**פרויקט- מבוא לראייה ממוחשבת**

**קודם כל רקע להרצת הפרויקט-**

**יש להוריד המודל והמשקולות מהקישור להלן(final\_model.json, (final\_model.h5:**

**final\_model.h5-**

<https://drive.google.com/file/d/1-2XLtBSAZwQmzJh9oaAJA0TxI7PU1Hm-/view?usp=sharing>

**final\_model.json-**

<https://drive.google.com/file/d/1pD_YyBVhc1W09XoMrn3qNhYgsvd9ZIcX/view?usp=sharing>

**לשים את שני קבצים אלו, וכן בנוסף את הקובץ של סט האימון שקיבלנו SynthText.h5 וכן סט הטסט שקיבלנו SynthText\_test.h5 בתיקייה TEST.**

**(לעדכן את projectTest.py בשמות הקבצים אם צריך)**

**הקבצים שצריכים להיות בתיקייה TEST בסופו של דבר:**

* **final\_model.h5(המודל שלי)**
* **final\_model.json(המודל שלי)**
* **SynthText.h5(שקיבלנו)**
* **SynthText\_test.h5(שקיבלנו)**
* **projectTest.py(הקוד טסט שלי)**

**ואז אפשר להריץ את הקובץ projectTest.py והוא יוציא את קובץ החיזוי ל- predictionsFonts.csv.**

לפרויקט הוספתי כמה קבצים חוץ מהקבצים הכתובים למעלה:

1. Epoch\_60.txt"" – כל הפלטים של הEPOCHים עד לסיום המודל המוגמר.
2. "README.md" – הסברים איך להריץ ועוד...
3. "final\_modelCode.py" – קובץ קוד של האימון ליצירת המודל הסופי.
4. "projectTest.py" – קובץ הטסט להרצה ליצירת קובץ אקסל סופי.
5. "predictionsFonts.csv" – קובץ אקסל לפי קובץ של הילה-החיזוי של הפונטים של הטסט.

**בברכה,**

**נופת**

מבוא לראייה ממוחשבת- הפרויקט:

בפרויקט נדרשנו לזהות 7 גופנים שונים לאותיות באנגלית וסימנים ע"י יצירת מערכת ללמידה עמוקה.

בניתי מודל שלי וכך הכרתי הרבה יותר טוב את שכבות המודל(לעומת מודל מוכן או מוגמר) עקב זה שהמחשב שלי לא חזק כל כך ואין לי כרטיס מסך חזק בחרתי לעבוד בGoogle Colab . (כמובן שבו יש בעיות שלמזלי קראתי לפני כן שיש מגבלת זמן על GPU וכן הRAM כל פעם צריך להסתכל שלא יגמר וכו.. כי אז צריך להריץ הכול מחדש).

בתחילת הפרויקט היה צורך לעשות היפוך עבור כל תמונה לתמונות האותיות עבורה, כדי לעשות זאת היה צורך בשימוש בפונקציות getPerspectiveTransform ו- warpPerspective של ספריית OpenCV כאשר הם עזרו לנו בכך שהשינוי בפרספקטיבה קשור לשינוי בנקודת המבט. הוא משמר קולינאריות ושכיחות. זה אומר שהקווים הישרים יישארו ישר גם לאחר השינוי. אנו עושים שימוש גם בפונקציה בשם warpPerspective() כדי להתאים לגודל של התמונה המתקבלת באמצעות הפונקציה getPerspectiveTransform()‎ לגודל התמונה המקורית הפונקציה warpPerspective() מחזירה לנו תמונה שגודלם זהה כגודל התמונה המקורית.

עבור כל תמונת של אות היה צורך לבחור בגובה ורוחב מכיוון שרוב האותיות חלק גבוהות יותר חלקן פחות היה צורך לחשוב גם על האותיות המאורכות לכן בחרתי ביחס של 3/4.

לאחר עבודה עם גובה ורוחב ראיתי שהרבה פעמים עם גבהים ורוחביים גדולים כגון 60 40 וכו'... הפרויקט קרס כי גם לגוגל קולאב יש איזשהו RAM שהוא מאפשר לי , ולכן בסופו של דבר בחרתי גובה ורוחב של 40 על 30.

