פרויקט מסכם
קורס מבוא לראייה ממוחשבת 22928

זיהוי פונט תווים בתמונה

שובל כהן

208748152

תוכן עניינים

איך מריצים?

אימון המודל

תוך שימוש בקובץ `train/train_model.py`.

<u>קלט:</u> נתיב לקובץ h5 של הDataset הנתון.
ניתן לשנותו בשורת הקוד הראשונה בקובץ (השדה file_path).

<u>פלט:</u> קובץ `saved_model.h5` בו נשמר המודל על משקלותיו.

מודל מאומן ניתן למצוא <u>כאן</u>

סיווג המידע

תוך שימוש בקובץ `test/test_model.py`.

כאשר ישנו מודל מאומן השמור בקובץ `resources/saved_model.h5`, בין אם משתמשים במודל שאומן כבר או שמאמנים אחד שכזה בכוחות עצמכם, ניתן למצוא את סיווגי הDataset אותו רוצים לסווג.

<u>קלט:</u> נתיב לקובץ h5 של הDataset שנדרש לסווג.
ניתן לשנותו בשורת הקוד הראשונה בקובץ (השדה file_path).

<u>פלט:</u> קובץ `results.csv` המכיל את סיווגי כל אות בכל תמונה ע"פ הפורמט הנדרש.

* ניתן באותו האופן לאמן ולסווג את המודל עם ולידציה (train_model_with_validation.py & test_model_with_validation.py), או בשיטת Multi Input (train_multi_input_model.py & test_multi_input_model.py)

את המודל המאומן כמו גם שאר המודלים המאומנים ניתן למצוא <u>כאן</u>

ניתוח הריצות

לאחר הרצת המודל, ניתן לנתח את ריצת המודל באמצעות הלוגים שנשמרו לתוך תיקיית הלוגים ע"י הרצת הפקודה הבאה בטרמינל:

```
> tensorboard --logdir=logs
```

איך זה עובד?

נפרט על כל שלבי התהליך מקבלת הDataset, דרך הכנת המידע שייכנס למודל, אימונו במודל הנבחר ועד להכרעת הסיווג ע"פ רוב במילה.

הכנת הקלט למודל

מכל תמונה שנקבל, יחד עם שאר המאפיינים שלה נייצר אוסף מאפיינים ייחודי לכל אות בתמונה.

1. תמונת התו – תוך שימוש בנקודות Boundary Box – נבצע הטלה פרספקטיבית[1] של תמונת האות לתמונת צבע ריבועית בגודל 64×64 פיקסלים, תוך שימוש במתודה cv2.warpPerspective.
2. תו האות – נלקח מתוך טקסט התמונה במיקום הindex של התו.
3. סיווג פונט התו – נלקח מתוך מערך הפונטים במיקום הindex של התו. (רלוונטי רק כאשר מדובר בDataset שנועד לאימון הרשת)

את התו ותמונתו נשלח כקלט לרשת לחיזוי או לאימון יחד עם סיווג הפונט של התו.

בנוסף נשמור עבור כל תמונה את המילים הנמצאות בה ואת שם התמונה, לצורך הכרעת הרוב במילה וליצירת קובץ התוצאות.

