מכון טכנולוגי חולון
Holon Institute of Technology

סמסטר קיץ – תשע"ח, מועד ג'
תאריך: 20.12.18
המרצה: דמיטרי פטשוב
קבוצות הרצאה: 0-21210-62/63

אלגוריתמים במולטימדיה ולמידת מכונה בסביבת פייתון

מס' זהות:

הוראות:

1. המבחן נעשה במחשב.
2. אין שימוש בחומר עזר.
3. מותר להשתמש במחשבון, דף ריק מתוכן וכלי כתיבה.
4. משך הבחינה - 3 שעות
5. במבחן שלוש חלקים:
   - חלק ראשון חובה - יש לענות על כל השאלות.
   - חלק שני בחירה - יש לענות על שתי שאלות מתוך שלוש.
   - חלק שלישי בונוס - לא חובה לענות על חלק זה.
6. אין לעזוב את המבחן לפני שהוגש כנדרש (אחרת הציון 0 אוטומטית)
7. שימו לב! לא ליצור את הפרויקט בקונן C (אם קיים קונן נוסף) אחרת יש סיכוי לעובדן מידע.
8. שימו לב! תוודאו שאתם עובדים בגרסת פייתון נכונה.
9. יש ליצור קובץ פונקציות אחד בלבד ובו לממש את כל הפונקציות הנדרשות.
10. כל פונקציה שאתם מתבקשים לכתוב חייבת להיות בעלת שם זהה לזה שבשאלה! כולל התייחסות לאותיות קטנות/גדולות. פונקציה שהשם שלה לא יהיה רשום במדויק, לא תקבל ניקוד.
11. אין להגיש שום סקריפט שלא התבקשתם לכתוב וזה כולל סקריפט הבדיקות שלכם.
12. כל קטע קוד שלא קשור לאחת הפונקציות שהתבקשתם לכתוב יגרור הורדה בציון וזה כולל פלטים שלא התבקשתם לעשות.
13. שימו לב! יש להתחשב במקרי הקצה שעלולים להיות. לדוגמה קלט לא תקין או כתובת שלא קיימת.
14. אתם יכולים להשתמש בכל פונקציה שתרצו לפתרון המבחן.
15. במהלך המבחן אין גישה לאינטרנט.
16. מי שמסיים את המבחן מחכה במקומו עד להגעת אוסף הבחינות בכדי להגיש את הקוד.
17. יש לרשום את מספרי ת.ז. על גבי טופס זה ולהגיש אותו בסוף המבחן.
18. את הסקריפט שהנכם מגישים יש לקרוא בשם המתחיל באות "s" וללא רווח מס' הת.ז. שלכם. לדוגמה: s123456789
19. במידה וישנה פונקציה שאינכם יודעים מה יעודה, נסאו פקודה:
    print FunName.__doc__

מכון טכנולוגי חולון
Holon Institute of Technology

חלק א' (חובה): 50%

שאלה 1 (5 נק.)

1. הגדירו פונקציה בשם : **myDashMatrix**
הפונקציה צריכה לקבל מספר טבעי $n > 2$.
הפונקציה צריכה להחזיר מטריצה ממימד n בה יש אחדים בכל עמודה זוגית (כולל 0).

דוגמה :

n = 4
myMat = **myDashMatrix** (n)

תוצר הפונקציה יהיה :

$$myMat = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

שאלה 2 (15 נק.)

2. הגדירו פונקציה בשם : **countMyStrings**
הפונקציה צריכה לקבל מחרוזת.
הפונקציה צריכה להחזיר מילון המכיל את כל המילים הקיימים במחרוזת ללא חזרות וללא חשיבות לאותיות גדולות/קטנות בתור המפתחות של המילון וערך עבורן יהיה מספר הופעות של כל אחת מהמלים במחרוזת שהתקבלה, כולל מקרים בהם מילה היא חלק ממילה אחרת.

לדוגמה :

myStr = 'Hello Hello We Do Hi no No no we Donot Hey HE he heY'
myDict = **countMyWords (myStr)**

תוצר הפונקציה יהיה :

myDict :

hello : 2,	we : 2,	do : 2,	hi : 1
no : 4,	donot : 1,	hey : 2,	he : 6

פונקציות עזר אפשריות :

numbers.Number, ord, isinstance, chr, np.mean, np.zeros, np.pi

מכון טכנולוגי חולון
Holon Institute of Technology

שאלה 3 (5 נק.)