בעיות שהיו לי בתמונות בסט האימון והטסט שהתפלגות 7 הגופנים לא אחידה כלל (נגיד גופן Michroma היית בכמות קטנה מאוד יחסית לשאר הגפנים).

היו כמה דרכים שעלו לי לראש לפני כתיבת המודל-

1. יצירת מודל אחד לכל גופן כלומר יצירת מודל ל- 7 מחלקות.
2. יצירת מודל לכל תו מודל עצמי שילמד לאיזה גופן התו שייך.

במודל השני (לכל תו) חסרונו יהיה הרבה פחות 'אוכלוסייה' יתרונו בכך שמחלקות לא התבלבלו מתווים אחרים של הגופנים השונים. מודל זה לא הוכיח את עצמו כיוון שיש המון תווים, אז הרבה מן המחלקות לא היה מספיק מדגם אוכלוסייה כדי ללמוד את הבדלי התו בגופן אחד מאותו תו בגופן אחר.

במודל הראשון יתרונו שיש יותר 'אוכלוסייה' לכל מחלקה כיוון שכל התווים מתחלקים ל7 מחלקות רק. לאחר ניסוי וטעיה ע"י מודל פשוט 2 שכבות בלבד conv2d של כמות קטנה שלפילטרים לקבלת תוצאה איזה מידול עדיף. במודל הראשון המופשט יותר הגעתי לאחוזי דיוק 0.95 טיפה יותר גבוהה ממודל השני כ0.2 הפרש ביניהם ולכן העדפתי ליצור עפ"י מודל 1.

בשלב הראשון של בניית המודל כמובן שהסתכלתי בהרצאות וכן עליהם ניסיתי להוסיף שכבות ומסננים.

ראיתי שניתן היה להוסיף תמונות מהקישור שנשלח אך עקב בעיות משום מה לא הצלחתי ולאחר יום כבר החלטתי להרחיב תמונות בצורה שונה. קראתי מאמר בנושא, ע"י סיבוב נתונים רנדומלי לתמונה בין 2 עד -2 מעלות ע"י פונקציית ()ndimage.rotate של ספריית scipy רוב האותיות יחסית היו מיושרות אך היו קצת אותיות מסובבות לאחד מהצדדים ולכן עשיתי סיבוב הנתונים רנדומלי שהרחיב לי בצורה יפה את הנתונים. לא ניתן כמעט לראות שינוי במעלות כיון שזה הבדל של עד 2 מעלות אך זה מאוד שיפר לי את המודל.

כדי לנתב את הנתונים ראיתי שאין יתרון בתוצאות עבור תמונות בRGB ולכן השתמשתי בגווני אפור, מרכזתי את הנתונים ע"י טווח בין 0-1 וחלוקה ב-255.

חישבתי את התמונה הממוצעת בהתחלה עבור כל תמונה וראיתי שלכולם יש פחות או יותר אותו ממוצע 0.461 ולכן פשוט חיסרתי מכל התמונות את הממוצע הכללי 0.461.

לאחר מספר רב של ניסוי וטעיה ושינויים במודל וניסיון של הרבה מודלים שונים עמוק יותר שכבות conv2d ושכבות של נורמליזציה וdropout שונים הגעתי למודל דיי מופשט ונוח שנתן תוצאות טובות מאוד, קרוב ל96% הצלחה.

יש שכבה MaxPooling2D שנתנה לי תוצאה גרועה לא השתמשתי בה. רציתי לראות מודל לא שלי כדי לראות אם התוצאות שלו יהיו טובות משלי.

חיפשתי וחיפשתי עד שראיתי בGithub מודל ארוך וטיפה מסובך (יותר עמוק בהרבה מהמודל שלי) אז השתמשתי בו אבל שיניתי בו מלא דברים והורדתי לו מלא דברים הקטנתי שכבות וגודל תמונה ופילטרים כדי שיוכל לרוץ לי בגוגל קולאב.

הקישור לגיטהב:

https://github.com/robinreni96/Font\_Recognition-DeepFont/blob/master/Font\_Rec(DeepFont).ipynb

התוצאות הגיעו ונראה כי אחוזי הדיוק בין המודל שלי לבין המודל בגיטהב כמעט אותו אחוזי דיוק אפילו שלי טיפה יותר.

