מבוא לראייה ממוחשבת -22928

פרויקט למידה עמוקה

ירדן פרנקל - 315329490

תקציר:

בפרויקט זה, היה עליי לפתור את בעיית הסיווג לפונטים של אותיות אשר קיימות על תמונות. דהיינו בהינתן תמונה עם טקסט ,עליי להכריז עבור כל אות בתמונה, לאיזה מבין שבעת הפונטים הנתונים הינה שייכת.

רשימת הפונטים הם:

Alex Brush, Michroma, Raleway, Russo One, Ubuntu Mono ,Roboto

במהלך העבודה נתקלתי במספר שאלות על המודל ,על אופן העבודה ועל הדרך הכי טובה לפתור את הבעיה.
במסמך זה אשתף את השאלות, ואנסה לענות על רובן, חלק מן השאלות נשארו פתוחות ועל חלקן הפתרון אינו מוכח באופן מדעי אלא נפתר על ידי "ניסוי וטעיה" ובכך הגעתי לתוצאות הטובות ביותר עבורי.

על מנת להקל על הקריאה:

- ? לצד כל שאלה יופיע הסמל
- לצד כל גרף יופיע הסמל
- ✓ לצד כל בדיקה יופיע הסמל
- ⏱ ולעיתים ישנם מצבים בהם באמצעות ניסוי וטעיה הגעתי למסקנה הרצויה ולכן יופיע הסמל (תשובות ככל הנראה יופיעו בהמשך)

בעבודתי התבססתי על המידע אשר ניתן בהרצאות ,ובקורס CNN for Visual Recognition by Standford University

כמו כן בפרויקט זה השתמשתי במבחר ספריות וטכנולוגיות כגון:
'Keras by Tensorflow' , 'Pandas' , 'Numpy', 'CV2'
וכלי ויזואליזציה כגון:
'Seaborn', 'Matplotlib' and others

מסמך זה מהווה סיכום של התהליך שנעשה אני יותר ממליץ להיכנס למחברת המצורפת על מנת לקבל את התמונה המלאה ועוד נתונים אשר אינם מצורפים במסמך זה

לינק למחברת

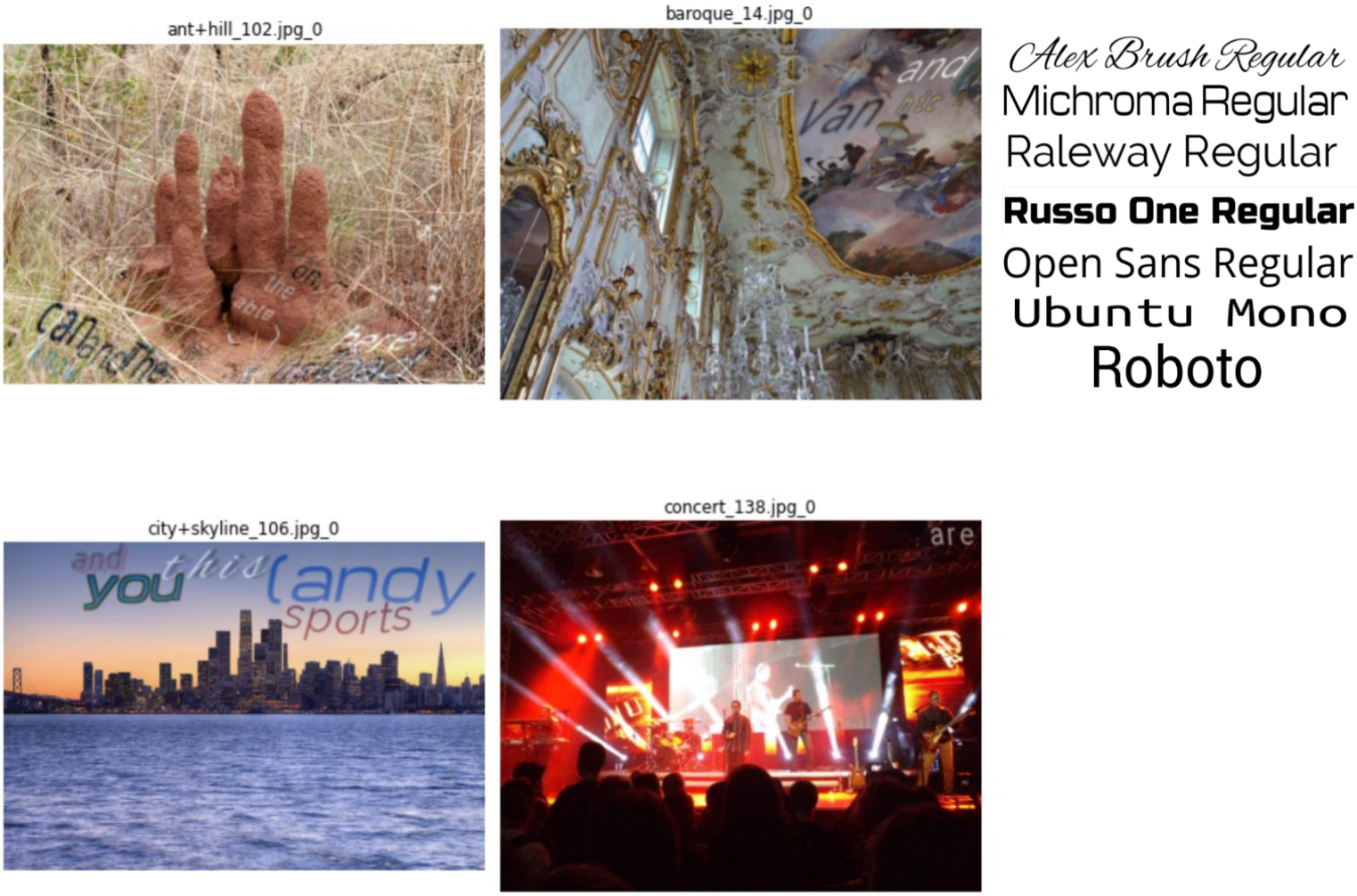
קריאה מהנה!

