שם בית הספר: תיכון ליאו באק

שם הפרויקט: זיהוי טקסט מתמונה

שם המגיש: אורי צמח

ת.ז: 327851010

שם המנחה: אולגה בוטמן

שם החלופה: למידת מכונה

תאריך הגשה: 9.4.23

תוכן העניינים :

מבוא 3
הרקע לפרויקט 3
תהליך המחקר 4
סקירת המצב הקיים בשוק 4
החידושים בפרויקט 4
שימוש במקורות המידע והתאמתם לפרויקט 5
טכנולוגיות מחוץ לתוכנית הלימודים 5
אתגרים מרכזיים 6
בעיות צפויות לפרויקט 6
שימושי הפרויקט 6
הצגת פתרונות לבעיה 7
מבנה וארכיטקטורת הפרויקט 8
איסוף הכנה וניתוח הנתונים 8
מאגר הנתונים 8
הכנת הנתונים לאימון 9
שלב בנייה ואימון המודל 10
המודל שבחרתי 13
השכבות במודל 15
פונקצית השגיאה 17
תוצאות האימון 18
שיפור הצלחת המודל ו Hyper Parameters 19
ייעול ההתכנסות 20
התמודדות עם Overfit 20
שלב היישום 22
מדריך למפתח 25
מדריך למשתמש 26
התקנה 26
השימוש באפליקציה 27
Screen Flow Diagram 31
רפלקציה 32
ביבליוגרפיה 33

מבוא

הרקע לפרויקט

במסגרת מגמת הנדסת תוכנה בבית הספר התבקשנו לבנות פרויקט על מנת להדגים את הידע שרכשנו במהלך הלמידה של תחום הלמידת מכונה, ובפרט למידה עמוקה.

בחרתי לבצע פרויקט הממ‌מש זיהוי טקסט אנגלי שנכתב על גבי דף חלק. פרויקט זה דרש ממני להרחיב את הידע שלי אל מעבר לתכנית הלימודים, וזו אחת מהסיבה שבחרתי בו.
הרעיון הראשוני היה לבצע את זיהוי הטקסט על כתב יד בעברית, אך נאלצתי לעבור לשפה האנגלית בשל אי זמינות של מאגרי מידע בעברית, וכן יצירת מאגר נתונים כזה כרוך בעבודה סיזיפית רבה.
לאחר מחשבה נוספת, הצלחתי למצוא מאגר נתונים המכיל אותיות בעברית. חשבתי שאוכל לחלק את המילה לאותיות המרכיבות אותה, ולאחר מכן לזהות כל אות בנפרד. הבעיה היא שהכתב שלנו, לעיתים קרובות, מחובר או צפוף, כך שבעיית ההפרדה לאותיות נעשית בלתי אפשרית. לאחר מחקר, ראיתי שהדרך המקובלת לזיהוי טקסט היא ביצוע הפרדה למילים וזיהוי כל מילה בבת אחת. הידע שהיה דרוש ליישום שיטה זו הוא מחוץ לתוכנית הלימודים, אך למרות זאת קיבלתי על עצמי את האתגר ועברתי למאגר נתונים בשפה האנגלית בכדי לממש את השיטה המדוברת.

בחרתי בפרויקט זה בעיקר כי לדעתי יש לו שימוש נרחב בחיי היום יום שלנו. ישנם מצבים רבים בהם אנו נתקלים בצורך להעביר מסמך הכתוב על גבי דף לצורה דיגיטלית - למשל, על מנת לשמור אותו, לשלוח, להימנע מכתיבה מייגעת ועוד. בנוסף, הכלים שלמדנו במהלך השנה היוו את הבסיס לפרויקט ולכן בחרתי בפרויקט זה גם בכדי שאוכל להמשיך לחקור ולהעמיק בחומר.

עבורי, מטרות הפרויקט הן להעשיר את הידע שלי בתחום מדעי המחשב וללמוד כיצד טכנולוגיות מתקדמות, הרלוונטיות בזמנינו, פועלות ; ללמוד להתמודד עם אתגרים הכרוכים בלמידה עצמית - במהלך הפרויקט אאלץ להתמודד עם מגוון שגיאות ובעיות, וכן להרחיב את הידע שלי על מנת לפתור אותן ; ליישם את הידע התיאורטי - במהלך שנות הלימוד במגמה, רכשנו ידע תיאורטי מקדים במתמטיקה ובלמידת מכונה. בפרויקט זה יישמנו את הידע שרכשנו ובחנו אותו בפועל.

קהל היעד של הפרויקט הוא כלל האוכלוסייה. יישום הפרויקט יהיה פשוט לתפעול וכולם (דוברי אנגלית) יהיו מסוגלים להשתמש בו. המגבלה היחידה היא גודלה של האפליקציה שצפוי להיות גדול בשל אוסף ספריות בהן אצטרך להשתמש. זה מה שאוכל להספיק במסגרת זמן הפרויקט, אך כמובן ניתן לקחת זאת כאתגר לפרויקט המשך עתידי.

כפי שכבר ציינתי, בשביל להדגים את מימוש הפרויקט אכין אפליקציית Android בה יהיה ניתן לצלם בעזרת המצלמה תמונה של מסמך טקסט. בעזרת המודל המאומן, האפליקציה תחזה את הטקסט שמופיע במסמך ותציג אותו למסך. אם אוכל, אוסיף אפשרות לעצב את התמונה (לחתוך, לסובב).

תהליך המחקר

סקירת המצב הקיים בשוק

כיום, זיהוי טקסט מתמונות, המכונה גם זיהוי תווים אופטי (OCR), הוא שוק צומח בשל הצורך הגובר בדיגיטליזציה של מסמכים ותמונות מבוססי נייר. השוק לזיהוי טקסט מתמונות מונע על ידי גורמים כמו הצורך בניהול יעלי של מסמכים והביקוש הגובר לאוטומציה בתעשיות שונות כמו בנקאות, בריאות ולוגיסטיקה.

נכון לעכשיו, השוק נשלט על ידי חברות גדולות כמו Google LLC, Adobe Inc., Microsoft, ABBYY ו-IBM Corporation. חברות אלו מציעות מגוון רחב של מוצרים ושירותים לזיהוי טקסט, הנותנים מענה לתעשיות ולצרכים עסקיים שונים.
ראוי לציין כי האינטרנט מוצף באפליקציות ואתרים המאפשרים לבצע OCR, למשל שירות " Google Lens" ואתרים נוספים התומכים במגוון שפות.

לפי [1], השוק צפוי לצמוח משמעותית בשנים הקרובות עקב הביקוש הגובר לאוטומציה בתעשיות שונות. גם האימוץ של פתרונות OCR מבוססי ענן צפוי לעלות בשל היתרונות של עלות-תועלת והשיפור בדיוק הטכנולוגיה.

עם זאת, רוב מציעי השירות הגדולים, דורשים תשלום עבור זיהוי מסמכים ותמונות, ולכן טכנולוגיות אלה לא זמינות בנוחיות לקהל הרחב, שרובו ירצה לבצע שימוש נקודתי בשירות, ועל כן לא ישלם על שירות זה. לכן, החברות הגדולות פונות לעסקים בהם ישנו צורך לבצע אוטומציה ודיגיטליזציה לכמות גדולה של מסמכים.

בסך הכל, שוק זיהוי הטקסט מתמונות צפוי להמשיך לצמוח עקב הצורך הגובר בדיגיטציה ואוטומציה בתעשיות שונות, וזמינותן של טכנולוגיות OCR מתקדמות.

החידושים בפרויקט

כפי שכבר ראינו, קיימים שירותים רבים המציעים פתרונות לבעיית ה OCR. למרות זאת, לפרוייקט שלי ישנם כמה יתרונות חדשניים.
ראשית כל, השימוש באפליקצייה המממשת את הפרויקט שלי יהיה חינמי. השימוש בשירותים האיכותיים אותם מציעות החברות הגדולות אינו חינמי, ולכן הפרויקט שלי יכול להיות פתוח וזמין יותר לקהל הרחב.
בנוסף, האפליקציה המיישמת את הפרויקט נותנת אפשרות לעיצוב בסיסי של התמונה לפני שליחתה לתהליך החיזוי. זוהי תכונה שיכולה לעזור לשפר את דיוק המודל ואת זמן החיזוי - דבר שלא קיים בכל שירותי ה .OCR
בנוסף, אני מקווה שאוכל לחזות מסמכים בדיוק טוב יותר מחלק מאתרי האינטרנט השונים שניסיתי.

כמו כן, הקוד שמממש פרויקטים רבים של OCR הינו מסובך וקשה להבנה. אני סבור שהקוד שלי יהיה קריא יותר ופשוט יותר, ובכך יהיה זמין ללמידה של OCR עבור מתחילים.

שימוש במקורות המידע והתאמתם לפרויקט

במהלך העבודה על הפרויקט נעזרתי במגוון מקורות מידע שהסבירו כיצד לממש את טעינת הנתונים, בניית המודל ובדיקתו (כל המקורות הרלוונטיים מופעים בחלק הביבליוגרפיה).
בכל פעם שהשתמשתי בקוד מהאינטרנט וידאתי שאני מבין אותו עד הסוף וניסיתי לפשט אותו עד כמה שאפשר. ברוב המקרים, החיפוש באינטרנט עזר לי למצוא את הפונקציה שהייתי צריך מספרייה מסוימת. כך יכולתי לראות דוגמא בה משתמשים בפונקציה ויכולתי לשלב אותה בתוך הקוד שלי.
היו מקרים בהם השתמשתי בקוד מהאינטרנט במלואו מכיוון שהוא היה מוכן בדיוק בשביל השימוש שלי. לכן וידאתי שאני מבין אותו ולאחר מכן השתמשתי בו.

