

האוניברסיטה הפתוחה

מערכת חכמה לסיווג פקודות קוליות

פרויקט מתקדם במדעי המחשב - 22997

מגיש :

בן זגגי

העבודה הוכנה בהנחייתה של ד"ר מיה הרמן

מאי 2019

תקציר

במסגרת פרויקט מתקדם במדעי המחשב פותחה מערכת שרת-לקוח חכמה לסיווג פקודות קוליות בעזרת רשתות נוירונים עמוקות.

המערכת מקיימת את תנאי התחרות "Tensorflow Speech Recognition Challenge" של Kaggle [1] שבמסגרתה יש לפתח אלגוריתם חזק כך שבעזרת שימוש במאגר האימונים בלבד נוכל לזהות ולסווג את קטעי הקול ממאגר הבדיקות.

במסגרת התחרות ניתנו שני מאגרי מידע גדולים וחופשיים אשר מוגדרים לשימוש בחוקי התחרות של Kaggle, מאגרים אלו משמעותיים ומכילים כמויות גדולות של קבצי קול:

- **מאגר קטעי קול הנועד לאימון המערכת (Train Set)** – מאגר קטעי קול מכיל 65,000 קבצים. קטעי הקול הוקלטו ע"י אלפי אנשים שונים, כאשר בכל קובץ קיימת פקודה בודדת. כמו כן הקבצים השונים מתויגים ע"פ התוכן שלהם. [2]
- **מאגר קטעי קול לבדיקת המערכת (Test Set)** – מאגר זה מכיל כמעט 160,000 קבצים אשר בדיוק כמו Training Set הוקלטו ע"י אלפי אנשים שונים ומכילים בתוכם פקודות קוליות שונות. בשונה ממאגר האימונים, קבצי הקול שבמאגר זה לא מתויגים ומקוטלגים לפי תוכנם. [2]

מכיוון שרוב האלגוריתמים הקיימים היום בעולם הלמידה העמוקה מכוונים לסיווג תמונות בלבד, ומכיוון שהמידע הנתון למערכת הפרויקט הינו אותות קוליים בלבד, המערכת מבצעת המרה של מרחב הבעיה ממרחב האותות והקול למרחב התמונה.

המערכת כוללת שני שלבים עיקריים:

- שלב הלמידה המתבצע ב Offline במהלכו מתבצעות הפעולות הבאות:
 1. המרת קבצי הקול ממאגר האימונים - מייצוג במרחב הקול לייצוג במרחב התמונה
 2. בעזרת הייצוג של קבצי מאגר האימונים במרחב התמונה מתבצע אימון בגישת למידה עמוקה לרשתות הנוירונים השונות.התוצר של שלב Offline הינו מודלי רשתות נוירונים מארכיטקטורות שונות אשר ביכולתן לסווג מידע קולי המיוצג במרחב התמונה לאחת מהפקודות הקוליות הנתמכות במערכת. ייחודיותו של הפרויקט הינה באנסמבל המשלב בין 6 רשתות למידה עמוקה שונות וביצוע ממוצע משוכלל בין סיווגיהן השונים על מנת לקבל סיווג מערכתי יחיד.
- שלב הסיווג המתבצע ב Online במהלכו המערכת מבצעת את הפעולות הבאות:
 1. המרת קבצי הקול ממאגר הבדיקה - מייצוג במרחב הקול לייצוג במרחב התמונה
 2. סיווג הקבצים שהומרו בעזרת רשתות הנוירונים השונות שנוצרו בשלב Offline
 3. השוואת הסיווגים השונים המתקבלים במערכת ביחס לסיווגים המקוריים אשר התקבלו עבור המערכת שזכתה באתגר של Kaggle וקבלת ציון המערכת הכולל ע"פ השוואה זו.תוצאות סיווג המערכת שפתוחה עבור כל אחד מהקבצים המופיעים בסט נתוני הבדיקה הושו למול התוצאות שהתקבלו מהאלגוריתם אשר ניצח בתחרות של Kaggle (עבור אותו סט נתוני בדיקה). התוצאות שהתקבלו מהשוואה זו הראו כי איכותה ורמת דיוקה של המערכת עומדות על כ 94%.

תוכן עניינים

2	תקציר
5	מבוא
6	המרת קול לתמונה לצורכי למידה עמוקה
7	מטרת הפרויקט
8	דרישות המערכת
8	על המערכת
8	פעולה #1 – אימון רשתות נוירונים עמוקות
9	פעולה #2 – סיווג קבצי הבדיקה ע"י הרשתות שאומנו
9	פעולה #3 – ביצוע ממוצע משוקלל וקבלת סיווג יחיד לכל קובץ בדיקה
9	פעולה #4 – השוואת סיווגי המערכת לסיווגים המקוריים וקבלת ציון
10	שלב Offline
10	מטרת שלב Offline
10	תיאור Offline
12	שלב ה Online
12	מטרת שלב Online
13	דרישות פונקציונליות
13	קלט המערכת
13	פלט המערכת
13	קלט/פלט של תתי מערכות
15	דרישות לא פונקציונליות
15	דרישות חומרה מינימלית להתקנת מערכת
16	אילוצי מימוש
16	מגבלות והנחות המערכת
17	עיצוב מערכת תוכנה
17	ארכיטקטורה
18	תיאור רכיבים עיקריים
22	תיאור תהליכים עיקריים
23	תהליך אימון המערכת
25	תהליך תשאול המערכת
28	תהליך המרת קבצי אודיו לתמונה
28	אימון רשתות נוירונים עמוקות
29	סוגי ארכיטקטורות רשתות נוירונים עמוקות המשמשות לאימון המערכת
31	קלטים לרשתות הנוירונים העמוקות שבמערכת שפותחה הם :

32.....	דיאגרמת מחלקות
32.....	המחלקות השונות בשלב האימון
36.....	המחלקות השונות בשלב יצירת קבצי הפלט
39.....	המחלקות השונות בשלב התשאול
45.....	כלי פיתוח
45.....	Librosa
46.....	Flask
47.....	PyTorch
48.....	ממשק המשתמש
48.....	הרכיבים השונים איתם מתקשר ממשק המשתמש
49.....	שימוש בממשק המשתמש
52.....	תוצאות המערכת
52.....	בדיקות
52.....	בדיקות יחידה
56.....	בדיקות אינטגרציה
60.....	משקלים נוספים שנבדקו לצורך מקרה הבוחן
61.....	ניתוח התוצאות והסקת מסקנות
62.....	סיכום והצעות לפיתוחים עתידיים
63.....	נספחים
63.....	נספח א' – סטטיסטיקות
79.....	נספח ב' – איורים
79.....	נספח ג' – קוד
80.....	נספח ד' – התקנת המערכת
81.....	נספח ה' – מבנה רשת הנוירונים העמוקה VGG-2d
89.....	נספח ו' – מבנה רשת הנוירונים העמוקה VGG-1d עבור Mel Spectrograms
92.....	נספח ז' – מבנה רשת הנוירונים העמוקה VGG-1d
99.....	נספח ח' – מבנה רשת הנוירונים העמוקה Senet
100.....	נספח ט' – מבנה רשת הנוירונים העמוקה Resnet
103.....	נספח י' – מבנה רשת הנוירונים העמוקה DenseNet
104.....	נספח י"א – מאפייני רשתות הנוירונים השונות
106.....	נספח י"ב – הצעת פרויקט הצעה לשיפור הביצועים
108.....	נספח י"ג – הצעת פרויקט
119.....	רשימת מקורות

מבוא

למידה עמוקה (Deep Learning) הוא תת תחום בלמידת מכונה (Machine Learning) כאשר שני התחומים שייכים לעולם הבינה המלאכותית. זהו תחום מחקר בעולם המחשבים שמטרתו לחקות באופן מלאכותי את פעולת המוח האנושי.

אחת הסיבות העיקריות לנסיקה של תחום הלמידה העמוקה קשורה בהתפתחות והתחזקות המעבדים הגרפיים החדשים שיצאו לשוק בשנים האחרונות אשר מגיעים עם יכולת לעבד כמויות של מידע במקביל בכמה ערוצים (בדומה למוח האנושי).

הייחוד במערכות ללמידה עמוקה הוא היכולת שלהן ללמוד ולהשתפר כל הזמן וזאת על סמך דוגמאות, ממש כמו המוח האנושי.

בעזרת גישה זו הוכח כי ניתן לפתור מגוון רחב של בעיות ומשימות חישוביות בהן התכנות הקלאסי אשר עובדת במתכונת של כללים ספציפיים ומוגדרים מראש לא אפשרי.

פעולת האימון של מודל ללמידה עמוקה הינה אינטנסיבית ודורשת משאבי חומרה גדולים מאוד מהמערכת עליה האימון מתבצע.

מדובר בעיקר על כוח עיבוד עצום הנדרש לבצע איטרציות חוזרות ונשנות בכדי לעדכן את המודל בהתאם לדוגמאות עליהן הוא עובד. החישובים המתמטיים המתרחשים בשלב האימון הם פשוטים יחסית אך ישנה כמות אדירה של חישובים פשוטים כאלו שיש לבצע.

בעוד ליחידות העיבוד המרכזיות הסטנדרטיות (CPU) ישנן בדרך כלל מעט מעבדים חזקים, ליחידות העיבוד הגרפיות הסטנדרטיות (GPU) ישנן הרבה מאוד (אלפים) מעבדים חלשים יחסית. מסיבה זו מקבול של תהליך אימון של מודל ללמידה עמוקה על גבי GPU הוא אופטימאלי – אימון מודל ללמידה עמוקה על גבי GPU שאורך כ-5 ימים יכול לקחת כמה חודשים אם יאומן על גבי CPU.

בטווח הרחוק מטרת תחום הלמידה העמוקה הינה לפתח מחשב אשר יוכל להחליף את החשיבה האנושית. מערכת שכזו תהיה מסוגלת לזהות תבניות ודפוסים בעולם האמיתי (כגון קול ותמונה) אשר המערכות כיום מתקשות לעבד ולזהות.

תחום הלמידה העמוקה מצליח כל כך מכיוון שהוא מייצר פתרונות לבעיות ולתהליכים שגרתיים ויומיומיים – מזהוי דיבור בטלפונים החכמים ועד לשדרוג של חלקים מעולם הרפואה וההתקדמות אל עבר תחבורה אוטונומית.

למידה עמוקה משתלטת על כל תחומי החיים

תחום מחקר	פעולה	שימוש בתעשייה
 זיהוי קול	הבנה וניתוח של דיבור	האינטרנט של הדברים, טלקום, מערכות Personal Assistant
 טקסט	למידת שפה – קריאה וכתיבה	תרגום, סנטימנט שיח, תקשורת
 תמונה	ראייה ממוחשבת	רשתות חברתיות, רפואה (רדילוגיה), רכב אוטונומי, רובוטיקה
 ניתוח נתונים	עיבוד נתונים	סייבר, פיננסים, ניהול ארגוני, מדעים מדויקים (גם לרפואה), פרסום

TheMarker

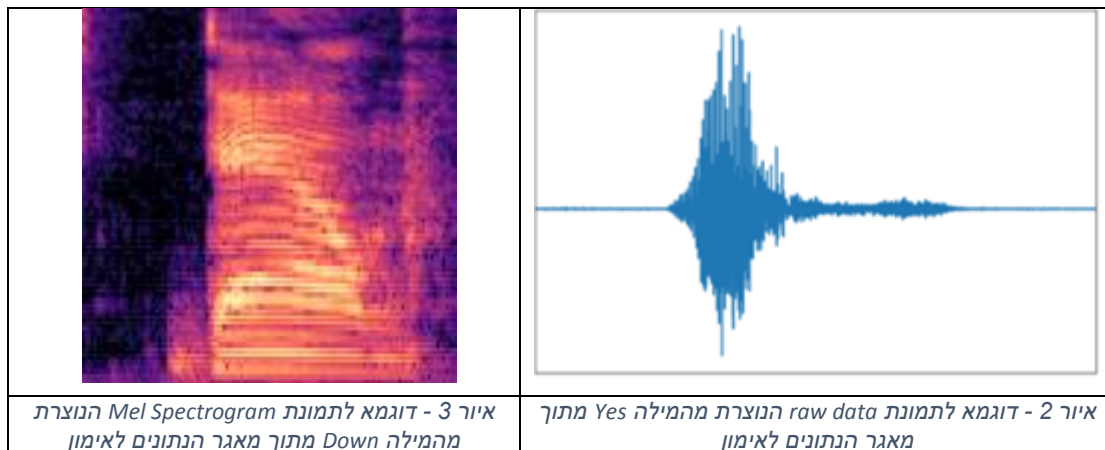
איור 1 – מתוך מגזין דה-מרקר – תחומי החיים והמחקר בהם נעשה שימוש בלמידה עמוקה

כיום, רוב רובן של העבודות בתחום למידה עמוקה עוסקות בסיווג תמונות.

המרת קול לתמונה לצורכי למידה עמוקה

בשנים האחרונות חלה התעוררות בנושא למידה עמוקה בתחום הקול. התעוררות זו קרתה בין היתר לאחר שהושגה ההבנה כי המרת מידע ממרחב הקול למרחב התמונה והזנתו לאלגוריתמים השונים אשר נועדו לתת פתרון לבעיות במרחב התמונה בעזרת למידה עמוקה, נותנת תוצאות טובות לא פחות מהתוצאות המתקבלות עבור בעיות ממרחב התמונה המוזנים לאותם אלגוריתמים.

עיצוב ובחירת הייצוג הנכון הינו מרכיב קריטי להצלחת תהליך הסיווג בעזרת אלגוריתמי למידה עמוקה. קבצי קול יכולים להיות מיוצגים בדרכים שונות כתמונות כדוגמת raw data (איור 2) או במרחב הייצוג הספקטרלי (איור 3).



שמע הינו גל קול אותו ניתן לייצגו כמערך חד-מימדי, איברי המערך מכילים את אמפליטודות גל הקול בנקודות זמן שונות, שהמרחקים ביניהן קבועים. מערך כזה קרוי: דגימה (sampling). זהו המבנה הבסיסי של קובץ מסוג wav.

כדי להפוך את המידע הקולי למתאים יותר כקלט לרשת עצבית, מפרקים את גל השמע המורכב – לחלקים המרכיבים אותו. מחלקים את גל השמע לחלקים עם גובה הצליל הנמוך ביותר, גובה הצליל הנמוך מעט פחות וכך הלאה, לבסוף מסכמים את כמות האנרגיה בכל אחד מן הצלילים וכך תתקבל מעין טביעת אצבע של קטע השמע.

הגישה הרווחת בהמרת אותות קוליים לצרכי למידה עמוקה הינה המרת האותות הקוליים למרחב הייצוג הספקטרלי כתמונה.

כפי שתיאר זאת Lonce Wyse מהאוניברסיטה הלאומית של סינגפור במאמרו משנת 2017

Audio spectrogram representations for processing with Convolutional Neural Networks – לייצוג הספקטרלי של קובץ קול ישנו תפקיד חשוב במימושים לסיווג קולי בעזרת רשתות נוירונים. הייצוג הספקטרלי מחזיק בחובו יותר מידע מרוב הייצוגים הקיימים עבור אותות קוליים במרחב התמונה אשר משמשים בדרך כלל לניתוח אותות קוליים ובעלי ממדים נמוכים יותר מאשר שמע גולמי (raw audio). [3]

מטרת הפרויקט

פיתוח מערכת חכמה לסיווג אותות קוליים תוך שימוש ברשתות נוירונים עמוקות.

נתוני הפרויקט הם מאגרי קול מתויגים ששימשו למטרת אימון ומאגרי קול לא מתויגים ששימשו לבדיקת נכונות הסיווג של המערכת.

