OCR of Handwritten Hebrew

תאריך ההגשה: 7.12.2020, שעה 23:55

בתרגיל זה תשתמשו באלגוריתם k-Nearest Neighbor כדי לסווג תמונות של אותיות ממאגר HHD_0, שמורכב מאותיות בכתב יד. המאגר HHD_v0 מכיל בסביבות 5000 תמונות של אותיות בודדות. תמונות אלו מחולקות לשתי קבוצות (תיקיות) TEST ו-TRAIN, כאשר כל אחת מקבוצות האלו מחוקלת ל-27 תתי-קבוצות (תתי-תיקיות). כל תיקייה מכילה תמונות של אות מסוימת מתוך האלפבית העברי. פרטים אודות המאגר HHD_v0 ניתן למצוא ב-[1].

איור 1: דגימה ממאגר HHD_v0 של אותיות בכתב יד

מטרת התרגיל היא לסווג את אותיות בצורה נכונה (סיווג בדיוק גבוה).
לשם כך תדרשו לאמן מסווג k-NN.

העבודה תחולק למספר צעדים:

1. עיבוד מקדים (pre-processing)

בשלב זה עליכם להעביר את כל האותיות לגודל אחיד.

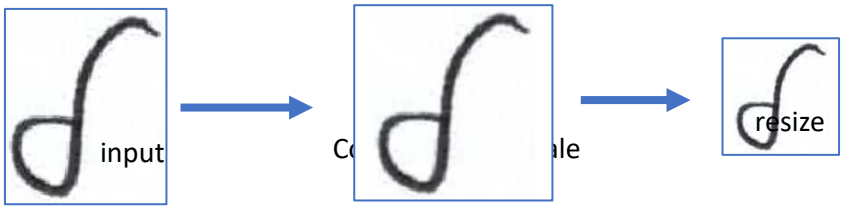
a. המירו את התמונה לגווני אפור (greyscale)

b. הוסיפו לתמונה ריפוד (padding) כדי שגודלה יהיה מרובע

- אם רוחב התמונה קטן מגובה, יש להוסיף Padding מימין ומשמאל
- אחרת, אם רוחב גדול מגובה, יש להוסיף Padding מלמעלה ולמטה

אפשר להיעזר בפונקציית `cv2.copyMakeBorder` של OpenCV.

c. העבירו את התמונה לגודל אחיד (40,40)



2. חילקו את ה-TRAIN set באופן אקראי לשתי קבוצות (training and validation sets).
החלוקה תהיה ביחס 90% ל-training ו- 10% ל-validation.
בשלב 4, אתם תשתמשו ב-training set כדי לאמן את k-NN, וב-validation set כדי למצוא את הערך הטוב ביותר של k (ערך שנותן דיוק הגבוה ביותר) ופונקצית מרחק הטובה ביותר.
לאחר שתמצאו את השילוב הטוב ביותר של k ופונקצית מרחק, תריצו את ה-k-NN על ה-TRAIN set המקורי עם שילוב פרמטרים הכי טוב שמצאתם, ותעריכו את התוצאות על TEST set.

3. Feature extraction – בשלב זה תחלצו HOG features כדי לייצג את התמונה של כל אות.
השתמשו בפרמטרים הבאים:

```
ch_hog = feature.hog(ch_im, orientations=9,
                     pixels_per_cell=(8, 8),
                     cells_per_block=(2, 2),
                     transform_sqrt=False,
                     block_norm="L2")
```

4. אימון (training).
בשלב זה יש לאמן את המסווג k-NN על ערכים שונים של k ופונקציות מרחק שונות, להעריך את התוצאות על validation set עבור כל ערך k ופונקציית מרחק, ולבחור את שילוב הטוב ביותר (שילוב שנותן הדיוק הגבוה ביותר על validation set).

- יש לאמן את המסווג על הערכים של k בין 1 ל-15.
- יש לאמן את המסווג על שתי פונקציות מרחק שונות: Euclidean distance ו- Chi-Square distance (χ^2).

χ^2 היא פונקציית מרחק שמתאימה במיוחד להשוואה בין שתי היסטוגרמות. בהינתן שתי היסטוגרמות H_1 ו- H_2, χ^2 מוגדרת באופן הבא:

$$\chi^2(H_1, H_2) = \frac{1}{2}\sum_{i=1}^{b}\frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}$$

כאשר b הוא מספר ה-bins בכל היסטוגרמה.

אתם יכולים לממש את k-NN באופן עצמאי (הוא מאוד פשוט) או להשתמש ב-k-NN מתוך הספרייה sklearn (תצטרכו להתקין ספרייה זו כמובן).

5. הערכת ה-k-NN על TEST set.
ברגע שמצאתם את השילוב האופטימאלי של k ופונקציית מרחק, יש להעריך את התוצאות של k-NN על TEST set ולדווח את התוצאות.

פלט התוכנית יכלול

1. קובץ טקסט בשם "results.txt" שיכיל:

a. ערך k ופונקצית מרחק שנותנים דיוק הכי גובה בפורמט
k = … , distance function is ….
b. דיוק אליו הגיע המסווג עבור כל אחת מהאותיות (27 אותיות שונות) בפורמט

Letter	Accuracy
0	…
1	…
…	
26	…

2. Confusion matrix עבור התוצאות בקובץ excel/scv בשם "confusion_matrix.csv"

הרצת התוכנית תתבצע משורת הפקודה בפורמט

```
> python knn_classifier.py    path
```

כאשר knn_classifier.py הוא שם התוכנית ו-path הו מסלול לתיקייה עם המאגר.

זמן ריצה של התוכנית אינו צריך לעלות על מספר דקות בודדות (5 דקות לכל היותר עבור מחשב ביתי טיפוסי). יש להדפיס על המסך את הזמן הנוכחי בתחילת ריצה של התוכנית ובסופה. כמו-כן, יש לדפיס את זמן ריצה הכולל של התוכנית.
זמן ריצה של התוכנית כולל את כל השלבים: קריאת תמונות של אותיות, עיבוד מקדים, חילוץ HOG, אימן k-NN ויצירת קצבי פלט.

שימו לב: על מנת לייעל זמן ריצה, השתמשו ב-vectorization והימנעו מהלולאות.
לדוגמה, במקום לבצע פעולה מסויימת על כל איבר של המערך באמצעות לולאה, ניתן לבצע פעולה זו בו זמנית על כל הערכים.

הגשה:
יש להגיש
- קובץ קוד עם התוכנית
- קובץ readme.txt
- קבצים "results.txt" ו- "confusion_matrix.csv" בפורמט שמתואר למעלה

אופן הבדיקה:
הבדיקה תתבצע בצורה פרונטלית (מקוונת). מועדי הבדיקה ייקבעו בהמשך.

בכל שימוש המאגר HHD_v0, יש לתת הפנייה ל-[1]

עבודה נעימה!

References
[1] I. Rabaev, B. Kurar Barakat, A. Churkin and J. El-Sana. The HHD Dataset. The *17th International Conference on Frontiers in Handwriting Recognition*, pp. 228-233, 2020.