טכנולוגיות מחוץ לתוכנית הלימודים

בכדי להשיג תוצאות ששווה לדבר עליהן נדרשתי ללמוד טכנולוגיות מחוץ לתכנית הלימודים. בפרויקט זה מימשתי פתרון נפוץ לבעיית ה - OCR והוא נקרא ארכיטקטורת CRNN. הרעיון הוא שהתמונה, המכילה מילה, עוברת שכבות CNN בהם אנו מוצאים את התכונות החשובות לסיווג ; לאחר מכן, התמונה עוברת דרך רשת (Long Short Term Memory) LSTM, המתמחה בסיווג מידע סדרתי, וחוזה את האות בכל מקום בתמונה. לבסוף, המידע מפוענח על ידי CTC decoder, כאשר פונקצית העלות היא CTC loss function.

אפרט בהרחבה על כל מושג בשלב הצגת המודל, הנמצא בחלק "מבנה וארכיטקטורת הפרויקט".

אתגרים מרכזיים

בעיות צפויות לפרויקט

בעת העבודה על הפרויקט צפויות בעיות בעת כתיבת הקוד ואולי אפילו בעת ניתוח התוצאות.
ראשית כל, טעינת הנתונים לתוך הקוד תהיה מאתגרת, בעיקר בשל העובדה שה labels של כל תמונה נמצאים בקבצי xml בצורה לא נוחה. אני אצטרך למצוא דרך לחלץ את המידע הרלוונטי מקבצי ה xml אל תוך הקוד.
שנית, יתכן שבעת בדיקת תוצאות האימון של המודל יווצר מצב של overfit שידרוש טיפול לא טריוויאלי. יתכן וזה יהיה הקושי המרכזי בעת הכנת הפרויקט.
כמו כן, בשביל ליישם את המודל שאבנה, אני מתכנן לבנות אפליקצית android בה ניתן יהיה להעלות תמונות ולחזות את הטקסט המופיע בהן. נכון לעכשיו, אין לי ידע בתחום חוץ מידע תכנות בשפת java. אני סבור כי הכנת האפליקציה תהיה מאתגרת.

שימושי הפרויקט

פרויקט זה הוא שימושי ביותר, שכן הוא נותן מענה לבעיות של דיגיטליזציה. כלומר, הפרויקט מאפשר אוטומציה של תהליך העברת מסמכים כתובים למסמכים מוקלדים על גבי המחשב.
בנוסף, הפרויקט מאפשר זיהוי של טקסט ולכן אפשר לבחון את הצלחתו בזיהוי טקסט המשמש לצורך ניתוח הסביבה במערכות מתקדמות.
כמובן, הפרויקט מיועד עבור שימוש אישי ולכן יכול להיות שימושי גם עבור המשתמש הפרטי.

הצגת פתרונות לבעיה

בחלק זה אתרכז בהצגת הפתרונות החינמיים העמודים לרשותינו ברשת על מנת ביצוע OCR.
הפתרון הראשון שבדקתי הוא שירות Google Lens. זוהי אפליקציה מבית Google המתעסקת ב OCR.
בעזרת האפליקציה ניתן לתרגם מסמכים, לזהות טקסט מתמונה ועוד. האפלקיציה היא חינמית ועובדת מצוין, לפי הבדיקות שערכתי.

בנוסף לאפליקציית Lens, ישנם אתרים רבים באינטרנט הטוענים לבצע OCR.מבדיקת האופציות הראשונות שעלו בחיפוש הגוגל "Online OCR", נוכחתי לדעת כי פתרונות אלה הם בדרך כלל לא מדוייקים, ומוציאים פלט שרחוק ממה שנכתב במסמך.

כמובן שישנם שירותי OCR שהם לא חינמיים ומיועדים לפריסה רחבה יותר בחברות גדולות. בין שירותים אלה נכללים Adobe Acrobat ו - ABBYY - FineReader. יש לציין, כי חברות אלה מוכרות חבילות גם למשתמשים פרטיים במחיר זול יותר.

מבנה וארכיטקטורת הפרויקט

איסוף הכנה וניתוח הנתונים

מאגר הנתונים

מאגר הנתונים [4] ששימש אותי לפרויקט זה הוא ה - IAM database. זהו מאגר נתונים פופולרי בתחום ה OCR. המאגר פורסם על ידי חברת ICDAR בשנת 1999 והוא פתוח בחינם לקהל הרחב.
המאגר מכיל 115,320 תמונות בשחור לבן של מילים באנגלית המורכבות מ 79 תווים (52 אותיות גדולות וקטנות ו 27 סימני פיסוק וספרות). המילים נאספו על ידי 657 כותבים שונים.
זו דוגמה לכ - 64 מילים מהמאגר :

הטקסט המופיע בכל תמונה מאורגן בקבצי xml נפרדים.

הכנת הנתונים לאימון

בכדי להכין את הנתונים לאימון יש צורך לבצע קידוד ל label של כל תמונה ויש לנרמל את התמונה לערכים בין 0 ל 1.
בקוד ישנה רשימה בשם vocab ששומרת את כל סוגי התווים אותם המודל יודע לסווג. קידוד ה labels נעשה על ידי הענקת וקטור, שגודלו כאורך המילה המקסימלי, לכל תמונה. בכל תא בוקטור אנו שמים את האינדקס של האות הנוכחית ברשימה vocab. את התאים הנותרים אנו ממלאים באינדקס של התו ε (הסבר על התו ε נמצא בחלק ההסבר על פונקצית השגיאה).

לדוגמה, האינדקסים של האותיות a, b ברשימה vocab הינם 54, 53 בהתאמה. לכן הקידוד של המילה 'aab' יהיה הוקטור:

$$\underline{y_{true}} = [53, 53, 54, 79, 79, \dots, 79]$$

כאשר אורך הוקטור y_{true} הוא 21 והאינדקס של התו ε הוא 79.

בנוסף, עלינו לנרמל את התמונות לצורך שיפור מהירות ויציבות האימון. אנו עושים זאת בצורה פשוטה, על ידי שינוי טווח הפיקסלים מ [255, 0] לטווח [1, 0], כלומר חלוקת כל ערך פיקסל ב 255.

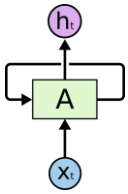
כמו כן, אנו מבצעים data augmentation לנתונים על ידי שתי דרכים עיקריות: סיבוב אקראי של התמונה בין ערך של -15 ל 15 מעלות ושינוי אקראי של בהירות התמונה מפקטור של 0.5 ל 1.5.

שלב בנייה ואימון המודל

מודל ה - *LSTM*

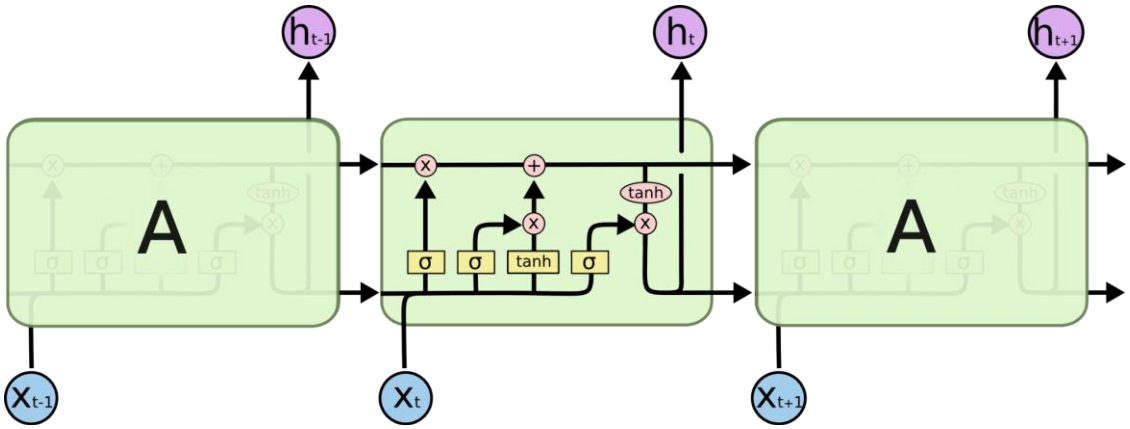
מודל ה – LSTM הניו שכבה חלק מרכזי במודל שבניתי. לכן, נתחיל מלהבין את אופן פעולתו.
מודל ה LSTMהינו אופטימיזציה של מודל ה - RNN - Recurrent Neural Network. נתחיל מלהבין את המודל הזה. (בהסבר אתבסס על [2]).

בבעיות סיווג מסוימות יש צורך שהמודל יתבסס בחיזוי על מידע קודם. בבעיה שלנו (OCR), המודל צריך להתבסס על חלקים קודמים של המילה בעת סיווג החלקים הבאים, כאשר כל חלק יכול להיות חתך של אות מסויימת. בשביל לפתור את הבעיה הזו, לאחר כל חיזוי נכניס את תוצאות המודל בחיזוי הקודם כקלט נוסף לחיזוי הבא, כמתואר בתרשים :

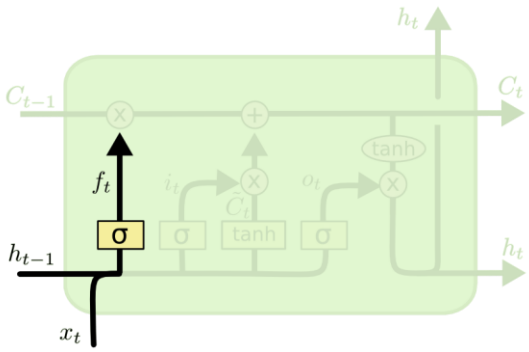


כאשר h_t הפלט בנקודת זמן t (נקרא גם time-step) ו - x_t הקלט בנקודת זמן t.

