

תרגיל Matlab מספר 1

הוראות כלליות

- יש לשמור ולטעון תמונות איתם תעבדו **בספרייה אחת מעל** הספרייה בה נמצאים קבצי הקוד. לדוגמה:
`>> I = imread('..\Lena.jpg');`
- יש להימנע ככל האפשר משימוש בלולאות בקוד, ולעשות שימוש בכתוביב מטריצי.
- יש להגיש את כל התרגיל כסקריפט יחיד. ניתן ורצוי לממש שאלות או חלקים מהשאלות כפונקציות (לא לשכוח לצרף אותם!!).
- יש להקפיד לתת כותרת מתאימה לכל תמונה או גרף, וגם לצירים כאשר זה רלוונטי. יש לדאוג שגרפים ותמונות יוצגו בגודל מספק להבנת תוכנם.
- יש לכתוב כל סעיף ב-cell שונה. יוצרים cell חדש ע"י כתיבת התווים '%' בתחילת שורה ריקה. (תווים אלו מבדילים בין cells). התווים הופכים להיות מודגשים ואז אפשר להפעיל בנפרד את ה-cell הנוכחי, הצבוע בצהוב, בעזרת Ctrl+Enter. לכל cell יש לרשום את מספר הסעיף ככותרת.
- יש לתעד את הקוד באופן סביר ובמיוחד במקומות בהן מבוצעות פעולות לא טריוויאליות.
- את התרגיל יש להגיש אלקטרונית **דרך Moodle**, בחלק של Matlab Assignments בצורה הבאה:
יש לשמור את קבצי התרגיל שלכם ללא תמונות קלט/פלט כלשהן, בקובץ ZIP כך:
ex1_<ID1>_<ID2>.zip
כאשר <ID1> הוא מספר הסטודנט של אחד המגישים ו-<ID2> הינו מספר הסטודנט של המגישים השני. לדוגמא: **ex1_012345678_987654321.ZIP** עבור תרגיל של מגישים בעלי מספרי הסטודנט 987654321, 012345678.
- ההגשה ב-moodle תעשה רק ע"י אחד מבני הזוג. יש לוודא כי גם שם קובץ ה-zip וכן גם העמוד הראשון של קובץ ה-pdf יכילו את מספרי ת.ז. של שני המגישים. יש ללחוץ על כפתור ההגשה עבור הגשה סופית.
קובץ ה-ZIP הנ"ל יכלול את הקבצים הבאים:
 - סקריפט הרצת התרגיל וכל יתר הקוד הרלוונטי הנדרש להשגת כל התוצאות.
 - קובץ PDF יחיד עם תשובות לשאלות התיאורטיות ותמונות/גרפים רלוונטיים.
 - אין צורך להגיש את התמונות או את קבצי העזר שניתנו לכם.
- את התרגיל יש להגיש (מומלץ בזוגות) עד ליום א' בתאריך **26.11.2017** בשעה **23:55**.
על כל יום איחור בהגשה (שלא אושר מראש ע"י סגל הקורס) יורדו 4 נקודות אוטומטית. הגשה מאוחרת תעשה ישירות לאחראי ה-Matlab ולא דרך Moodle.
- התייעצות עם חברים מותרת ואף מומלצת, אולם את הקוד עליכם לכתוב בצורה עצמאית. הסגל יתייחס בחומרה המקסימלית להענקות.
- מספר קבצי עזר ב-Matlab מצורפים באתר. כמו כן ניתן להיעזר כמובן ב-help של Matlab.
- שאלות לגבי התרגיל ניתן להפנות לאחראי ה-Matlab דרך הפורום המתאים ב-Moodle או במייל.

בהצלחה!

הצעות לפונקציות שימושיות בגליון זה:

imshow, imread, im2double, rgb2gray, figure, plot, subplot, randperm, hist, imhist, cumsum.

הערות כלליות

הפונקציה imshow:

הפונקציה imshow מצפה לקבל מטריצה שאבריה הם unit8 בתחום 0,255 "בדיד" או בייצוג Double בקטע [0,1] "רציף" (התחום הדינאמי נקבע ע"פ הפורמט). אלו פורמטים מקובלים לתמונות. שימוש בסוגריים ריקים, כך: imshow(), בוחר את טווח התמונה בפועל כתחום הדינאמי המוצג. אפשר גם לציין במפורש את גבולות התחום ע"י [low,high]. יש לשים לב כי ערכים הנמצאים מחוץ לתחום יקבלו את ערכי הקצוות.

1. קוונטיזציה וקטורית אופטימלית

בשאלה זו אין להשתמש בפונקציות מובנות ב-Matlab המבצעות קוונטיזציה וקטורית.