תוכן עניינים

על הדאטה:

מערך הנתונים עליו הפרויקט מבוסס (SynthText) מורכב מתמונות סצנה טבעיות, שעליו ממוקמים מילים ואותיות אשר הורכבו באופן סניטתי תוך התחשבות בפריסת הסצנה בתמונה.

מערך הנתונים זה, מורכב מ973 תמונות. אשר לכל תמונה מצורפות גבולות הגזרה של המילים והאותיות הנמצאות בתוכו. כל מילה בתמונה מורכבת בדיוק מפונט אחד מהפונטים הבאים:



1תמונות לדוגמה

בניית מערך הנתונים:

על מנץ לקבל את מירב המסקנות שאנו יכולים לקבל מין המידע עלינו לשמור על כל אות בכל תמונה את התכונות הבאות:

1. כותרת התמונה.
2. מספר האות במילה.
3. האות עצמה.
4. מספר המילה בתמונה.
5. המילה עצמה.
6. שם הפונט.
7. גובה התמונה.
8. רוחב התמונה.
9. התמונה הגזורה של האות מתוך התמונה המקורית.

כעת לפני שנתחיל בבניית מסד הנתונים עלינו לעבור על כל אות בכל תמונה ולגזור אותה מהתמונה המקורית.
כאשר ניגשתי לתחילת עבודת ייצוא המידע נתקלתי במספר שאלות שעוררו את סקרנותי. הרי ידוע כי המודל שאני מעוניין לבנותו צריך ללמוד מתמונות אשר גודלן קבוע, אך לא כל האותיות בתמונות מקיימות תנאי זה, ולכן בהינתן והמודל העתידי יתאמן על תמונות בגודל (X,Y) , כל אות נופלת לאחד מתוך שלושה דליים:

1. התמונה בגודל הרצוי (האפשרות האופטימלית).

2. התמונה קטנה מהגדול הרצוי – במקרה זה התמונה תזדקק לתהליך upscaling וכאשר תמונות עוברות תהליך זה, הן מייצרות עוד מידע שאינו היה קיים לפני כן, ובכך "הורסות" את אמינות הנתונים שהיו לתמונה לפני תהליך זה. כמו כן מודל המתאמן על תמונות גדולות יותר זקוק לכוח חישוב גדול יותר על מנת להתאמן.

3. התמונה גדולה מהגדול הרצוי – במקרה זה התמונה תזדקק לתהליך downscaling וכאשר תמונות עוברות תהליך זה, הן מאבדות מידע שהיה קיים לפני כן, ובכך גם כן "הורסות" את אמינות הנתונים שהיו לתמונה וגם עלולות לאבד מידע חיוני של התמונה המקורית.

ראשית נסביר באילו דרכים ניתן לגזור את התמונות מהתמונה המקורית:

⏱ במהלך חיפוש אחר פתרון מצאתי שתי שיטות עיקריות שבהן בחרתי להשתמש לגזירת התמונות:

1. גזירת מלבן (ירוק).

2. גזירת טרנספורמציה (אדום).

גזירת טרנספורמציה:

הגזירה מתבצעת באופן הבא:

1.בהנתן גבולות גזרה של אות, נחשב את הטרנספורמציה הלינארית בין התמונה לצירים.
2. נגזור את המלבן לפי הגבולות הנתונים.
3. נבצע יישור של התמונה הגזורה לצירים לפי הנתונים מסעיף 1

יתרונות

- האותיות הגזורות מיושרות עם הצירים.