[1] העתקה פרויקטיבית – Wikipedia

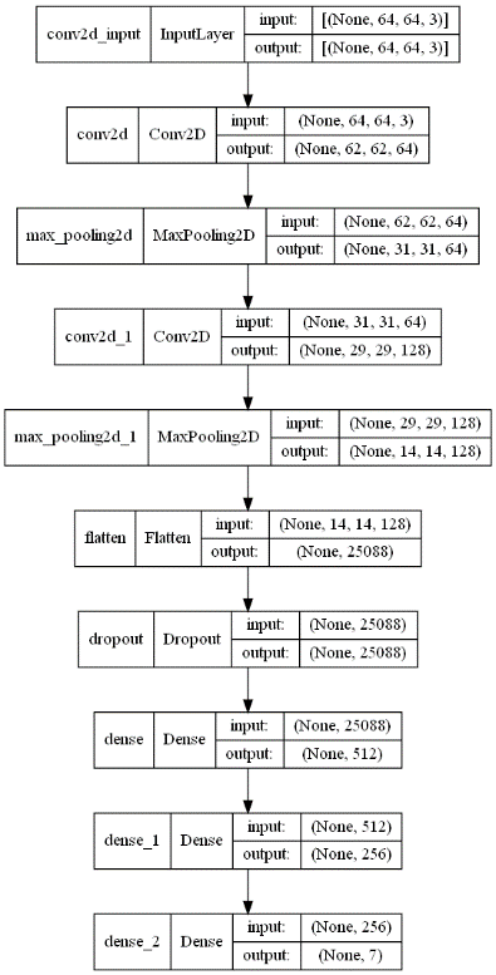
בניית המודל

רשת CNN בסיסית

ראשית בניתי מודל המצליח להוציא את הפיצ'רים מהתמונה בצורה יעילה:

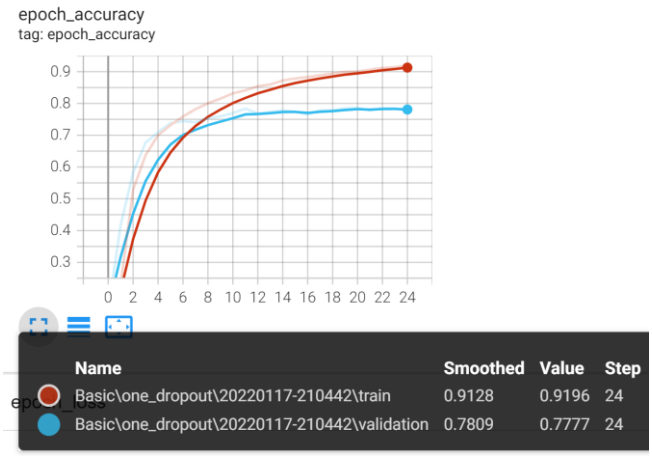
- שתי שכבות קונבולוציה בלבד עם פונקציית הפעלה מסוג relu
  - כאשר לאחר כל אחת מהן מבצעים MaxPooling.
- שכבה המשטחת את הפלט.
- שכבת Dropout בהסתברות חצי
- שתי שכבות Fully Connected עם פונקציית הפעלה מסוג relu
- ולבסוף שכבת הסיווג - שכבת Fully Connected בעלת 7 נוירונים עם פונקציית הפעלה מסוג Softmax.

מצורף משמאל שרטוט של המודל הראשוני ⇐



תוצאות הרצת המודל ב24 ריצות הסתכמו בכ91% דיוק על האימון, אך על הValidation Data – תוצאות הדיוק היו כ77%. מצב זה הינו Overfitting.

ניתן לראות בגרף שהוצאתי בעזרת שימוש בTensorBoard את אחוזי הדיוק:



