

## תרגיל מחשב מספר 1

### הוראות כלליות

1. יש לשמור ולטעון את התמונות שקשורות לתרגיל **בספרייה אחת מעל** הספרייה בה נמצאים קבצי הקוד.
2. יש להימנע ככל האפשר משימוש בלולאות בקוד, ולעשות שימוש בכתוב מטרצי.
3. יש להגיש את כל התרגיל כסקריפט/מחברת יחידה. ניתן ורצוי לממש שאלות או חלקים מהשאלות כפונקציות (לא לשכוח לצרף אותם!!).
4. יש להקפיד לתת כותרת מתאימה לכל תמונה או גרף, וגם לצירים כאשר זה רלוונטי. יש לדאוג שגרפים ותמונות יוצגו בגודל מספק להבנת תוכנם.
5. יש להפריד ויזואלית בין סעיפים שונים בקוד.
6. יש לתעד את הקוד באופן סביר ובמיוחד במקומות בהן מבוצעות פעולות לא טריוויאליות.
7. את התרגיל יש להגיש אלקטרונית דרך **Moodle** בצורה הבאה:  
יש לשמור את קבצי התרגיל שלכם ללא תמונות קלט/פלט כלשהן, בקובץ ZIP כך:  
**code1\_<ID1>\_<ID2>.zip**  
כאשר <ID1> ו-<ID2> הם מספרי ת.ז. של המגישים. לדוגמה: **code1\_012345678\_987654321.ZIP**  
עבור תרגיל של מגישים בעלי מספרי ת.ז. אלו.  
קובץ ה-ZIP הנ"ל יכול את הקבצים הבאים:  
א. קובץ readme.txt עם שמות המגישים ותעודות הזהות שלהם.  
ב. סקריפט/מחברת הרצת התרגיל וכל יתר הקוד הרלוונטי הנדרש להשגת כל התוצאות.  
ג. קובץ PDF יחיד עם תשובות לשאלות התיאורטיות. יש לרשום את פרטי המגישים במסמך.  
ד. אין צורך להגיש את התמונות או את קבצי העזר שניתנו לכם.
8. את התרגיל יש להגיש עד לתאריך **8.1.2026 בשעה 23:59**.
- איחור בלתי סביר בהגשה וללא סיבה מוצדקת יפגע בציון התרגיל.
9. התייעצות עם חברים מותרת ואף מומלצת, אולם את הקוד עליכם לכתוב בצורה עצמאית. הסגל יתייחס בחומרה המקסימלית להעתקות.
10. שאלות לגבי התרגיל ניתן לרשום בפורום המתאים ב-Moodle או במייל.

**בהצלחה!**

## 1. דגימה ושחזור

$$\text{נתונה הפונקציה } f_1(x, y) = \frac{\sin(\pi \frac{50}{21} y)}{\pi y} = \text{sinc}(\frac{50}{21} y)$$

ברצוננו לדגום את הפונקציה לקבלת תמונה ספרתית בגודל  $201 \times 201$  פיקסלים.

- א. דגמו את הפונקציה במרווחי דגימה  $\Delta = \Delta x = \Delta y = 0.1$ , ובאופן סימטרי. כלומר יש לדגום בקטע  $[-100\Delta, +100\Delta]$ .
- ב. הציגו את התמונה המתקבלת כתמונת גווני אפור. הקפידו לנרמל את ערכי התמונה לפני ההצגה. הסבירו את התמונה המתקבלת. מהו המחזור בפיקסלים של החלק המחזורי בפונקציה? כיצד הדבר תואם למחזור התיאורטי של החלק המחזורי בפונקציה?
- ג. הציגו את התמונה במישור התדר כתמונת גווני אפור (אמפליטודה בלבד). הסבירו את התמונה המתקבלת. התייחסו בהסבר למיקום האנרגיה בפיקסלים בתמונה, וכיצד הדבר מתיישב עם התיאוריה.
- ד. חזרו על סעיפים א'-ד' עבור  $\Delta = 1$ . הסבירו את ההבדלים בין המקרים.
- ה. האם ישנן הגבלות על מרווח הדגימה בפונקציה הנתונה ובגודל התמונה הנתון? אם יש, ציינו מהן. בכל מקרה, נמקו תשובתכם.