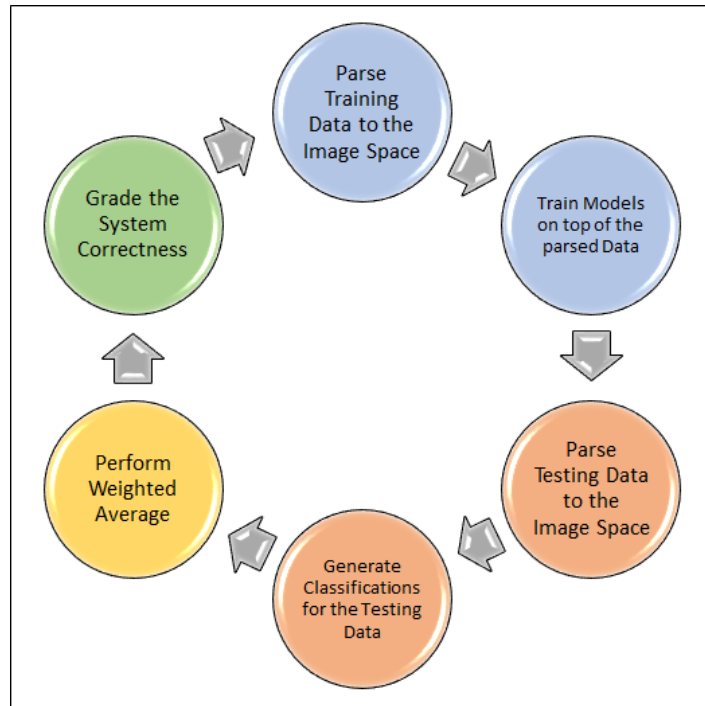
ייחודיות הפרויקט בהשוואה לעבודות הקיימות כיום הינו השילוב של מספר רשתות נוירונים עמוקות לצורך סיווג. זאת מול הגישה הסטנדרטית בעולם רשתות הנוירונים – אימון רשת אחת ושימוש בסיווג המתקבל ממנה באופן בלעדי.

בפרויקט נעשה אנסמבל של מספר רשתות יחידות. הרשתות בהן נעשה שימוש בפרויקט (ויתוארו במפורט בהמשך המסמך) ילמדו ויאומנו מראש על מילים שונות הנאמרות ע"י בני אדם שונים מתוך מאגר קבצי האימון המתויג. המטרה לשמה נוצרו רשתות אלו הינה לסווג נכונה את מאגר קבצי הבדיקה.

דרישות המערכת

על המערכת

המערכת פותחה בהתאם לארבע פעולות שונות המתוארות באיור 4. כל פעולה צבועה בצבע שונה והתוצר של כל פעולה חיוני לפעולה הבאה אחריה בשרשרת. מטרת העל של המערכת הינה לקבל ציון סיווג מקסימלי. טווח הציון הוא בין 0 ל 100.



איור #4 - מחזור החיים של המערכת

פעולה #1 – אימון רשתות נוירונים עמוקות

(מסומנת בכחול באיור 4)

פעולה זו מתחלקת לשני חלקים -

1. המרת קבצי הקול שנועדו לאימון המערכת למרחב התמונה
2. אימון רשתות נוירונים בעזרת למידה עמוקה על קבצים אלו

התוצרים של פעולה זו הינם 6 מודלים של רשתות נוירונים, אשר נוצרו בעזרת למידה עמוקה וביכולתם לסווג את תוכן קבצי הקול (לאחר המרתם לייצוג במרחב התמונה) לאחת מהאפשרויות הבאות:

Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Silence, Unknown

פעולה #2 – סיווג קבצי הבדיקה ע"י הרשתות שאומנו

(מסומנת בכתום באיור 4)

הפעולה השנייה מתחלקת לשני חלקים -

1. המרת קבצי הקול שנועדו לבדיקת המערכת למרחב התמונה
2. קבלת ושמירת סיווגי רשתות הנוירוניים שנוצרו בשלב הקודם עבור כל אחד מקבצי הבדיקה המומרים.

התוצרים של פעולה זו הינם תוצאות הסיווגים של כל אחד מהמודלים שנוצרו בשלב הקודם עבור כל אחד מקבצי הבדיקה

פעולה #3 - ביצוע ממוצע משוקלל וקבלת סיווג יחיד לכל קובץ בדיקה

(מסומנת בצהוב באיור 4)

בפעולה השלישית יתבצע ממוצע משוקלל עבור סיווגי המודלים השונים עבור כל אחד מקבצי הבדיקה.

התוצרים של פעולה זו הינם תוצאות הסיווגים הכוללים של המערכת עבור כל אחד מקבצי הבדיקה. בסיום פעולה זו, יהיה קיים סיווג אחד ויחיד לכל קובץ בדיקה (בשונה מסיום פעולה #2 בה לכל קובץ בדיקה היו 6 סיווגים שונים ע"פ כמות המודלים שאומנו).

פעולה #4 – השוואת סיווגי המערכת לסיווגים המקוריים וקבלת ציון

(מסומן בירוק באיור 4)

בפעולה הרביעית תתבצע השוואה בין תוצאות סיווגי המערכת שהתקבלו בפעולה #3 לבין תוצאות הסיווגים הנכונים שהתקבלו מהמערכת אשר ניצחה בתחרות של Kaggle.

התוצר של פעולה זו הינו ציון סיווג סופי עבור המערכת. ציון זה ישקף את טיב הסיווג של המערכת שפותחה במסגרת הפרויקט.

בחלקים הבאים בספר הפרויקט יתוארו שני שלבי המערכת העיקריים המשלבים בתוכם את ארבעת הפעולות המתוארות בחלק זה -

- שלב Offline – בשלב זה מתבצעות פעולות #1 ו #2. פעולות אלו קורות פעם אחת בלבד – בעת בניית המערכת הראשונית, לאחר מכן ניתן יהיה להשתמש בתוצרים שלהן לכל אורך החיים של המערכת.
- שלב Online – בשלב זה מתבצעות פעולות #3 ו #4. פעולות אלו אינן מעמיסות על המחשב עליו נמצאת המערכת ויכולות להתבצע פעמים רבות שכן העבודה הנכללת בשלב Online הינה בעיקר שימוש בתוצרים של פעולות #1 ו #2 בשלב Offline

שלב Offline

מטרת שלב Offline

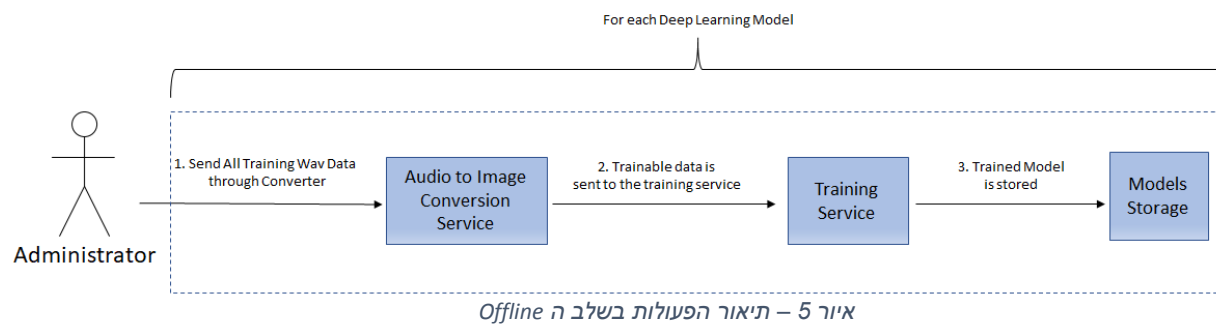
בשלב Offline מתבצעים חישובים כבדים הקורים פעם אחת בלבד לאורך חיי המערכת, מטרתו של שלב Offline הינה להכין תוצרים בדמות מודלי רשתות נוירונים המאומנים על קבצי האימון וסיווג קבצי הבדיקה בעזרת מודלים אלו. בעזרת תוצאות הסיווגים הללו, המשתמש יוכל לעבוד בשלב Online בצורה מהירה וחלקה על מנת לקבל את ציון המערכת הכולל (ובכך להגשים את מטרת המערכת הכוללת).

תיאור Offline

להלן תיאור התהליכים של שלב ה-offline כמודגם באיור 5 :

1. שליחת כל קבצי הקול שנועדו לאימון המערכת לשירות הממיר את קבצי הקול לתמונות במרחב הספקטרלי.
ההמרה למרחב הספקטרלי נעשית בעזרת ספריה מיועדת להמרה (Librosa הפיתונית), ספריה זו מקבלת קבצי קול מסוג wav ומחזירה תמונה במרחב הספקטרלי אשר מייצגת את אותו קובץ wav.
2. התמונות מוזנות לשירות האימונים המייצר מודלים שונים של רשתות נוירונים בעזרת למידה עמוקה.
תהליך אימון הרשת מתבצע באמצעות אלגוריתם back-propagation (חלחול לאחור) המקבל כקלט הראשוני שלו את התמונה במרחב הספקטרלי, בכל שכבה של הרשת מוזנים פיצורים ספציפיים מהקלט המוזנים לשכבה הבאה לעיבוד נוסף. כל שכבה ברשת הנוירונים מכילה משקלים שונים אשר מתכווננים לאט מאוד עד הגעתם לביצוע סיווג נכון. זה קורה ע"י כך שבכל איטרציית אימונים הרשת מקבלת פידבק חיובי אם צדקה ופידבק שלילי אם טעתה (ובכך משנה את משקלי השכבות השונות). כך שבמצב אידיאלי, לאחר איטרציות אימון רבות מאוד, הרשת מצליחה לסווג נכון יחסית את תמונות הקלט.
כלומר, אותם משקלים התחלתיים שנקבעו כמשקלים אקראיים לגמרי יתכווננו לאט ובזהירות במהלך ביצוע איטרציות רבות מאוד למשקלים האופטימליים המסווגים נכונה את קבצי הקול המוזנים להם.
3. המודלים ללמידה עמוקה מאוכסנים במקום משותף בו ייעשה שימוש בשלב Online על מנת לסווג ולקטלג קבצים נוספים.

הערה : שלב Offline מתבצע מראש על מחשב חזק שאינו תלוי במחשב בו ירוץ הזיהוי העתידי של קבצי קול אחרים.



שלב ה Online

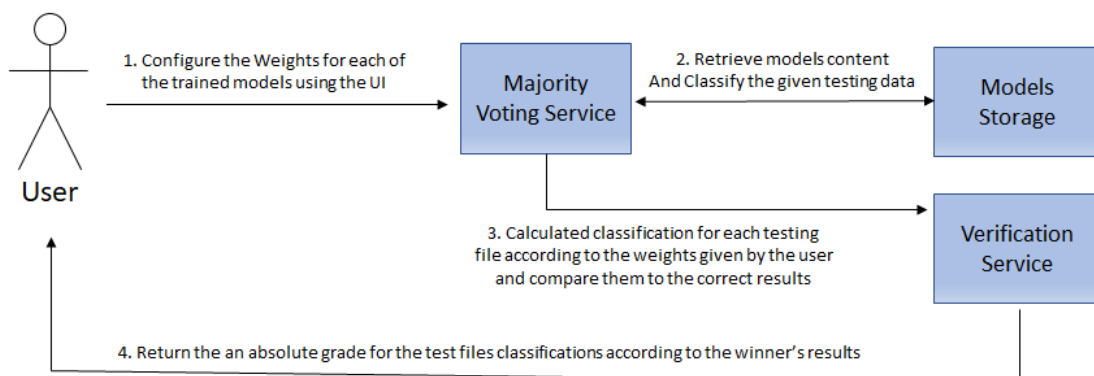
מטרת שלב ה Online

מטרת שלב ה Online הינה להחזיר למשתמש את ציון הסיווג הכולל של המערכת (ציון בין 0 ל 100) אשר בעזרתו ניתן לאמוד את טיב הסיווג של המערכת.

תנאי מקדים לשלב ה Online הוא כמובן סיום שלב ה Offline בהצלחה, שכן ללא היכולת לסווג את קבצי הבדיקה בעזרת המודלים שנוצרו בשלב ה Offline לא נוכל לדעת מהו טיב הסיווג של המערכת.

להלן תיאור התהליכים של שלב ה- online כמודגם באיור 6 :

1. בעזרת הממשק הגרפי המשתמש בוחר את המשקל וההשפעה של כל אחד מהמודלים שיוצרו בשלב ה Offline.
2. המודלים השונים שיוצרו בשלב ה Offline נשלפים ממקום האחסון שלהם, קבצי הבדיקה השונים מקוטלגים ע"פ כל אחד מהמודלים השונים (התוצאה המתקבלת עבור אותו הקובץ דרך מודלים שונים לא חייבת להיות זהה).
3. שירות Majority Voting מבצע ממוצע משוקלל על תוצאות הזיהוי שהתקבלו ע"פ המשקלים השונים שהמשתמש הזין דרך הממשק הגרפי.
4. השוואת תוצאות שלב Majority Voting לתוצאות המקוריות של הזוכה באתגר וקבלת ציון המערכת.



איור 6 – תיאור הפעולות בשלב Online

• הערות :

1. שלב ה Online מתבצע ב"Live" על המחשב של המשתמש ואינו תלוי מכונה.
2. פירוט על הדרכים בהן שלב ה Online ושלב ה Offline משתלבים במערכת יינתן במסגרת החלק בספר הפרויקט הכולל את תיאור תהליכי המערכת השונים המתחיל בעמוד 22.

דרישות פונקציונליות

קלט המערכת

בעזרת הממשק הגרפי, המשתמש בוחר את המשקל וההשפעה של כל אחד מהמודלים שיוצרו בשלב Offline, לאחר מכן הוא מפעיל את שלב Online אשר בסופו המערכת מחזירה פלט מתאים.

פלט המערכת

ציון באחוזים (מ1 עד 100) אשר מתאר את איכות הזיהוי של המערכת עבור קבצי הבדיקה בהינתן משקלי המודלים שהמשתמש הזין דרך הממשק הגרפי קודם לכן.

פלט זה נשען על השוואה הסיווגים השונים שניתנים לקבצי הבדיקה למול הסיווגים הנכונים שלהם אשר התקבלו מתוצאות הזוכה המקורי באתגר של Kaggle [7].

המערכת מחשבת את הציון באופן הבא :

$$Grade = \frac{\sum_{i=0}^{(test\ data\ files\ amount)} \begin{cases} SystemClassification[i] == CorrectClassifications[i] : 1 \\ else : 0 \end{cases}}{(test\ data\ files\ amount)}$$

הציון הכולל של המערכת מחושב לפי סכום הפעמים שהסיווג שהמערכת נתנה היה זהה לסיווג שהמנצח בתחרות נתן חלקי כמות הקבצים הכוללת לסיווג, בצורה זו ניתן לכמת ציון בין 0 ל 100 המבטא את טיב הסיווג של המערכת.

קלט/פלט של תתי מערכות

Audio to Image Conversion Service

קלט

1. קבצי קול מסוג WAV

עיבוד

1. המרת קבצי קול לתמונה בעזרת ספרייה חיצונית

פלט

1. ייצוג במישור התמונה לקבצי הקול מסוג Mel Spectrogram

Training Service

קלט

1. קבצי האימון אשר עברו המרה למרחב התמונה
2. סוג הרשת אותה נרצה לאמן

עיבוד

1. הפעלת אלגוריתמים ללמידה עמוקה ע"פ סוג הרשת שנבחרה

פלט

1. מודלים מאומנים של רשתות הניורונים העמוקות שבמערכת

Test Data Output Generator

קלט

1. כל מודלי רשתות הניורונים שאומנו
2. קבצי הקול ממאגר הבדיקות לאחר שהומרו לייצוג במישור התמונה

עיבוד

1. קטלוג כל אחד מקבצי האימון לפי כל אחד מהמודלים
2. שמירת תוצאות הקטלוג בקבצי CSV כאשר עבור כל מודל רשת נורונים יישמר קובץ CSV במערכת.

פלט

1. קבצי CSV לכל אחת מרשתות הניורונים שאומנו. בכל קובץ כזה קיימות שורות כמספר קבצי הבדיקות (158,536 שורות). כל שורה מכילה את שם הקובץ ואת הסיווג שלו עבור המודל אותו מתאר הקובץ הנוכחי.

Majority Voting Service

קלט

1. המשקלים (באחוזים) השונים שהוזנו ע"י המשתמש לכל אחד מ-6 המודלים, משקלים אלו יילקחו בחשבון בעת סיווג כל אחד מקבצי הבדיקה על מנת לקבל את הסיווג הסופי בעזרת ממוצע משוקלל.
2. קבצי ה-CSV שנוצרו בשלב הקודם עבור כל אחד מרשתות הניורונים

עיבוד

1. טעינת כל קבצי ה-CSV של כל אחת מרשתות הניורונים לזיכרון
2. בכל קובץ CSV, כזה, מספר השורות הינו כמספר קבצי הבדיקה, מתבצע מעבר על כל קובץ בדיקה (כל שורה)

3. חישוב ממוצע משוקלל עבור כל קובץ בדיקה שזוהי ע"פ המשקלים שהוזנו ע"י המשתמש
4. קבלת הסיווג הסופי עבור כל קובץ בדיקה כזה לאחר ביצוע הממוצע המשוקלל
5. החזרת הסיווג הסופי של המערכת עבור כל אחד מקבצי הבדיקה.