הבעיה בשכבות ה RNN מתגלה כאשר ישנה תלות ארוכת טווח בין הפלטים. שכבת ה LSTM פותרת בעיה זו. זוהי הדיאגרמה המתארת בלוק LSTM באיטרציה t :

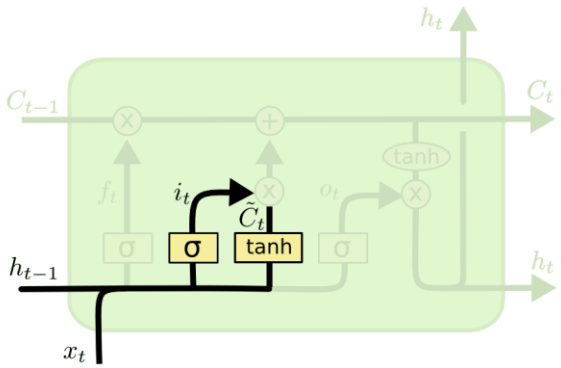


כעת, נעבור שלב שלב ונבין כיצד הבלוק הנ״ל בנוי :



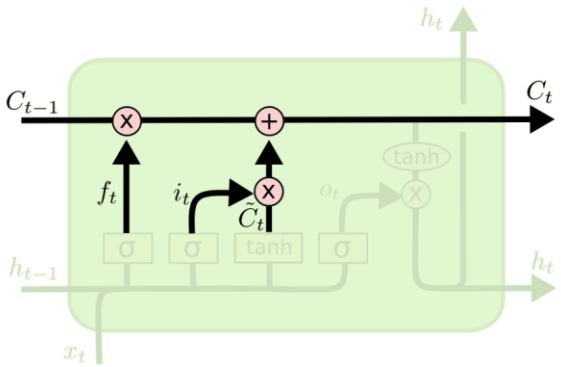
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

בשלב הראשון המידע החדש, בשילוב עם תוצאת האיטרציה הקודמת, עובר דרך בלוק ה - σ המודגש. בלוק זה מכיל שכבה של רשת נוירונים פשוטה שמטרתה להחליט מהי האינפורמציה החשובה ואיזו אינפורמציה צריך לשכוח. בלוק זה נקרא "שער השכחה" (forget gate). ה'שכחה' מתבצעת באמצעות כפל - אם תוצאת הרשת היא 1 אז האינפורמציה נשמרת במלואה ואם היא 0 אז היא נשכחת לגמרי.



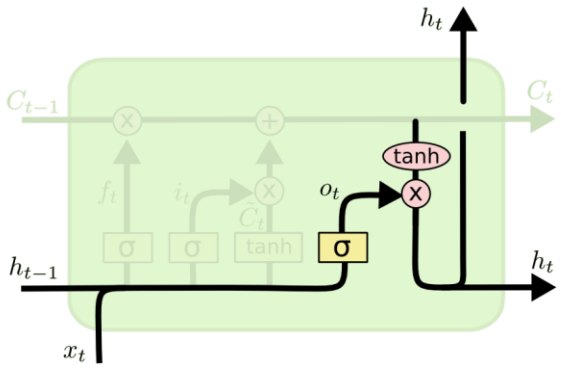
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

כעת נרצה לעדכן את האינפורמציה הנוכחית בתא C_{t-1}.
לשם כך, המידע עובר דרך שכבת נוירונים עם אקטיבצית sigmoid שמטרתה להחליטה איזה מידע אנחנו צריכים לעדכן (כלומר בהינתן הקלט $[h_{t-1}, x_t]$, הרשת צריכה להחליט איזה כניסות בוקטור C_{t-1} צריך לשנות). כמו כן אנחנו מעבירים את המידע דרך רשת עם אקטיבצית tanh שיוצרת מועמד חדש שמטרתו לעדכן את C_{t-1} - נסמנו $\tilde{C}_t$.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

בשלב הבא נשלב את תוצאות השלבים הקודמים על מנת לעדכן את האינפורמציה ב C_{t-1}.
'נשכח' את המידע הלא רלוונטי על ידי הביטוי $f_t * C_{t-1}$ ונבצע את עדכון המידע לקראת האיטרציה הבאה על ידי $i_t * \widetilde{C}_t$.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

לבסוף, נרצה להוציא את החיזוי שלנו עבור האיטרציה t. נרצה להוציא גרסה כלשהי של C_t . קודם נעביר את C_t דרך tanh בכדי לנרמל את הערכים בין -1 ל 1 ונכפול בתוצאה של ה sigmoid, על מנת להוציא כפלט רק את החלקים הרלוונטיים לחיזוי הנוכחי. בלוק ה o_t אחראי על תוצאת ה sigmoid ולמעשה מחשב על סמך הקלט ותוצאת האיטרציה הקודמת, איזה מידע צריך לעבור שינוי ב C_t לפני שנוכל להחזיר את התוצאה.

המודל שבחרתי

בחירה נכונה של המודל היא הכרחית להשגת תוצאות טובות ודיוק גבוהה.
במהלך כתבית קוד הפרויקט התבססתי על פרויקט git של pythonlessons הנמצא ב [5]. בפרויקט זה נעשה שימוש באותו מאגר נתונים ולכן קיוויתי שאוכל להשיג תוצאות דומות או טובות יותר אם אשתמש בפרויקט זה כנקודת התחלה.
המודל שבחרתי לצורך זיהוי המילים הוא מודל CRNN פשוט המתבסס על הרשת [6] ResNet.
כאשר אנו מוסיפים שכבות רבות למודל CNN פשוט אנו נתקלים בבעיית ה Vanishing Gradients.
למעשה בתהליך ה Backward Propagation, ככל שיש יותר שכבות, כך בחישוב הגרדיאנטים מופיעות המשקולות בחזקה גבוהה יותר. כאשר המשקולות קטנות, זה עלול לגרום לגרדיאנטים להיות קטנים מאוד ולבסוף אפס. מטרתה של רשת ResNet היא למתן בעיה זו ולאפשר רשת עם מספר גדול של שכבות.
הרשת מורכבת למעשה מאוסף של Residual Blocks כאשר כל בלוק נראה כך :

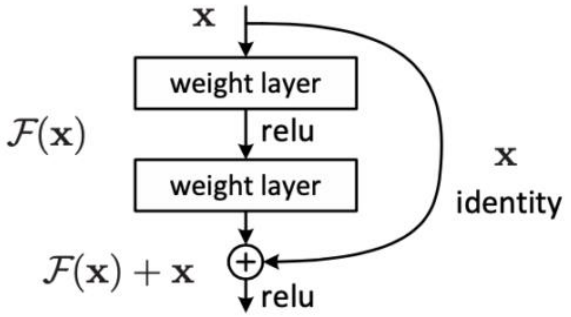


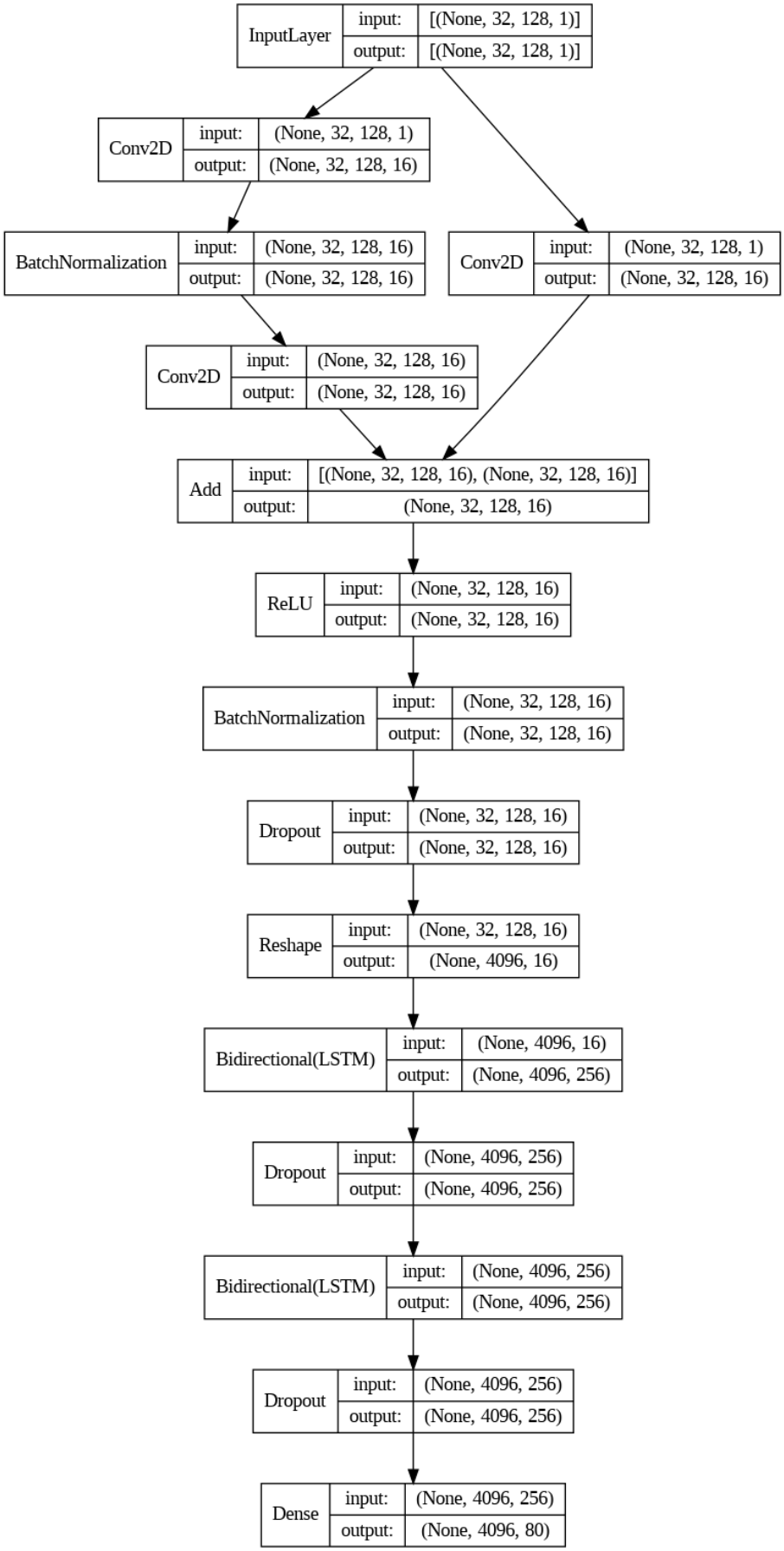
Figure 2. Residual learning: a building block.