א. פתחו את הקובץ scenary.mat בעזרת load. הקובץ מכיל מטריצה 4 ממדית שהיא למעשה סרטון. הריצו אותו בעזרת פקודת imshow.

הגדרת הממדים במטריצה:

מימד 1 – שורות בתמונה.

מימד 2 – עמודות בתמונה.

מימד 3 – ייצוג הצבע בתמונות (3 מטריצות R,G,B).

מימד 4 – זמן.

ב. ממשו קוונטייזר Max-Lloyd עבור תמונות צבע. המימוש יעשה ע"י פונקציה ובה המשתנים:

data	–	מטריצה תלת-ממדית אשר מייצגת תמונת צבע (RGB).
levels	–	מספר ווקטורים מייצגים רצוי (גודל המילון).
meps	–	השיפור היחסי המינימלי.
dataout	–	מטריצה תלת-ממדית המכילה את התמונה לאחר הקוונטיזציה.
distortion	–	וקטור באורך של מספר האיטרציות, המכיל את העיוות הממוצע שהתקבל בכל איטרציה (ע"פ שגיאה ריבועית ממוצעת).
QL	–	מטריצה בגודל (levels x 3) המכילה את וקטורי המילון (codebook).

חשבו את העיוות הריבועי הממוצע D_m בכל איטרציה ואת ערך השיפור היחסי $\varepsilon = \frac{|D_m - D_{m+1}|}{D_m}$. קבעו את תנאי העצירה ל- $\varepsilon \leq meps$.

הדרכה: לשם המימוש ניתן להיעזר בחוברת Image Processing Booklet המופיעה ב- Moodle. קוונטיזציית צבע ווקטורית מוסברת בפרק 4 החל מעמוד 71, ובעמודים 72-73 מוסבר האלגוריתם. שימו לב **שאין לבצע קוונטיזציה לכל ערוץ בנפרד**, אלא להתייחס לפיקסלים כנקודות במרחב תלת ממדי (R,G,B) כמצוין לעיל. יש להשתמש במטריקה אוקלידית. שימו לב שבניגוד למימוש עבור תמונה דו-ממדית, כאן מקטעי ההחלטה הם למעשה אזורי החלטה, ומוצאים אותם באמצעות ערכי התמונה עצמם.

הערות לכל סעיפי שאלה זו:

- (1) על הערכים של dataout להיות שלמים.
- (2) כאשר אתם מאתחלים את הקוונטייזר, בחרו את רמות האתחול בצורה אקראית, אך ודאו שהן שייכות לצבעי התמונה המקורית (data) ואין רמות זהות.
- (3) על מנת לממש אלגוריתם איטרטיבי תצטרכו לממש לולאה. קבעו מראש מספר איטרציות מירבי הגיוני (נניח 20), כך שהקוד יסיים לרוץ גם אם יש לכם טעות בתנאי העצירה (וגם תוך זמן סביר).

טיפים למימוש:

- על מנת לסדר את הווידאו בצורה נוחה לעבודה כתמונה, מומלץ להשתמש בפקודות permute (שינוי סדר ממדי המטריצה) ו- reshape (פריסת המטריצה כאובייקט תלת-ממדי).
- על מנת למצוא שיוך של פיקסלים לרמות ייצוג (QL), מומלץ להשתמש בפקודות repmat (להגדלת מטריצות ע"י שכפול) ו-min (למציאת מרחק מינימלי). שימו לב שתוצאת פעולת ה-min יכולה לשמש אתכם גם לחישוב רמות הייצוג הבאות.
- יש לשמור בין איטרציות רק אובייקטים רלוונטיים (על מנת למנוע בעיה של עומס על הזיכרון וכמו כן תרגול של בניית קוד נכון יותר לאלגוריתמים "סקלביים" - scalable).

ג. ראשית נבדוק נכונות לפונקציה באמצעות תמונה בודדת. טענו את התמונה scotland.jpg והפעילו עליה את הקוונטייזר עבור $levels = [4, 8, 16, 32]$. השוו באופן איכותי עם התוצאות הבאות:

original



quantized to 4 colors



quantized to 8 colors



quantized to 16 colors



ד. הפעילו את הקוונטייזר שמימשתם על הווידאו scenario.mat (להלן "נאיבי"), כאשר השיפור המינימלי נקבע להיות $meps=0.02$ ומספר הרמות הינו $levels=10$. הציגו גרף של העיוות הממוצע כפונקציה של האיטרציות. הערה: על מנת להריץ את הפונקציה שמימשתם בסעיף ב' על סרטון ווידאו יש לפרוש את הווידאו כ-"תמונה ארוכה". כלומר, שרשרו את המסגרות (תמונות הווידאו) לקבלת מטריצה תלת-ממדית אחת. לאחר הקוונטיזציה, החזירו את המידע למבנה של סרטון והציגו אותו. הציגו את ערכי השגיאה על כל מסגרת ועל כל הסרטון.