פלט

1. תוצאת הסיווג הסופית לאחר ממוצע משוקלל לגבי הזיהוי והסיווג של כל המערכת לגבי כל אחד מקבצי הבדיקה

Verification Service

קלט

1. הסיווגים הסופיים לאחר הממוצע המשוקלל עבור כל אחד מקבצי הבדיקה
2. תוצאות כל סיווגי הקבצים הנכונות (אשר התקבלו ע"י המערכת של המנצח בתחרות של Kaggle)

עיבוד

1. השוואת כל אחת מתוצאות הסיווגים השונים של קבצי הבדיקה למול תוצאת הסיווג הנכונה שהתקבלה ע"י המנצח בתחרות של Kaggle, מטרת ההשוואה היא מציאת הדמיון באחוזים בין תוצאות הסיווג של המערכת שלי לבין תוצאות הסיווג הנכונות של המנצח בתחרות של Kaggle.
2. חישוב הציון הסופי של המערכת – חישוב זה מתקבל ע"פ הנוסחה הבאה:

$$Grade = \frac{\sum_{i=0}^{(test\ data\ files\ amount)} \begin{cases} SystemClassification[i] == CorrectClassifications[i] : 1 \\ else : 0 \end{cases}}{(test\ data\ files\ amount)}$$

יתבצע מעבר על כל קבצי הבדיקה ותחשב את כמות הפעמים בה הסיווג של המערכת זהה לסיווג הנכון, לאחר מכן נחלק את מספר זה במספר קבצי הבדיקה בסה"כ – כך נוכל לקבל ציון אמין ונכון לאיכות המערכת.

פלט

1. תוצאה סופית באחוזים המתארת את איכות ונכונות הסיווג של המערכת.

דרישות לא פונקציונליות

דרישות חומרה מינימלית להתקנת מערכת

- מעבד – 4 ליבות.
- זיכרון RAM – 4 GB.
- דיסק קשיח – 350 MB.

אילוצי מימוש

- מערכת הפעלה תומכת python גירסא 3.
- לפחות מעבד גרפי חזק אחד לצורך אימון הרשתות.
- (זמן אימון הרשתות תלוי בכמות ואיכות המעבדים הגרפיים שעל המחשב).

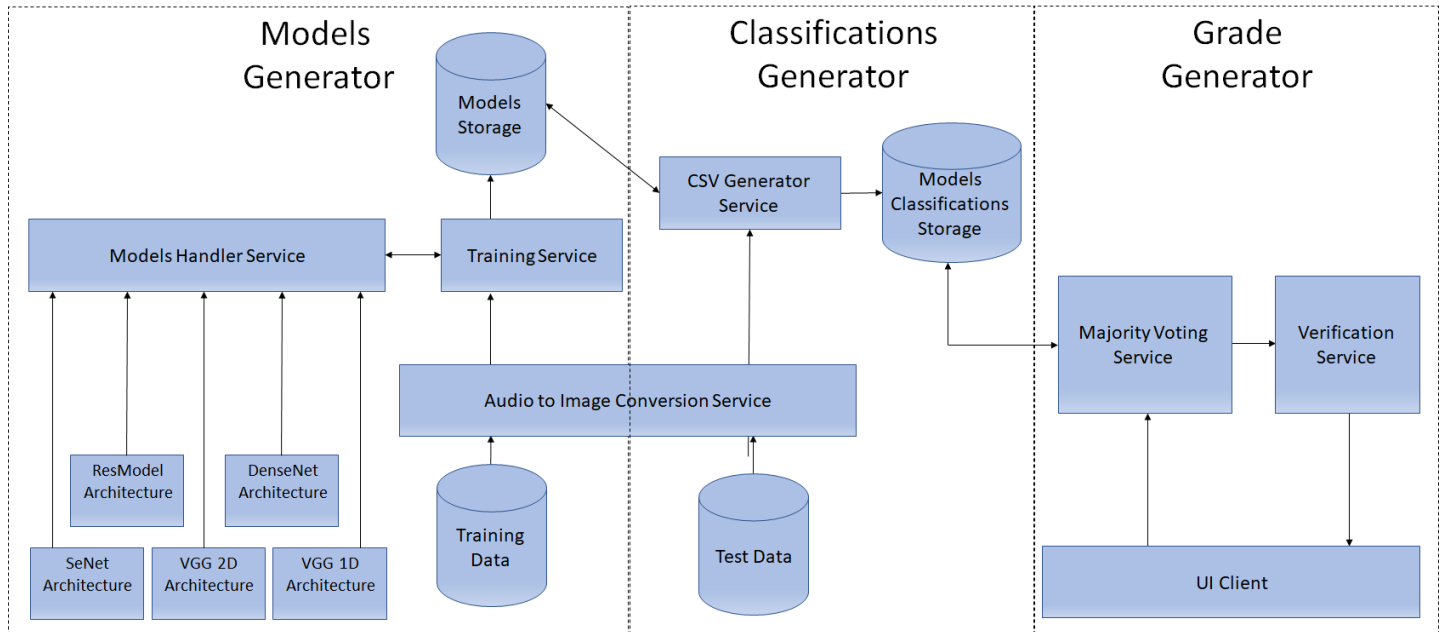
מגבלות והנחות המערכת

- קבצי הקלט של המערכת הינם מסוג WAV.
- מערכת הפעלה תומכת python גירסא 3.
- סה"כ אחוזי המשקלים אשר מוזנים ע"י המשתמש בשלב ה Majority Voting הינו 100.

עיצוב מערכת תוכנה

תכנון ועיצוב המערכת בוצע בגישת OO תוך כדי שימוש במספר דיאגרמות UML כדוגמת דיאגרמות מחלקות ותרשימי רצף.

ארכיטקטורה



איור 7 – ארכיטקטורת המערכת

המערכת מורכבת משלושה יישומים עיקריים (איור 5) :

1. Models Generator – תפקידו של היישום הוא לקבל קבצי קול ממאגר האימונים כקלט ולייצר מודלים (אשר אומנו בעזרת למידה עמוקה) ומסווגים קבצי אודיו על פי תוכנם לאחת מן האפשרויות הבאות :
 - Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Silence, Unknown
2. Classifications Generator – תפקידו של היישום הוא לקבל קבצים ממאגר הבדיקות כקלט ולהשתמש במודלים אשר נוצרו ביישום Models Generator על מנת לסווג את קבצי הבדיקה לאחת מן המילים החוקיות ע"פ תוכנם. סיווג זה נשמר כקובץ CSV נפרד עבור כל מודל בנפרד.
3. Grade Generator – תפקידו של היישום הוא לספק ציון כולל בין 0 ל-100 אשר משקף את יכולת הסיווג המערכתית הכוללת.

- **Training Data Storage**

כאן מאוחסן כל המידע הנדרש ל-Training, מדובר על 64,732 קבצי אודיו מסוג wav המקוטלגים לפי תוכן המילה שנאמרת בהם לתיקיות שונות, המילים האפשריות (ע"פ חוקי הפורמט של התחרות של Kaggle) הן:

Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Silence, Unknown ○

- **Test Data Storage**

כאן מאוחסן כל המידע הלא מקוטלג הנדרש על מנת לבדוק את איכות המערכת שפותחה בפרויקט, מדובר על 158,536 קבצי אודיו מסוג wav.

- **Audio to Image Conversion Service**

שירות הממיר תוכן קבצי אודיו מסוג wav לפורמט תמונה. ההמרה מבוצעת תוך שימוש בספריית Librosa לפייתון הכתובה בקוד פתוח. ספרייה זו מבצעת המרה מקובץ קול ל-Mel Spectrogram למרחב התמונה. לאחר ההמרה הספרייה תחזיר את התמונה המייצגת את קובץ הקול במרחב התמונה. [10]

שירות זה, מזין בתמונות מסוג Mel Spectrogram, הן את שירות האימון (Training Service) בתמונות להתאמן עליהם לצורך יצירת מסווגים (כפי שיתואר בהמשך) והן את השירות אשר מסווג ושומר את פלט הסיווג (CSV Generator Service).

אומנם האפשרות לשמור את קבצי התמונה הנוצרים בשירות זה בדיסק הקשיח המקומי קיימת, אך אין צורך בכך – כל השימוש בקבצי התמונה הנוצרים בשירות זה נעשה on-the-fly והתמונה המומרת נמצאת בזיכרון ה-RAM כאשר יש צורך בה. בסוף תהליך האימון והסיווג התמונה שהומרה תמחק מזיכרון ה-RAM.

• **Models Handler Service**

שירות זה מספק את ארכיטקטורות המודלים איתם המערכת מאמנת את המסווגים לשירות האימונים (Training Service).

ארכיטקטורות המודלים השונות מקושרות לשירות זה אך הוא היחיד המתקשר איתן מתאמי פשוטות ומודולריות. ניתן בעתיד להוסיף בקלות מודלים מארכיטקטורות חדשות. במקרה זה, יש לחבר את אותן ארכיטקטורות לשירות זה וכל שאר המערכת תישאר בדיוק כפי שהייתה.

במידה והתקבלה בקשה בשירות האימונים לאמן מסווג כלשהו מארכיטקטורה מסוימת, שירות האימונים ייפנה לשירות Models Handler בבקשה לקבל את ארכיטקטורת המודל לצורך אימון מסווג חדש.

ארכיטקטורות המודלים המקושרות לשירות זה:

- VGG 2D Architecture – מבנה הארכיטקטורה מפורט בנספח ו'
- VGG 1D Architecture – מבנה הארכיטקטורה מפורט בנספחים ז' ו-ח'
- SeNet Architecture – מבנה הארכיטקטורה מפורט בנספח ט'
- ResNet Architecture – מבנה הארכיטקטורה מפורט בנספח י'
- DenseNet Architecture – מבנה הארכיטקטורה מפורט בנספח יא' (הארכיטקטורות השונות וההבדלים ביניהם יתוארו בהמשך)

• **Training Service**

שירות זה הינו לב המערכת, מטרתו הינה לאמן את שלל המסווגים של המערכת. בכל פנייה לשירות זה יש לציין את ארכיטקטורת המודל עליה המסווג הנוכחי מיועד להתבסס. בשירות זה מיוצרים המודלים השונים המאומנים על פי:

1. ארכיטקטורת המודל עליו המערכת מתאמנת ובונה מסווג, ארכיטקטורה זו הגיעה משירות ניהול המודלים (Models Handler Service).

ארכיטקטורה זו נקבעת ע"פ נתוני ההרצה הנוכחיים של שירות האימונים.

2. תמונות במרחב ה Mel Spectrogram אשר הומרו מקבצי הקול לאימונים [11] בעזרת שירות המרת הקול לתמונה (Audio to Image Conversion Service)

אין הגבלה על מספר המודלים של רשתות הנוירוניים בהם ניתן להשתמש, כל עוד הארכיטקטורה המתארת את תוכן המודל המבוקש קיימת במערכת ומחוברת לשירות ניהול המודלים. לאחר ניסוי וטעייה נמצאו 6 המודלים אשר לאחר אימוןם מתקבל ציון הסיווג הטוב ביותר.

מודלי רשתות הנוירוניים העמוקות הנוצרים בתהליך זה יופעלו בשלב הסיווג, כאשר מטרתם הסופית הינה לסווג בהצלחה את קבצי הבדיקה. בשירות זה נעשה שימוש בספריה לאימון מודלים של PyTorch שייצרו בפייסבוק, זוהי ספריה חדשנית ונוחה לשימוש בסביבת python.

• **Models Storage**

מודול זה מאחסן את המודלים אשר נוצרו בשלב ה Training. המודלים הללו הינם בפורמט הסטנדרטי של PyTorch (בעלי סיומת "pth"), כמו כן בשלבים המאוחרים יותר, ניתן יהיה לסווג בעזרתם את קבצי הבדיקה השונים.

- **CSV Generator Service**

מטרתו של שירות זה הינה לייצר קבצי CSV (קובץ לכל מודל רשת נוירונים עמוקה) המכילים את תוצאות הסיווג של כל רשת עבור כל אחד מקבצי הבדיקה.
קובץ ה CSV יכול שורות כמספר קבצי הבדיקה וכמו כן שתי עמודות :
העמודה הראשונה תכיל את שם קובץ הבדיקה
העמודה השנייה תכיל את תוצאות הסיווג של המודל הנוכחי עבור אותו קובץ
לדוגמא :

עבור המודל "vgg1d_mel", נוצר הקובץ הבא : "vgg1d_mel_prediction.csv",
כך נראה תוכנו בשורות 60 ו 61 :

```
up      clip_0018ff8e9.wav  
  
stop    clip_001a5ce9c.wav
```

כלומר : עבור קובץ הבדיקה clip_0018ff8e9.wav, המודל vgg1d_mel שנוצר, מזהה את המילה up.
עבור קובץ הבדיקה clip_001a5ce9c.wav, המודל vgg1d_mel שנוצר, מזהה את המילה stop.

ההשלכות ממידע זה הן כי כצפוי וע"פ ההנחה המקורית שממנה יצאה המערכת לדרך – אין מודל אחד שמסוגל לסווג את כל קבצי הבדיקה נכון – המערכת צריכה להיבנות (ונבנתה) על בסיס כמה מודלים שונים אשר לכל אחד חוקות וחולשות שונות (הנובעות מהארכיטקטורה ממנה נוצר אותו מודל), כך שרק שילוב בין הרבה מודלים כאלו יוכל לתת תוצאות סיווג נכונה עבור כל (או רוב) קבצי הבדיקה.

- **Models Classifications Storage**

כאן מאוחסנות תוצאות הקטלוג של כל אחד מהמודלים שנבנו בצורת CSV (כאשר לכל מודם קיים קובץ CSV מתאים).

- **Majority Voting Service**

מטרת שירות זה הינה לקבל את תוצאת הסיווג הסופית המתקבלת מהמערכת עבור כל אחד מקבצי הבדיקה השונים. תוצאה סופית זו הינה תוצר של פלט הסיווג של כלל המודלים עבור כל קובץ וביצוע ממוצע משוכלל עבור פלט זה ע"פ משקלי הקלט אשר הוזנו.
הצורך בשירות זה נובע מארכיטקטורת המערכת - מכיוון שהמערכת מכילה יותר ממודל רשתות נוירונים עמוקות אחד וכיוון שמודלים שונים יכולים להניב סיווגים שונים. על מנת לקבל סיווג אחד ויחיד עבור קובץ בדיקה מסוים, יש צורך בביצוע ממוצע משוקלל על שלל תוצאות הסיווגים של המודלים השונים שבמערכת אשר משקלם גדול מ 0.
שירות זה מקבל כקלט את המשקלים השונים (האחוזים השונים) של מודלי רשתות הנוירונים העמוקות השונות אשר יילקחו בחשבון בעת בחירת הסיווג הסופי של כל אחד מקבצי הבדיקה במערכת.
לדוגמא, בהינתן :

- קובץ אודיו לדוגמא בשם sample.wav.

- 3 מודלים לדוגמא בשמות model1,model2,model3.

- 3 משקלים באחוזים עבור המודלים הנ"ל:

model1	49%
model2	11%
model3	40%

- הסיווגים השונים המתקבלים מהמודלים האלו עבור קובץ הדוגמא הינם:

model1	Up
model2	Go
model3	Go

הסיווג אשר יתקבל מתוצאות הממוצע המשוכלל עבור הנתונים הנ"ל הינו Go שכן סיווג זה מחזיק ב-51% לעומת הסיווג Up אשר מחזיק ב-49%.

• Verification Service

מטרת שירות הverification הינה להשוות את תוצאות הסיווג הסופיות אשר התקבלו מהמערכת עבור כל אחד מקבצי הבדיקה למול תוצאות הסיווג הנכונות עבור כל אחד מקבצי הבדיקה [7]. שירות זה מחזיר ציון באחוזים בין 0 ל100 המתאר את איכות ונכונות תוצאות המערכת.