כפי שניתן לראות, כל בלוק מורכב משתי שכבות קונבולוציה (CNN), חיבור של תוצאת הקונבולוציה, $F(x)$, עם הקלט, x , הפעלת אקטיבציה, כאשר באמצע ניתן להוסיף שכבת Batch Normalization.
בנוסף, הוספתי שכבת Dropout לאחר כל בלוק על מנת לצמצם אף יותר את בעיית ה Overfit.

שיטה זו, המשתמשת ב Residual Blocks, עוזרת משמעותית לצמצום בעיית ה Vanishing Gradients וכן מאפשרת לאינפורמציה לעבור לשכבות עמוקות יותר בקלות.
במודל שלי בחרתי להשתמש ב 12 בלוקים כאלה מה שאומר 24 שכבות CNN ובנוסף עוד 4 שכבות CNN נוספות הדואגות להתאמת מימדים. יכולתי להשתמש ביותר שכבות, אך מכיוון שאימון המודל התבצע על מחשבי האישי וארך זמן רב גם ככה, החלטתי שלא להוסיף עוד שכבות.

לאחר מודל ה ResNet, נעשה שימוש בשתי שכבות של LSTM, כאשר אחרי כל שכבה הוספתי שכבת Dropout נוספת.
לבסוף, שכבת ה LSTM השנייה מחוברת לשכבת Dense רגילה עם 80 נוירונים (79 עבור כל התווים אותם צריך לסווג ועוד מקום אחד עבור התו הריק, ε) עם אקטיבצית softmax.

זהו גרף המתאר את המודל עם residual block יחיד :



השכבות במודל

Dense

שכבת ה Dense היא שכבת נוירונים פשוטה המתאימה N נוירונים לטנסור הקלט. שכבה זו מחברת בין כל איבר בטנסור לכל N הנוירונים בעזרת משקלים ופונקצית אקטיבציה.

Conv2D

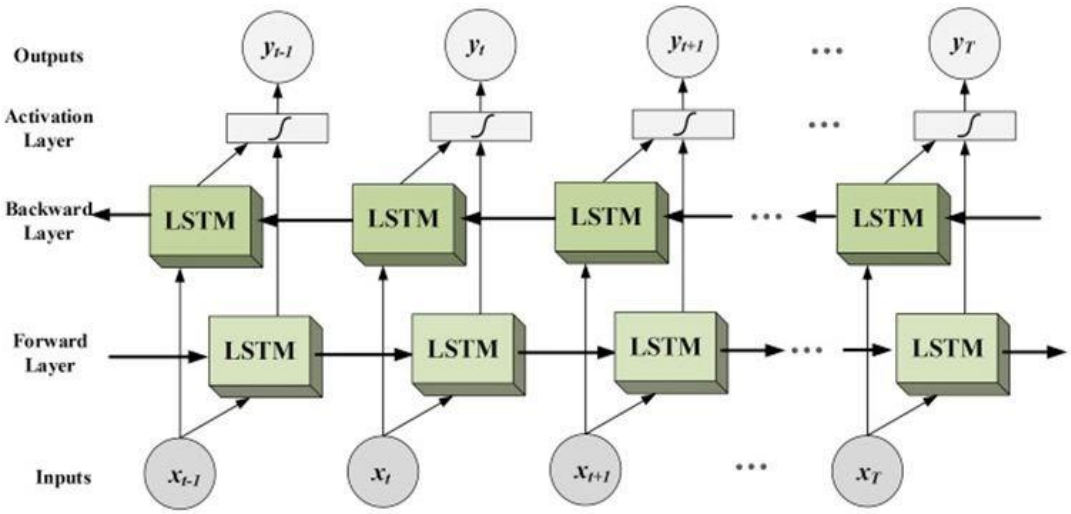
שכבת הקונבולוציה היא אבן היסוד של המודל. למעשה שכבה זו מבצעת את פעולת הקונבולוציה על הקלט:

בהינתן מטריצת הקלט בגודל MxN וטנסור פילטרים בגודל $KxKxL$ נסתכל על (פילטר) בגודל KxK.
טנסור התוצאה של שכבת הקונבולוציה הוא אוסף של L מטריצות. האיבר ה i,j במטריצה הזו הוא התוצאה של כפל element-wise בין איברי הפילטר לאיברי החלון המתחיל באינדקס $K * i, K * j$.

שכבת הקונבולוציה מאפשרת לנו להחליף את שכבת הנוירונים הרגילה בשכבה יעילה הרבה יותר - פחות פרמטרים ותפיסה מרחבית של התמונה שהולכת וגדלה עם השכבות. תפיסה זו מאפשרת זיהוי תכונות בסיסיות בשכבות הראשונות וזיהוי תכונות מתקדמות בשכבות המאוחרות.

LSTM/Bidirectional

על שכבת ה LSTM כבר הסברתי בפירוט רב בחלק בניית המודל.
שכבת ה Bidirectional ב Keras נותנת אפשרות, בין היתר, לבצע שימוש דו כיווני ב LSTM.
בניגוד ל LSTM רגיל, השכבה הדו כיוונית מאפשרת למודל להתבסס על המידע 'מהעתיד' ומהעבר בשביל לעשות חיזוי באיטרציה t:



כפי שניתן לראות באיור, ישנן שתי שכבות של LSTM. אחת עוברת על המידע מההתחלה לסוף ואחת מהסוף להתחלה. ובשביל לחזות את y_t נשתמש באקטיבציה על סכום התוצאות של שתי השכבות.

שימוש זה ב LSTM מאפשר תוצאות טובות יותר שכן לעיתים בשביל לחזות אות באיטרציה t, נצטרך להסתמך על מידע שמקורו באיטרציה t+1 וכו'. יתר על כן, התוצאות תואמות לאינטואיציה - אחוז הדיוק על קבוצת הבדיקה עם LSTM חד כיווני הוא 72.9% כאשר אחוזי הדיוק עם LSTM דו כיווני הינם 79.8%

Add

זוהי שכבה פשוטה מאוד המקבלת שני טנסורים בעלי מימדים זהים כקלט ומחזירה את טנסור הסכום שלהם.

שכבה זו משמשת אותנו בעת מימוש ResNet.

Dropout

שכבה זו מקבלת כקלט טנסור והסתברות p. השכבה מחזירה את הטנסור, כאשר לכל איבר יש הסתברות p להיות 0.

זוהי דרך רגולריזציה נפוצה. במקרה זה השימוש היה הכרחי בכדי להימנע מ Overfit.

השימוש ב Dropout מאפשר למודל להסתמך על כל הנוירונים או המשקלים בכדי ללמוד את הדפוס האמיתי בנתונים.

Batch Normalization

בעת האימון, ובפרט בתהליך ה backward propagation, עדכון המשקולות משנה את המאזן (distribution) בין המשקולות, דבר היוצר מצב של חתול הרודף אחרי הזנב שלו. תפקיד ה Batch Normalization הוא לדאוג שטווח המשקולות ישאר ללא שינוי, עד כמה שאפשר, לאורך תהליך האימון. פשוטו כמשמעו, שכבה זו שומרת על טווח קבוע בעבור המשקולות על ידי נורמליזציה לקלט (איפוס הממוצע והשונות). במודל שלי, לאחר כל שכבת קונבולוציה ישנה שכבת נורמליזציה.

כפי שמצביעה התאוריה, שמגובת בניסויים רבים, התהליך מקצר משמעותית את זמן האימון, מאפשר למידה אגרסיבית יותר עם learning rate גדול, מזניח את השפעת אתחול המשקולות על תוצאות האימון ואף יכול לשמש כשיטת רגולריזציה.

במהלך הפרויקט ניסיתי שילובים שונים בין Dropout ל .Batch Normalization מסקנותי הן שהשימוש בשתי השיטות יחד משפר את תהליך הלמידה.

פונקצית השגיאה

פונקצית השגיאה המתאימה עבור פרויקט זה היא פונקצית ה - CTC - Connectionist Temporal Classification . פונקציה זו אחראית על שתי פעולות עיקריות חישוב ה loss בעבור תהליך הלמידה ופענוח הפלט של ה - LSTM.