חסרונות

- המידע בתמונה עשוי לעבור עריכה כלשהי בעקבות הטרנספורמציה.
- האות חתוכה בדיוק לפי גבולות הגזרה.

גזירת מלבן:

הגזירה מתבצעת באופן הבא:

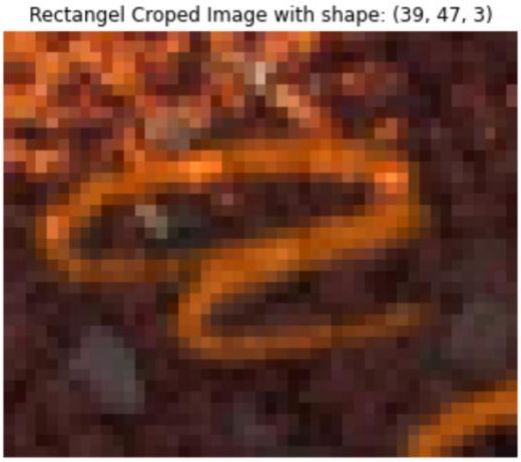
1.בהנתן גבולות גזרה של אות, נחשב את המינימום והמקסימום לפי הצירים של גבולות הגזרה.
2. נציב את גבולות המלבן לפי המינימום והמקסימום שמצאנו בסעיף 1
3. נגזור את המלבן לפי גבולות אלו.

יתרונות

- האות בשלמותה נמצאת בתוך המלבן.
- התמונה לא עברה עריכה כלל.

חסרונות

- האותיות הגזורות אינן מיושרות עם הצירים (כפי שניתן לראות בתמונה להמחשה).



בעקבות כך, תהליך בניית המסד מעלה מספר שאלות שבמהלך הפרויקט ננסה לענות עליהן:

❓ 1. מהו גודל התמונה האופטימלי עבור המודל?

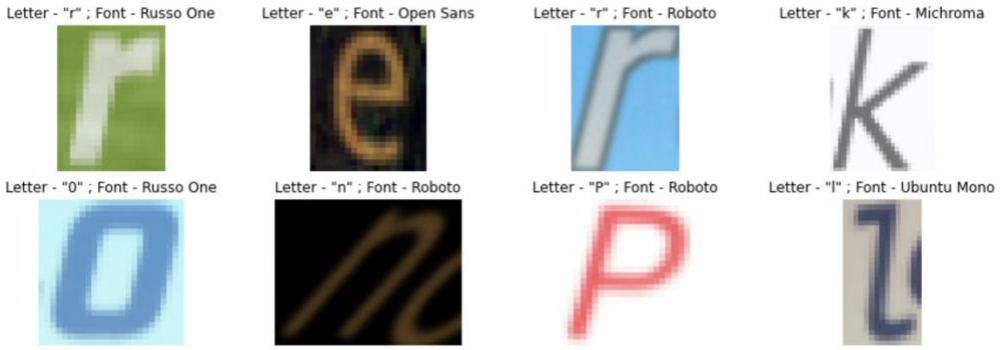
❓ 2. האם למודל יהיו תוצאות טובות יותר עם אותיות מיושרות כלפי הצירים?

❓ 3. האם למודל יהיו תוצאות טובות יותר עם תמונות בגווני אפור?

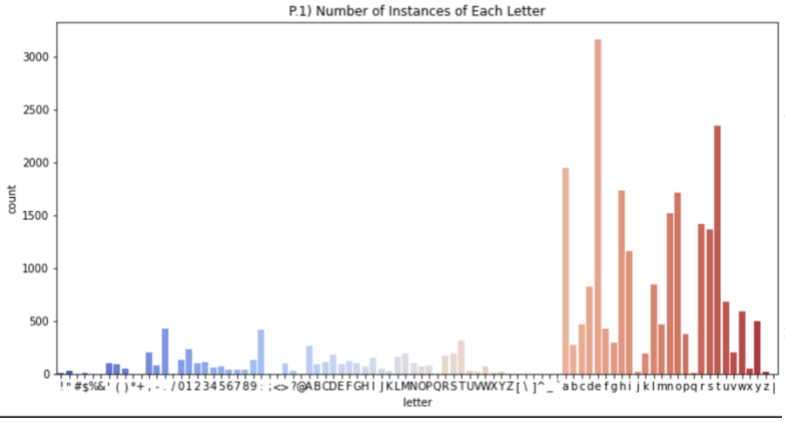
⏱ *החלטתי תחילה לבנות את מסד הנתונים באמצעות "גזירת טרנספורמציה*

ניתוח הדאטה:

