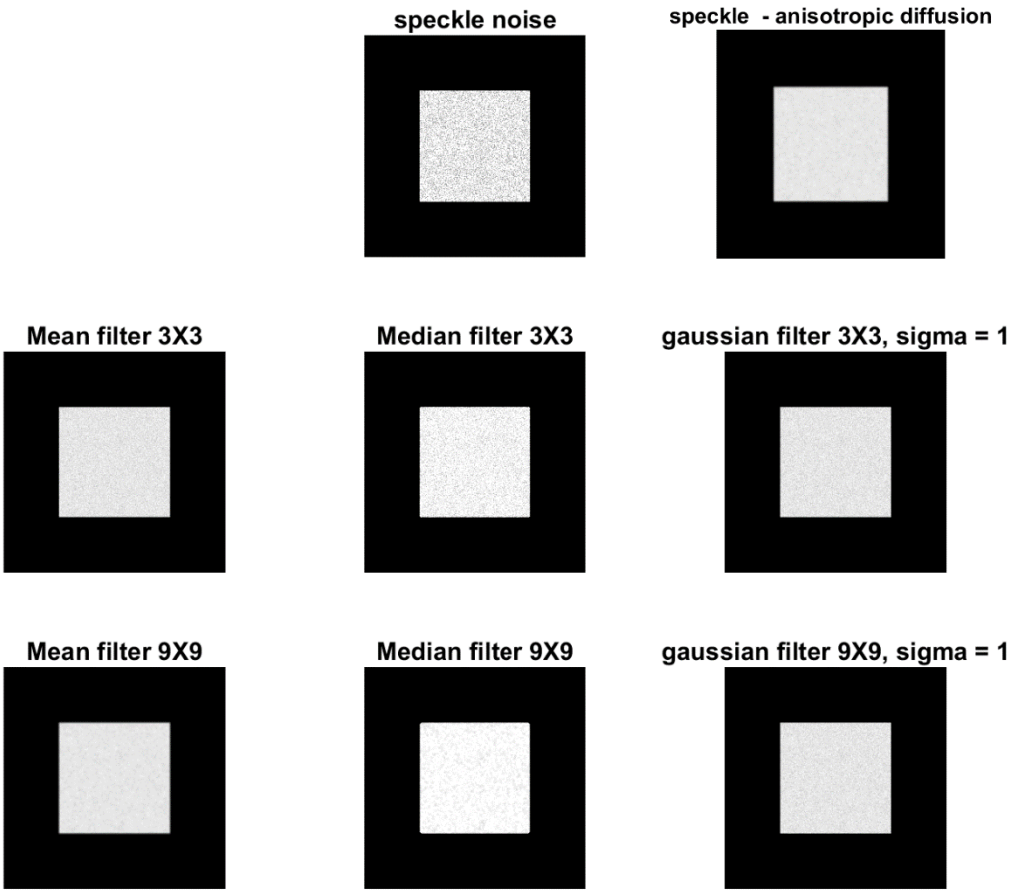


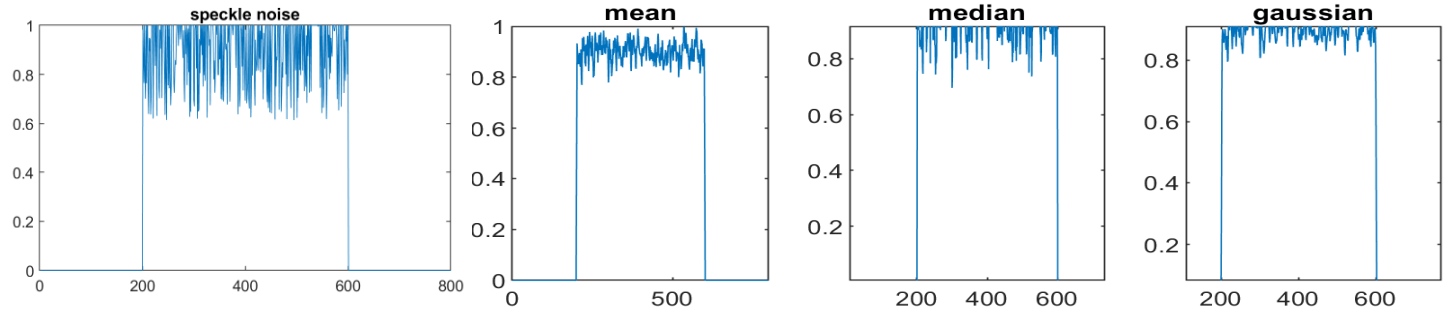


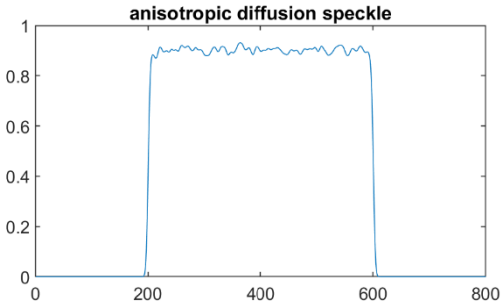
אם נסתכל על הגרף נראה שעבור **Anisotropic filter** נקבל חלון עם תנודות ממש קלות ולא חדות ובנוסף תמונת הפלט של מסנן זה נראית כמעט חסרת רעש. נתונים אלה יכולים לגרום לנו להסיק שזה הוא המסנן הכי טוב לניקוי ראש זה, אך בתמונה עם הכלב היה נראה שדווקא המסנן הגאוסי הוא המסנן הטוב ביותר, אז מה קורה פה בעצם?

נסיק שעבור ראש זה קיים סוג של trade off בין עוצמת הרעש לבין טשטוש התמונה. אמנם עבור **Anisotropic filter** נראה שהתנודות נורא קלות והרעש זניח אך בפועל התמונה מרוחה, לעומת זאת עבור המסנן הגאוסי אנו כן רואים את הראש בצורה בולטת אך התמונה יותר נראית כמו התמונה המקורית בתוספת רעש.

נציג את תמונת הריבוע עבור רעש **speckle**:







ניתן להבחין בגרפים שבהשוואה למסנן התוחלת והמסנן הגאוסי, למסנן החציון יש רעש פחות צפוף אך עם זאת תנודות הרעש יותר חדות, כלומר גם כאן יש trade off בין צפיפות הרעש לבין העוצמה שלו.

לסיכום:

- ראינו שיש מקרים רבים של trade off בין עוצמת הרעש לבין הטשטוש של התמונה, לעתים תמונה שתראה לנו פחות מורעשת תהיה מטושטשת, ולהפך – נראה תמונות מאוד ברורות עם רעש מפוזר עליהן.
- אין באמת מסנן יותר טוב, הכל תלוי במוצא שאנו רוצים לקבל, בסוג הרעש ובגורמים אחרים.
- ברוב המקרים לא הצלחנו לקבל תמונה שלא ניתן להבחין ברעש שלה, אך כן היה מקרה יוצא מין הכלל שבו קיבלנו תמונה שלא ניתן היה לראות הבדל בינה לבין התמונה המקורית וזה קרה עבור מסנן החציון שקלט תמונה עם רעש salt & pepper.