## חלק ב'

- ו. פתחו את הקובץ `SpongeBob.mat`, הקובץ מכיל מטריצה תלת ממדית בה שני הממדים הראשונים הינם השורות והעמודות בתמונה והממד השלישי הינו הזמן. צפו בסרטון.
- ז. מצאו מהו זמן מחזור הריצה של בוב-ספוג (ביחידות של מסגרות (frames) לסיבוב) ומכך חשבו מהו תדר הסיבוב. הציגו מחזור ריצה שלם.
- ח. מהו תדר Nyquist לדגימה בזמן של סרטון זה? התייחסו לתדר הסיבוב כתדר המקסימלי. הערה: על מנת לבאר את התהליך, ניתן להכפיל את התדרים שמצאתם ב-N מסגרות לשנייה, כך שיתקבל תדר ביחידות של סיבוב לשנייה (N הוא קצב המסגרות לשנייה בקוד המציג את הסרטון).
- ט. דגמו את הסרטון כך שבוב-ספוג ירוץ אחורה. מהו תדר הדגימה המתאים לכך? על מנת לראות את הסרטון טוב יותר, ניתן לשנות את מספר המסגרות לשנייה.
- י. הסבירו מה מתרחש בדגימה אשר גורם לבוב-ספוג לרוץ אחורה.

## 2. אינטרפולציה

הערה: בשאלה זו **אסור** להשתמש בפונקציית ספריה המבצעת אינטרפולציה או בפונקציית סיבוב תמונות. 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  $\alpha$  ו- $\beta$  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  $dy, dx$  בקטע  $[0,1]$ .

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

ג. טענו את התמונה cameraman.jpg והפכו אותה לתמונת אפור בקטע  $[0,1]$ . הפעילו את הפונקציה שממשתם 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.

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

אשר מקבלת תמונה כלשהי ומסובבת אותה בזווית  $\alpha$  סביב מרכז התמונה. שימו לב שכאשר מבצעים את הסיבוב ע"י מעבר על כל האינדקסים בתמונת המקור יהיו מקרים בהם יהיו אינדקסים בתמונת הסיבוב שלא יקבלו כל ערך, ולכן ייווצרו חורים בתמונה המסובבת. לכן, על מנת לכסות את כל הפיקסלים בתמונה החדשה יש להשתמש במטריצה ההופכית למטריצת ההעתקה שיצרתם, ולמצוא את ערך המקור על פי היעד. 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  $0^\circ$ ,  $45^\circ$  ו- $90^\circ$ . הציגו את התוצאות. מהן הבעיות המתקבלות בתמונה המסובבת ומדוע?

ז. 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? האם ישנם מקרים בהם זה לא משנה?

### 3. קוונטיזציה וקטורית אופטימלית

בשאלה זו מומלץ להיעזר בפרק 4, עמוד 72 בחוברת של מיכאל אלעד המופיעה ב-moodle. אין צורך לממש את הקוונטיזציה בצורה יעילה (אבל מומלץ). מי שמעוניין בכך יכול לדלג על מימוש עצמי ולהיעזר בספריית Scikit-Learn.

- א. ממשו קוונטיזציה Max-Lloyd לוקטורים עבור תמונות צבע, כאשר:
- data – מטריצה תלת-ממדית אשר מייצגת תמונת צבע (RGB).
  - levels – מספר וקטורים מייצגים רצוי (גודל המילון).
  - meps – השיפור היחסי המינימלי.
  - dataout – מטריצה תלת-ממדית המכילה את התמונה לאחר קוונטיזציה.
  - distortion – וקטור באורך של מספר האיטרציות, המכיל את העיוות הממוצע שהתקבל בכל איטרציה (עפ"י שגיאה ריבועית).
  - QL – מטריצה בגודל (levelsX3) המכילה את וקטורי המילון (codebook).
- חשבו את העיוות הממוצע  $D_m$  בכל איטרציה ואת ערך השיפור היחסי  $\varepsilon = \frac{|D_m - D_{m+1}|}{D_m}$ . קבעו את תנאי העצירה להיות  $\varepsilon \leq \text{meps}$ .