פענוח הפלט - CTC decode

נניח שנרצה לפענח את המילה "hi". עבור time steps = 5 יתכן ותוצאת ה softmax תהיה משהו כמו "hhhii". נשים לב שאם נתעלם מאותיות חוזרות נקבל את המילה הרצויה - "hi". אבל, שיטה זו אינה מספיקה, שכן אם נרצה לחזות את המילה "hello" אז אין לנו דרך המאפשרת לחזות את הרצף "ll" מכיוון שהוא יהיה מאוחד לכדי "l". בשביל כך נוסיף אות מיוחדת הנקראת blank ונסמנה ב - ε. כעת נדרוש שבין כל אות שחוזרת על עצמה יהיה ε. כך למשל הרצף "hhhheeellεloo" שקול לרצף "hello". לסיכום, פעולת הפיענוח מורכבת משני חלקים: הסרת אותיות חוזרות והסרת האות ε.

חישוב העלות - Loss

ראינו שבהינתן רצף באורך n ישנן אפשרויות רבות לייצוג מילה באורך k. למשל את המילה tea ניתן לייצג בעזרת הרצפים הבאים באורך 4:

ttea, teea, teaa, tεea, teεa, teaε

הדרך בה פונקצית העלות פועלת היא פשוטה: בעזרת תוצאות ה - softmax נחשב את ההסתברות לכל אחת מהאפשרויות. למשל:

P(ttea) = P(t אות ראשונה היא) * P(t אות שנייה היא) * P(e אות שלישית היא) * P(a אות רביעית היא)

באופן כללי אם נסמן את המילה $\underline{w} = (w_1, w_2, \ldots, w_n)$ ואת מטריצת התוצאה של ה softmax ב A אז מתקיים

$$P(\underline{w}) = \prod_{i=1}^{n} P(w_i \mid A_i)$$

לבסוף נסכום את כל האפשרויות ונסמן את הסכום P_{word} . נגדיר את פונקצית העלות להיות:

$$Loss(y\_true, y\_pred) = -log(P_{word})$$

ההתנהגות של פונקצית ה log מתאימה במקרה זה. ככל ש - $P_{word} \rightarrow 0$ נרצה עלות גדולה יותר כי המודל נכשל בחיזוי המילה, ובאמת נקבל $Loss \rightarrow \infty$. באופן דומה ככל ש - $P_{word} \rightarrow 1$ נרצה עלות קטנה יותר כי המודל הצליח, ובאמת $Loss \rightarrow 0$.

תוצאות האימון

במהלך הפרויקט ניסיתי מגוון רחב של קונפיגורציות של רשתות. ראשית אסביר על המדדים באמצעותם מדדתי את הצלחת הרשת :

Word Error Rate (WER)

זהו מדד לאחוז השגיאה בסיווג מילים, כלומר $WER = 1 - ACCURACY$. מדד זה טוב להערכת המודל בכלליות, אך הוא לא מושלם, שכן למשל עבור המילה ABC נרצה למצוא מדד בו המילה ACB בעלת שגיאה קטנה יותר מהמילה ACBBBBB.

Character Error Rate (CER)

זהו מדד המשפר את הערכת החיזויים אותם מבצע המודל. עבור מחרוזת בודדת, מדד זה למעשה מחשב את מרחק לוינשטיין בין המחרוזת הנכונה לבין המחרוזת אותה המודל חזה ומחלק באורך המחרוזת הנכונה.

מרחק לוינשטיין בין שתי מחרוזות מוגדר כמספר המינימלי של פעולות עריכה שיש לבצע על מחרוזת אחת כדי להגיע למחרוזת השנייה, כאשר פעולות העריכה המותרות הן : הוספת אות, מחיקת אות או שינוי אות לאות אחרת. את המרחק הזה נחלק באורך המילה. לכן למשל מרחק לוינשטיין בין המילה ABC ל ACB הוא 2/3 - נחליף את C ב B ונחליף את B ב C – כלומר שתי פעולות עריכה חלקי אורך המילה = 3. לכן, ניתן לחשוב על מדד ה CER כאחוז האותיות שהמודל סיווג לא נכון עבור המילה הממוצעת.

סקירת התוצאות

אלו הן התוצאות הכי טובות אליהן הצלחתי להגיע :

$$1 - WER(test) = ACC(test) = 0.7977$$

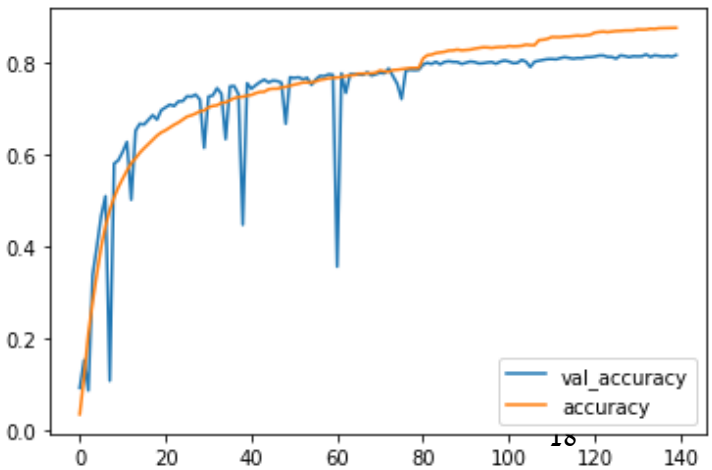
$$1 - WER(train) = ACC(train) = 0.8034$$

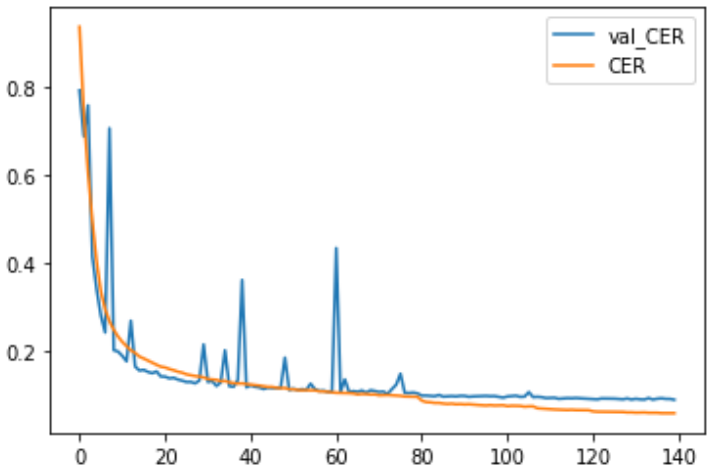
$$CER(average, test) = 0.0944$$

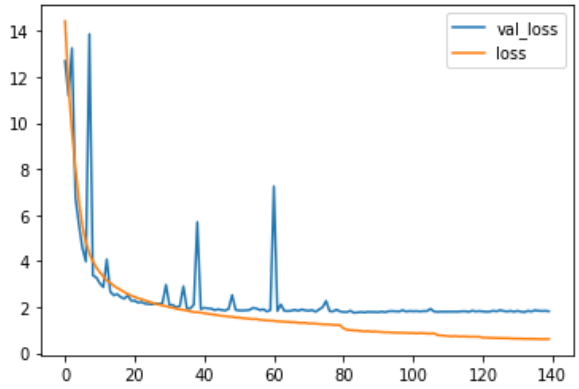
$$Loss(test) = 1.6224$$

אלו תוצאות לא רעות. המודל השיג אחוז דיוק של כ - 79.77%. בנוסף, ממדד ה - CER אנו רואים שבממוצע 9.44% מהמילה שהמודל חזה היא שגויה, כלומר 91.56% מהמילה נכונה בממוצע.

אלו הגרפים של המדדים הללו (accuracy (1-WER), CER, Loss) :







בסופו של דבר לא היה Overfit גדול, גם אם זה נראה כך. למשל באיטרציה האחרונה, ההפרש בין val_CER ל train_CER היה 0.03, כלומר 3%.

שיפור הצלחת המודל ו Hyper Parameters

במקרה שלי, בעיקר בגלל גודלו העצום של מאגר הנתונים, כל אימון מלא של המודל ארך בין 7 ל 10 שעות. לכן, הייתי צריך לבחור בקפידה את השינויים בהיפר פרמטרים.
במהלך הפרויקט ביצעתי 20 הרצות עם קונפיגורציות שונות של פרמטרים. התחלתי עם מודל בסיסי מ - [5] שהגיע ל 70% הצלחה וניסיתי לשפר את התוצאות. הדבר הראשון שעשיתי ששיפר את התוצאות הוא דבר שעוזר ועובד בהרבה פרויקטים, והוא ביצוע Batch Normalization לאחר שכבת האקטיבציה ולא לפניה. לאחר השינוי הזה הצלחתי להגיע ל 73.6% דיוק. לאחר מכן החלטתי להעלות את הסיבוכיות של המודל מכיוון שראיתי שהמודל סובל מעט מ underfit. הוספתי בלוקים (residual blocks) נוספים למודל והדיוק עלה ל - 77% בקבוצת הבדיקה. לאחר מכן בחנתי את השפעתן של פונקציות האקטיבציה.
מסקנותי היו כי פונקצית ה ReLU טובה יותר במקרה זה (הגעתי ל 78%) מפונקצית ה Leaky ReLU.
כמו כן בדקתי כמספר אופטימייזרים לתהליך ה Gradient Descent ומצאתי ש ADAM הוא האופטימייזר בעל הביצועים הטובים ביותר ביחס ל SGD ו RMSprop.
לבסוף, החלטתי להוסיף שכבה נוספת של LSTM דו כיווני בכדי להעלות שוב את סיבוכיות המודל וקיבלתי כ - 79.77% דיוק בקבוצת הבדיקה.