לאחר שינויים רבים והתאמות החלטתי כדי שיהיה לי יותר מעניין לאחד בין שני המודלים ובסופו של דבר ליצר מהן מודל סופי יחדיו.

בשלב הסופי של המודל המשכתי לאמן עםlearning rate יותר קטן עד למיטוב סופי להחלטתי עפ"י הנתונים של תוצאות המודל.

תמונת המודל הסופי שלי(ראה בסוף הדו"ח- ערך נספח 1).

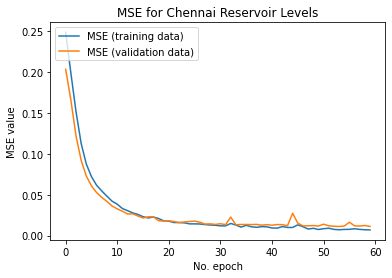
עקב זה שנזכרתי מאוחר לא עשיתי משהו שרציתי לעשות, משהו יחסית פשוט של בדיקת פונט לכל מילה ושינוי לפונט לפי הכרעת הרוב.

ערכי הEPOCHים האחרונים:

Epoch 60/60

**705/705 [==============================] - 71s 101ms/step - loss: 0.0069 - accuracy: 0.9710 - val\_loss: 0.0112 - val\_accuracy: 0.9512**

|  |  |  |
| --- | --- | --- |
| Accuracy | Loss |  |
| 0.9710 | 0.0069 | Train |
| 0.9512 | 0.0112 | Validation |

**תמונת MSE (טעות ריבועית ממוצעת-LOSS) לפי ולידשיין ואימון עבור 60 הEPOCHים:**

לאחר מכן רציתי לבדוק באמצעות הפונקציה- evaluate() אשר מיועדת להערכת המודל שכבר עבר הכשרה באמצעות נתוני האימות (או הבדיקה) והתוויות המתאימות. מחזירה את ערכי ההפסד והערכים עבור המודל. וזוהי התוצאה שיצאה לי:

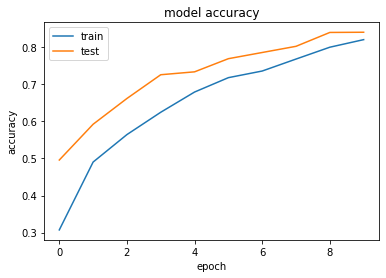
[0.012241094373166561, 0.9464539289474487]

['loss', 'accuracy']

**גרפי accuracy+ loss עבור תחילת אימון המודל (10 הEPOCHים הראשונים)-**

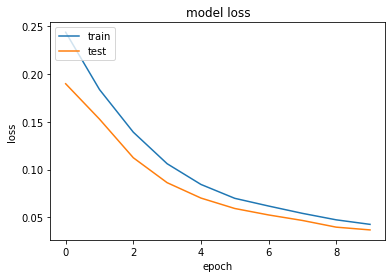
**גרף accuracy תחילת אימון המודל (10 הEPOCHים הראשונים) והדיוק:**

**רואים איך יש לו לאן לעלות והוא עולה יפה מאוד...**

****

**גרף Loss תחילת אימון המודל (10 הEPOCHים הראשונים) והדיוק:**

**רואים איך יש לו לאן לרדת ויורד בהתחלה ירידה דרסטית...**

****

בשלב הסופי של הפרויקט במודל שלי קיימים בסט של הטסט:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Russo One** | **Alex Brush** | **Michroma** | **Ubuntu Mono** | **Roboto** | **Open Sans** | **Raleway** |
| **8040** | **8159** | **5108** | **10930** | **8101** | **7376** | **8451** |

בסה"כ יש לנו 56,165 שורות של אותיות ותווים.

הפרויקט היה מעניין ומאתגר מאוד, למדתי המון על ספריות שונות בפיתון(בייחוד ספריית KERAS), למדתי המון על למידה עמוקה. למדתי כמה בפיתון הספריות שלה מועילות מאוד ומקצרות תהליכים רבים אם זה באיחוד מודלים ואם זה ביצירת גרפים יפים ויזואליים.

עמוד הבא- המודל שלי ויזואלי->

תמונה שמכילה טקסט, קבלה

התיאור נוצר באופן אוטומטינספח 1- המודל הסופי שלי ויזואלית: