

אוניברסיטת אריאל מבוא לחישוב – סמסטר א', תשע"ף מטלה מספר 2.

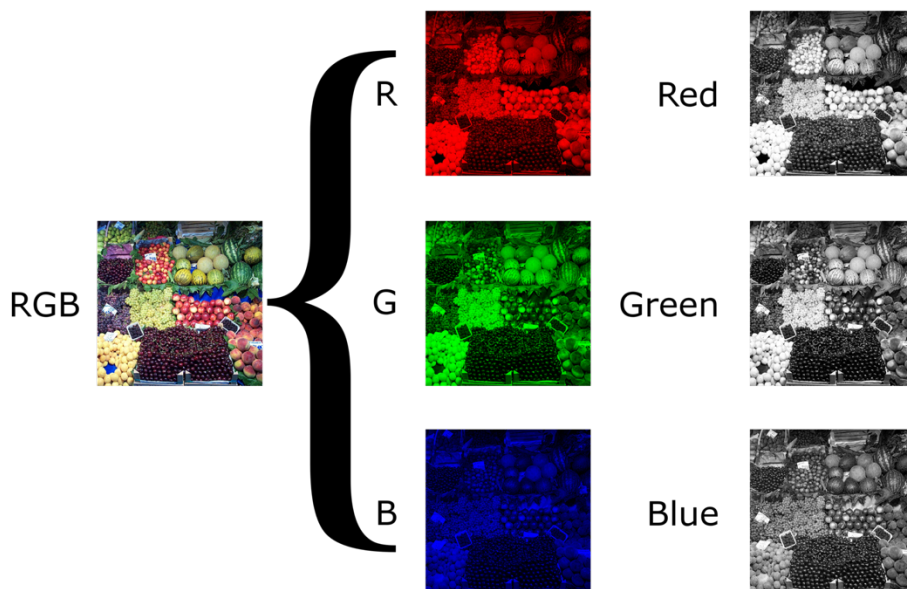
- ההגשה הינה ביחידים בלבד. אין למסור קוד לתלמיד אחר. המטלות עוברות בדיקות נגד העתקה.
- יש להגיש למודל קובץ ZIP בלבד (ולא כל דחיסה אחרת, לדוגמה rar או 7z), ששמו יהיה מס' ת"ז של התלמיד. קובץ ה-ZIP יכיל קובץ אחד בלבד בשם Ex2.java. סטייה מההדרכות האלו יגרור הורדת ציון.

במטלה זו נלמד מעט על עקרונות עיבוד תמונה.

רקע:

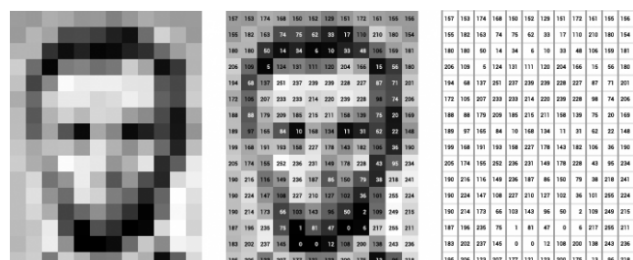
כל תמונה צבעונית היא אוסף של פיקסלים (פיקסל = יחידת מידע גרפית בסיסית). לכל פיקסל יש צבע המיוצג במחשב ע"י שילוב של 3 עוצמות של הצבעים היסודיים: אדום, ירוק, כחול (המכונה שיטת RGB). כאשר: 0 – ללא עוצמה. 255 – עוצמה מלאה. את שלושת הצבעים מסמנים בדרך כלל באופן הבא (R, G, B) כאשר R מסמן את עוצמת הצבע האדום בפיקסל, G – מסמן את עוצמת הירוק, B – מסמן את עוצמת הכחול. כל צבע אחר הוא שילוב של הצבעים בעוצמות שונות. לדוגמא: את הצבע הלבן ניתן לקבל ע"י: (255,255,255). ואת הצבע השחור ניתן לקבל ע"י: (0,0,0). תמונה באורך H וברוחב W מיוצגת ע"י מטריצה $H \times W$ של פיקסלים.

דוגמה:



תמונה ברמות אפור היא אוסף של פיקסלים אפורים, כאשר כל פיקסל מיוצג על ידי מספר אחד בלבד – עוצמת הצבע האפור. לכן תמונה כזו מיוצגת על ידי מערך דו ממדי.

דוגמה:



ניתן להמיר תמונה צבעונית לתמונה בצבעי שחור לבן על ידי שיטה שתראו בהמשך.