• UI Client

ממשק הUI נועד לטעינת אחוזי המשקלים השונים לכל מודל אשר יוזנו כקלט לשירות ה Majority Voting מאחורי הקלעים (ב Backend). לאחר הזנת המשקלים השונים למודלים השונים, ממשק המשתמש יציג את התוצאה הסופית (ציון המערכת) אשר התקבלה בשקלול של כל המערכת (הפרטים המלאים לגבי אופן קבלת ההחלטה על הציון נמצאים בחלקים המתאר את שלבי ה Majority Voting וה Verification בהמשך המסמך). הממשק כתוב בhtml ומתקשר באמצעות ארכיטקטורת Flask עם ספריות הpython שבפרויקט. [11]

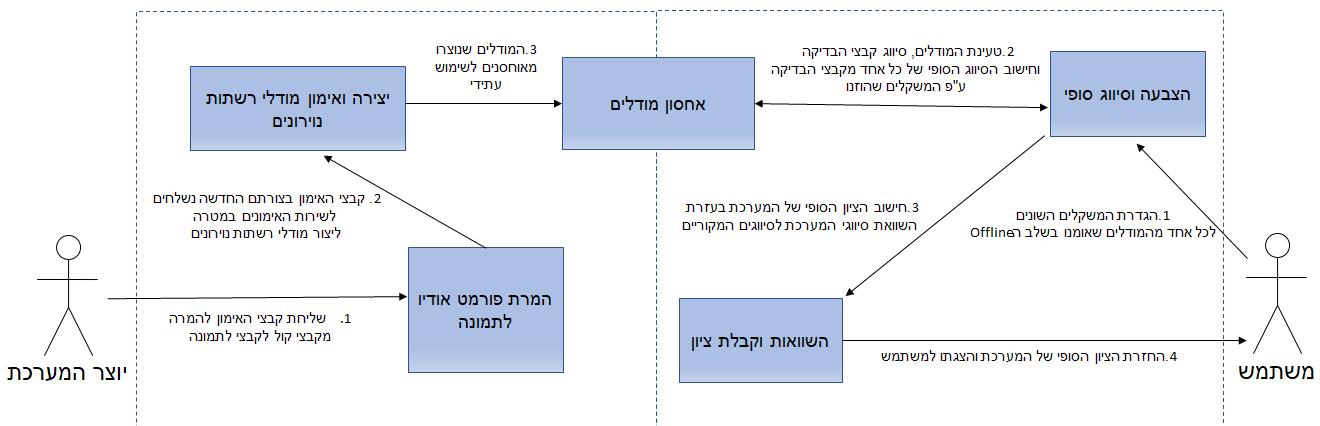
תיאור תהליכים עיקריים

קיימים שני תהליכים עיקריים במערכת(איור 4) :

- Offline – יצירת מודלים של רשתות נוירונים עמוקות מתאימים ועבודה מול קבצי האימון (אימון המערכת) .
- Online – החזרת תוצאות איכות הסיווג של המערכת בעבודה מול קבצי הבדיקה (תשאול המערכת).

תהליך אימון המערכת ב Offline

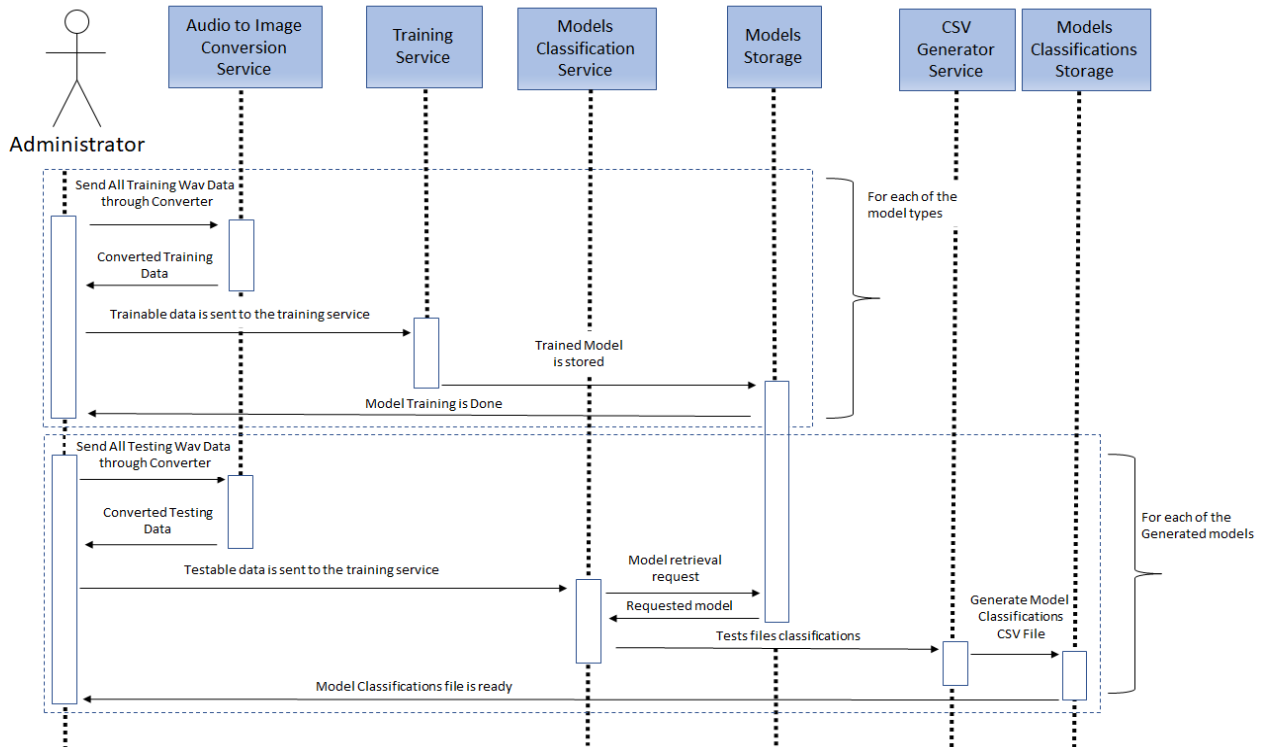
תהליך תשאול המערכת ב Online



איור 8 – התהליכים העיקריים במערכת

תהליך אימון המערכת

באיור 9 מודגמת דיאגרמת רצף עבור תהליך אימון המערכת



איור 9 – דיאגרמת רצף עבור תהליך אימון המערכת

הסבר תרשים הרצף של תהליך אימון המערכת –

האחראי ליצירת המערכת שולח את כל קבצי האימון בצורתם הראשונית כקבצי אודיו מסוג wav לשירות הממיר אותם לקבצי תמונה אשר בעזרתם ניתן ללמד את מערכת הלמידה העמוקה. [9]

לאחר מכן המידע שנוצר נשלח לשירות האימונים שתפקידו לייצר מודלים בעזרת ספריית PyTorch מבית פייסבוק. [10]

כאשר שלב האימונים מסתיים (שלב זה כפי שצוין קודם, יכול להמשך זמן רב – בין שעות במחשב חזק לימים ושבועות במחשב ביתי), יאוחסן המודל החדש שנוצר בתיקייה המכילה את מאגר המודלים של המערכת.

לאחר ניסיונות שונים ושיטות בפורומים המיועדים לפתרון בעיית סיווג הפקודות הקוליות, הגעתי למסקנה כי עבור אימון של 6 מודלי רשתות נוירונים ספציפיים (עליהם אפרט בהמשך) המערכת מצליחה להחזיר את הסיווג הנכון עבור המספר הרב ביותר של קבצי קול מתוך מאגרי הבדיקה והאימון בצורה הטובה ביותר – כלומר עבור 6 המודלים הללו מתקבלות התוצאות הטובות ביותר.

ארכיטקטורות המודלים וכמותם הגיעו לאחר לא מעט ניסוי וטעיה, הן שלי והן של מפתחים אחרים שניסו לפתור את בעיית סיווג הפקודות הקוליות.

מודלים אלו נבדלים זה מזה במודל הבסיס עליו הם מתאמנים כפי שאתאר בהמשך.

לאחר בניית 6 סוגי המודלים השונים בצורה המתוארת יחל החלק בו המערכת מייצרת 6 קבצי CSV המכילים את הסיווגים של כל מודל.

ראשית המערכת ממירה את קבצי הבדיקה הנתונים באותה הדרך בה המירה את קבצי האימון קודם לכן (מקבצי אודיו מסוג WAV למידע אשר מודל הרשת העמוקה שבנינו יכול לסווג – כלומר לפורמט של קבצי תמונה). [9]

עבור כל מודל המערכת תייצר את קובץ ה CSV שלו בצורה הבאה :

בהינתן קבצי הבדיקה המומרים למרחב התמונה, המודל יתושאל לגבי הסיווג שלו עבור כל אחד מקבצי הבדיקה. המודל יחזיר עבור כל קובץ בדיקה מומר את הסיווג המתאים לקובץ זה.

כל שורה בקובץ ה CSV תכיל את שם קובץ הבדיקה המקורי ואת הסיווג שלו ע"י המודל הנוכחי.

כאמור האפשרויות השונות לסיווגים הינן המילים : Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Silence, Unknown.

קבצי ה CSV עבור המודלים השונים נשמרים בתיקיה אשר תשמש בשלב הבא כמאגר הסיווגים של המערכת.

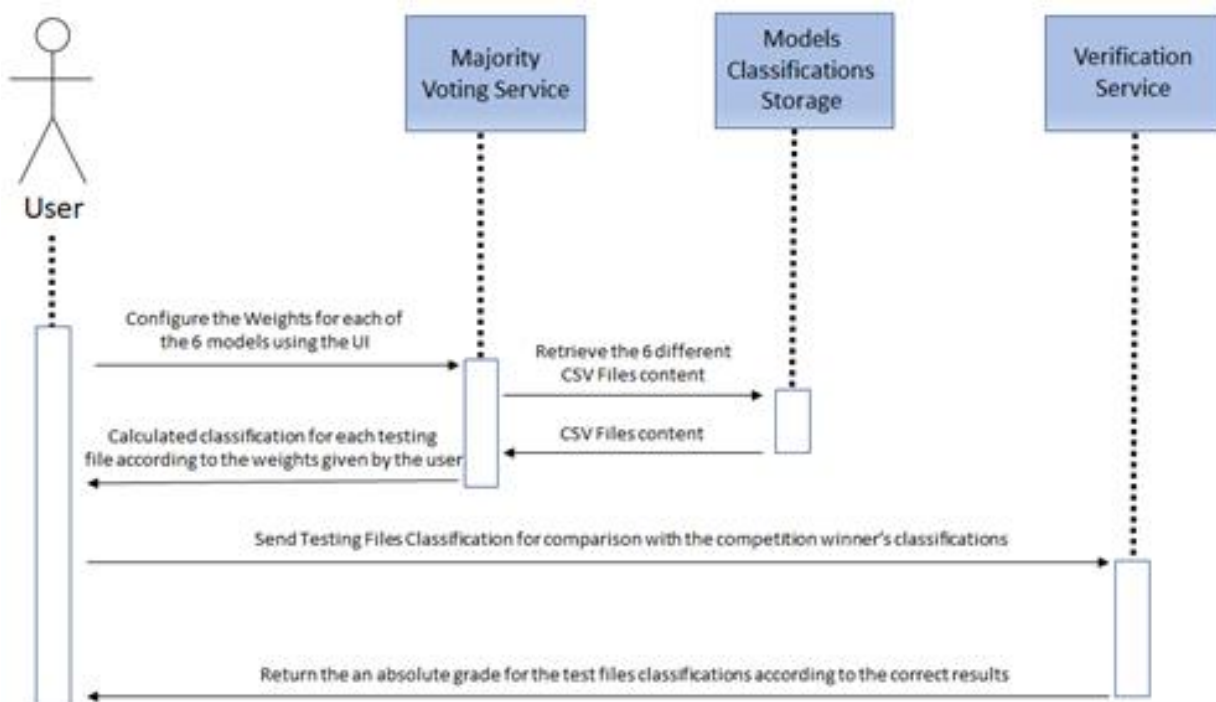
דוגמא לכמה שורות מתוך קובץ ה CSV שנוצר עבור מודל ה VGG1D אשר סיווג תמונות מסוג Mel Spectrograms :

unknown	clip_00147bbb6.wav
yes	clip_0014ea384.wav
no	clip_0014ed3d5.wav
no	clip_0014f2f18.wav
unknown	clip_00150496f.wav

unknown	clip_0015fa156.wav
up	clip_00169a7f7.wav
go	clip_0017365f5.wav
unknown	clip_0017714af.wav

תהליך תשאול המערכת

באיור 10 מודגמת דיאגרמת רצף עבור תהליך אימון המערכת



איור 10 – דיאגרמת רצף עבור תשאול המערכת

הסבר דיאגרמת הרצף עבור תהליך תשאול המערכת –

תשאול המערכת מתבצעת דרך ממשק המשתמש.

ממשק המשתמש יישלחו אחוזי המשקלים של כל אחד מששת הסיווגים שנוצרו מששת המודלים בשלב הקודם אל שירות Majority Voting.

שירות Majority Voting יבצע ממוצע משוקלל לפי תוצאות הסיווגים השונים של כל מודל וייתן לכל סיווג כזה את המשקל שהמשתמש הזין לאותו המודל.

לדוגמא :

בהינתן :

- קובץ אודיו לדוגמא בשם sample.wav.
- 6 מודלים לדוגמא בשמות M1, M2, M3, M4, M5, M6.
- 6 משקלים באחוזים עבור המודלים הנ"ל:

M1	25%
M2	20%
M3	15%
M4	15%
M5	15%
M6	10%

- הסיווגים השונים המתקבלים מהמודלים האלו עבור קובץ הדוגמא הינם :

M1	Up
M2	Go
M3	Go
M4	Down
M5	Go
M6	Go

הסיווג אשר יתקבל מתוצאות הממוצע המשוכלל עבור הנתונים הנ"ל הינו Go שכן סיווג זה הינו החזק ביותר ומחזיק ב-60% מהמשקל הכולל. (לעומת הסיווג Up אשר מחזיק 25% מהמשקל הכולל והסיווג Down שמחזיק ב-15% מהמשקל הכולל).

פעולה זו תתבצע עבור כל אחד מ-158,536 קבצי הבדיקה ובסופה יוחזר סיווג המערכת עבור כל אחד מקבצי הבדיקה.

בשלב הבא יישלחו הסיווגים שהתקבלו לשירות נוסף אשר משווה את סיווגי המערכת לסיווגים הנכונים של קבצי הבדיקה (אשר התקבלו בעזרת המערכת של המנצח בתחרות של Kaggle).

לאחר ההשוואה שירות זה יחזיר למשתמש את הציון אשר המשקלים שהוא בחר נותנים ביחס לסיווגים של המנצח בתחרות. [7]

המערכת מחשבת את הציון באופן הבא :

$$Grade = \frac{\sum_{i=0}^{(test\ data\ files\ amount)} \begin{cases} SystemClassification[i] == CorrectClassifications[i] : 1 \\ else : 0 \end{cases}}{(test\ data\ files\ amount)}$$

הציון הכולל של המערכת מחושב לפי סכום הפעמים שהסיווג שהמערכת נתנה היה זהה לסיווג שהמנצח בתחרות נתן חלקי כמות הקבצים הכוללת לסיווג, בצורה זו ניתן לכמת ציון בין 0 ל 100 אשר משקף בצורה טובה את טיב המערכת.

תהליך המרת קבצי אודיו לתמונה

קלט: קבצי אודיו בפורמט wav.

תהליך: המרת קבצי האודיו מסוג wav למרחבים אשר ניתנים לאימון וקטלוג ע"י רשתות למידה. בספרות [3,4] מתוארים שלושה סוגי מידע עיקריים של מרחבי מידע עליו רשתות נוירונים עמוקות יכולות להתאמן ולהשיג תוצאות טובות. בשניים מהסוגים הללו המערכת שפותחה עורכת שימוש:

- המרה למישור התמונה מסוג Mel Spectrogram, זהו המרחב העיקרי בו המערכת משתמשת.
- המרה לRaw Data כלומר - טעינת קובץ האודיו לזיכרון באמצעות python ושליחת ייצוגו בזיכרון כפי שהוא גם לצרכי אימון המודל וגם לצרכי סיווג באמצעות מודל קיים.

שתי ההמרות בוצעו בעזרת ספריית הקוד הפתוח Librosa [9].

אימון רשתות נוירונים עמוקות

הדרך האפקטיבית ביותר לאימון מודל רשת נוירונים עמוקה על קלט חדש הינו אימון של רשת קיימת בעלת משקלים שונים על קבצי הקלט לאימון הקיימים, גישה זו נקראת Transfer Learning [8].

במאמר פורץ הדרך CNN Features off-the-shelf: an Astounding Baseline for Recognition [21] שפורסם ע"י חוקרים מאוניברסיטת שטוקהולם, הוכח כי רשת אשר למדה לסווג בעיה מסוימת יכולה בתהליך המרה בסיסי ופשוט לעבור התאמה לפתרון בעיית סיווג אחרת.

גישה זו מצליחה בזכות העובדה כי רשת נוירונים עמוקה מכילה בתוכה שכבות רבות של אבסטרקציה ומידול עבור הקלט שקיבלה. שכבות אלו מתמקדות בתכונות שונות וחלקיות מהתמונה המקורית. תכונות אלו נקבעות ע"פ ארכיטקטורת כל רשת ואינן תמיד תכונות הנראות לעין אנושית – אלו מאפיינים בתמונות שהרשת למדה אשר התמקדות בהן עזרו לרשת לשפר את איכות הסיווג שהיא מחזירה.

