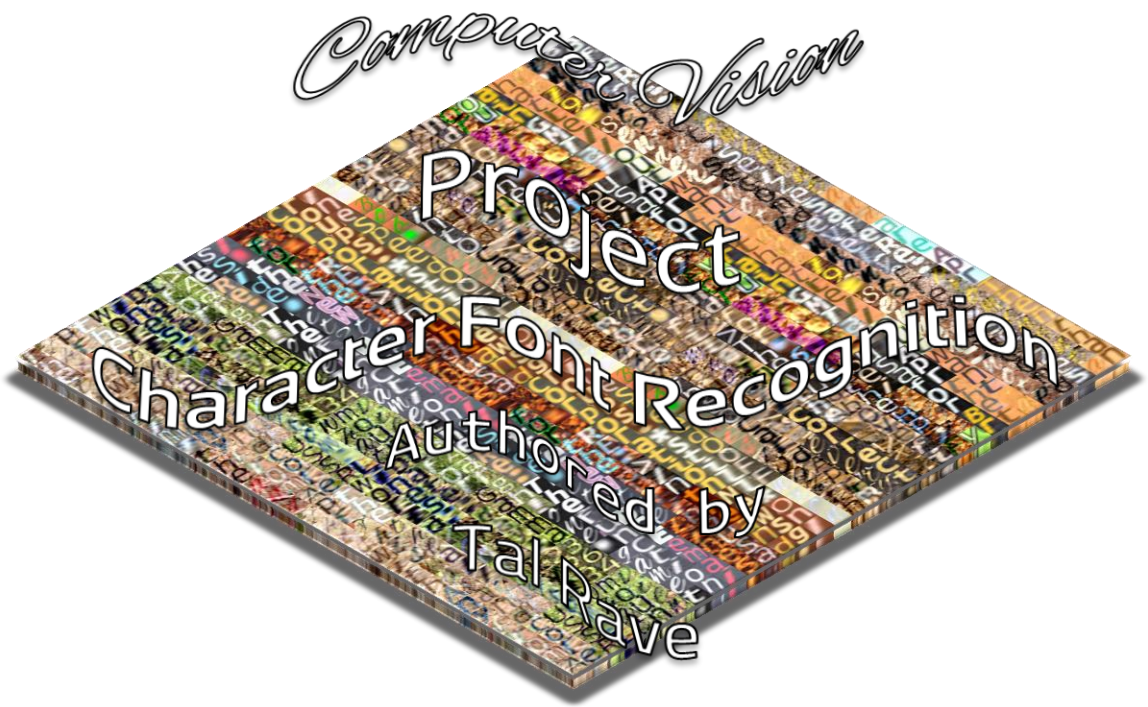

Table of Contents

קבצי הפרויקט

בנוסף לקובץ זה, קיימים בתיקיית הפרויקט הקבצים הבאים:

1. The Project.ipynb
2. head.py
3. prep.py
4. test.py
5. pred.py
6. labels_verify.csv
7. readme.pdf

הקובץ הראשון מכיל את כל שלבי האימון\ולידציה של המודל\ים תוך מתן הסברים תמציתיים והצגת גרפים מתאימים.

הקובץ השני מכיל את הספריות והקבועים הנדרשים עבור הקבצים השונים.

הקובץ השלישי מכיל שיטות אשר מתמקדות בהכנת הנתונים (חילוץ האותיות, ניקוי, יצירת מאפיינים, וכד').

הקובץ הרביעי מכיל שיטות לבחינת ביצועי המודל.

הקובץ החמישי משתמש במודל (יש ללחוץ על הלינק) ליצירת קובץ csv. כנדרש.

הקובץ השישי מכיל את החיזויים המבוקשים על קבוצת המבחן לשימוש ישיר או לצורכי אימות.

הקובץ האחרון מכיל הוראות לשחזור התוצאות על קבוצת המבחן כנדרש.