המטלה :

למטלה זו כתבנו עבורכם ספרייה בשם MyImageIO (מצורפת) שבתוכה ישנן שתי הפונקציות :

1. `public static int[][][] readImageFromFile(String fileName)` – פונקציה המקבלת מיקום של תמונה במחשב, ומחזירה מערך תלת ממדי $3 \times H \times W$, כאשר H – אורך התמונה, W – רוחב התמונה, ושלושת המטריות הן שלושת השכבות של אדום, ירוק, וכחול עבור האינדקסים 0,1,2 בהתאמה.
2. `public static void writeImageToFile(String fileName, int[][][] pixels)` – פונקציה המקבלת שם של מיקום חדש במחשב ומערך תלת ממדי (המייצג תמונה צבעונית), ושומרת את התוצאה של המערך שהתקבל כתמונה במחשב במיקום שהתקבל.
3. `public static void writeImageToFile(String fileName, int[][] pixels)` – פונקציה המקבלת שם של מיקום חדש במחשב ומערך דו ממדי (המייצג תמונה ברמות אפור), ושומרת את התוצאה של המערך שהתקבל כתמונה במחשב במיקום שהתקבל.

דוגמאות להרצת הפונקציות הנ"ל :

```
int [] [] [] image=readImageFromFile("C://Documents and Settings/user1/Desktop/cat.jpg");
writeImageToFile("C://Documents and Settings/user1/Desktop/catCopy",image);
```

- שימו לב לכך שאינכם נדרשים (ולא אמורים) להוסיף את הסיומת "jpg" לשם של הקובץ בפונקציה ששומרת את התמונה.
- מומלץ לשמור ולעבוד עם קובץ שנמצא בקונן C ישירות (כמו : C://images).
- חובה לכתוב את המיקום המלא של התמונה בשתי הפונקציות.

הפונקציות למימוש :

1. תמונה ברמות אפור :
ממשו את הפונקציה :

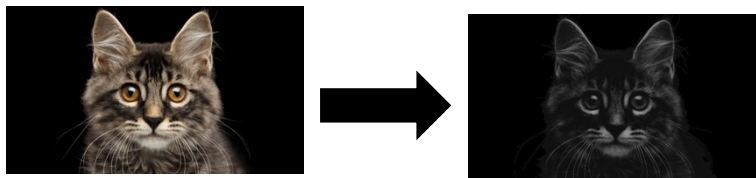
```
public static int[][] rgb2gray(int[][][] im)
```

הפונקציה מקבלת מערך תלת ממדי (המייצג תמונה צבעונית) ומחזירה מערך דו ממדי (המייצג אותה תמונה ברמות אפור).
פיקסל במקום i,j ברמת אפור יהיה הממוצע הממושקל הבא של אדום, ירוק, וכחול :

$$Grey_{i,j} = [(0.3 * R_{i,j}) + (0.59 * G_{i,j}) + (0.11 * B_{i,j})] \cdot 255$$

כאשר $R_{i,j}$, $G_{i,j}$, $B_{i,j}$ הם ערכי האדום, ירוק, והכחול של הפיקסל בהתאמה.

לדוגמה :



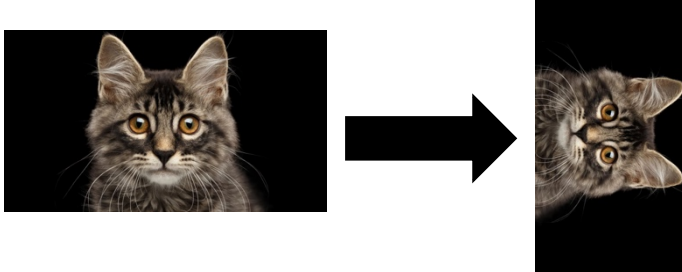
2. סיבוב התמונה ב 90 מעלות :

ממשו את הפונקציה :

```
public static int[][] rotate90(int[][] im)
```

הפונקציה מקבלת מערך תלת ממדי המייצג תמונה צבעונית, תסובב את התמונה ב 90 מעלות ימינה, ומחזירה את התוצאה (לא לשנות את התמונה המקורית).

לדוגמה :



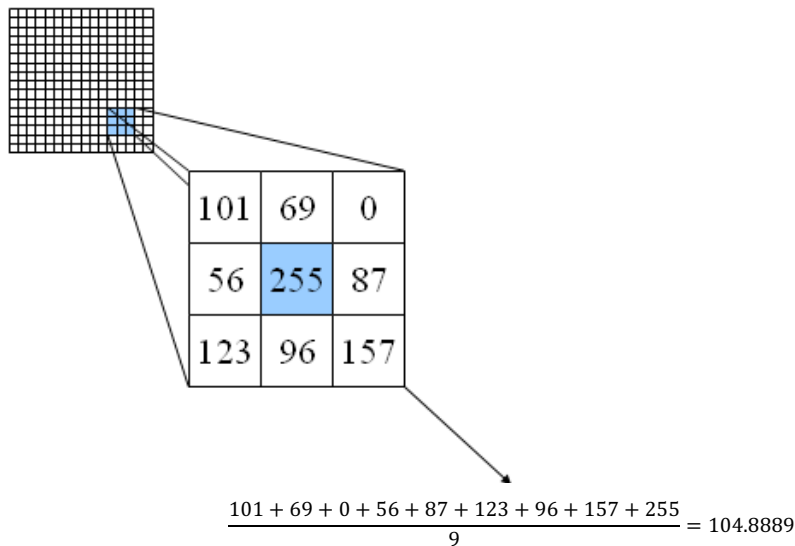
3. החלקת תמונה :