ה. כעת נציע דרך לשפר את תוצאות הקוונטיזציה. יש לחלק את הווידאו לשני חלקים ובכל חלק לבצע קוונטיזציה למספר רמות $levels=10$ (להלן "משופר"), כאשר השיפור היחסי המינימלי הינו $meps=0.02$. מצאו באופן אוטומטי מהי המסגרת שבה יש לחלק את הווידאו לשני חלקים על מנת לקבל תוצאה מיטבית (שגיאה מינימלית), ורשמו אותה. הציגו גרף של העיוות הממוצע כפונקציה של האיטרציות (עבור מסגרת החלוקה האופטימלית שהתקבלה), וגרף של העיוות הממוצע הכולל כפונקציה של מסגרת החלוקה (עיוות ממוצע סופי). הערות:

- מדד האיכות נשאר שגיאה ריבועית ממוצעת. שימו לב שיש למדוד את השגיאה על כל הווידאו.
- מציאת המסגרת נעשית באופן אוטומטי כדי שהקוד יוכל לעבוד גם על סרטוני וידאו נוספים.
- זמן הריצה עלול להיות ארוך! מומלץ להשקיע בכתיבת קוד נכון ובדיקתו (גם ע"י הרצה על מידע חלקי ומוקטן) לפני הבדיקה על הסרטון המלא ולא לבזבז הרבה זמן על הרצות.

ו. הציגו את הווידאו לאחר הקוונטיזציה המשופרת עם מסגרת החלוקה האופטימלית. השוו בין שתי השיטות ("נאיבי", "משופר"). בהשוואתכם התייחסו למדד איכותי (מראה הווידאו) ולמדד כמותי (שגיאה ריבועית ממוצעת).

ז. הסבירו באופן איכותי כיצד ניתן לאתחל את האלגוריתם לקוונטיזציית צבע בצורה חכמה יותר (לא אקראית).

2. פעולות נקודה ועיצוב היסטוגרמה

בשאלה זו אינכם רשאים להשתמש בפקודות histeq, adapthisteq, brighten, imadjust כמו כן, אין להשתמש ב-scaling בהצגת תמונות המוצא.

פעולת Power-Law Transformation מוגדרת ע"י הפונקציה $s(r) = cr^\gamma$.

כאשר: r – עוצמת הפיקסל (ערכים ב-[0,1]). γ, c – קבועים חיוביים.
 s – עוצמת הפיקסל לאחר הטרנספורמציה.

א. נרצה לבחון את תכונות הטרנספורמציה. איירו את הפונקציה s עבור ערכי הקבועים

$$\gamma = [0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1, 2, 3, 5, 10, 20], c = 1$$

עבור רמות האפור [0,255]. שימו לב כי יש לנרמל את רמות האפור לתחום [0,1] לפני הטרנספורמציה, ולהחזירם לתחום [0,255] לאחריה.

צינו מה מבצעת הטרנספורמציה עבור $\gamma > 1, \gamma < 1, \gamma = 1$ ומה השפעתו של הקבוע c ?

ב. טענו את התמונה Maria.jpg ובחנו את ההיסטוגרמה שלה.

קעת הפעילו את Power-Law Transformation על התמונה.

במימושכם, לצורך יעילות חישובית, בנו ראשית LUT (lookup table) של כל ערכי הטרנספורמציה עבור γ, c מסוימים בכל תחום רמות האפור [0,255] והפעילו את הטרנספורמציה על התמונה באמצעות שימוש ב-LUT. מצאו את ערכי γ, c שיביאו לתוצאות אופטימליות לדעתכם.
הציגו את תמונת המוצא לצד התמונה המקורית. בחנו את ההיסטוגרמה של תמונת המוצא.

המלצה: לצורך ייעול של תהליך כתיבת הקוד, מומלץ לקרוא ולהבין קודם את הנדרש בסעיפים ג'-ד' לפני המימוש בקוד. ניתן לממש פונקציה אחת שתשמש אתכם בסעיפים ג'-ד'.

ג. ממשו פונקציה המבצעת יישור היסטוגרמה (histogram equalization) לתמונה. הפעילו פונקציה זו על התמונה Maria.jpg. הציגו את ה-LUT, את התמונה המתקבלת ואת ההיסטוגרמה שלה. האם אכן קיבלתם היסטוגרמה אחידה? הסבירו.