תהליך העבודה

במבט ראשון על הדאטה ניתן היה לראות מיד כי התיבות המכילות את האותיות אינן מיושרות ביחס לצירי התמונה. ככל הנראה מדובר במלבנים\מקביליות מסובבים. בהתאם לכך, תיבות אלו נתונות ע"י ארבע נקודות במישור ולא כדאי לשלוף את האותיות באופן גס ע"י שליפת תתי-תמונות אשר עלולות להכיל רעש רקע נוסף שעלול לפגוע בביצועי המודל. כמו כן, ניתן לראות כי המילים שונו קצת ע"י העתקות מגוונות כגון: סיבוב, מתיחה, עיוות, וכד'. לכן כדי לקבל צורה נורמלית לכל אות, החלטתי להתאים בין שלוש קואורדינטות של התיבה לשלוש קואורדינטות בפינה השמאלית-עליונה של התמונה (ניתן לשלוט על גודל התמונה של האות המחולצת ע"י בחירה מתאימה של הנק' הנ"ל). כלומר, בהינתן מטריצות A ו- B המכילות בעמודותיהן את הקואורדינטות (הומוגניות) של היעד (הפינה השמאלית-עליונה) והמקור (שלוש נק' של תיבה של אות), בהתאמה, יש לחשב את המטריצה M המקיימת $MA = B \Rightarrow M = BA^{-1}$ (יש לשים לב כי A הפיכה לפי הגדרתה). אציין כי מדובר בשיטת ה- inverse warping מכיוון ש- M מעבירה מהיעד למקור.

כעת בהינתן האותיות המחולצות והמטא דאטה שלהן, ניתן ליצור קבוצות נתונים לאימון מודל סיווג כלשהו. המודל הראשון שחשבתי עליו הוא SVM מכיוון שהשתמשנו בו בסוף המטלה השנייה. מצד שני, בבעיה הנ"ל יש יותר משתי מחלקות ועשרות אלפי דוגמאות של אותיות – במקרה כזה, SVM לא כל כך פרקטי כי הוא מיועד לדאטה קטן יחסית כאשר הסיווג הוא בינארי [1]. כמו כן, עקב הדמיון הרב בין פונטים שונים (האותיות משותפות לכל הפונטים) סביר להניח כי הם לא ניתנים להפרדה לינארית באופן מועיל, ומכאן שיש צורך בשימוש בגרעין מיוחד, מה שהופך את SVM לעוד יותר לא פרקטי (במובנים של סיבוכיות זמן\זיכרון) [1]. לכן, המודל הפשוט והמוכר הבא שלא סובל מבעיות אלו הוא KNN – מודל זה מהווה את הבסיס לפתרון שלי.

לצורך קבלה מהירה של תוצאות, השתמשתי תחילה בתמונות קטנות של האותיות (32x32x3) אשר שוטחו לווקטור, ללא כל עיבוד נוסף. כצפוי, התוצאות לא היו מרשימות ולכן התחלתי לחשוב על דרכים פשוטות לשיפור. למשל, שימוש באינטרפולציה מעוקבת תוך ביצוע ההעתקה שמופעלת על התמונה לצורך חילוץ האותיות הביא לשיפור קל בדיוק. אין ספק שיש צורך בחילוץ מידע יותר מועיל מתמונות האותיות.

כעת נשאלת השאלה אם יש צורך במידע צבעי – במחשבה ראשונה, התשובה היא "לא" מכיוון שהפונטים לא תלויים או קשורים באיזשהו אופן לצבע שנבחר באופן מקרי למילים או לצבעים של הרקע הנותר, אך במחשבה שנייה, מידע זה יכול להיות מועיל