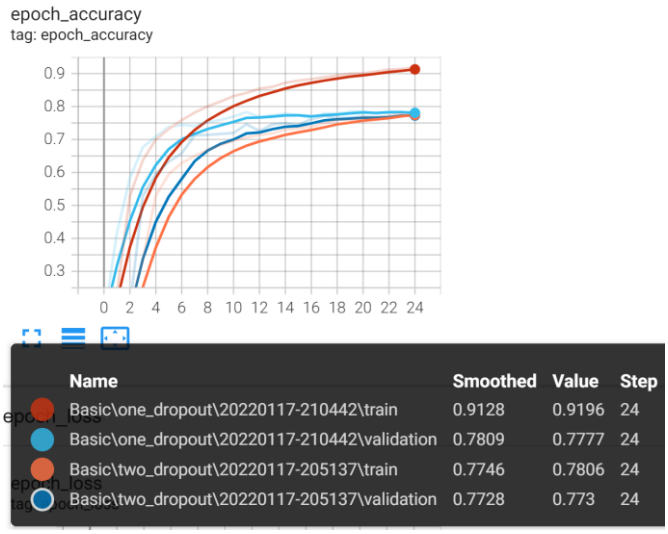
שיפורי המודל

טיפול בOverfitting

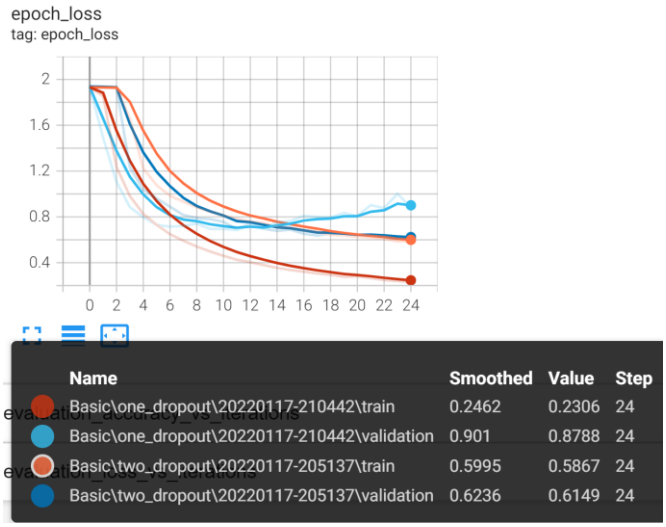
ע"מ לפתור את בעיית הOverfitting החלטתי לשנות את המודל כך שידע להתמודד טוב יותר עם Data שהוא לא מכיר.

הוספתי עוד שכבת Dropout ולא הוספתי עוד שכבות קונבולוציה מיותרות שיגרמו למציאת פיצ'רים ספייציפיים מדי – מה שיגרום לOverfitting.

ניתן לראות שהוספת הDropout אמנם פגעה באחוזי הדיוק של הtrain אך אחוזי הדיוק בValidation נותרו די דומים.



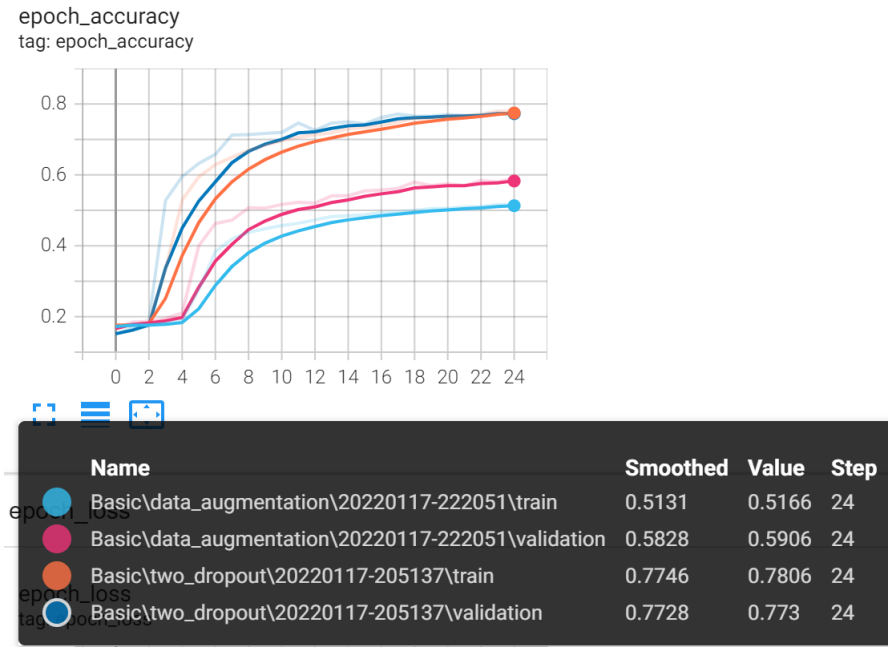
הנתון היותר מעניין הוא נתון הLoss. ניתן לראות שבמצב המודל הבסיסי הLoss של מידע הValidation היה במגמת ירידה עד להרצה ה12 ומשם לא ירד ואפילו עלה לכשהתמשכו הריצות. כאשר הוספנו שכבת Dropout נוספת – נראה כי הLoss קטן במשך הרצות המודל בדומה לLoss של הTrain:



דרך נוספת למנוע Overfitting היא הוספת שכבת Data Augmentation המשנה את מידע האימון בכל הרצה כך שידע להתמודד עם מגוון רחב יותר של תמונות.