ההיפר פרמטרים אותם בחרתי הם להלן :

Parameter	Value
learning rate	0.001
batch size	64
dropout probability	0.2
LSTM units	128

64	time steps

ייעול ההתכנסות

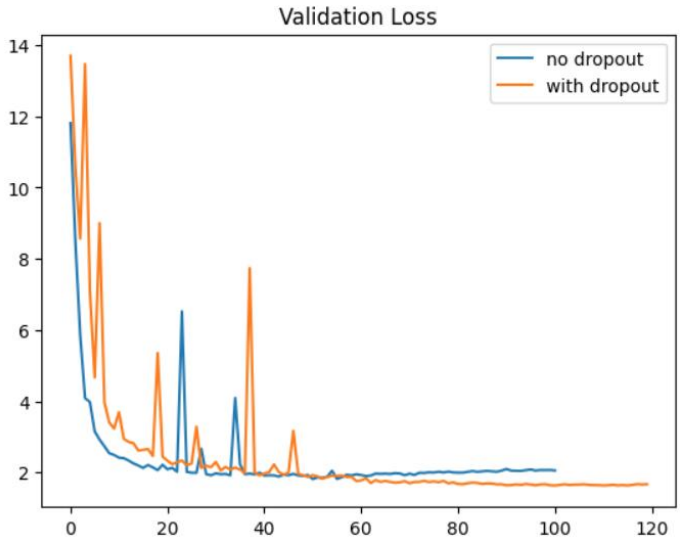
בשביל לייעל את התכנסות אלגוריתם ה - Gradient Descent. בחנתי אלגוריתמי ייעול התכנסות שונים כמו SGD with momentum, RMSprop ו ADAM.לאחר ניסוי וטעייה, גיליתי כי ADAM הוא האלגוריתם שהביא לתוצאות הכי טובות.
אלגוריתם ADAM למעשה משלב את שני האלגוריתמים שהזכרתי לעיל לכדי אלגוריתם יחיד :

1. האלגוריתם מנסה להקטין את ההתקדמות המשקולות בכיוון שהוא לא ישיר למינימום, כלומר מנסה לצמצם 'זיגזגים' אל עבר המינימום הגלובלי.

2. נעשה שימוש בפרמטר $\beta \leq 1$ שקובע את חשיבותם של הגרדיאנטים הקודמים בעת החישוב של הגרדיאנט הנוכחי. השקלול של הגרדיאנטים הקודמים יוצר תחושה של תנע (momentum) המאפשרת להחליק במהירות על חלקים בהם הגרדיאנטים קטנים מאוד ולהגיע למינימום מהר יותר.

התמודדות עם Overfit

למעשה המודל מתמודד עם overfit בשתי דרכים חשובות - Batch Normalization, Dropout.
שני אלה מאפשרים רגולריזציה, כפי שהסברתי בחלק על שכבות המודל. אציין כי ברוב המאמרים שקראתי לא היה מומלץ להשתמש ב Batch Normalizaion ו Dropout יחד, אבל במודל שלי, בניסויים בהם לא שמתי את אחד מהם, היה Overfit גדול מאוד.
כך למשל כאשר הפסקתי את השימוש ב Dropout ההפרש בין אחוזי הדיוק של ה train ל test הגיע לכדי 20%! בגרף הבא ניתן לראות את ההבדלים בקבוצת האימון באימון עם dropout לעומת בלי :



בנוסף, כאשר העליתי בצורה משמעותית את כמות הפרמטרים של המודל היה Overfit גדול שלא הצלחתי למתן. זהו דבר לא מפתיע כל כך, שכן ככל שסיבוכיות המודל עולה, יש לו אפשרות להתמקד בצורה מדויקת יותר על נתוני האימון, אך צורה זו אינה בהכרח מבטיחה להציג את המגמה האמיתית בנתונים.
לבסוף, הצלחתי למצוא מאזן אידאלי בין כמות הרגולריזציה לסיבוכיות המודל. גם הגרפים שהצגתי בחלק הצגת התוצאות וגם השימוש במודל באפליקציה מעידים כי הצלחתי להימנע, עד כמה שאפשר, מבעיית ה Overfit.

שלב היישום

בשביל להדגים יישום המשתמש במודל המאומן, יצרתי אפליקציית Android לטלפון. באפליקציה, המשתמש יכול לבחור תמונה מקבצי הטלפון או לחלופין, לצלם תמונה ישירות מהמצלמה, של מסמך המכיל טקסט. תפקידה של האפליקצייה יהיה לבצע לתמונה הכנה מקדימה (preprocessing), לחלק את התמונה למילים ולהשתמש במודל על מנת לחזות את המילים הללו לפי הסדר.

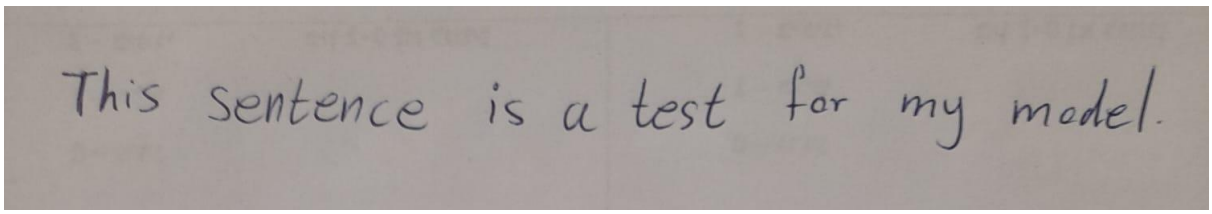
את האפליקציה, מימשתי באמצעות ספריות Android המגוונות בשפת Java וגם בעזרת ספריות נוספות כמו OpenCV [7] ו Image Cropper [8], בכדי לעשות עיבוד מקדים למסמך. כמו כן, השתמשתי בספריית Tensorflow Lite המיועדת להרצת מודל מאומן באופן מקומי על מכשיר הטלפון הנייד.

קליטת הנתונים מהמשתמש

הדבר הראשון שיש לעשות הוא לקבל תמונה מהמשתמש עליה נבצע את החיזוי. באפליקציה ישנה אופציה לבחור תמונה מן הגלריה ולבחור תמונה מן המצלמה. בספריות Android יש מימוש פשוט מאוד לכל אחת מהאפשרויות הנ"ל, ולכן פשוט השתמשתי בו.

עיבוד מקדים של המסמך

עכשיו, יש לנו תמונה מהמשתמש, אך עדיין נותרה עבודה רבה. אנו צריכים למזער עד כמה שניתן את הרעש, להעלות את הניגודיות בתמונה ולהפריד את המסמך למילים.
את עיבוד המסמך אנחנו עושים בעזרת הספרייה OpenCV. זוהי ספרייה נוחה וחינמית לעיבוד תמונות.
אדגים את עיבוד התמונה עבור המסמך הבא:



אנחנו הופכים את התמונה לשחור לבן, מבצעים Denoising ולבסוף מבצעים thresholding לתמונה.
תהליך ה Denoising מבטל עד כמה שניתן רעש לא רצוי בתמונה כמו נקודות על הדף, תאורה לא טובה וכו'; תהליך ה - thresholding מבצע ניקוי רעש נוסף לתמונה ויתר על כן, מוצא רק את החלקים החשובים בתמונה והופך את הניגודיות ביניהן למקסימלית. תהליך זה מתממש על ידי הקריאה לפונקציה:

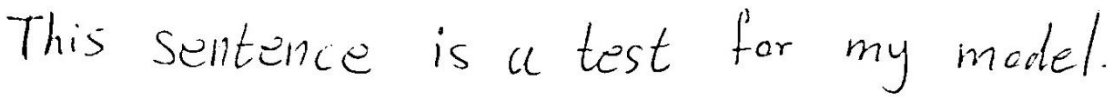
```
cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,
    cv2.THRESH_BINARY,block_size,C);
```

זהו סוג מיוחד של thresholding שלמעשה מפעיל את הפונקציה המתמטית הבאה על הפיקסל ה - i,j:

$$thresh_{i,j} = mean\ across\ block\ size\ amount\ of\ neighbors\ - C$$

$$img[i][j] = 255 \;\; if \; img[i][j] \geq thresh_{i,j} \;\; , 0 \; otherwise.$$

בצורה זו, אנו נפטרים מעוד שכבת רעש ובמקביל, יכולים לספק למודל תמונה נקייה יותר המורכבת רק משחור ולבן. בסוף השלב הזה התמונה נראית כך:

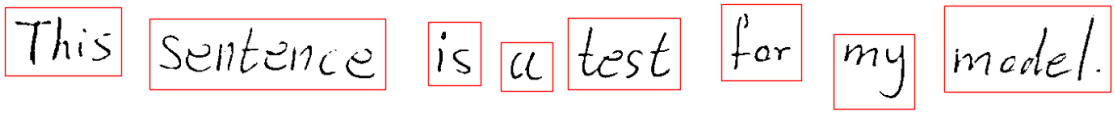


עכשיו כל שנותר הוא לחלק את התמונה הזו למילים ולשלוח אותם למודל.
בשביל לחלק את התמונה למילים, ראשית נרצה לעבות את האותיות כך שכל מילה תוצג כגוש אחד בתמונה (באופן הזה נוכל להבדיל בקלות רבה יותר בין מילים). כאן למשתמש יש אפשרות לבחור כמה צפוף הוא כתב. ככל שהכתב פחות צפוף נצטרך להרחיב יותר את האותיות בכדי שיתאחדו למילה אחת, ואותו דבר להפך. לאחר השלב הזה התמונה נראית כך:

לפי מענה המשתמש אנו מבצעים את הרחבת האותיות ולאחר מכן קוראים לפונקציה:

```
findContours(img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE);
```

הפונקציה הנ"ל של ספריית OpenCV יודעת להפריד את התמונה לחלקים ולמצוא את קווי המתאר של כל 'גוש' אותיות (כלומר מילה) ולהחזיר את המלבן החוסם את המילה:



לבסוף אנו מבצעים פעולות אחרונות על כל מלבן שקיבלנו - שינוי גודל התמונה ונרמול הערכים לטווח [0,1], בכדי להיות תואמים למה שהמודל מצפה.