3. הגדירו פונקציה בשם: **myListSum**
הפונקציה מקבלת רשימה המכילה מספרים ורשימות נוספות במבנה דומה לרשימה עצמה.
הפונקציה צריכה להחזיר סכום של כל האיברים הנמצאים ברשימה שהתקבלה כולל כל תתי הרשימות.

לדוגמה:

myList = [1, [2], 3, [1, -1, [2, 3, [1.0], 1], 2], -3, 1.0]
mySum = **myListSum (myList)**

תוצר הפונקציה יהיה:

mySum = 13.0

שאלה 4 (5 נק.)

4. הגדירו פונקציה בשם: **myListEdit**
הפונקציה מקבלת רשימה, ערך ופרמטר בחירה עם הגדרה של בררת מחדל שווה לאפס.
הפונקציה צריכה להחזיר רשימה לאחר הוספת איבר המכיל את הערך הנשלח לפונקציה.
אם פרמטר בחירה שווה לאפס, הפונקציה צריכה להוסיף איבר "by reference" אחרת "by value".

לדוגמה:

myL = [1,2,3]
myL1 = **myListEdit (myL, 10, 0)**
myList = [1,2,3]
myList1 = **myListEdit (myList, 10, 1)**

תוצר הפונקציה יהיה:

myL = myL1 = [1,2,3,10], כתובות זהות בזיכרון
myList = [1,2,3], myList1 = [1,2,3,10], כתובות שונות בזיכרון

פונקציות עזר אפשריות:

np.eye, id, np.argmax, cv2.copyMakeBorder, hex

מכון טכנולוגי חולון
Holon Institute of Technology

שאלה 5 (20 נק.)

5. הגדירו פונקציה בשם : **myTwoLiner**

```
def TwoLiner(img):

    if img.__class__ != np.ndarray:
        return None

    im = img.copy()
    D = im.shape
    th = np.int64(np.round(D[0]/3.))
    wi = np.int64(np.round(D[0]/30.))
    if len(D) != 3 or D[0] < 30:
        return None

#-------------------------------------------------------
    for m in range(th - wi, th + wi + 1):
        for n in range(D[1]):
            im[m, n, 0] = 150
            im[m, n, 1] = 255
            im[m, n, 2] = 0
    for m in range(2*th - wi, 2*th + wi + 1):
        for n in range(D[1]):
            im[m, n, 0] = 150
            im[m, n, 1] = 255
            im[m, n, 2] = 0
#-------------------------------------------------------

    return im
```

תחליפו את קטע הקוד שבין הקווים השבורים בשורת קוד אחת בלבד.
פונקציות עזר אפשריות :

np.sort, np.logical_or, np.logical_and, np.reshape

מכון טכנולוגי חולון
Holon Institute of Technology

חלק ב' (בחירה): 50%

שאלה 6 (25 נק.)

6. הגדירו פונקציה בשם : **myKNNClasification**

הפונקציה צריכה לקבל מספר טבעי k, מטריצה (ndarray) המכילה דגימות בתור שורות (שורה אחת זו דגימה אחת), וקטור (עמודה - ndarray) הסוגים באורך שווה לכמות השורות במטריצה ודגימה אחת לסיווג אשר תנתן בצורת וקטור (ndarray) שורה שאורכו זהה לכמות העמודות במטריצה.
הפונקציה צריכה להחזיר את הסוג (ערך המתאים לאחד הערכים הנמצאים בוקטור הסוגים) של הדגימה הבודדת.
את המרחק יש לחשב על ידי המרחק האוקלידי.
אין צורך להתייחס למקרה בו יש יותר מסוג אחד המתאים לדגימה שהתקבלה.

דוגמה :

myLabel = **myKNNClasification (k, Data, Labels, Sample)**

פונקציות עזר אפשריות :

np.reshape, np.arcsin , np.sort, cv2.copyMakeBorder, np.argsort, cv2.filter2D

שאלה 7 (25 נק.)

7. הגדירו פונקציה בשם : **myLinearDependancyRemoval**

הפונקציה צריכה לקבל מטריצת ndarray של נתונים בייצוג float64.
הפונקציה צריכה לחשב ממוצע בין כל השורות שבמטריצה ולהחסיר אותה מכל שורה ושורה.
לאחר מכן יש לחשב את מטריצת שונות המשותפת של המטריצה המתקבלת.
למטריצת שונות המשותפת יש לחשב ערכים עצמיים ווקטורים עצמיים.
יש להעתיק את מטריצת הנתונים למרחב פורש בעל שונות מקסימלית וללא תלות לינארית (להיפתר מוקטורים עצמיים של ערכים העצמיים ששווים ל 0). תניחו שכל ערך הקטן מ 10^{-9} הוא שווה ל 0.