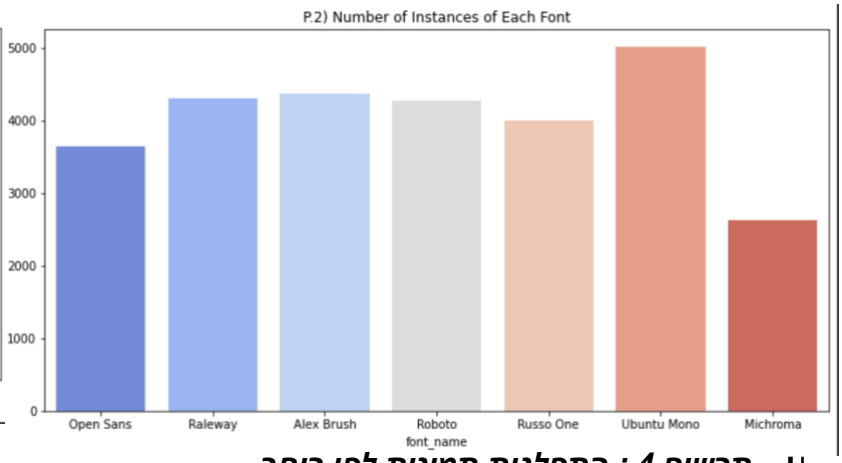
אותיות גזורות לדוגמא.



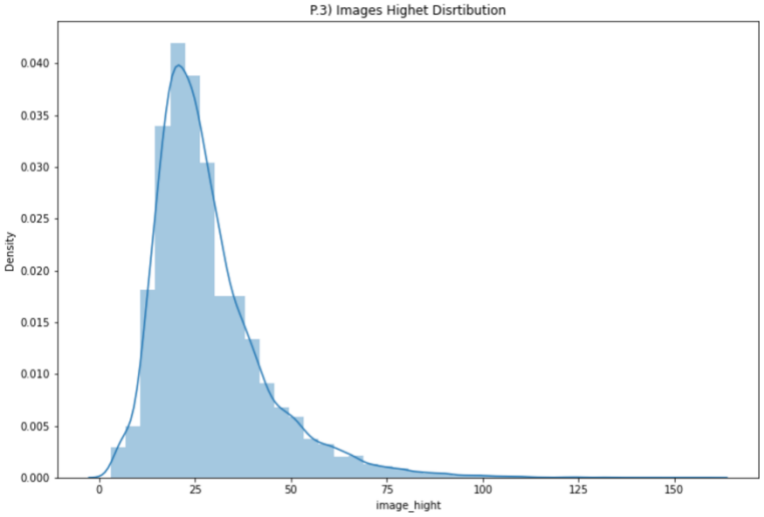
תרשים 1 : התפלגות אותיות לפי אות.



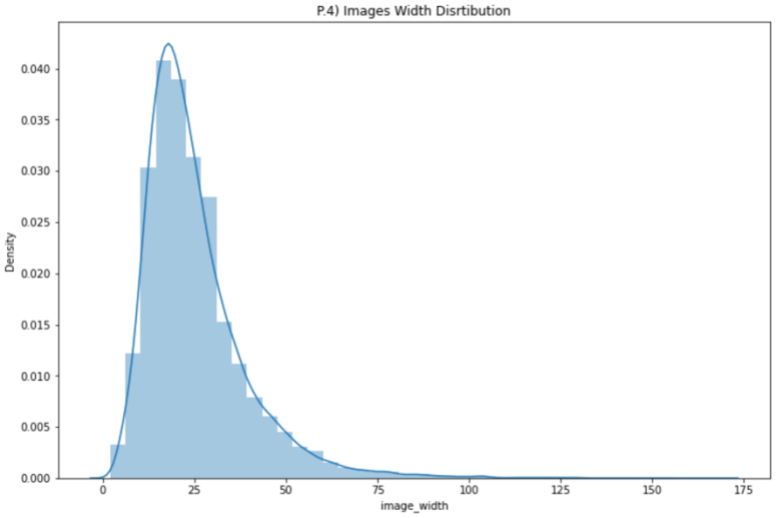
תרשים 2 : התפלגות אותיות לפי פונט.



תרשים 3 : התפלגות תמונות לפי גובה.



תרשים 4 : התפלגות תמונות לפי רוחב.