הערה: עיבוד התמונה שיישמתי הוא אינו מושלם, ולכן לא מביא לתוצאות טובות כל כך בעת החיזוי. ניתן לשפר את תוצאות החיזוי על ידי שימוש באפליקציה שעושה עיבוד מקדים לתמונה כמו למשל CamScanner.

השימוש במודל

לאחר שיש לנו את התמונה של המילה זה הזמן להשתמש במודל כדי לחזות אותה. בכדי לעשות זאת, השתמשתי בספריית Tensorflow Lite. ספרייה זו מקבלת גרסה דחוסה של המודל ומממשת את תהליך ה - forward propagation בצורה חכמה המאפשרת הרצה מקומית על גבי הטלפון.
השימוש בספרייה הוא די פשוט. צריך לטעון את התמונה ל Input Buffer ולקרוא לפונקציה tflite.run.
תוצאת הפונקציה היא Output Buffer המכיל את תוצאות ה softmax בעבור הקלט. עבור כל time step אנו בוחרים את האות עם ההסתברות הגבוהה ביותר ולבסוף שולחים את המערך ל CTC decoder בכדי לקבל את התוצאה כ String.

לבסוף, נוכל להדפיס את התוצאה על המסך.
לנוחיות המשתמש, הוספתי כפתור המאפשר העתקה של התוצאה ל clipboard

מדריך למפתח

את כל קוד הפרויקט (כולל היישום) תוכלו למצוא כאן תחת וב [9]. אלו הם הקבצים בתוך תקיית הדרייב :

Project.ipynb
זהו הקובץ הראשי המכיל את כל קוד הפרויקט.

CreateCsvData.ipynb
זהו הקובץ האחראי לסידור הכתובת של כל תמונה עם ה label המתאים לה. התוצאה נשמרת בקובץ **data.csv** בתור טבלה.

test.jpg, word.jpg
שתי תמונות המשמשות לצורך בדיקה ומופיעות כדוגמה בקוד.

models.xlsx
טבלת סיכום המכילה את תוצאות כל ההרצות שעשיתי במהלך העבודה על הפרויקט.

train history
תיקייה המכילה את היסטוריית האימון של כל המודלים (מסודרת לפי המודלים ב models.xls). כל קובץ מכיל את ה loss, WER, CER של קבוצת האימון והבדיקה בכל epoch.

model
תיקייה המכילה את היסטוריית המודלים המאומנים שהכנתי במהלך הפרויקט (מסודרת לפי המודלים ב models.xls).

Lite Model
תקייה המכילה גרסאות של מודל ה tflite הארוז והמוכן לשלב היישום.

app
תיקייה המכילה את כל קבצי הקוד של האפליקציה כ zip ומכילה את קובץ ההתקנה של האפליקציה (apk.) הנקרא app-debug.apk

dataset, labels
שתי תקיות המכילות את כל מאגר התמונות וכל מאגר קבצי ה - xml בהם נמצאים ה labels של כל התמונות.

מדריך למשתמש

התקנה

דרישות

- לפני התקנת האפליקציה, מומלץ לעדכן את הטלפון לגרסת ה Android העדכנית ביותר המומלצת על ידי מערכת ההפעלה של הטלפון.
- כמו כן, יש לוודא כי מכשיר הטלפון תומך במצב Portrait Mode (שימוש אנכי במסך).
- האפליקציה תומכת אך ורק במסמכים הכתובים על דף חלק ולבן.

מדריך ההתקנה

המדריך הבא יהיה עבוד טלפון מסוג Samsung, אך ניתן לעקוב אחריו עם כל מכשיר androrid אחר. יתכן ששמות ההגדרות יהיו שונים בטלפונים אחרים.

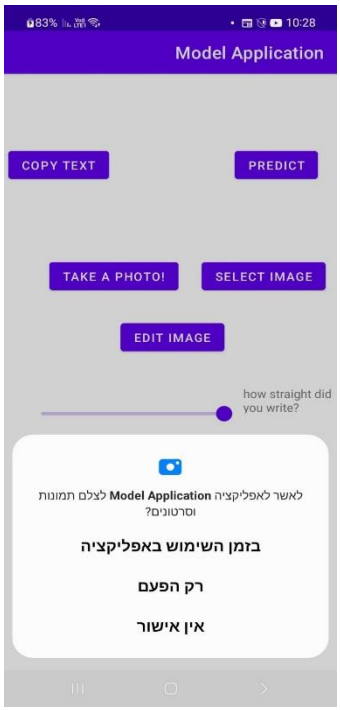
בכדי להתקין את האפליקציה, ראשית יש להותיר התקנת אפליקציות ממקורות לא ידועים. הכנסו להגדרות והקלידו בתיבת החיפוש: "התקן יישומים לא מוכרים". לאחר מכן, סמנו את האפשרות "הקבצים שלי".

כעת לחץ על קובץ ההתקנה (ניתן למצוא אותו ב [9] תחת תיקיית app) ולחץ התקנה.

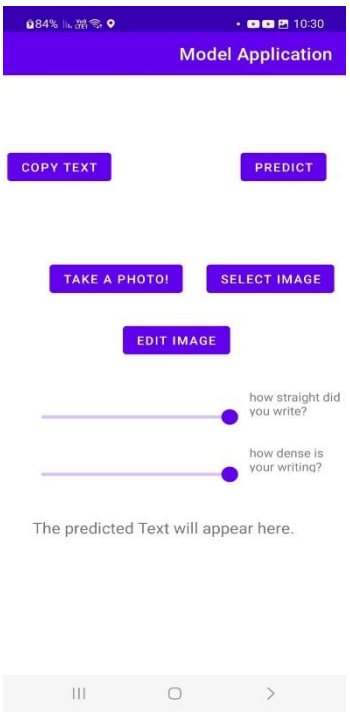
לאחר כמה שניות יופיע חלון המודיע כי האפליקציה הותקנה בהצלחה.

השימוש באפליקציה

בעת פתיחת האפליקציה תקבלו הודעת בקשה לאישור גישה אל המצלמה. אישור זה הוא הכרחי בשביל להיות יכולים לצלם את המסמך אותו תרצו לסרוק. אשרו את השימוש.

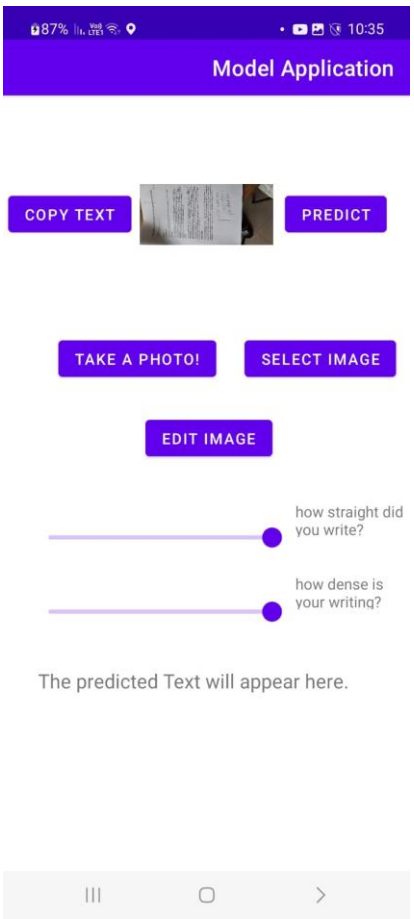


כעת הגענו אל חלון הבית :

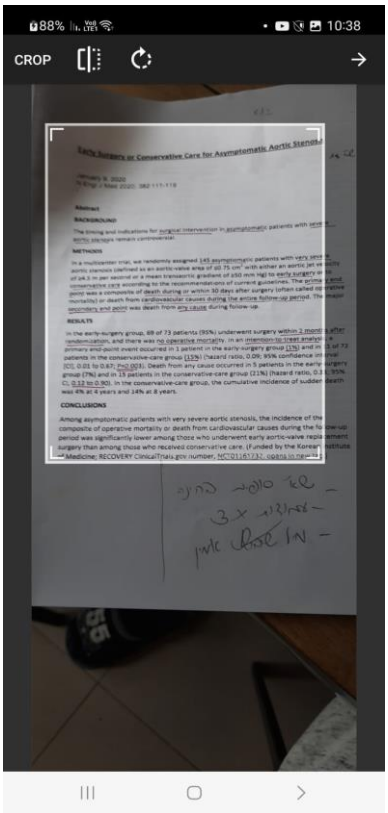


כאן ניתן לבחור אפשרויות שונות. בשביל להתחיל, לחצו על כפתור "Select Image" בשביל לבחור תמונה מהגלריה, או לחלופין לחצו על "Take A Photo" בשביל לצלם תמונה ישירות מהמצלמה. אם תבחרו באפשרות זו, חלון המצלמה יפתח ותוכל לצלם. לאחר הצילום ישנה אפשרות לנסות שוב, או לאשר. כאשר התמונה טובה לטעמכם, לחצו על אישור.