בהפרדת האות מהרקע. לאור זאת, לא מיהרתי לוותר על הצבע של התמונות. עם זאת, לא כל כך ברור איך ניתן לעשות שימוש בצבע באופן ישיר מכיוון שלאותיות וגם לרקע יש מגוון צבעים רחב. במילים אחרות, הדבר העיקרי שמבדיל בין הרקע לאות הוא הצורות הייחודיות של האותיות השונות. בדומה, הפונטים גם נבדלים זה מזה בצורות של האותיות. לכן, חשבתי מיד על שימוש ב- SIFT לצורך קבלת מידע מהסוג הנ"ל עקב האופן שבו הוא מייצר את ה- descriptors [2], אך אז יתקבל מס' שונה של נקודות מעניינות עבור כל אות כאשר ה- descriptors מכילים מידע לוקאלי שעלול להיות משותף להרבה אותיות דומות בפונטים שונים. יתכן כי הגישה הנ"ל מועילה לזיהוי האותיות עצמן, אך במקרה של זיהוי פונטים כמדומני שיש צורך במידע יותר גלובלי שנותן מעין תיאור של הצורה הכללית של אותיות השייכות לפונט מסוים. כידוע, ה- descriptors של SIFT מבוססים על מאפיינים מסוג HOG [3] – ניתן לחשב מאפיינים כאלו באופן גלובלי עבור כל תמונה של אות ללא שימוש בנק' מעניינות – גישה זו הביאה לעלייה דרמטית בביצועי המודל ומהווה את החלק המהותי השני בפתרון שלי.

אחד מהשלבים החשובים לפני שליחת נתונים לאימון הוא ביצוע פעולות נרמול\סטנדרטיזציה. לחלופין, ניתן להשתמש במרחקי cosine בעת הגדרת KNN במקום ביצוע נרמול לאחר יצירת מאפייני HOG – שיטה זו מועילה במיוחד כאשר המאפיינים דלילים [4] (זוויות הגרדיאנטים בתא מסוים בד"כ מתאימות למס' מצומצם של סלים, ולכן יש הרבה ערכים מאופסים\קטנים יחסית בייצוגי HOG), ולכן אין זה פלא כי הדיוק השתפר שוב באופן משמעותי בהוספת המרכיב הנ"ל. אלמנט אופציונאלי נוסף במודל KNN הוא שימוש במשקלים בהתאם למרחקים מ- k השכנים הקרובים ביותר. קרי, מחשבים את סכום המרחקים ההופכיים ואז התווית עם הסכום הגדול ביותר תבחר – תהליך זה אמנם הוביל לשיפור קל בלבד בדיוק, אך יש לו יתרון נוסף שיבוא לידי ביטוי בהמשך – עקב כך, השימוש במרחקים ההופכיים כמשקלים לצורך קביעת התיוגים מהווה חלק נוסף בפתרון.

בהתבוננות נוספת על המטא דאטה שמתי לב כי לא עשיתי עדיין שימוש בתיבות של המילים. סביר להניח כי מודל טוב יחסית יידע לזהות נכונה את רוב האותיות של מילה נתונה, ולכן סביר להניח כי שימוש בהצבעה לקביעת התיוג של כלל האותיות יביא לשיפור נוסף בדיוק. לצורך כך, אני מריץ תהליך נוסף על הפרדיקציה של המודל המבצע את האמור לעיל – תהליך זה הביא לקפיצה דרמטית נוספת בביצועים, ולכן מדובר במרכיב חשוב בפתרון המוצע.

אזכיר כעת את השימוש במרחקים ההופכיים לצורך קביעת התיוגים – מסתבר שניתן לקבלם מהמודל ולבצע את הסיווג באופן "ידני" במקום להשתמש בשיטה predict. לאור זאת, החלטתי להשתמש בהם בתור משקלים עבור מרכיב ההצבעה שציינתי קודם לכן. קרי,

ה- score שאות תורמת בהצבעה ביחס לפונט t מסוים הוא סכום המרחקים ההופכיים שלה מ- $k_t \leq k$ השכנים הקרובים ביותר אליה השייכים לפונט t – תהליך ההצבעה המשופר הנ"ל הביא לשיפור משמעותי נוסף בדיוק ולכן הוא גם מהווה עוד אלמנט בעל ערך בפתרון.