מסקנות מתרשימים אלו : (תובנות נוספות במחברת)

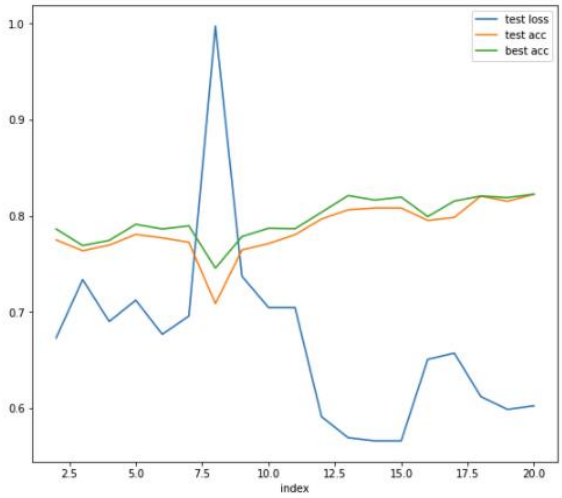
1. ישנן אותיות שהייצוג שלהן נמוך במאגר התמונות.
2. ישנו ייצוג יחסית טוב של כלל הפונטים.
3. ממדיי התמונות מתפלג בעיקר סביב (25,25) בממוצע.
4. התמונות בעלי ממדים קטנים מ (10,10) הן בעיקר סימנים מיוחדים.

בניית המודל:

במהלך בניית המודל ביצעתי בדיקות על מגוון רשתות בארכיטקטורות שונות. לטובת בניית המודל שלי השתמשתי בספרייה Keras של Tensorflow. ובאובייקטים הבאים:
צ׳ק פוינט – אובייקט אשר מקבל מדד מסוים, ושומר את המודל בעל המדד הטוב ביותר בזמן הריצה.
עצירה מוקדמת – אובייקט אשר מקבל פרמטרים ״סבלנות״ ומדד מסוים .ועוצר כאשר את אימון המודל כאשר המדד לא השתפר כמספר הפעמים ששודר בפרמטר הסבלנות.
אנסה לתאר באמצעות הטבלה הבאה את סדר הבדיקות שבוצעו על מנת להגיע למודל הסופי:
(אנא התייחסו לטבלה כאל ״השתלשלות אירועים״)

מספר המודל	הניסוי	השינוי מהמודל הקודם	הסיבה לשינוי	זמן ריצה	אחוזי דיוק
1	ראשית חיפשתי מודל מוכר לכל -והעתקתי את הארכיטקטורה של Alexnet	-	-	ארוך מאוד	40%
2	אותו המודל	הקטנתי את כמות הפילטרים	זמן הריצה הארוך בתוספת התוצאה שאינה מספקת	עדיין ארוך מאוד	אותו סדר גודל
3	מודל קטן 2 שכבות קונבולוציה ושיטוח	ארכיטקטורה שונה לגמרי	זמן הריצה היה ארוך מידי עבור תוצאות שאין מספקות כלל והעדפתי להתחיל דווקא ממודל קטן יותר ומשם להתקדם	קצר	55%
4	אותו המודל	הוספת 2 שכבות pooling	ניסוי לשיפור המודל	קצר	60%
5	אותו המודל	Batch normalization	ניסוי לשיפור המודל	קצר	63%
6	אותו המודל	הוספת עוד שכבת Fully connected	ניסוי לשיפור המודל	קצר	70%
7	אותו המודל	הוספת עוד שכבת conolution	ניסוי לשיפור המודל	קצר	75%

⏱ כעת לאחר שהגעתי למודת יחסית מספק מבחינת אחוזי הדיוק, התחלתי לבצע כיוונון לפרמטרים במודל את תוצאות הריצה של כל מודל שידרתי לקובץ עד שבסוף התהליך קיבלתי את הטבלה הבאה:



	index	activation	optimizer	learning-rate	loss function	batch size	epoch	early stoped at	test loss	test acc	best acc
0	2	relu	adadelta	1	categorical_crossentropy	128	50	38	0.6732	0.77510	0.78647
1	3	relu	adam	default	categorical_crossentropy	128	50	37	0.7338	0.76380	0.76920
2	4	relu	adam	default	categorical_crossentropy	64	50	48	0.6902	0.76970	0.77460
3	5	prelu	adam	default	categorical_crossentropy	64	50	33	0.7124	0.78080	0.79130
4	6	prelu	adadelta	1	categorical_crossentropy	64	50	48	0.6769	0.77720	0.78640
5	7	leakyRelu	adadelta	1	categorical_crossentropy	64	100	44	0.6959	0.77270	0.78970
6	8	relu	adam	default	categorical_crossentropy	64	50	26	0.9976	0.70880	0.74570
7	9	leakyRelu	adam	default	categorical_crossentropy	64	50	42	0.7373	0.76460	0.77860
8	10	prelu	adam	default	categorical_crossentropy	32	50	26	0.7048	0.77130	0.78720
9	11	prelu	adam	default	categorical_crossentropy	32	50	30	0.7047	0.78050	0.78670
10	12	prelu	adam	default	categorical_crossentropy	32	50	42	0.5912	0.79690	0.80390
11	13	prelu	adam	default	categorical_crossentropy	32	100	60	0.5693	0.80640	0.82120
12	14	prelu	adam	default	categorical_crossentropy	64	50	50	0.5661	0.80830	0.81660
13	15	prelu	adam	default	categorical_crossentropy	254	50	50	0.5661	0.80830	0.81960
14	16	relu	adam	default	categorical_crossentropy	32	50	32	0.6509	0.79540	0.79940
15	17	prelu	adam	default	categorical_crossentropy	32	50	34	0.6573	0.79850	0.81550
16	18	prelu	adam	default	categorical_crossentropy	64	50	38	0.6121	0.82070	0.82070
17	19	prelu	adam	default	categorical_crossentropy	64	50	34	0.5988	0.81530	0.81910
18	20	prelu	adam	default	categorical_crossentropy	64	50	40	0.6025	0.82259	0.82259