הערות לכל סעיפי שאלה זו:

- (1) על הערכים של dataout להיות שלמים.
- (2) כאשר אתם מאתחלים את הקוונטיזציה, 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 מרבי הגיוני, כך שהקוד יסיים לרוץ מתישהו גם אם יש לכם טעות בתנאי העצירה.

- ב. הפעילו את הקוונטייזר על התמונה colorful.tif, ובצעו קוונטיזציה ל-6 צבעים ול-15 צבעים כאשר האתחול נעשה בצורה אקראית. חשבו את העיוות הממוצע  $D_m$  (ע"פ שגיאה ריבועית) בכל איטרציה ואת השיפור היחסי  $\varepsilon = \frac{|D_m - D_{m+1}|}{D_m}$ . קבעו את תנאי העצירה להיות  $meps = 0.02$ . הציגו גרף של העיוות הממוצע כתלות במספר האיטרציות ואת התוצאות.
- ג. חזרו על סעיף ב' מספר פעמים עבור 6 צבעים (אין צורך להציג את התוצאות). האם בכל פעם מתקבלת אותה תוצאה? הסבירו.
- ד. חזרו על סעיף ב' עם אתחול לפי הצבעים: אדום, ירוק, כחול, סגול, צהוב, תכלת, שחור, לבן, אפור:  
[0.9 0 0], [0 0.9 0], [0 0 0.9], [0.6 0 0.6], [0.7 0.7 0], [0 0.8 0.8], [0 0 0], [1 1 1], [0.5 0.5 0.5]  
השוו את התוצאות בשתי השיטות. היכן התקבלה תוצאה טובה יותר? נסו להסביר מדוע. השוו את כמות האיטרציות שנדרשו ע"מ להגיע לתוצאה בשתי השיטות. על מה זה מעיד?

## 4. דחיסה

- א. טענו את התמונה heisenberg.jpg. חשבו ורשמו את האנטרופיה של התמונה בעזרת פונקציית ספרייה מתאימה והסבירו.
- ב. כתבו פונקציה שמבצעת קידוד Huffman לתמונה im ומחזירה את הווקטור הבינארי code, את המילון המתאים לו dict ואת קצב המידע הממוצע במילון avglen. לצורך כך, הפכו את התמונה לווקטור עמודה, בנו עבורו היסטוגרמה והשיגו וקטור של ערכים ייחודיים בו באמצעות פונקציית ספרייה מתאימה. אז, בנו מילון Huffman לפי האלגוריתם שנלמד בכיתה וקודדו את העמודה באמצעות המילון. אפשר ומומלץ למצוא מימוש טוב לקוד Huffman ברשת (Github), אבל בדקו אותו קודם על מקרה פשוט.
- ג. בנו קוד Huffman לתמונה שטענתם, וחשבו את אורך הקוד שהתקבל עבור התמונה. מהו יחס הדחיסה המתקבל?
- ד. פענחו את הקוד ושחזרו את התמונה. חשבו את ה-MSE (שגיאה ריבועית ממוצעת) של התוצאה ביחס לתמונה המקורית. הסבירו את התוצאה.
- ה. טענו את התמונה mauritius.jpg (המציגה את דגל מאוריציוס), המירו אותה לתמונת רמות אפור ובצעו את הפעולות הבאות:
- חשבו את האנטרופיה של התמונה.

- מהו קצב המידע הממוצע שמתקבל מבניית קוד Huffman? השוו אותו לאנטרופיה והסבירו את התוצאה.