בשלב זה ביצועי המודל נראים די מבטיחים. כמו כן, נדמה שזמן הריצה מהיר למדי למרות הכמות הגדולה של הנתונים. מצד שני, הדיוק ככל הנראה חשוב יותר ולכן התחלתי לחשוב על דרכים יקרות בזמן ריצה\זיכרון עם פוטנציאל לשיפור נוסף של הדיוק. ניתן לשים לב כי האותיות נוצרות באופן סינטטי ובגדלים שונים, ומכאן יתכן כי שימוש בתמונות קצת יותר גדולות עבור האותיות בעת חילוצן מהתמונה ישמר מידע מדיוק יותר של צורתן, אם כי אין טעם להגדיל מעבר לגודל התמונה המקורית. שימוש בתמונות בגודל 64 (במקום 32) לא הביא לשיפור בהתחלה, אך לאחר הוספת מרכיב נוסף שאתייחס אליו מיד, ניתן היה לראות כי שימוש בתמונות בגודל האמור דווקא כן הביא לשיפור מסוים. כאמור, בשלב זה הדיוק כבר די גבוה ולכן גם שיפור קטן נראה מרשים. לכן, החלטתי לשלם את המחיר הנוסף בזמן העיבוד המקדים כדי למקסם את הדיוק ככל הניתן. יש לציין כי גודלו של ייצוג HOG אינו משתנה עקב כך הודות לפעולת ה- binning, ולכן אין למרכיב זה השפעה על זמן ההסקה של KNN.

ככל הנראה הגדלת התמונות הביאה לשימור של מידע חשוב, אך גם להרבה מידע מיותר לחלוטין – הרקע של התמונה שנמצא מאחורי האות – ברור שכדאי לנסות להסיר מידע זה באופן סלקטיבי לפני הוצאת המאפיינים לקבלת ייצוג נקי ככל האפשר. מצד שני, כל פעולת ניקוי עלולה להסיר שוב חלק מהמידע ששילמתי עליו מחיר יקר. לאחר שלל ניסיונות כושלים עם פילטרים ופעולות סף שונות, מצאתי פעולה אחת שהצליחה להביא לשיפור קל נוסף בדיוק – opening/closing morphology [5] – פעולה זו הצליחה להסיר כמות משמעותית של רעש רקע מבלי לפגוע במידע החשוב – עקב כך, גם פעולה פשוטה זו (בשילוב עם הגדלת התמונות) מהווה מרכיב נוסף, אך לא הכרחי, בפתרון שלי.

לצורך בחירת הערך האופטימאלי עבור הפרמטר העיקרי של KNN (מספר השכנים k), אני מבצע תהליכי 10-fold cross-validation כאשר ה- k המשרה את הדיוק הממוצע הגבוה ביותר נבחר. יש לשים לב כי הנתונים בבעיה הם בעצם התמונות המקוריות ולכן החלוקות השונות לקבוצות אימון\ולידציה הן ביחס לתמונות אלו ולא ביחס לתמונות של האותיות הבודדות. עקב סיבה טכנית זו (וגם עקב תהליך ההצבעה הייחודי), נאלצתי לכתוב שגרה לביצוע תהליך הוולידציה במקום להשתמש בשיטות מוכנות של sklearn. אציין אנקדוטה מעניינת בהקשר הזה – בשלב מסוים באימון המודלים השונים שמתי לב כי ערכי הדיוק בהרצת תהליך הוולידציה הלכו וירדו

ככל שהגדלתי את קבוצות האימון (למשל, ע"י הגדלת מס' ה- folds). מסתבר שאנומליה זו נבעה מכך שלא ביצעתי את החלוקה של הנתונים לקבוצות באופן שכרגע ציינתי. מדוע? מכיוון שתהליך ההצבעה עבור האותיות הולך ונפגם ככל שקבוצת המבחן קטנה עקב ההסתברות ההולכת וגדלה של אות מקרית כלשהי בקבוצת המבחן להיות "מנותקת" מהאותיות האחרות (בקבוצת האימון) ששייכות למילתה.

לבסוף, התאמתי את המודל לכל הנתונים בתקווה למקסם את הדיוק שיתקבל בתוצאות הסופיות של התחרות.