ביצעתי Data Augmentation ע"י הפיכתו באופן רנדומלי בכיוון אופקי או אנכי והטיה רנדומלית של התמונה.
לא ראיתי צורך בהגדלת והקטנת התמונה מכיוון שההטלה שינתה את כלל התווים לאותה הפרספקטיבה.
[בכל זאת הטלה פרספקטיבית (:]

תוצאות הוספת הData Augmentation:



השימוש בData Augmentation רק הוריד את ביצועי הדיוק של המסווג ספייציפי זה ועל כן לא השתמשתי ביכולת זו.

נעזרתי במאמר הזה לשיפור הOverfitting:

[5 Techniques to Prevent Overfitting in Neural Networks]

[More on CNNs & Handling Overfitting]

מודל רב קלטים - Multiple Input Model

חשבתי מה עוד אני יודע על התמונה שאני יכול להביא למודל בכדי שיקל עליו להבדיל בין הפונטים.

הבנתי שיש לי בעצם לכל תמונה של תו פיצ'ר נוסף שאני יכול להשתמש בו ונתון לי – והוא ערך התו המוצג בתמונה. כאשר ישנם מצבים בהם התמונה נראית די דומה ויש צורך בהכרעה בין שני פונטים – הוספת ערך התו יכולה להקל על ההכרעה במיוחד.

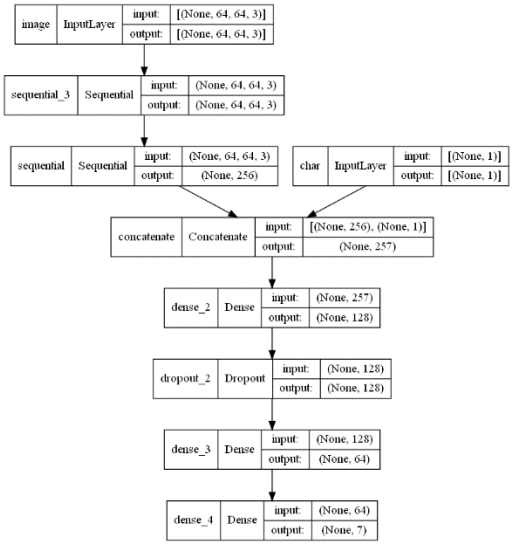
לדוגמה:

התו j בפונט Open Sans

אל מול

התו s בפונט Alex Brush

כאן לו היה נתון לי מראש שהאות היא S – הבחירה היתה קלה ללכת על פונט Alex Brush.



חמוש ברעיון זה ניגשתי להוסיף את המידע הזה לרשת.

מצאתי דוגמה לשימוש מעשי ביכולת ה Multi Input Neural Network במאמר הזה:

[A Multiple-Input Neural Network Model for Predicting Cotton Production Quantity: A Case Study]

אך לא מצאתי מידע רב לכך ברשת.

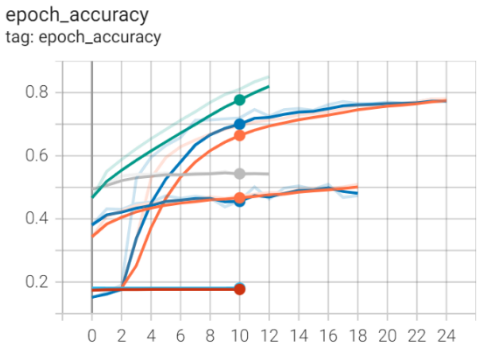
לאחר שלב זה המודל נראה כך ⇐

לאחר שהרצתי את המודל כך – ראיתי שנתון זה רק פגע בביצועי הדיוק והביא דיוקים נמוכים משמעותית ועל כן החלטתי לדבוק במודל הפשוט יותר.