הגישה הרווחת בקרב חוקרי הלמידה העמוקה אשר קיבלה חיזוק בין היתר במאמר המדובר הינה שהאבסטרקציה בשכבות השונות של רשת נוירונים הינה גנרית ודומה מאוד גם עבור בעיות ממרחבים שונים, ההשלכות לגבי המערכת שנבנתה לצורך הפרויקט הינה כי ניתן להשתמש ברשתות נוירונים עמוקות אשר אומנו למטרות אחרות שאינן סיווג אותות קוליים על מנת לסווג אותות קוליים – וכך באמת נבנתה המערכת.

סוגי ארכיטקטורות רשתות נוירונים עמוקות המשמשות לאימון המערכת

על מנת לבחור את הארכיטקטורה של רשתות הנוירונים השונים עליהם תתבסס המערכת התבצע תחקור ומעבר על פורום התחרות ועל פרויקטים שונים אשר זכו במקומות גבוהים בתחרות. [12]

התנאים הבסיסיים לבחירת הארכיטקטורות במערכת שפותחה:

1. הארכיטקטורות שתיבחרנה חייבות להיות מוכחות שביכולתן לזהות ולסווג בצורה איכותית תמונות המכילות מידע מהמרחב הספקטרלי.
2. הארכיטקטורות שתיבחרנה חייבות להיות מוכחות ובעלות מאגר רחב של תמיכה וידע ברשת, שכן עבודה ברשתות נוירונים עמוקות גוררת בעיות חישוביות אשר לא נתקלים בהם בפרויקטים אחרים בעולם התוכנה. לכן חשיבות מאגרי המידע, הפורמים והתמיכה ממקורות שונים ברשת היוו תנאי הכרחי לבחירת מודל לשימוש במערכת.

בסופו של תהליך החקירה הוחלט להשתמש במודלים הבאים כבסיס למערכת -

VGGNet

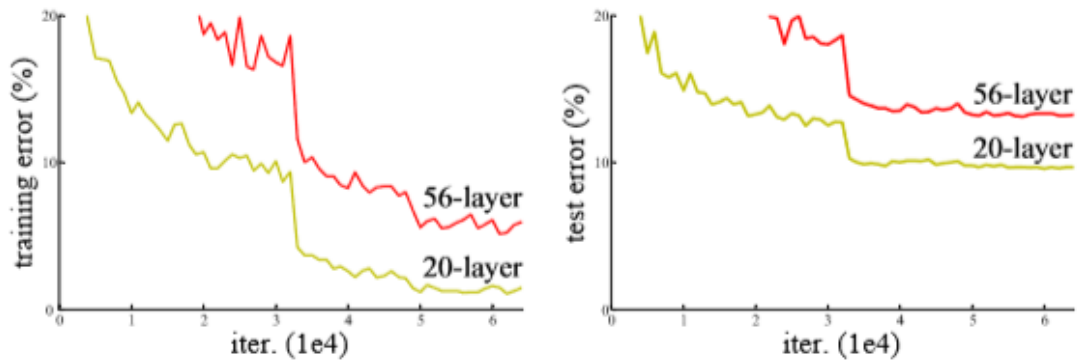
הוצגה לראשונה ב-2014 ע"י Karen Simonyan ו Andrew Zisserman [1] המאמר המתאר את הרשת הזו זכה ליותר מ-10,000 ציטוטים מאז שעלה לאוויר. המאמר הני"ל מתייחס לחלק חשוב מאוד בארכיטקטורה של רשתות למידה עמוקה – **עומק**. [13]

הארכיטקטורה של רשת זו היוותה התקדמות טכנולוגית מרשימה בשנת 2014, שכן ברשת זו נעשה שימוש בפילטרים קטנים מאוד בגודל 3×3 פיקסלים לתמונות הקלט בשכבות הפנימיות שברשת, פעולה ששיפרה את דיוקה ביחס לרשתות שהיו קיימות עד אותה העת בצורה משמעותית.

ResNet

הוצגה לראשונה ב-2014 ע"י Kaiming He ממיקרוסופט, המאמר המתאר את הרשת הזו זכה ליותר מ-8,000 ציטוטים מאז שעלה לאוויר. [14]

הייחודיות של רשת זו מגיעה בדמות **כמות שכבות גדולה** משמעותית מרשתות שקדמו לה וכמו כן בניסוח מחדש של **שכבות הביניים ברשת כפונקציות דינמיות המתאימות את עצמן לקלטי השכבות שברשת**. זאת בשונה מרשתות שהיו עד אותה העת אשר הכילו רשתות ביניים סטטיות.



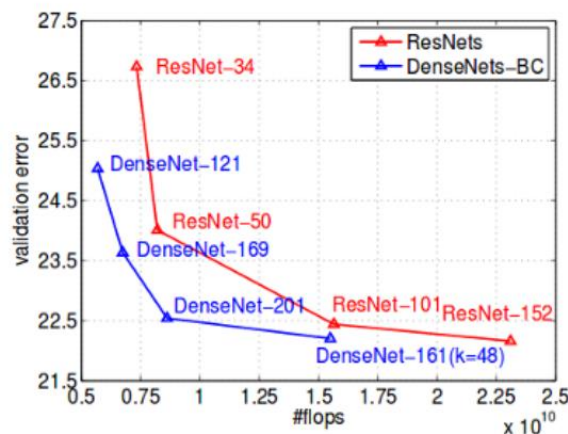
איור 15 – השוואת כמות איטרציות לאימון רשת למול איכות הזיהוי של הרשת תוך כדי האימון [14]

איור 15 נלקח מהמאמר אשר הציג את הרשת לראשונה [14], ניתן לראות את איכות הרשת המתקבלת תוך כדי אימון כתלות באיטרציות האימונים. בירוק מתוארת רשת ResNet ובאדום רשת CiFarn אשר הייתה נפוצה באותה תקופה, ניתן לראות בבירור כי רשת ResNet מגיעה לתוצאות אימון טובות הרבה יותר מהר ובהרבה יותר יעילות מהרשת המתחרה.

DenseNet

הוצגה לראשונה ב־2016 ע"י Gao Huang, Zhuang Liu, Kilian Q. Weinberger, המאמר המתאר את הרשת הזו זכה לקרוב ל-1000 ציטוטים מאז שעלה לאוויר, שמה של הרשת מתייחס ל DenseNet Connected convolutional [15].

רשת זו מתבססת על מחקרים המראים כי **רשתות בעלות חיבורים קצרים בין שכבות הקלט לשכבות הפלט יכולות להגיע לרמת דיוק גבוהה מרשתות בעלות חיבורים ארוכים יותר.**



איור 16 – השוואת כמות איטרציות לאימון רשת למול איכות הזיהוי של רשתות DenseNet ו ResNet תוך כדי האימון על מאגר הסיווג ImageNet [15]

באיר 16 מוצגת השוואה בין DenseNet (בכחול) ResNet (באדום) תוך כדי אימון על מאגר הסיווג ImageNet. ניתן לראות כי רשת DenseNet מקדימה את רשת ResNet בכמה איטרציות מבחינת איכות הסיווג. [15]

SeNet

הוצגה לראשונה ב-2017 ע"י Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, המתאר את הרשת הזו זכה לקרוב ל-100 ציטוטים מאז שעלה לאוויר, רשת זו נבחרה למנצחת האתגר של ImageNet לשנת 2017. שמה של הרשת מתייחס ל Squeeze and Excitation [16].

רשת זו מציגה שיפור משמעותי בתלויות הפנימיות שבין שכבות הרשת כמעט ללא עלות. הגישה עליה מתבסס היתרון המשמעותי של רשת זו הינו הוספת פרמטרים בכל שכבה ברשת כך שבעזרת התצורה הנכונה וההגדרות הנכונות, רשת זו תוכל להסתגל לקליטים שונים בשכבות שונות ובכך לשפר את איכות הזיהוי משמעותית.

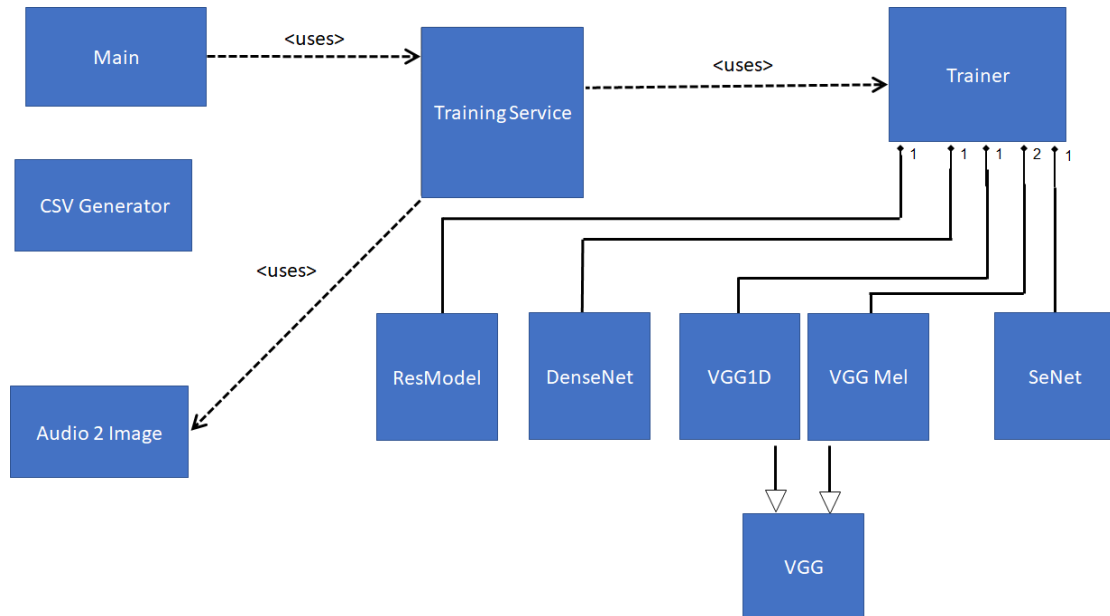
קליטים לרשתות הנוירוניים העמוקות שבמערכת שפותחה הם:

- **ResNet** – הקלט לרשת זו הינו תמונות מסוג MEL SPECTROGRAMS הנוצרות מקובץ WAV.
- **SeNet** – הקלט לרשת זו הינו תמונות מסוג MEL SPECTROGRAMS הנוצרות מקובץ WAV.
- **DenseNet** – הקלט לרשת זו הינו תמונות מסוג לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV.
- **VGG2d** – הקלט לרשת זו הינו תמונות מסוג MEL SPECTROGRAMS הנוצרות מקובץ WAV.
- **VGG1d** – הקלט לרשת אחת כזו הינו תמונות מסוג MEL SPECTROGRAMS הנוצרות מקובץ WAV.
- **VGG1d on raw data** – הקלט לרשת השנייה מסוג VGG1d הינו Raw Data הנוצר מקבצי ה WAV ללא המרה נוספת.

דיאגרמת מחלקות

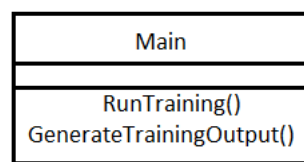
המחלקות השונות בשלב האימון

באיור 17 מודגמת דיאגרמת מחלקות העל ליישום אימון המערכת



איור 17 – דיאגרמת מחלקות על לאימון המערכת

Main



איור 18 – תוכן המחלקה Main

זוהי נקודת ההתחלה לאימון רשתות הנוירונים העמוקות, המחלקה הראשית מכילה 2 פונקציות עיקריות:

1. הפעל את אימוני הרשתות השונות עבור כל אחת מרשתות הנוירונים העמוקות אותן נרצה לאמן (6 במספר – 6 הסוגים נקבעו לאחר לא מעט ניסוי וטעייה יחד עם מעבר המלצות בפורומים השונים עד גיבוש סוגי וכמות המודלים המומלצים למערכת מסוג זה)

התוצאה מהרצה כזו היא היווצרות המודלים(בעזרת שימוש בשירותים ממחלקות האימון החיצוניות למחלקה זו) ושמירת הפלט שלהם לתיקיית המודלים הצמודה לתוכנה. בכל עת מאומנת רשת אחת בלבד, לא אומנו כמה רשתות במקביל שכן האימון של כל רשת דורש משאבי מערכת גדולים מאוד(גם CPUs וגם GPUs). הפלט מפעולת הפונקציה הנ"ל הינו קבצים פיזיים מסוג `pth` המכילים עבור את תוכן של 6 הרשתות השונות לאחר נוצרו ואומנו כהלכה.

2. ייצור קבצי פלט מסוג CSV בעזרת מחלקת `CSV_Generator` עבור החיזוי של כל מודל עבור כל אחד מקבצי `test`.
 בסיום הרצת הפונקציה הזו הפלט הנוצר הוא 6 קבצי `csv` המכילים את החיזויים השונים לכל קבצי הבדיקה עבור כל אחת מ-6 הרשתות שאומנו.

Audio2Image

Audio2Image
<code>add_noise</code> <code>background_noises</code> <code>is_id : bool</code> <code>label_words_dict</code> <code>mode</code> <code>n_silence : int</code> <code>preprocess_fun</code> <code>preprocess_param : dict</code> <code>resize_shape : NoneType</code> <code>sr : int</code> <code>wav_list</code>
<code>get_mix_noises()</code> <code>get_noisy_wav()</code> <code>get_one_noise()</code> <code>get_one_word_wav()</code> <code>get_silent_wav()</code> <code>timeshift()</code>

איור 19 – תוכן המחלקה *Audio2Image*

מחלקה זו נלקחה מספריית הקוד הפתוח `Librosa`.

מטרת המחלקה היא להמיר קבצי הקול הנתונים למידע אותו ניתן להזין לארכיטקטורת `Pytorch` על מנת להתממשק עם הרשתות השונות, נעשה שימוש במחלקה זו גם בשלב האימון וגם בשלב הסיווג.

מחלקה זו מהווה חלק חשוב במערכת, שכן רק בעזרת ההמרות המתקבלות ממחלקה זו ניתן לאמן את המודלים השונים ולייצרם וכמו כן רק בעזרת ההמרות המתקבלות ממחלקה זו ניתן לבדוק את טיב המודלים השונים.

Training Service

TrainingService
<code>run_training()</code> <code>train_DenseNet_mel_data()</code> <code>train_ResNet_mel_data()</code> <code>train_SeNet_mel_data()</code> <code>train_all_models()</code> <code>train_vgg1D_mel_data()</code> <code>train_vgg1D_raw_data()</code> <code>train_vgg2D_mel_data()</code>

איור 20 – תוכן המחלקה TrainingService

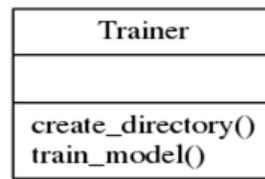
מחלקה זו אחראית על אימון המודלים השונים. הפונקציות אותן היא מכילה הן :

1. פונקציה לאימון כל המודלים בזה אחר זה אשר נקראת מבחון (ממחלקת Main)
2. 6 פונקציות לאימון ששת המודלים אשר בכל אחת מוגדר סוג המודל וסוג הקלט אותן צריך לאמן על מנת לייצר את המודל המבוקש
3. פונקציית אימון כללית בהינתן סוג המודל והגדרותיו מאחת מ-6 הפונקציות שתוארו בסעיף הקודם.

מחלקה זו משתמשת במשתנה מוגדר מראש הנקרא Bagging (בעל הערך 4) -

בעולם הלמידה העמוקה bagging מתאר Bootstrap aggregating – כלומר בהינתן קבוצת קבצי אודיו לאימון D בגודל N, bagging מייצר 4 קבוצות כאלו D1,D2,D3,D4 כולן בגודל N.

ע"י דגימה מ D המקורית תוך כדי החלפת סדר האיברים בקבוצה בצור רנדומלית (shuffling) – לאחר מכן בעת בניית כל אחת מהרשתות, הרשתות ייבנו עם כל הקבוצות שנוצרו גם כן תוך כדי בחירה רנדומלית. ע"פ הוגה הגישה ליאו בריימן, "שימוש בBagging מוביל לשיפור עבור הליכים לא יציבים" הכוללים בין היתר אימון רשתות עמוקות. [17]



איור 21 – תוכן המחלקה Trainer

מחלקה זו מאמנת את כל אחד מהמודלים של 6 סוגי רשתות הנוירונים העמוקות בנפרד, עבור כל מודל היא מבצעת את הפעולות הבאות:

1. אתחול על פי סוג הרשת וההגדרות אותן יש לאמן על אותה הרשת.

פרטים אלו מוגדרים מראש במחלקת Main, לכל אחת מהרשתות השונות, מידע זה מועבר ממחלקת Main למחלקה הזו ונטען בשלב זה לכל אחת מהרשתות השונות.