📊 לפי הטבלה השתדלתי לקחת את המודל בעל ה- loss הכי נמוך וה- acc הכי גבוה.

שיפור המודל באמצעות הדאטה:

בחלק זה נשתדל לענות על השאלות שנשאלו בתחילת המסמך, ונראה כי באמצעות מספר פעולות על הדאטה ניתן לשפר את המודל גם מבלי לשנות את המבנה שלו. המסקנות המובאות להלן מבוססות על סטטיסטיקות ונתונים שניתן לראות את תהליך החשיבה עליהן בתוך המחברת ולכן מוצגות כאן באופן תמציתי. (ייתכן ומקרה זה הוא פרטי למבנה המודל שלי ולכן אין לייחס לכך משמעות גורפת).

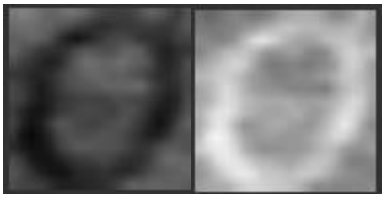
✓ 1. גודל התמונה האופטימלי עבור המודל בעל הביצועים הטובים ביותר מבין הגדלים הבאים (32,32) , (48,48) , (64,64) הוא **(32,32)** ובנוסף ככל שהגדלתי את הממדים תוצאות המודל הרעו.

✓ 2. המודל שרץ על התמונות הגזורות בשיטת גזירת הטרנספורמציה הניב תוצאות טובות יותר מהגזירה המלבנית.

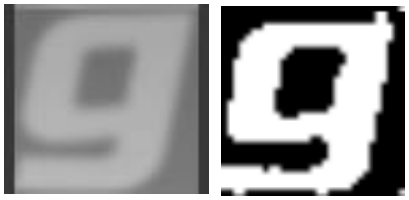
✓ 3. המודל שרץ על תמונות בגווני אפור הניב תוצאות טובות יותר מהתמונות הצבעוניות.

✓ 4. בנוסף ,באמצעות מספר פעולות העשרה על הדאטה ניתן היה להשיג תוצאות טובות יותר ולהגיע לאחוזי דיוק גבוהים יותר. הפעולות שביצעתי הן :

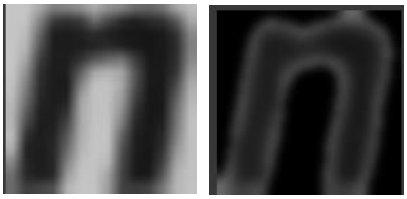
- העשרת הדאטה באמצעות ניגודי התמונות (כל פיקסל בתמונה המקורית =- 255).



- העשרת הדאטה באמצעות ת׳רשולדינג. (Thresholding)



- העשרת הדאטה באמצעות מסכה בינארית. קישור להסבר



✓ 6. בנוסף נשתמש בנתון נוסף אשר יעלה את אחוזי החיזוי של המודל. ידוע כי לכל מילה בתמונה פונט זהה כלומר כל האותיות השייכות לאותה המילה הן מאותו הפונט, ויש ברשותנו את המידע הזה בבסיס הנתונים. לפיכך, לאחר חיזוי המודל, נבצע אגרגציה לפי המילים בתמונה ונספור מהי הפרדיקציה שחזרה על עצמה הכי הרבה פעמים בעבור כל מילה, ולפיה נוכל לבצע תיקון לאותיות שהפרדיקציה שלהם הייתה לא נכונה. פעולה זו תשמש כבדיקה אחרונה ותיקון סופי לתוצאות

על מנת להשתמש בכלי זה ישנן 2 אפשרויות מימוש, נזכיר כי עבור כל אות המודל מחזיר מערך בגודל 7 כמספר הקטגוריות שלנו ועליהם ציונים בין 0 ל1 כאשר גם סכום האיברים במערך הוא 1, דהיינו ככל שהמספר בתא מסוים גבוהה יותר כך המודל "חושב" שמדובר בפונט זה.
כעת האפשרויות הן:

1. עיגול הפרדיקציות לפני ולאחר מכן בדיקת מספר ההופעות המקסימלי. לדוגמה:

image_title	word	letter	predictions	predictions_value	max_pred
kerala_113.jpg_0	Epilepsy	E	[4.6745e-08 1.8359e-02 2.8098e-03 5.0994e-05 6.1496e-01 2.2222e-01 1.4160e-01]	4	6
kerala_113.jpg_0	Epilepsy	p	[2.8378e-08 1.0933e-05 1.3764e-02 9.3623e-07 1.3643e-01 1.5693e-02 8.3411e-01]	6	6
kerala_113.jpg_0	Epilepsy	i	[1.3923e-04 2.7913e-01 6.9767e-02 2.1173e-01 1.3372e-01 4.5331e-03 3.0099e-01]	6	6
kerala_113.jpg_0	Epilepsy	l	[1.9978e-04 1.8767e-01 2.5207e-01 9.3634e-04 9.0501e-02 5.4965e-04 4.6808e-01]	6	6
kerala_113.jpg_0	Epilepsy	e	[6.3645e-12 8.6500e-07 2.9792e-08 1.7030e-07 8.4281e-01 1.5691e-01 2.7206e-04]	4	6
kerala_113.jpg_0	Epilepsy	p	[3.6049e-11 5.6361e-08 4.8695e-03 3.0089e-09 1.5891e-01 1.1137e-03 8.3510e-01]	6	6
kerala_113.jpg_0	Epilepsy	s	[1.1930e-07 5.7356e-07 1.6216e-03 5.6723e-05 2.4473e-01 7.5346e-01 1.3783e-04]	5	6
kerala_113.jpg_0	Epilepsy	y	[4.1106e-07 1.0958e-04 6.3780e-04 5.9699e-06 6.3629e-01 3.4394e-01 1.9016e-02]	4	6

לפיכך כפי שמסומן בצהוב המספר שחזר הכי הרבה בפרדיקציות הוא 6 ולכן הפרדיקציה האגרגטיבית תהיה 6 (באדום)

2. סכמת וקטורי הפרדיקציות ולאחר מכן עיגול לבחירה הנכונה, כך לדוגמא סכום הוקטורים בדוגמא שלעיל הוא הוקטור :

[3.3961e-04 4.8528e-01 3.4554e-01 2.1278e-01 2.8583e+00 1.4984e+00 2.5993e+00]

וכפי שניתן לראות האינדקס המקסימלי הוא 4 ולכן השיטה השנייה תשנה את כלל הפרדיקציות להיות 4.
ולפי בדיקה שערכתי בין השיטות השיטה הראשונה משפרת את המודל מ83.1 אחוזי דיוק ל91.8 אחוזי דיוק. ואילו השיטה השנייה משפרת את המודל מ83.1 אחוזים ל95.1 אחוזים.

ולכן נבחר להשתמש בשיטה השנייה ולהלן סיכום של שתי התוצאות ביחד עם הפונט המקורי:

image_title	word	letter	predictions	font_no	predictions_value	max_pred	pred_val
kerala_113.jpg_0	Epilepsy	E	[4.6745e-08 1.8359e-02 2.8098e-03 5.0994e-05 6.1496e-01 2.2222e-01 1.4160e-01]	4	4	6	4
kerala_113.jpg_0	Epilepsy	p	[2.8378e-08 1.0933e-05 1.3764e-02 9.3623e-07 1.3643e-01 1.5693e-02 8.3411e-01]	4	6	6	4
kerala_113.jpg_0	Epilepsy	i	[1.3923e-04 2.7913e-01 6.9767e-02 2.1173e-01 1.3372e-01 4.5331e-03 3.0099e-01]	4	6	6	4
kerala_113.jpg_0	Epilepsy	l	[1.9978e-04 1.8767e-01 2.5207e-01 9.3634e-04 9.0501e-02 5.4965e-04 4.6808e-01]	4	6	6	4
kerala_113.jpg_0	Epilepsy	e	[6.3645e-12 8.6500e-07 2.9792e-08 1.7030e-07 8.4281e-01 1.5691e-01 2.7206e-04]	4	4	6	4
kerala_113.jpg_0	Epilepsy	p	[3.6049e-11 5.6361e-08 4.8695e-03 3.0089e-09 1.5891e-01 1.1137e-03 8.3510e-01]	4	6	6	4
kerala_113.jpg_0	Epilepsy	s	[1.1930e-07 5.7356e-07 1.6216e-03 5.6723e-05 2.4473e-01 7.5346e-01 1.3783e-04]	4	5	6	4
kerala_113.jpg_0	Epilepsy	y	[4.1106e-07 1.0958e-04 6.3780e-04 5.9699e-06 6.3629e-01 3.4394e-01 1.9016e-02]	4	4	6	4