במצב הזה התמונה אמורה להופיע למעלה :



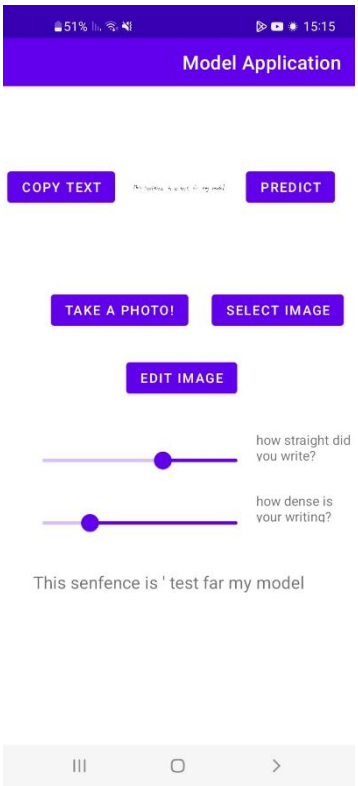
יתכן והתמונה תהיה מסובבת או שתרצו לחתוך אותה כדי שיראו רק את הדף. בשביל לעשות זאת, לחצו על כפתור "Edit Image". חלון חדש יפתח המאפשר לכם לעשות זאת:



לאחר שאתם מרוצים מאיך שהתמונה יצאה, לחצו על כפתור ה - Crop, או לחלופין אם התחרטתם ולא תרצו לערוך את התמונה, לחצו על כפתור החץ בפינה הימנית למעלה בשביל לחזור אחורה.
כעת וודאו כי התמונה החדשה המוצגת בראש דף הבית היא התמונה הערוכה.

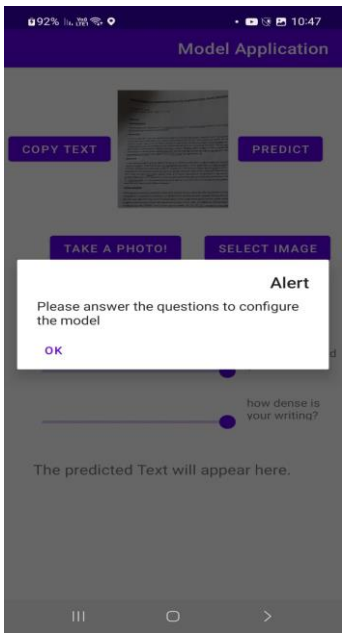
כל שנשאר הוא לענות על שתי שאלות קצרות המשמשות לשיפור תוצאות הזיהוי.
גררו את הסמן משמאל לשאלה "how straight did you write?" כמענה לשאלה כמה ישר כתבתם על גבי הדף. כאשר הסמן נמצא במצב הימני ביותר - אתם מצהירים כי כתבתם בצורה לא ישרה בכלל, וכאשר הסמן במצב השמאלי ביותר - אתם מצהירים כי כתבתם בצורה ישרה מאוד.
באותו אופן, גררו את הסמן משמאל לשאלה "how dense is your writing?" כמענה לשאלה כמה צפוף הכתב שלכם. כאשר הסמן נמצא במצב הימני ביותר - אתם מצהירים כי כתבכם אינו צפוף כלל (יש מרווח גדול בין אותיות), וכאשר הסמן במצב השמאלי ביותר - אתם מצהירים כי כתבכם צפוף מאוד (יש מרווח קטן מאוד בין אותיות).
אם אינכם בטוחים במה לענות, יש לשים את שני הסמנים, בקירוב, בנקודת שלושת רבעי הדרך.

זהו! הכל מוכן ללחיצה על כפתור ה "Predict". לאחר הלחיצה, העזרו בסבלנות. עיבוד התמונה עלול לקחת כמספר שניות, ולאחר מכן תוצאת החיזוי תופיע למטה :



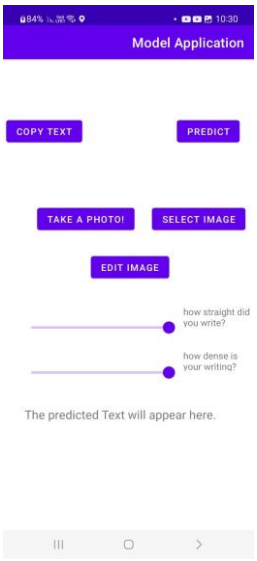
תוכלו להעתיק אותה על ידי לחיצה על הכפתור Copy Text.

אציין כי אם לא תענו על אחת מהשאלות, יופיע מסך עם ההודעה הבאה :



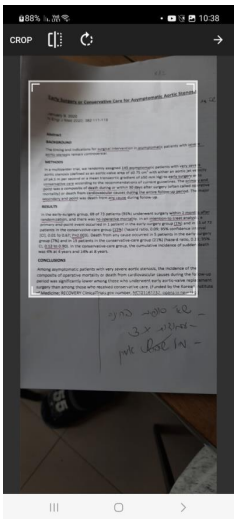
במקרה זה, לחצו על OK ומלאו מחדש את תשובותיכם על השאלות.

Screen Flow Diagram



מצלמה

גלריה



רפלקציה

במהלך השנה עמלתי על הכנת הפרויקט בהנדסת תוכנה בתחום שסקרן אותי ושהיה מעבר לתכנית הלימודים. במבט לאחור, אני שמח שבחרתי את נושא הפרויקט הזה. למדתי רבות על הבסיס לרשתות הנוירונים עליהם מבוססים כל מודלי ה AI ששוחררו לציבור בזמן האחרון. יתר על כן, למדתי כיצד להתמודד עם שגיאות וכיצד להתמודד עם ההרגשה של להיות תקוע על בעיה ללא כל מענה מועיל באינטרנט. לדעתי, זוהי יכולת חשובה עבורינו בעידן המודרני, ובפרט עבור תכנתים.

העבודה על הפרוקיט היתה מלווה בפתרון אתגרים רבים. פתירה של כל אתגר כזה הייתה מאוד מספקת, אך לעיתים גם מתסכלת. למשל, קרה לי לא פעם שלאחר שעות של debugging הבנתי שפשוט לא השתמשתי בגרסה העדכנית ביותר של ספרייה כלשהי. למרות זאת, בזכות הפרויקט, הבנתי לעומק את החומר התיאורטי שלמדנו במהלך השנתיים האחרונות וגם למדתי כיצד ניתן לפתח אפליקציה בסיסית ב Android.

אני מרוצה מכך שביצעתי את הפרויקט הזה וקיבלתי הזדמנות מעולה להתנסות בעיסוק היום יום בתחום הלמידת מכונה.
למרות זאת, יכולתי לעשות את העבודה הזו בצורה טובה יותר ויעילה יותר, שהייתה חוסכת לי זמן רב.
אם הייתי משקיע יותר זמן בתחילת העבודה למחקר ולניהול זמנים, הייתי יכול להיות יעיל יותר בהמשך.
לדוגמה, השקעתי חודש שלם מתחילת העבודה על נסיון לממש פרויקט זהה בשפה העברית. הייתי שמח אם פרויקט כזה היה עובד, אך אם הייתי עורך מחקר מקדים ויסודי, הייתי מגלה כי אין מאגרי מידע המתאימים לבעיה זו באינטרנט (בחינם).
עבודה יעילה יותר הייתה חוסכת לי זמן רב שהושקע בפרויקט שיכל להיות מושקע בדברים אחרים, או לחלופין להיות מושקע בשיפור התוצאות של הפרויקט.

אם הייתי יכול להתחיל מחדש את הפרויקט, הייתי מבצע את השינויים האלה ובנוסף, הייתי שוקל מחדש האם לבחור בנושא הפרויקט. לדעתי, נושא ה OCR הוא מעניין ומאתגר למימוש. בהחלט היה מאתגר לגרום לפרויקט זה לעבוד כמתוכנן ולהסביר את כל הטכנולוגיות החדשות בהן נעשה שימוש. הייתי יכול להרוויח זמן רב אלמלא בחרתי בנושא זה כנושא העבודה והייתי מתמקד בנושא פשוט יותר הדורש שימוש פשוט ב CNN. בכל מקרה, עכשיו, כשהתוצאה מונחת לפני, אני שמח שבחרתי בפרויקט זה, אך אני לא יודע אם הייתי מבצע את אותה בחירה שוב.
לסיכום, אני מרוצה מהפרויקט שכתבתי ואין לי ספק שהשקעתי את מלא המאמצים בכדי לבצע אותו על הצד הטוב ביותר. למדתי המון מהפרויקט וזהו ידע שאקח איתי לעתיד.

ביבליוגרפיה

1. GVE. (2022). Optical Character Recognition Market Size, Share & Trends Analysis Report.

2. Olah, C. (2015). Understanding LSTM Networks.

3. Bhat, R. S. (2022). Text Recognition With CRNN-CTC Network.

4. ICDAR. (1999). The IAM dataset.

5. Python Lessons. (2023). Handwriting words recognition with TensorFlow.

6. Shorten, C. (2019). Introduction to ResNets.

7. OpenCV. (2000). OpenCV.

8. Teplitzki, A. (2013). Android Image Cropper.

9. Zemach, O. (2023). Google Drive Folder.