שימוש בTransfer Learning

לאחר מכן חשבתי איך ניתן לשפר עוד את הוצאת הפיצ'רים מתמונת התו, וניסיתי להפעיל מספר רשתות מאומנות המסווגות בצורה טובה מאוד כמות עצומה של תמונות כך שיודעות למצוא את הפיצ'רים משמעותיים כמעט בכל תמונה באופן שהוכח כמעולה.

ביצעתי כמה הרצות בהן החלפתי את מודל הCNN הבסיסי בInceptionV3, ResNet50, VGG19 ועוד, תוך מחיקת השכבה האחרונה שלהם והחלפתה בשכבת הסיווג שלי ל7 הפונטים.



Name	Smoothed	Value	Step
Basic\transfer_learning\InceptionV3\20220117-230230\train	0.7769	0.8105	10
Basic\transfer_learning\InceptionV3\20220117-230230\validation	0.543	0.5388	10
Basic\transfer_learning\ResNet50\20220117-233510\train	0.1768	0.1768	10
Basic\transfer_learning\ResNet50\20220117-233510\validation	0.181	0.181	10
Basic\transfer_learning\VGG19\20220117-231320\train	0.4666	0.4729	10
Basic\transfer_learning\VGG19\20220117-231320\validation	0.4554	0.4573	10
Basic\two_dropout\20220117-205137\train	0.6644	0.6959	10
Basic\two_dropout\20220117-205137\validation	0.6999	0.7197	10

ניתן לראות שרוב האלגוריתמים בהם השתמשנו היו פחות טובים משמעותית מהמודל הבסיסי, מלבד הגרף שהשתמש בInceptionV3 שחטא בOverfitting שהשאיר כ50% דיוק לValidation שלא השתפרו עם הרצת עוד ועוד epochs.

ועל כן נראה היה שלבעיית הסיווג הספייציפית הזו – המודל הבסיסי הפשוט הביא תוצאות טובות יותר משימוש בTransfer Learning ולכן החלטתי לדבוק במודל זה.

נעזרתי במאמר זה:

[Font Recognition in Natural Images via Transfer Learning]

הכרעת הרוב

לאחר כלל השיפורים קיבעתי את המודל שלי, כך שייתן פרדיקציה באחוזים טובים על כל תו ותו באיזה פונט הוא נכתב.