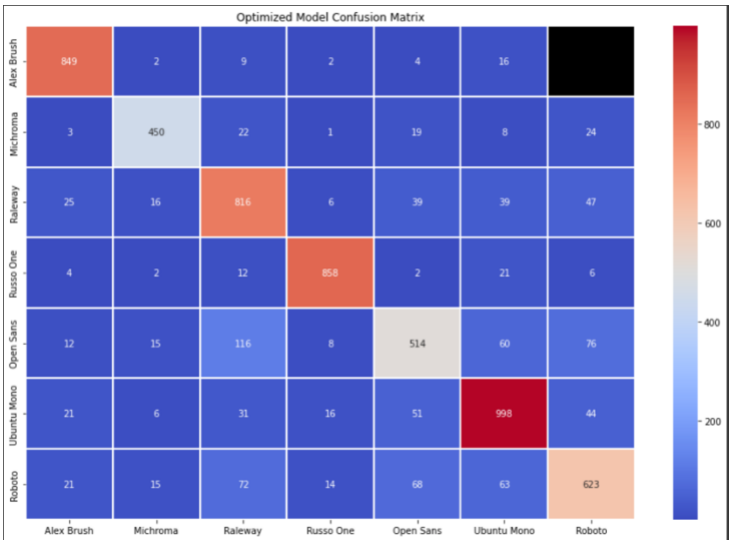
שיערוך המודל וסיכום:

לסיכום ניתן לומר שלאחר צפייה בסרטוני הדרכה, קריאת מאמרים וחקירת דוקומנטציות – למדתי דברים רבים מפרויקט זה. אם מדובר בפרמטרים השושנים שבאמצעותם ניתן לשפר את המודל כגון:
פונקציות אקטיבציה, מספר השכבות, שכבות pooling and batch normalization
טיב הדאטה איכותו ,אוגמנטציה שלו על מנת להגיע לדיוק מקסימלי, שיטות preprocessing
כלי ויזואליזציה וניתוח של דאטה ועוד..

ולתוצאות המודל: (הן בעבור קבוצת הטסט שלי ולכם אחוזים פחות או יותר צריכים להיות זהים)

לפני פונקציית ההקבצה:

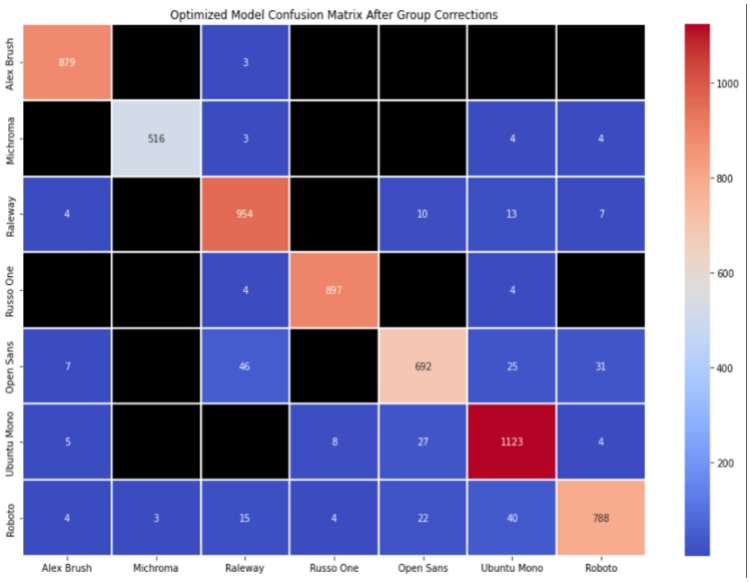
האלה הן הנתונים:




```
Test loss: 0.4704877734184265
Test accuracy: 0.8311096429824829
```

	precision	recall	f1-score	support
Alex Brush	0.91	0.96	0.93	882
Michroma	0.89	0.85	0.87	527
Raleway	0.76	0.83	0.79	988
Russo One	0.95	0.95	0.95	905
Open Sans	0.74	0.64	0.69	801
Ubuntu Mono	0.83	0.86	0.84	1167
Roboto	0.76	0.71	0.73	876
accuracy			0.83	6146
macro avg	0.83	0.83	0.83	6146
weighted avg	0.83	0.83	0.83	6146

לאחר פונקציית ההקבצה:




```
Model accuracy is: 95.168%
```

	precision	recall	f1-score	support
Alex Brush	0.98	1.00	0.99	882
Michroma	0.99	0.98	0.99	527
Raleway	0.93	0.97	0.95	988
Russo One	0.99	0.99	0.99	905
Open Sans	0.92	0.86	0.89	801
Ubuntu Mono	0.93	0.96	0.95	1167
Roboto	0.94	0.90	0.92	876
accuracy			0.95	6146
macro avg	0.95	0.95	0.95	6146
weighted avg	0.95	0.95	0.95	6146