ו. כעת נתנסה בקידוד הפרשים:

1. טענו את התמונה scotland.jpg והעבירו אותה לווקטור עמודה באמצעות שיטות הסידור הבאות:
  - סידור עמודה רגיל (Column stack).
  - סידור זיג זג. (הזכירו למרצה לפרסם את הקוד לסידור זה).
2. עבור כל סידור, חשבו וקטור הפרשים (עם הפרש של 1) באמצעות הפרש אחורי (האיבר הראשון בווקטור יישאר ללא שינוי) והפכו אותו למטריצה בעלת הממדים המקוריים של התמונה. הציגו את ההיסטוגרמה המתקבלת בכל שיטת סידור. הסבירו את התוצאות.
3. בנו קוד Huffman עבור כל תמונת הפרשים באמצעות הפונקציה מסעיף ג'. בחנו את קצב המידע הממוצע שהתקבל בכל שיטה. איזו מבין שיטות הסידור עדיפה ולמה? הסבירו.

ז. האם קיימות תמונות אותן לא משתלם לקודד לפי קידוד הפרשים? הסבירו מדוע.

## 5. התאמת תבניות

לאורך תרגיל זה נשתמש במדד ה-SSD (Sum of Squared Differences) למדידת הדמיון בין תמונות ה-Template (התבנית שנרצה למצוא) ובין התמונה המרכזית. מדד זה נתון על ידי:

$$S[x, y] = \sum_{i=1}^M \sum_{j=1}^N \left( T[i, j] - I \left[ x + i - \frac{M}{2} - 1, y + j - \frac{N}{2} - 1 \right] \right)^2$$

כאשר  $T$  היא התבנית,  $I$  היא התמונה ו- $(M, N)$  הם ממדי התבנית. ההתאמה המיטבית מתקבלת עבור ערך SSD מינימלי.

הערה: על מנת לבצע שימוש נכון במדד ה-SSD הפכו את כל התמונות בתרגיל זה ל-double.

א. כתבו פונקציה המקבלת תמונה ותבנית ומחשבת את מדד ה-SSD בכל הפיקסלים בתמונה. בקצוות התמונה, היכן שהתבנית ממילא אינה נכנסת, אפשר להחזיר ערך גדול מאוד.

ב. טענו את התמונה Text.jpg. תמונה זו מכילה טקסט קצר של Edgar Allan Poe.

בסעיף זה עליכם לגלות מהו גודל הגופן בו כתוב הטקסט. לשם כך השתמשו בתמונות

E10.jpg, E11.jpg, E12.jpg, E14.jpg, E16.jpg התואמות לגדלי גופן 10, 11, 12, 14, 16.

- הציגו את חמש התמונות יחדיו.
- ממשו אלגוריתם המשתמש במדד ה-SSD ומוצא לאיזו תמונת Template התאמה מיטבית לטקסט. מהי מסקנתכם לגבי גודל הגופן בטקסט?

ג. בסעיף זה עליכם לגלות כמה פעמים מופיעה האות  $a$  בטקסט וכמה פעמים מופיעה האות  $t$  בטקסט (כולל capitals). חתכו מתוך הטקסט את התבניות הדרושות לחישוב וקחו מסגרת הצמודה לאות ככל הניתן (הסתכלו על E10,...,E16 כדוגמה). השתמשו במדד ה-SSD להתאמה, קבעו סף לערכי ה-SSD על מנת להפריד בין התאמה וחוסר התאמה וספרו כמה פעמים מופיעה כל אות. יש לקבוע סף אחיד לשתי האותיות וה- $\text{capitals}$  שלהן (תנאי או פונקציה כלשהי הפועלים על ערכי ה-SSD ולא מספר קבוע); בחנו את וקטורי ה-SSD על מנת למצוא סף בצורה מתאימה.

- הציגו את התבניות שבהן השתמשתם.
- הסבירו את הסף הנבחר.
- רשמו את מספר המופעים של כל אות בשני הגדלים. האם הצלחתם לזהות את כל מופעי האותיות?

עזרו ל-Mr. Smoke-too-much שאינו מסוגל להגיד את האות  $c$ , להקריא את הטקסט. עשו זאת ע"י זיהוי כל מופעי האות  $c$  בטקסט והחלפתם באות  $k$ . לשם כך השתמשו בתבניות  $c.jpg$ ,  $k.jpg$ . הציגו את הטקסט החדש. האם הוא נראה טבעי?