2. אימון הרשת ע"פ המימוש המקובל בעזרת ספריית האימונים של Pytorch.

מכיוון שהרשתות שנבחרו בקפידה עבור המערכת הינן רשתות נפוצות מאוד בעולם הלמידה העמוקה, קיימים מימושים מוכנים מראש שלהן בPytorch. כך שבשלב זה ובהתאם לקריאה מהTrainingService מחלקה זו מתאימה את ההגדרות לאימון והמימוש הנכון של הרשת שהתבקשה לאמן על מנת להתחיל בתהליך אימון רשת הנוירונים המבוקשת.

3. שמירת קובץ הפלט של הרשת.

רשתות אשר אומנו בעזרת Pytorch נשמרות כקובץ בעל סיומת .pth. בעתיד המערכת תוכל לתשאל את הקובץ הנ"ל על מנת לקבל את סיווג הרשת עבור קובץ תמונה כלשהו הנמצא במרחב הספקטרי.

הרשתות השונות במימוש פיתוני - VGG, SeNet, DenseNet, ResNet

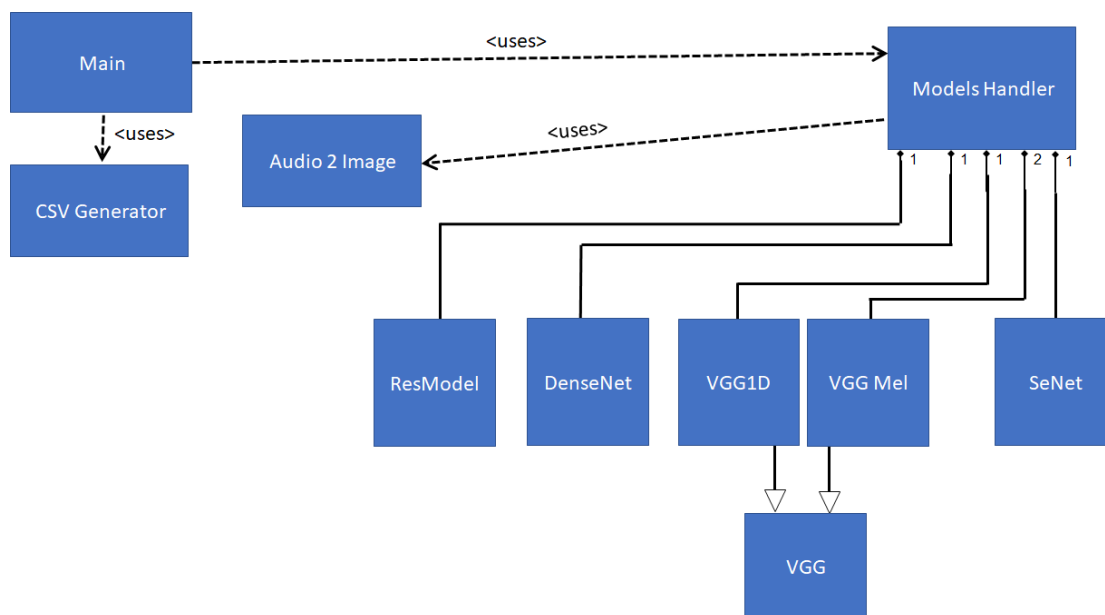
על מנת להתאים את המימוש בpython של רשתות נוירונים עמוקות שונות לרשת אותה ניתן לאמן באמצעות pytorch, יש להשתמש בירושה מאובייקט המודל של pytorch (torch.nn.Module).

על מנת להציג את הגרפים עבור הרשתות השונות השתמשתי במימוש של netron [22] לוויזואליזציה של רשתות נוירונים עמוקות.

כל הרשתות שאומנו יורשות את אובייקט המודל של Pytorch ובנויות בצורה הפיתונית המתאימה למימוש כל רשת ע"פ המאמר המתאר אותה. מאפייני הרשתות השונות מתוארים בנספח י"ב.

המחלקות השונות בשלב יצירת קבצי הפלט

באיור 22 מוצגת דיאגרמת מחלקות העל של המחלקות השונות המיישמות את יצירת קבצי הפלט מסוג CSV



איור 22 – דיאגרמת מחלקות על בשלב יצירת קבצי הפלט

כפי שניתן לראות, רוב המחלקות כבר תוארו והיו בשימוש בשלב הקודם :

- מחלקת Main
- מחלקת Audio2Image
- מחלקות המודלים השונות

בשלב זה נוספו למערכת שתי מחלקות – Models Handler ו CSV Generator .

Models Handler

ModelsHandler
<pre> get_labels_dictionaries() get_model() load_all_models() </pre>

איור 23 – תוכן המחלקה ModelsHandler

תפקידה של מחלקה זו הוא לדאוג לספק את הנתונים על המודלים השונים עבור מחלקות חיצוניות אשר תעשה בהם שימוש (מחלקת CSV Generator).

מחלקה זו מכילה 3 פונקציות עיקריות :

1. טעינת כל המודלים לשימוש בהמשך כמסווגים עבור קבצי הבדיקה.
על מנת לסווג קובץ כלשהו לאחת מהאפשרויות השונות, ראשית יש לטעון את המודל המאומן אשר בעזרתו הקובץ יסווג.
לאחר טעינת המודלים, המידע והתוכן שלהם מוחזק ב-GPU .
2. החזרת כל אופציות הסיווג שהמודלים השונים יכולים לסווג :
'yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop', 'go', 'unknown', 'silence'
בסה"כ יוחזר מערך עם 12 האפשרויות הנ"ל.
3. החזרת מודל ספציפי יחד עם קבצי הבדיקה שלו.
קבצי הבדיקה של המודל המתקבלים מפונקציה זו נמצאים כבר בשלב שאחרי ההמרה מקובץ אודיו. כלומר, הקבצים במצבם בשלב זה מוכנים לסיווג וניתן לתשאל את המודל ולקבל את תחזית הסיווג שלו עבורם ולא נדרשת כל המרה נוספת עבורם.
בשלבים הבאים ניתן יהיה להשתמש במידע המוחזר ממחלקה זו על מנת לסווג את קבצי הבדיקה הרלוונטיים.

CSV Generator

CSV_Generator
GenerateCSVs()

איור 24 – תוכן המחלקה CSV Generator

מחלקה זו נקראת ע"י מחלקת העל Main לאחר שלב האימון.
מחלקה זו משתמשת במחלקת Models Handler על מנת לקבל את המודלים השונים וקבצי הבדיקה השונים אותם יש לסווג.
לאחר קבלת המודל וקבצי הבדיקה השונים במצב בו ניתן לסווגם, מחלקה זו מייצרת קבצי CSV בהתאם לכל מודל בצורה הבאה :

1. מתבצע מעבר על כל קבצי הבדיקה
2. עבור כל קובץ בדיקה מתבצע סיווג אך ורק ע"פ המודל הנוכחי
3. שורה המכילה את שם קובץ הבדיקה ואת תוצאת הסיווג תתווסף לקובץ הפלט.

בכל קובץ פלט מסוג CSV ייווצרו 158,536 (כמספר קבצי הבדיקה) שורות כאלו

לדוגמא :

עבור המודל "vgg1d_mel", נוצר הקובץ הבא : "vgg1d_mel_prediction.csv",

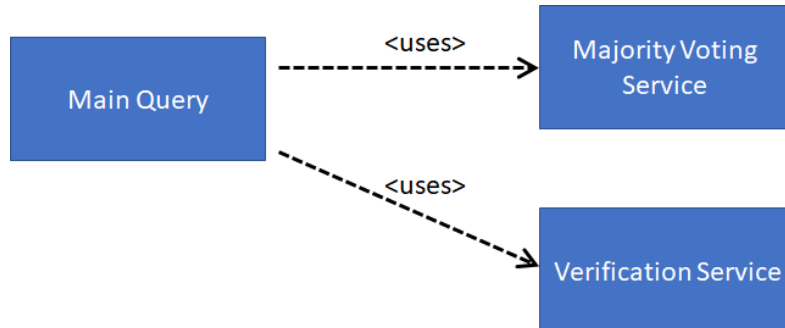
כך נראה תוכנו בשורות 60 ו 61 :

up clip_0018ff8e9.wav

stop clip_001a5ce9c.wav

המחלקות השונות בשלב התשאול

באיור 25 מודגמת דיאגרמת העל של המחלקות ליישום תשאול המערכת



איור 25 – דיאגרמת מחלקות על בשלב התשאול

שלב התשאול מתחלק לארבעה תת שלבים עיקריים

1. המשתמש מזין את המשקלי המודלים השונים עבור קבלת ציון הסיווג הכולל של המערכת
2. המערכת מבצעת ממוצע משוכלל על תוצאות הסיווג של המודלים שבה ע"פ המשקלים שהוזנו ע"י המשתמש לכל אחד מקבצי הבדיקה
3. המערכת משווה בין תוצאותיה עבור כל אחד מקבצי הבדיקה לבין התוצאות המקוריות
4. המערכת מחזירה את הציון הכולל המכמת את איכותה.

בסופו של שלב התשאול מתקבל ציון המכמת את טיב הסיווג של המערכת.

Main Query
ApplyMajorityVoting() CompareToWinner()

איור 26 – תוכן המחלקה Main Query

מחלקה ראשית אליה מתממשק גם ממשק המשתמש, מחלקה זו מכילה 2 פונקציות עיקריות –

- הפונקציה הראשונה משתמשת במחלקת Majority Voting. דרך ממשק המשתמש נקבעים האחוזים (המשקלים) אשר יינתנו לכל מודל בעת ביצוע הממוצע המשוכלל של המערכת עבור תוצאות הסיווגים השונים

הפלט של הפונקציה הזו הינו קובץ CSV המכיל את הממוצע המשוכלל עבור כל אחד מקבצי הבדיקה לאחר מעבר על המודלים השונים (אשר אומנו בשלב האימון) והסיווגים שהתקבלו מהם.

דוגמא לפלט חלקי הנוצר מהפונקציה הזו :

label	Fname
unknown	clip_7067d2303.wav
no	clip_000044442.wav
unknown	clip_0000adecb.wav
unknown	clip_0002d9b83.wav
go	clip_000373a5b.wav
unknown	clip_0003c7122.wav
off	clip_0003e6aee.wav
unknown	clip_00049951d.wav
silence	clip_0004c6707.wav

כאשר כל label הינו תוצאת הממוצע המשוכלל מכל המודלים שאומנו ע"פ המשקלים שהוזנו ע"י המשתמש.

- הפונקציה השנייה משתמשת במחלקת Verification.

מטרת הפונקציה הזו הינה להשוות בין תוצאות הפלט של Majority Voting במערכת למול תוצאות הסיווגים הנכונות עבור כל אחד ואחד מקבצי הבדיקה השונים.

פונקציה זו מחזירה את הציון באחוזים בדמיון בין תוצאות המערכת לתוצאות של הסיווגים הנכונות.

Majority Voting

Majority_Voting_Service
getGradeByModel() runMajorityVotingLogic()

איור 27 – תוכן המחלקה Majority Voting

מחלקה זו אחראית על ביצוע הממוצע המשוקלל בין תוצאות המודלים.

ראשית נטענים האחוזים מהזיהוי עבור כל אחד מהמודלים אשר נקבעו ע"י המשתמש והתקבלו דרך מחלקת הMain Query.

לאחר מכן ישנן שתי פונקציות אשר נמצאות בשימוש :

1. Get Grade By Model – בהינתן מודל עליו נרצה לקבל תוצאה, פונקציה זו תחזיר את הערך באחוזים אשר נקבע בממשק המשתמש עבור אותו מודל – בהמשך ערך זה ישפיע על תוצאות החיזוי הכוללת של המערכת.

2. Run Majority Voting Logic – פונקציה זו מבצעת את הפעולות הבאות :

יצירת מילון סיווגים, מילון הסיווגים יכיל את השדות הבאים:

שדה הKey : יכיל את שם הקובץ עליו התקבל הסיווג, למשל file1.wav

שדה הValue : יחולק ל6 שדות בעצמו, כל שדה יכיל את הסיווג שכל מודל ייתן לקובץ.

דוגמא לשורה אחת במילון הסיווגים :

Key	Value					
	Model1	Model2	Model3	Model4	Model5	Model6
File1.wav	"go"	"go"	"no"	"go"	"no"	"unknown"

מעבר על קבצי הבדיקה ומילוי מילון הסיווגים

כעת הפונקציה תמלא את מילון הסיווגים שנוצר קודם לכן, כך שיכיל את הסיווגים השונים של 6 המודלים לכל אחד מ-158,536 קבצי הבדיקה.

עבור כל קובץ בדיקה הפונקציה תבדוק מהו הסיווג אותו כל אחד מ-6 המודלים השונים שנוצרו בשלב Offline נתנו לו, הפונקציה תעשה זאת בעזרת קבצי CSV שנוצרו מראש המכילים את הסיווג של כל מודל עבור כל קובץ בדיקה.

ביצוע ממוצע משוקלל ע"פ המשקלים שהוזנו ע"י המשתמש

כעת לאחר שכל קבצי הבדיקות וסיווגיהם טעונים בזיכרון התוכנה, הפונקציה תבצע ממוצע משוקלל ע"פ הציונים שהוזנו ע"י המשתמש.

לדוגמא,

עבור הסיווגים הבאים לקובץ File1.wav

Model1	Model2	Model3	Model4	Model5	Model6
"go"	"go"	"no"	"go"	"no"	"unknown"

ועבור המשקלים הבאים שהוזנו ע"י המשתמש :

Model1	Model2	Model3	Model4	Model5	Model6
30%	25%	20%	15%	5%	5%

יתקבל הסיווג הסופי "go" ע"פ חישוב הממוצע המשוקלל באופן הבא :

$$\text{Maximum} \begin{cases} "go" = 60 \\ "no" = 20 \\ "unknown" = 20 \end{cases} = "go"$$

שמירת התוצאות

שמירת התוצאה של כל אחד מהממוצעים המשוקללים שנעשו על כל אחד מקבצי הבדיקה בשורה חדשה בקובץ CSV חדש שנוצר עבור ההרצה הנוכחית, המטרה הסופית היא להשוות את הקובץ שנוצר עם קובץ התוצאות הנכונות בשלבים הבאים ובכך לקבל את ציון המערכת.

Verification Service

Verificaion Service
compareToWinner() csvToDic()

איור 32 – תוכן מחלקת ה Verification

מחלקה זו אחראית על השוואת תוצאות סיווגי המערכת לתוצאות הסיווגים המקוריים.

מחלקה זו מקבלת כקלט עבור הבנאי הראשי שלה את מיקומי שני קבצי ה CSV בהם היא מתמקדת:

1. קובץ CSV המכיל את התוצאות המקוריות עבור החיזוי של כל אחד מקבצי הבדיקה
2. קובץ CSV המכיל את תוצאות המערכת עבור החיזוי של כל אחד מקבצי הבדיקה

כמו כן מחלקה זו מכילה 2 פונקציות עיקריות:

1. טעינת קבצי ה CSV לזיכרון
2. השוואה בין שני קבצי ה CSV לאחר שנטענו לזיכרון והחזרת ציון לממשק המשתמש ע"פ הנוסחה הפשוטה:

$$\text{Grade} = \frac{\sum_{i=0}^{(\text{test data files amount})} \begin{cases} \text{SystemClassification}[i] == \text{CorrectClassifications}[i] : 1 \\ \text{else} : 0 \end{cases}}{(\text{test data files amount})}$$

כך ש :

- ה הינו מספר קבצי הבדיקה (158,536)
- רצים על כל אחד ואחד מקבצי הבדיקה ומשווים האם תוצאת החיזוי שווה בין תוצאות הסיווג של המערכת לבין תוצאות הסיווג הנכונות.
- על מנת לקבל את הציון הסופי יש לחלק את תוצאת השלב הקודם במספר קבצי הבדיקה.

כלי פיתוח

הפרויקט מומש בעזרת טכנולוגיות וכלים שונים שיפורטו בהמשך סעיף זה.

הקוד נכתב בשפת Python , גירסא 3 בסביבת PyCharm.

רשימת הספריות הנדרשות להתקנת המערכת מתוארות בנספח ד

Librosa

זוהי ספריית קוד פתוח בשפת Python, ספרייה זו משמשת לעיבוד של אותות קוליים. [9]

ספרייה זו מספקת מבחר רב של פונקציות שימושיות המשמשות בעולם עיבוד האותות והקול.