הפתרון המוצע

להלן סיכום כרונולוגי של הפתרון המוצע. קרי, בהינתן תמונה עם מטא דאטה מתאים מתבצע:

1. חילוץ תמונות 64x64x3 של האותיות (תמונות צבע) באמצעות שיטת ה- inverse warping, תוך שימוש באינטרפולציה מעוקבת.
2. ניקוי התמונות משלב 1 באמצעות פעולות מורפולוגיה. קרי, ביצוע opening ולאחר מכן ביצוע closing עם structuring element מלבני בגודל 5.
3. חישוב מאפייני HOG (גרסה ייחודית שלי) על התמונות משלב 2. קרי, מעבר עם פילטר Sobel בגודל 7x7 (פילטר גדול הוא פחות רגיש לרעשים) על כל אחד מערוצי הצבע לצורך חישוב הנגזרות החלקיות, חישוב הגדלים והזוויות של הגרדיאנטים תוך בחירת הגדולים ביותר (מבחינת הנורמה של הגרדיאנט) ביחס לממד הצבע, כאשר לכל תא (ישנם 16x16 תאים סה"כ) מתבצעת חלוקה ל- 8 סלים עפ"י הזוויות המתאימות כאשר הגדלים המתאימים ממלאים את הסלים. לבסוף, כל ההיסטוגרמות הנ"ל משורשרות לווקטור מאפיינים אחד בגודל $8 \cdot 16^2 = 2048$.
4. מודל KNN (עם פרמטר k שנבחר באמצעות 10-fold cross-validation) המבוסס על מרחקי cosine שהותאם לכל נתוני האימון מופעל על ייצוגי האותיות שהתקבלו בשלב 3.

5. המרחקים שחושבו בשלב 4 והמטא דאטה הנתון לגבי המילים נשלחים לתהליך המבצע את ה- weighted majority voting לצורך חיזוי התיוגים.

אציין כי בחירת הגרדיאנטים הגדולים ביחס לממד הצבע משפרת את הדיוק ביחס לשימוש בתמונות grayscale [3], ובכך בעצם בא לידי ביטוי השימוש בצבע לצורך שיפור היכולת להבדיל בין האות לרקע. אציין כי ישנן דרכים אחרות לחישוב הגדלים והזוויות הנ"ל במקרה של תמונות צבע [6], אך מצאתי כי דרך זו היא הטובה ביותר בבעיה הנדונה.

ניסיונות כושלים

להלן מקבץ של ניסיונות שלא העלו פרי:

- שימוש בתמונות grayscale או במודלי צבע שונים כדוגמת HSV.
- שימוש בשיטות סף כדוגמת Otsu's method לקבלת תמונות בינאריות המכילות את האותיות בלבד.
- ניקוי נוסף באמצעות שילובים שונים של פילטרים כגון: Gaussian, median, bilateral.
- שימוש במאפייני SIFT בשיטת BoW בדומה לנעשה במטלה השנייה.
- הורדת ממד באמצעות PCA.
- סטנדרטיזציה של התמונות\מאפייני HOG.
- שימוש במידע לגבי האותיות עצמן לצורך הסרת חריגים ע"י התאמת מודל GMM לקבוצות של אותיות מאותו סוג תוך סילוק אותיות מקבוצת האימון אשר יש להן log-likelihood קטן יחסית.
- שימוש במודלים שונים מ- KNN כדוגמת random forest.
- שימוש ב- boosting\bagging.

רעיונות נוספים

להלן מקבץ של רעיונות מעניינים שלא היה זמן לנסות:

- שימוש במידע לגבי האותיות עצמן לצורך בניית מסווג ייעודי עבור כל אות.
- אימון רשת עמוקה ושימוש במאפיינים הנוצרים בשכבה פנימית (למשל, זו שלפני שכבת הפרדיקציה) במקום HOG והתאמת מודל KNN להם.
- שימוש ברשת עמוקה שמקבלת כקלט גם את התמונות וגם את ייצוגי HOG.