ד. כדי לשפר את תוצאות הסעיף הקודם נרצה לבצע יישור היסטוגרמה רק עבור פיקסלים החל מערך סף מסוים. הסבירו מדוע צפוי שיפור כאשר נעשה זאת.

שנו את האלגוריתם של הסעיף הקודם כך שיבצע את היישור רק עבור פיקסלים הגדולים מערך סף T , ואילו פיקסלים הקטנים או שווים לערך הסף יישארו כפי שהם בתמונת המקור. קבעו ערך סף T המתאים לתמונה, והציגו את התוצאות. השוו לתוצאות סעיפים ב' ו-ג' והסבירו באופן איכותי עבור אילו תמונות שיטה זו טובה.

3. הזזה וסיבוב במקום

בשאלה זו אֵין להשתמש בפונקציה `imrotate`.

ניתן לבצע הזזת תמונה בחלקי פיקסלים ע"י שימוש באינטרפולציה בילינארית. באינטרפולציה זו, כאשר מתקבלים ערכי אינדקסים לא שלמים. משתמשים בסכום משוקלל של ארבעת הפיקסלים הסמוכים לנקודה לא שלמה. החישוב מתבצע באופן הבא:

$$(*) \quad g(x,y) = [1-\alpha, \alpha] \begin{bmatrix} f(i,j) & f(i,j+1) \\ f(i+1,j) & f(i+1,j+1) \end{bmatrix} \begin{bmatrix} 1-\beta \\ \beta \end{bmatrix}$$

כאשר α ו- β הם קבועים בקטע $[0,1]$ המהווים את החלק הלא שלם באינדקס.

א. כתבו פונקציה המבצעת הזזה ב- $[dx,dy]$ בעזרת אינטרפולציה בילינארית, כאשר ידוע כי dx, dy בקטע $[0,1]$.

ב. היעזרו בפונקציה שמימשתם בסעיף א' כדי לממש פונקציה אשר מבצע על התמונה הזזה ציקלית באינדקס לא שלם כלשהו $[dx,dy]$ (לאו דווקא בקטע $[0,1]$).

ג. טענו את התמונה `cameraman.jpg` והפכו אותה לתמונת אפור בטווח $[0,1]$. השתמשו ב-`rgb2gray` וב-
`im2double`. הפעילו את הפונקציה שמימשתם בסעיף ג' על תמונת האפור עם $[dx,dy] = [170.3, 130.8]$.
הערה: שימו לב כי גודל התמונה לא אמור להשתנות כתוצאה מהאינטרפולציה הבילינארית.

כעת טענו את התמונה `Brad.jpg` בעזרת הפקודה `imread`.

ד. צרו מסכה אשר מכילה את הערך 1 בפנים ואת הערך 0 בחוץ. המסכה תהיה בצורה של חצי עיגול תחתון סביב פני הדמות. קראו למסכה זו `mask1` והציגו אותה.

ה. הציגו את התמונה רק במקומות שבהם ערך המסכה הוא 1. תמונה זו תקרא `brad_win`.

1. מטריצת ההעתקה הדרושה כדי לסובב תמונה בזווית כלשהי α היא $\begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$.

כתבו פונקציה אשר מקבלת תמונה כלשהי ומסובבת אותה בזווית α סביב מרכז התמונה. שימו לב שכאשר מבצעים את הסיבוב ע"י מעבר על כל האינדקסים בתמונת המקור יהיו מקרים בהם יהיו אינדקסים בתמונת המסובבת שלא יקבלו כל ערך, ולכן ייווצרו בה חורים. לכן, על מנת לכסות את כל הפיקסלים בתמונה החדשה יש להשתמש במטריצה ההפכית למטריצת ההעתקה שיצרתם. במקרה שבו מתקבלים אינדקסים לא שלמים יש לעגל אותם לערכים השלמים הקרובים ביותר (nearest neighbor). הפעילו את הפונקציה על התמונה של brad_win. השתמשו בזוויות 45° , 60° ו- 90° . הציגו את התוצאות. מהן הבעיות המתקבלות בתמונה המסובבת ומדוע?

2. כעת נעזר באינטרפולציה ביליניארית בתהליך סיבוב התמונה. שוב השתמשו במטריצת ההעתקה שמצאתם בסעיף הקודם. כעת, כאשר מתקבלים ערכי אינדקסים לא שלמים, השתמשו בסכום משוקלל של ערכי ארבעת הפיקסלים הסמוכים לנקודה שמצאתם לפי נוסחה (*) שבתחילת השאלה. השוו לתוצאות שקיבלתם בסעיף הקודם. איזו תוצאה טובה יותר ומדוע? **הציגו דוגמאות לתשובתכם.** האם ישנם מקרים בהם זה לא משנה?