ממשו פונקציה :

```
public static int[][] Smooth(int[][] img, int n)
```

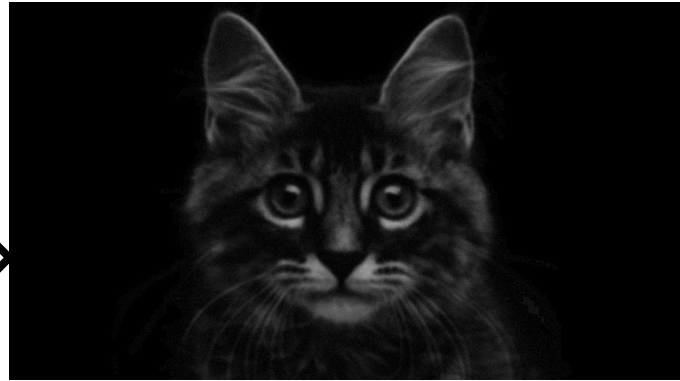
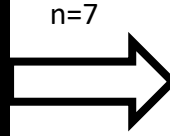
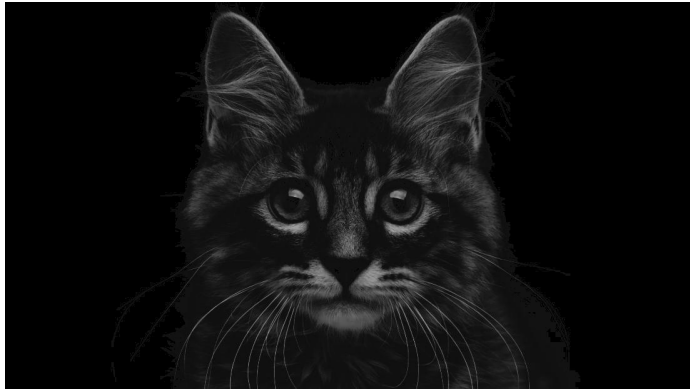
המקבלת תמונה ב-RGB, פרמטר ההחלקה, ומחליקה את התמונה. החלקת התמונה עבור פרמטר n , הינה מתן כל פיקסל בתמונה להיות **ממוצע השכונה שלו שמורכבת מריבוע בגודל $(2n + 1) \times (2n + 1)$** (להסתכל על ריבוע $(2n + 1) \times (2n + 1)$ שהפיקסל הוא מרכזו, ולתת לפיקסל להיות הממוצע של התאים האלו).

לדוגמה, עבור המטריצה הבאה ופרמטר ההחלקה **1**, ההחלקה עבור הפיקסל הכחול תהיה :



- יש לשים לב לכך שישנם פיקסלים שנמצאים בקצוות – ולכן, לא כל האיברים שיש בריבוע הנ"ל נמצאים בתוך גבולות התמונה!
- מומלץ ורצוי שעבור שאלות 2+3, לממש פונקציה שמבצעת את המבוקש עבור מטריצה אחת, ואז לקרוא לה 3 פעמים עבור כל אחת מהשכבות (האדומה, הירוקה, והכחולה).

דוגמה לשאלה 3 :



4. מתיחת תמונה:
ממשו את הפונקציה:

```
public static int[][] scaleUp(double factor_h, double factor_w, int[][] im)
```

הפונקציה מקבלת מערך דו ממדי שמייצג תמונה ברמות אפור, ושני מספרים **ממשיים** המייצגים את פקטור המתיחה בשני הממדים (רוחב ואורך), ומחזירה תמונה חדשה (מערך דו ממדי) שמייצג את התמונה מתוחה בגודל:

$$[m[0].length * factor_h] \times [m.length * factor_w]$$

המתיחה תעשה באמצעות שיטת השכן הקרוב ביותר.

שיטת השכן הקרוב ביותר הינה:

$$im[i \cdot factor_h + f_h][j \cdot factor_w + f_w] = im[i][j]$$

כאשר:

f_w - רץ מ-0 עד $factor_w$ (הפרמטר)

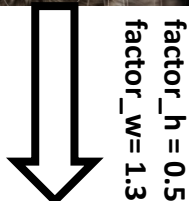
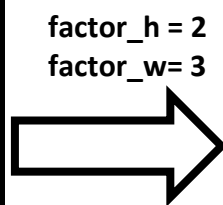
f_h - רץ מ-0 עד $factor_h$ (הפרמטר)

וכן:

$$0 \leq i \leq H$$

$$0 \leq j \leq W$$

לדוגמה:



בהצלחה!