תזכורת :

חישוב מטריצת שונות משותפת :

$CovMat = A^t A$ כאשר מטריצה A מסדר $m\ x\ n$: $A_{(m,n)}$.

חישוב ערכים עצמיים :

$$C \cdot \bar{u} = \lambda \bar{u}$$
$$U = [\bar{u}_1, \dots, \bar{u}_n]$$

העתקה לינארית :

$$F = A \cdot V$$

דוגמה :

Field = **myLinearDepandacyRemoval(**DataMat**)**

פונקציות עזר אפשריות :

np.mean, np.linalg.eigh, np.median, cv2.BORDER_REFLECT, plt.plot

מכון טכנולוגי חולון
Holon Institute of Technology

שאלה 8 (25 נק.)

8. הגדירו פונקציה בשם : **myHalfEdgeMaskKernel**

הפונקציה צריכה לקבל תמונה ומסכה. שתי המטריצות חייבות להיות מסוג ndarray.
הפונקציה צריכה להחזיר את המטריצה המתקבלת כתוצאה מתהליך המיסוך שמופעל אך ורק על השורות בעלות אינדקס זוגי (כולל 0).
על השורות הזוגיות יש להפעיל מסכת סובל (S_x, S_y) ולחשב מרחק אוקלידי (מראשית הצירים) עבורן (Intensity Gradient). את השורות שעברו מיסוך יש לנרמל לטווח בין 0 ל-255, לעגל ולהחזיר ליצוג של uint8.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad S_y = S_x^T \qquad I_g = \sqrt{G_x^2 + G_y^2}$$

כאשר G_x ו- G_y הם תוצרי המיסוך של סובל.

דוגמה :

mImg = **myHalfEdgeMaskKernel (img)**

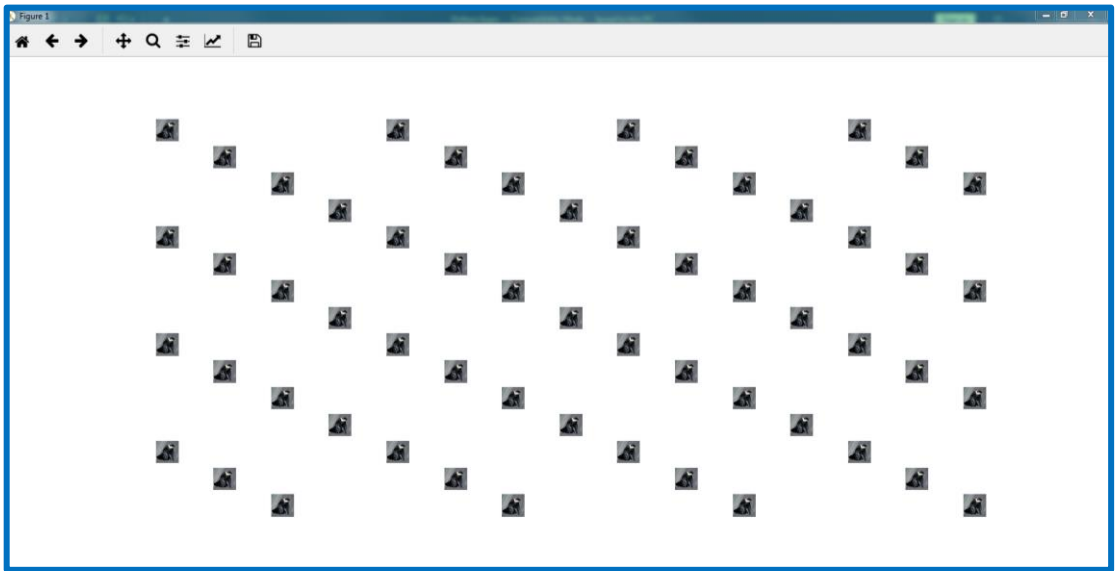
פונקציות עזר אפשריות :
cv2.copyMakeBorder, cv2.BORDER_REPLICATE, np.round, np.ceil, np.floor

חלק ג' (בונוס): 15%

שאלה 9 (15 נק.)

9. הגדירו פונקציה בשם : **myPlotShape**

הפונקציה צריכה לקבל תמונה. תמונה חייבת להיות מסוג ndarray.
הפונקציה צריכה לפלוט גרף של ספריית matplotlib שנראה כך :



בהצלחה!