מבנה הקוד

הקובץ The Project.ipynb משתמש בשיטות ובקבועים המופיעים בקבצי הקוד השונים והוא נועד בעיקר לצרכי ויזואליזציה, הצגת תוצאות של תהליך העבודה, ושמירת המודל הסופי. קרי, אין בו צורך כדי לייצר את החיזויים עבור קבוצת המבחן. המחברת הנ"ל מלווה בהסברים לצד תוצאות מפורטות, ולכן לא ארחיב עליה כאן.

הקובץ head.py מכיל את הספריות העיקריות ומס' קבועים. לכן, רצוי לעיין בו ובקבצי הקוד הנוספים לפני קריאת המחברת.

הקובץ prep.py מכיל את השיטות:

- extChr – מקבלת תמונה שלמה ותיבה של אות, ומחלצת אותה.
- clnChr – מקבלת אות שחולצה ע"י extChr ומבצעת תהליך ניקוי.
- hog – מקבלת תמונה ומייצרת את ייצוג HOG.
- getDatPt – מקבלת שם של תמונה ומחזירה את כל הנתונים הרלוונטיים.
- prepData – מקבלת פונקציות חילוץ, ניקוי, ויצירת מאפיינים, ומכינה שתי רשימות X_data ו- y_Data של נתונים. ניתן למצוא פירוט לגבי מבנה אברי הרשימות הנ"ל בהסברים המופיעים במחברת, ולכן לא אחזור עליהם כאן.

לשם הנוחות, הקובץ מכיל גם שיטות לשינוי גלובלי של גודל התמונות המחולצות, שיטת האינטרפולציה, וגודל פילטר המורפולוגיה.

הקובץ test.py מכיל את השיטות:

- split – מקבלת נתונים ואינדקס של fold, ומחלקת אותם לקבוצת אימון וקבוצת מבחן בהתאם.
- majVot – מקבלת חיזויים ומטא דאטה מתאים, ומבצעת את תהליך ההצבעה (מתבצעת הצבעה רגילה במקרה שבו מערך החיזויים המתקבל הוא מסוג int).
- distScrs – מקבלת מודל KNN מותאם ונתוני מבחן, ומחשבת scores עבור כל נתון בהתאם למרחקים ההופכיים כפי שהוסבר – חיזויים אלו ישמשו את השיטה הקודמת לצורך חישוב ה- weighted majority voting.
- cv – מקבלת מודל, נתונים, מספר folds, שיטת פרדיקציה, ושיטת הצבעה, ומבצעת תהליך cross-validation מתאים. לבסוף, שיטה זו מחזירה את ממוצע הדיוק וסטיית התקן.

התוכנית pred.py מכילה שיטת main ומבקשת שם של קובץ h5. מהמשתמש בעת הפעלתה (ראה איור).

התוכנית תטען את המודל ותתחיל בקריאה ועיבוד של הדאטה הנמצא בקובץ h5. המבוקש, ולאחר מכן תפעיל את המודל על הדאטה המעובד לקבלת המרחקים אשר ישמשו לצורך ההצבעה המשוקללת אשר תקבע את התיוגים הסופיים. לבסוף, תבנה טבלה בפורמט המבוקש ותישמר בקובץ csv., כנדרש (קובץ כזה כבר נמצא בתיקיית הפרויקט לצורכי אימות).

מקורות

[1] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[2] Distinctive Image Features from Scale-Invariant Keypoints, David G. Lowe

https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[3] Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection. International Conference on Computer Vision & Pattern Recognition (CVPR '05), Jun 2005, San Diego, United States. pp.886–893, ff10.1109/CVPR.2005.177ff. ffinria-00548512f

https://hal.inria.fr/file/index/docid/548512/filename/hog_cvpr2005.pdf

[4] J. Han, M. Kamber & J. Pei, Data Mining: Concepts and Techniques, 3rd ed. (see 2.4.7 Cosine Similarity)

http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[5] https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

[6] A Note on the Gradient of a Multi-Image, SILVANO DI ZENZO

https://people.csail.mit.edu/tieu/notebook/imageproc/dizenzo86.pdf