במערכת שפותחה ספרייה זו משמשת בעיקר להמרת קבצי אודיו למרחב התמונה. השימוש העיקרי היה בפונקציות הבאות :

- `librosa.load` – טעינת קובץ אודיו לשימוש בסביבת Python
- `librosa.feature.melspectrogram` – המרת קובץ אודיו לתמונה במרחב הספקטרי המייצגת את אותו הקובץ.

 librosa/librosa is licensed under the ISC License A permissive license lets people do anything with your code with proper attribution and without warranty. The ISC license is functionally equivalent to the BSD 2-Clause and MIT licenses, removing some language that is no longer necessary.	Permissions ✓ Commercial use ✓ Distribution ✓ Modification ✓ Private use	Limitations ✗ Liability ✗ Warranty	Conditions ⓘ License and copyright notice
This is not legal advice. Learn more about repository licenses.			

איור 12 – רישיון הקוד הפתוח של ספריית Librosa להמרת קבצי אודיו
בשפת python

ספריית Flask היא ספריית קוד פתוח קטנה יחסית וקלה לתפעול המשמשת לפיתוח מערכת שרת לקוח פשוטה בגישת web. [11]

במקור הספרייה עוצבה כדי לתת למפתחי Python את היכולת לפתח אפליקציית Web בצורה יעילה ומהירה אך בפועל היא מכילה בתוכה את האפשרות להגדיל ולהגביר את תוכן וסיבוכיות האפליקציה.

ספרייה זו החלה כמעטפת לספריות אחרות הכתובות בPython (בין היתר Werkzeug [18]) אר נועדו להקמת שרת מקומי אך מהר מאוד נהפכה לפופולארית מהן בעיקר בשל הנוחות והקלות לייצר בעזרתה אפליקציית Web מקצה לקצה. בשונה מספריות אחרות, Flask אומנם מציעה להתקין חבילות אחרות אשר יעזרו וישפרו את מהירות הקמת הפרויקט אך היא לא דורשת ולא תלויה בהן, בשונה מספריות Web אחרות.

ל Flask יש קהילת מפתחים רחבה וגדולה אשר דואגת להוסיף ולהרחיב את הפונקציונליות והfeatures השונים של Flask על בסיס יומי.



Flask

web development,
one drop at a time

[overview](#) // [docs](#) // [community](#) // [extensions](#) // [donate](#)

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed](#)!

Flask is Fun Latest Version: [1.0.2](#)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

And Easy to Setup

```
$ pip install Flask
$ FLASK_APP=hello.py flask run
* Running on http://localhost:5000/
```

Interested? Star 40,505

- » [Download latest release \(1.0.2\)](#)
- » [Read the documentation](#)
- » [Join the mailinglist](#)
- » [Fork it on github](#)
- » [Add issues and feature requests](#)

איור 13 – דף הבית המתאר את הוראות ההתקנה והשימוש בFlask

זוהי ספריית קוד פתוח מבית פייסבוק.

הספרייה נועדה לאימון ושימוש ברשתות למידה עמוקה בשפת Python [10].

המקבילה של PyTorch מבית פייסבוק בתחום הלמידה העמוקה הינה Tensorflow אשר מתוחזקת ע"י גוגל [19].

ע"פ המאמר PyTorch or Tensorflow : The force is strong with which one [20] אשר מוצג כחלק מסדרת הכתבות והקורסים של חברת Udacity Indica, לא קיימת הכרעה חד משמעית מי משתי הספריות עדיפה, ישנם כמה פרמטרים עיקריים המשווים במאמר זה.

הראשון הוא עלות השימוש- אין הכרעה שכן שתי הספריות הינן ספריות קוד פתוח.

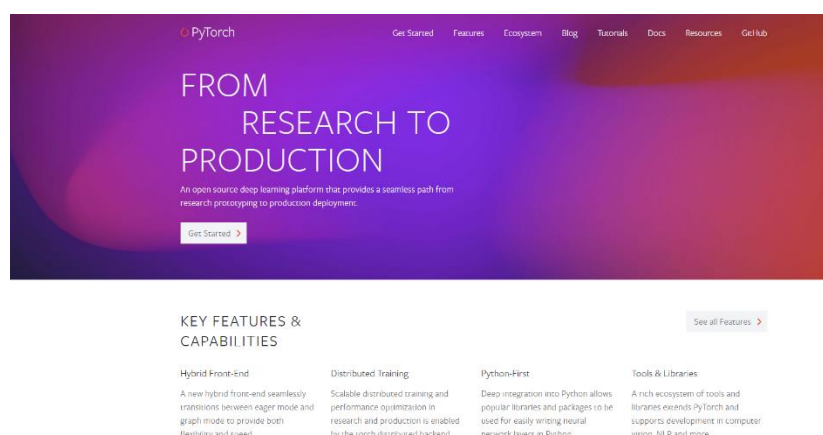
כמו כן מתוארים המנועים עליהם מתבססת כל ספרייה.

PyTorch מבוססת על מנוע Torch מבית פייסבוק לעומת Tensorflow המבוססת על מנוע Theano מבית גוגל.

יתרון חשוב שיש ל Tensorflow על PyTorch הינו Tensorboard. זהו כלי חשוב אשר עוזר לביצוע ויזואליזציה של מודלי רשתות הנוירוניים ישירות על דפדפן המשתמש בצורה נוחה.

מלבד ההבדל בכלים החיצוניים התומכים בTensorflow, ההבדל המשמעותי ביותר בין שתי הספריות היא הצורה בהן כל ספרייה מגדירה את הגרף החישובי לעיבוד המידע.

בעוד Tensorflow עובדת מול גרף סטטי שאינו משתנה, הגישה הרווחת בPyTorch הינה לעבוד מול גרף דינמי אשר משתנה תוך כדי תנועה, הגישה הדינמית הינה שימושית במיוחד אם המידע אותו מזינים לרשת על מנת ללמד ולאמן אותה עלול להשתנות תוך כדי העבודה ואינו קבוע.

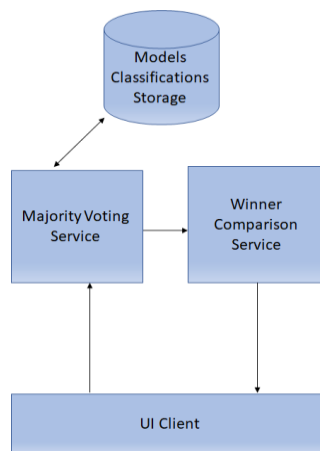


איור 14 – עמוד הבית של PyTorch מבית פייסבוק

הרכיבים השונים איתם מתקשר ממשק המשתמש

ממשק המשתמש מתקשר ישירות עם שניים משירותי המערכת:

- הראשון הוא שירות ה Majority Voting אשר מקבל את משקלי הקלט שהמשתמש הזין ומבצע בעזרתם ועל פיהם את הממוצע המשוקלל.
- השני הוא שירות ה Verification, מולו מתקשר ממשק המשתמש על מנת לקבל את הציון הסופי של המערכת לאחר ביצוע פעולות הממוצע המשוקלל בשלב ה Majority Voting.



איור 29 – הרכיבים הקשורים לממשק המשתמש

ניתן להיכנס לממשק המשתמש מכל דפדפן מודרני (גוגל כרום\מוזילה פיירפוקס\מיקרוסופט אדג') מממשק המשתמש מורכב ממסך אחד ובו שישה שדות קלט וכפתור לחישוב הציון.

The screenshot shows a web browser window with the title 'Speech Recognition x' and the address bar showing 'localhost:5000'. The main heading of the page is 'Speech Recognition Project By Ben Zagagy'. Below the heading is a form titled 'Running Mel Spectrograms Images as Input for Models:'. This form contains five input fields for percentages: 'Resnet Model Percentage' (15%), 'Senet Model Percentage' (15%), 'Densenet Model Percentage' (15%), 'VGG2D Model Percentage' (15%), and 'VGG1d Model Percentage' (10%). Below these is another section titled 'Running Raw Data as Input for Models:' which contains one input field: 'VGG1d Model Percentage' (30%). At the bottom of the form is a button labeled 'Calculate Score'.

איור 30 – ממשק המשתמש לשליטה במשקל Majority Voting וקבלת הציון הסופי של המערכת

המשתמש יוכל להזין לכל מודל שנוצר בשלב האימון את המשקל אשר הסיווג שלו יקבל בעת חישוב איכות המודל הסופית.

הקלט ממשתמש הקצה משפיע בצורה ישירה על איכות תוצאות המערכת בכך שבכוחו לקבוע כמה אחוזים מתוך הציון הסופי יינתנו לכל אחד מהמודלים השונים.

הערכים שנבחרו כברירת המחדל בממשק המשתמש מחזירים ציון של 94% (בהשוואה כמובן לתוצאות הנכונות כפי שצינתי קודם לכן).

בממשק המשתמש ישנן 6 תיבות טקסט להזנת אחוזים, כל תיבת טקסט יכולה לקבל מספר בין 0 ל-100 בלבד, כל תיבת טקסט מקושרת למשקל אותו יקבל המודל הקשור אליה בתהליך קבלת הציון הסופי.

לדוגמא עבור השורה הראשונה –

Resnet Model Percentage: %

איור 31 – דוגמא לדרך הזנת המשקל למודל מסוג Resnet אשר נלמד באמצעות Mel Spectrograms Images

לאחר הזנת כל 6 תיבות הטקסט במשקלים אשר סכומם יחדיו מגיע ל-100% בדיוק, המשתמש יילחץ על כפתור ה"Calculate Score".

Calculate Score

איור 32 – כפתור ה"Calculate Score" שבממשק המשתמש

לאחר הלחיצה על הכפתור, המערכת תחל בחישוב הציון של איכותה ע"פ המשקלים שהוזנו ע"י המשתמש.

לאחר סיום החישוב ע"י המערכת, המשתמש יקבל הודעה המכילה את מספר הזיהויים הנכונים ואת הציון הסופי שהמערכת קיבלה.

localhost:5000 says
Grade : 0.9403991472076095 ,
Correct Classifications: 149089/158538

OK

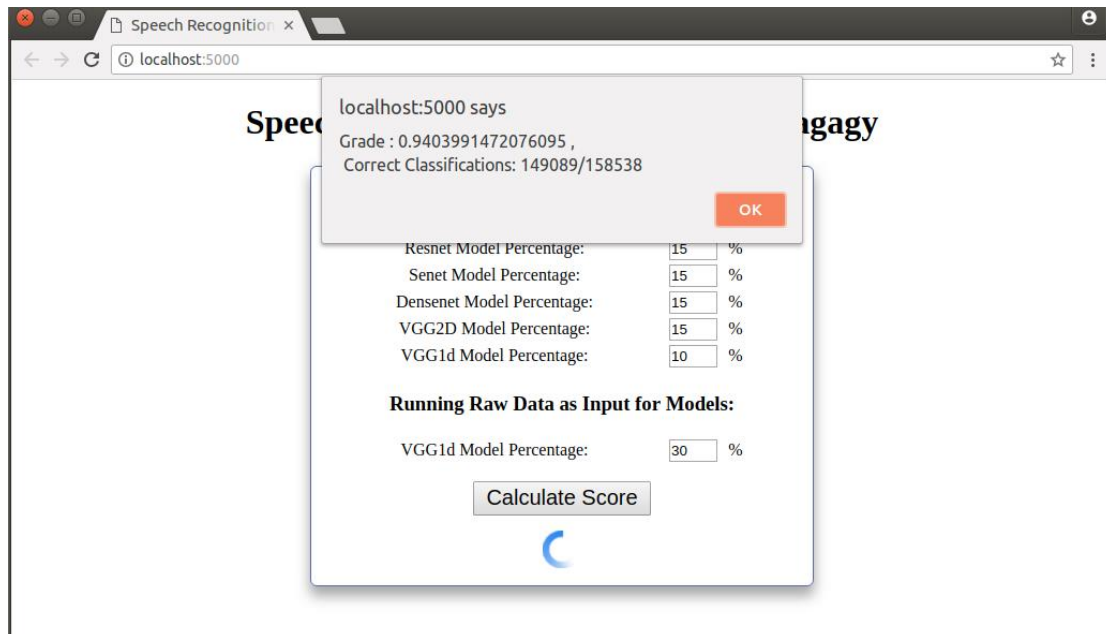
איור 33 – דוגמא להודעת תשובה מהמערכת לגבי הציון והזיהויים הנכונים של המערכת עבור ערכי המשקלים שהוזנו

ניתן לראות באיור 33 כי הציון מתקבל בשורה הראשונה בהודעת המערכת וכמות הזיהויים הנכונים מתקבלת בשורה השנייה.

המשתמש יוכל ללחוץ על כפתור ה "OK" ולנסות להזין משקלים חדשים על מנת לקבל את תוצאות וציון איכות המערכת עבור המשקלים החדשים.

תוצאות המערכת

לאחר החישוב יוצגו תוצאות המערכת(הציון הסופי) על גבי ממשק המשתמש.



איור 34 – הודעת המערכת עם הציון המתקבל מערכי ברירת המחדל

בדיקות

במערכת 6 מודלים עיקריים :

- מודול אימון המודלים
- מודול המרת אותות האודיו
- מודול הייצור של קבצי CSV
- ממשק המשתמש
- מודול Majority Voting
- מודול ההשוואה לקובץ המנצח

בדיקות יחידה

בדיקות יחידה הן בדיקות ברמת מודול(יחידת המערכת הקטנה ביותר) אשר מאמתות את פעילות התקינה של היחידה

קלט:

- רשימה של מיקומים של קבצי אודיו
- סוג המידע אותו נרצה לקבל שנרצה לקבל – Raw Data או Mel Spectrogram image

שלבי הבדיקה:

1. אטען רשימה קצרה של שניים עד שלושה קבצי אודיו שונים למודול וארצה לקבל פלט בדמות Mel Spectrogram image
2. בדוק שונות בין התמונות שנוצרו
3. טען קובץ אחד מתוך רשימת קבצי האודיו ששלחתי בסעיף 1
4. בדוק זהות בין תמונה זו לתמונה שנוצרה עבור אותו הקובץ בסעיף 2
5. טען את רשימת כל קבצי האימון (64,732 במספר) על מנת לבדוק ולוודא כי המודול מסתדר גם עם כמויות גדולות של מידע, בקש לקבל פלט בדמות Mel Spectrogram
6. בדוק כי הפלט עבור קבצי האימון נוצר והמחשב לא קפא ולא קרס בתהליך.
7. חזרה על שלבים 1-6 אך הפעם האלגוריתם יחזיר מידע מסוג raw data.

פלט:

- ייצוג קבצי האודיו שברשימת הקלט כ Raw Data או Mel Spectrogram image

בדיקות יחידה למודול ייצור קבצי CSV

קלט:

- רשימת מיקומי קבצי הקול המשמשים לבדיקת המערכת
- מיקומי המודלים השונים שנוצרו בשלב האימון

שלבי הבדיקה:

1. אתן למודול לייצר את קבצי ה CSV כפי שעושה בברירת המחדל שלו
2. בדוק כי קיימים 6 קבצי CSV בתיקיית היעד
3. בדוק כי 6 הקבצים שונים זה מזה
4. בדוק כי לכל אחת מהרשתות אותן אימנתי קיים קובץ CSV
5. בדוק כי בכל קובץ CSV קיימים הסיווגים לכל 158,536 קבצי הבדיקה

פלט:

- קובץ CSV לכל אחד מהרשתות אותן אימנתי, קובץ זה מכיל בין היתר את שם הרשת עליה סיווג הקבצים נבדק וכן שורות כמספר קבצי הבדיקה

קלט:

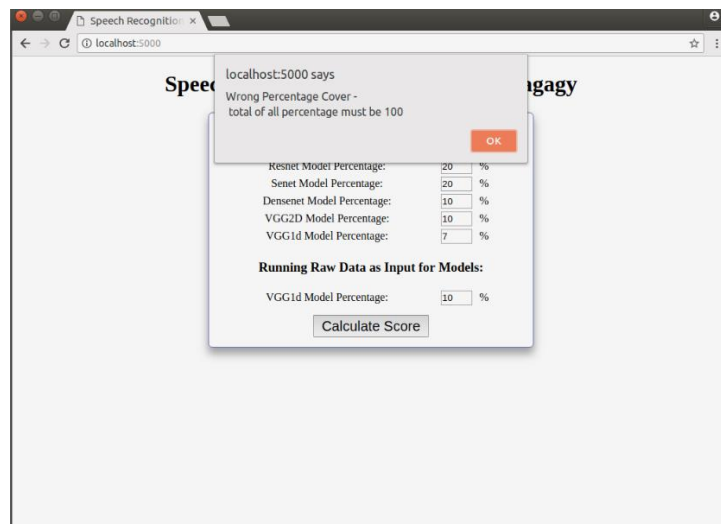
- אחוזי המשקל של כל מודל בהחלטה על סיווג סופי

שלבי הבדיקה:

1. הזן נתונים לא הגיוניים כקלט – אותיות במקום אחוזים
2. בדוק כי לא ניתן להתקדם ולשלוח את הנתונים האלו בלחיצה על כפתור ה Calculate Score
3. הזן נתונים לא הגיוניים אשר סה"כ הערכים שלהם גדול מ100
4. בדוק כי לא ניתן להתקדם ולשלוח את הנתונים האלו בלחיצה על כפתור ה Calculate Score
5. הזן נתונים הגיוניים כקלט
6. בדוק כי לאחר לחיצה על כפתור ה Calculate Score מתקבל ציון הגיוני
7. הזן נתונים הגיוניים אחרים כקלט
8. בדוק כי לאחר לחיצה על כפתור ה Calculate Score מתקבל ציון הגיוני השונה מהציון שהתקבל בסעיף 6
9. הזן בשנית את אותם הנתונים שהוזנו בסעיף 5
10. בדוק כי לאחר לחיצה על כפתור ה Calculate Score מתקבל אותו הציון שקיבלתי בסעיף 6.