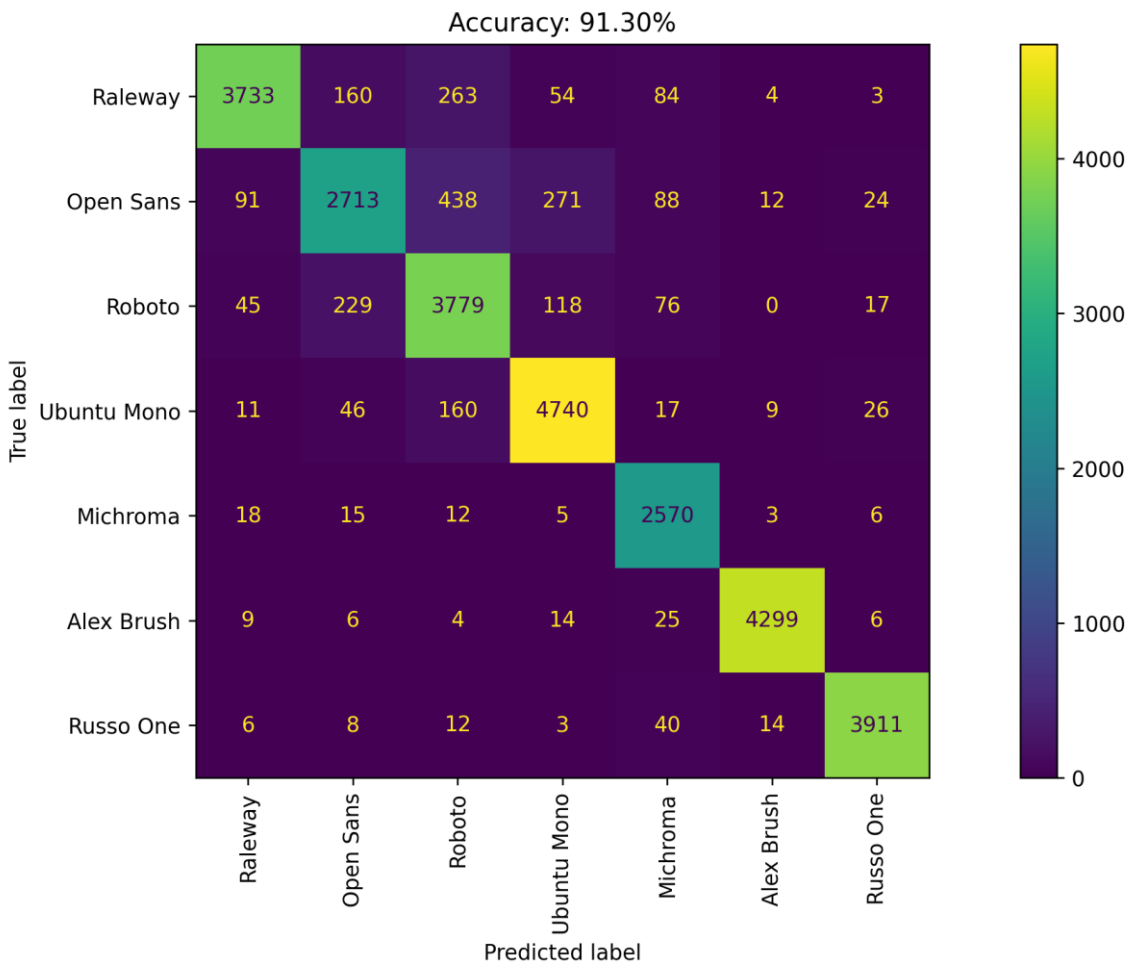
לאחר מכן החלטתי להשתמש בעובדה כי כל מילה שנכתבה על תמונה מסוימת, נכתבה באותו הפונט. ולכן אוכל להכריע ע"פ המודל שלי מהו הפונט שבו רוב תווי מילה מסוימת נכתבו – ואסווג בפונט זה את כל תווי המילה, גם אם המודל שלי הוציא סיווג מעט אחר.

כזכור, הרצת המודל הבסיסי הביאה ליכולת דיוק המתקרבות ל80% דיוק.

בכדי לבדוק את המודל שלי וסיווגו יחד עם הכרעת הרוב, כאשר דאגתי שהמודל לא יחטא באופן מובהק בOverfitting – הרצתי את קובץ הבדיקה שלי על גבי הDataset המתויג המקורי שקיבלנו, והשוותי את סיווגי המודל על מידע זה אל מול מידע האמת שקיבלנו.

אחוזי הדיוק לאחר שימוש בהכרעת הרוב הגיעו ל91.3%!

מצורפת מטריצת הבלבול (confusion matrix):



ניתן לראות כי חכמת המונים זו שיפרה בהחלט את ביצועי המסווג שלי.

[CNN - Train a network of images considering the image group classification]

[Multi-View Image Classification]

פלט המסווג

לבסוף, לאחר הכרעת הרוב לכל תו ותו מהו הפונט שלו – איגדתי את כלל המידע לכדי דו"ח מפורט המוציא את פלט המסווג.

בזכור לכל תמונת תו אני יודע את מספרה הסידורי בתמונה, את ערך התו שלה ובאיזו תמונת רקע היא נכתבה.

ע"פ דרישת הפרויקט הוצאתי מסמך המאגד את כלל הפרטים הנדרשים כולל הסיווג שהמסווג שלי ביצע (ע"פ הכרעת הרוב) וניתן לראותו בקובץ results.csv או להשתמש במסווג ולייצר קובץ זה מחדש.

תודה רבה!

שובל כהן

208748152