פלט:

- הציון עבור הקלט הנתון



איור 35 – הודעת השגיאה המתקבלת כאשר מנסים לשלוח ערכים שאינם שווים ל100 דרך ממשק המשתמש

קלט :

- אחוזי המשקל של כל מודל בהחלטה על סיווג סופי

שלבי הבדיקה :

1. טען את אחוזי המשקל של אחד מהמודלים בהחלטה הסופית למודול
2. בדוק את היווצרות קובץ התוצאות tmp_results.csv הזמני בסיום פעולות הממוצע המשוכלל
3. שמור את קובץ התוצאות הזמני בצד לצורכי השוואה בהמשך הבדיקה
4. טען נתונים שונים עם משקלים שונים עבור כל אחד מהמודלים
5. בדוק היווצרות קובץ תוצאות נוסף, בדוק שוני מהקובץ הקודם שנוצר בשלב 2 ושמור גם אותו בצד לקראת בדיקות היחידה למודול ההשוואה לקובץ המנצח.
6. טען בשלישית נתונים עבור משקלי המודלים השונים בהחלטה הסופית אך הפעם טען בשנית את אותם המשקלים שהעברנו בקריאה בשלב 1
7. בדוק היווצרות קובץ תוצאות נוסף
8. בדוק כי הקובץ החדש זהה לקובץ שנוצר בשלב 2.

פלט :

- קובץ המכיל את תוצאות הממוצע המשוכלל לפי האחוזים הנתונים

בדיקות יחידה למודול ההשוואה לקובץ המנצח

קלט :

- קובץ csv להשוואה מול הקובץ של המנצח בתחרות

שלבי הבדיקה :

1. טען כקלט את הקובץ שנוצר בבדיקות היחידה ל Majority Voting בשלב 2 בבדיקות היחידה של מודול Majority Voting
2. בדוק קבלת ציון הגיוני
3. טען כקלט את הקובץ שנוצר בבדיקות היחידה ל Majority Voting בשלב 5 בבדיקות היחידה של מודול Majority Voting
4. בדוק קבלת ציון הגיוני ושונה מהציון בשלב 2
5. טען כקלט את הקובץ שנוצר בבדיקות היחידה ל Majority Voting בשלב 7 בבדיקות היחידה של מודול Majority Voting
6. בדוק קבלת ציון הגיוני אשר זהה לציון בשלב 2

פלט :

- הציון ומספר סיווגים הנכונים עבור קובץ הקלט

בדיקות אינטגרציה

בדיקות אינטגרציה מבוצעות על מנת לבדוק את ההשפעה הקיימת בין חלקי המערכת השונים אשר משיקים וקשורים זה לזה. על מנת לבצען יש לחבר חלקים בודדים במערכת זה לזה או למערכת חיצונית לחלוטין

בדיקות אינטגרציה למודל אימון המודלים ולמודל המרת אותות האודיו

אימון 6 מודלים על בסיס קבצי אימון בודדים

קלט :

120 קבצי אודיו לאימון – עשרה קבצים לכל אחת מ-12 המילים שאותם אנו לומדים

['yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop', 'go', 'unknown', 'silence']

כל עשרה קבצים נמצא בתיקייה ע"פ שם ה-class המתאימה לו.

פלט :

- 6 קבצי מודלים של pytorch מסוג pth נוצרו

תוצאות צפויות לאחר האימון :

- סיווג קבצי אודיו נוספים ע"פ המודלים הנ"ל לא ייתן תשובה טובה אלא אם הסיווג מתבצע על אחד מ-120 קבצי הבדיקה אותם המודל למד שכן המידע עליו המודלים התאמנו היה קטן מידי

שלבי הבדיקה :

1. בדיקת קיום של 6 קבצי המודלים השונים בסיומות pth
2. הרץ קוד להמרת קובץ אודיו למel spectrogram בעזרת מודל המרת אותות האודיו
3. הרץ קוד לסיווג התמונה שהומרה בעזרת המודלים השונים שנוצרו
4. בדיקת קבלת תוצאה שגויה עבור סיווג קובץ אודיו שלא נלמד במודל
5. בדיקת קבלת תוצאה נכונה עבור סיווג קובץ אודיו שכן נלמד במודל

בדיקת קבצים מתוך מאגר קבצי הבדיקה :

קלט :

- 6 קבצי המודלים השונים המקוריים שאומנו כראוי
- 12 קבצי אודיו לסיווג ע"י המודלים השונים מתוך קבצי הבדיקה

שלבי הבדיקה :

1. הרץ את תהליך יצירת ה CSV בעזרת המודלים הקיימים עבור 12 קבצי הבדיקה
2. בדוק את היווצרות קבצי CSV בעלי הסיווגים השונים
3. בדוק את תוצאות הסיווג עבור כל אחד מ 12 קבצי הבדיקה
4. בדוק את תוצאות הסיווג עבור 12 הקבצים האלו בכל אחת מ 6 המודלים למול אותן התוצאות בקבצי ה CSV המקוריים שנוצרו בתהליך המקורי ויודוי נכונות עבורם.

פלט :

- תוצאות הסיווג לכל אחד מ 12 הקבצים המתפרסות על גבי 6 קבצי CSV המתאימים לכל מודל

בדיקת קבצים מתוך מאגר קבצי האימון :

קלט :

- 6 קבצי המודלים השונים המקוריים שאומנו כראוי
- 12 קבצי אודיו לסיווג ע"י המודלים השונים מתוך קבצי האימון

שלבי הבדיקה :

1. הרץ את תהליך יצירת ה CSV בעזרת המודלים הקיימים עבור קבצי האימון
2. בדוק את היווצרות קבצי CSV עם הסיווגים השונים
3. בדוק את תוצאות הסיווג עבור כל אחד מ 12 קבצי האימון
4. מכיוון שהקלט הינו קבצים שהמסווגים מכירים, בדיקת הנכונות במקרה הזה תהיה בין תוצאות המסווגים לבין התוכן שבאמת הוקלט ומופיע בקבצי האימון – הפעל והקשב לכל קובץ ובדוק כי הסיווגים השונים מתאימים לו.

פלט :

- תוצאות הסיווג לכל אחד מ 12 הקבצים המתפרסות על גבי 6 קבצי CSV המתאימים לכל מודל

מטרות הצגת מקרה הבוחן הן :

1. הצגת המצב האופטימלי והציון האופטימלי של המערכת.
2. הדגמת הקלות בה ניתן להשתמש בממשק המשתמש על מנת לבדוק משקלים שונים עבור המודלים השונים שאומנו ובכך גם להדגים את קלות קבלת הציון המעיד על איכות המערכת.

שלבי מקרה הבוחן :

1. פתח את טרמינל שורת הפקודה ונווט לתיקייה המכילה את קבצי המערכת.
2. הרץ את השרת המקומי המפעיל את המערכת בעזרת פקודת `"python3 main.py"`.
3. היכנס בעזרת דפדפן האינטרנט לכתובת <http://localhost:5000>.

**Speech Recognition Project By Ben Zagagy,
Guided By Dr. Maya Herman**

Running Mel Spectrograms Images as Input for Models:

Resnet Model Percentage:	15	%
Senet Model Percentage:	15	%
Densenet Model Percentage:	15	%
VGG2D Model Percentage:	15	%
VGG1d Model Percentage:	10	%

Running Raw Data as Input for Models:

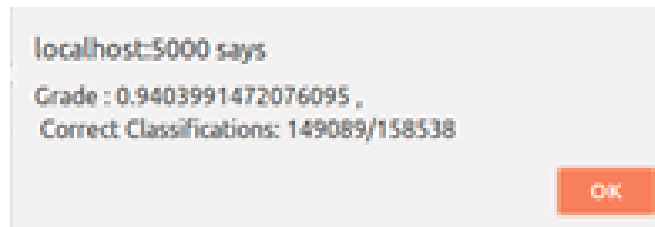
VGG1d Model Percentage:	30	%
-------------------------	---------------------------------	---

4. הזן את המשקלים השונים לכל מודל ע"פ הטבלה הבאה :

מודל	אחוזים עבור שלב ה Majority Voting
ResNet on Mel Spectrogram images	15%
SeNet on Mel Spectrogram images	15%
DenseNet on Mel Spectrogram images	15%
VGG-2D on Mel Spectrogram images	15%
VGG-1D on Mel Spectrogram images	10%
VGG-1D on Raw Data	30%

5. לחץ על "Calculate Score"

6. קבלת הציון 94.03



איור 37 – קבלת הציון 94.03 בעת מקרה הבוחן

משקלים נוספים שנבדקו לצורך מקרה הבוחן

בהינתן מצבים אחרים בהם נשתמש באחוזים אלו לביצוע Majority Voting של כל מודל נקבל את הציונים הבאים

(ציונים נוספים עבור משקלים נוספים
ניתן לראות בנספח א')

Total Grade	VGG-1D on Raw Data	VGG-1D on Mel Spectrogram images	VGG-2D on Mel Spectrogram images	DenseNet on Mel Spectrogram images
94.03%	30	10	15	
93.46%	10	30	10	
93.19%	5	35	15	
91.43%	10	50	5	
93.56%	15	15	30	
93.68%	15	15	10	
92.47%	50	10	10	
91.43%	10	50	10	
92.47%	10	10	50	
92.44%	10	10	10	
91.49%	10	10	10	
90.29%	10	10	10	
93.62%	30	10	10	
93.51%	10	20	20	
91.55%	100	0	0	
90.71%	0	100	0	
92.04%	0	0	100	
92.13%	0	0	0	
92.05%	0	0	0	
89.93%	0	0	0	

ניתוח התוצאות והסקת מסקנות

מניתוח התוצאות נראה שלא קיים מודל דומיננטי אחד. ההיפך הוא הנכון.

קיים מרחב פתרונות אפשריים גדול מאוד ולכן רק בעזרת שילוב של סיווגי המודלים השונים ניתן לקבל סווג וזיהוי אופטימליים.

יתרה מכך – כאשר בודקים את הציון של מודל בודד(עבור כל אחד מששת המודלים) ניתן לראות כי הציון הגבוה ביותר הינו 92.04, קטן בכמעט שני אחוזים מהציון האופטימלי שהושג ע"י שימוש בכל המודלים יחדיו גם אם באחוזים מעט שונים זה מזה.

ניתן לחשוב כי 2% אינו הבדל המצדיק את כל המערכת ה"מורכבת" שנבנתה אך אין זה נכון, שכן 2% מכמות קבצי הבדיקה (158539 קבצים) שווה ל3170, כלומר בעקבות השימוש ב6 מודלים ובמנגנון הממוצע משוכלל, המערכת משיגה 3170 זיהויים נכונים יותר משהייתה משיגה בלעדיהם.

עלות השימוש בכמה מודלים שונים הינה בעיקר בשלב Offline בו יש לאמן כמות גדולה של מודלים (וכפי שצוין, אימון של מודל הוא תהליך איטי שלוקח בין מספר שעות במחשב חזק למספר ימים במחשב חלש). בשלב Online אומנם החישוב הוא פחות אינטואיטיבי אך מבחינת משאבים וזמן, אין עלות משמעותית לשימוש בכמה מודלים למול מודל בודד.

התועלת בשימוש ב6 מודלים שונים הינה ברווח של כמות גדולה מאוד של זיהויים נכונים אשר היו נאבדים לו היה נעשה שימוש במודל בודד. שכן בעולם בו מודלים של רשתות נוירונים יכולים להוות חלק ממערכות קריטיות, כמות כזו של זיהויים נכונים הינה מהותית.

ניתן לראות כי התוצאות הטובות ביותר מתקבלות כאשר ישנו מודל אחד בעל 30% ושאר 70% אחוזי התוצאה מורכבים מחלוקה אחידה יחסית בין שאר המודלים, במקרה הפרטי בו המערכת הגיעה לתוצאה המקסימלית שלה ניתן לראות כי התפלגות האחוזים הינה 30% למודל VGG שעבד על raw data, 10% למודל שנבנה בארכיטקטורת SENET ו15% לשאר 4 המודלים האחרים.

הציון הטוב ביותר למודל אחד בודד מתקבל עבור מודל VGG 2D.

בנימה האישית, ציפיתי שהוא יהיה החזק ביותר. מניסיוני בעבודה מחוץ לאקדמיה עם רשתות נוירונים עמוקות, ארכיטקטורת VGG-2D הינה הארכיטקטורה הנפוצה ביותר בעולם ה deep learning נכון לשנה האחרונה.

סיכום והצעות לפיתוחים עתידיים

במסגרת פרויקט מתקדם במדעי המחשב פותחה מערכת לזיהוי פקודות קוליות העומדת בתנאי האתגר "Tensorflow Speech Recognition Challenge של Kaggle.

לאחר למידה, ניתוח ומציאת פתרון גיבשתי הצעה המגדירה את המטרות לפרויקט. במהלך הפרויקט הגדרתי את הדרישות, סקרתי את עיצוב המערכת, כלי הפיתוח בהם נעשה שימוש, ממשק משתמש והדרכים בהם ניתן להשתמש במערכת. בשלבים האחרונים הצגתי את הבדיקות שביצעתי כדי לבדוק את הנכונות ההתנהגות של המערכת, הבדיקות כללו גם מקרה בוחן. המקרה הבוחן מציג את הנתונים הנדרשים על מנת לקבל את הציון הטוב ביותר שהמערכת יכולה לקבל עבור יכולות הסיווג שלה.

המערכת שפותחה הינה מערכת שרת-לקוח והיא נכתבה כולה בשפת פייתון תוך כדי שימוש בספרייה החדשה Pytorch מבית Facebook לאימון רשתות נוירונים בלמידה עמוקה.

המערכת מכילה בין היתר ספריות לאימון מודלים של רשתות נוירונים בגישת למידה עמוקה אשר אומנו על בסיס מאגר קבצי קול וכמו כן מיועדים לסווג קבצי קול. זאת בשונה מרוב המודלים הנפוצים כיום אשר אומנו על תמונות ומיועדים לסיווג של תמונות בלבד.

הפרויקט תרם לי אישית בחשיפה לאתגרים וסוגיות של אימון רשתות נוירונים עמוקות על מידע לא סטנדרטי מסוג אודיו. עד היום בעבודתי ובפרויקטים השונים שבניתי בשנים האחרונות, הזנת ואימון מודלי למידה עמוקה תמיד נעשתה בעזרת מידע ויזואלי של תמונות.

הפרויקט יכול להוות בסיס לעבודות המשך כגון:

- אימון מודלים נוספים על קבצי האימונים ושיפור תוצאות המערכת
- אימון מערכות אחרות הלומדות קבצי קול ומזהות קטעי קול אחרים בעזרת למידה עמוקה.
- אימון מערכות אחרות לזיהוי וסיווג ע"פ מודלים שונים אשר לא מקבלים בהכרח מידע קולי(תמונות או אפילו סרטים יוכלו להשתלב בקלות במערכת שבניתי תוך כדי שינויים מינוריים בקלט המודלים והמסווגים)
- חישוב אוטומטי של המשקלים אופטימליים (פירוט בנספח א')
- שיפור ביצועי המסווג (פירוט בנספח יג')
- שיפור ביצועי האימון (פירוט בנספח יג')

פרויקט זה מהווה בסיס לעבודה המסכמת שלי שכותרתה "סקירה השוואתית של הגישות השונות מעולם הלמידה העמוקה עבור זיהוי קולי".

נספחים

נספח א' – סטטיסטיקות

סטטיסטיקות של זמנים וציונים עבור ערכים שונים המתקבלים מערכים שונים עבור חלוקות משקלים שונות בשלב Majority Voting

RESNET on Mel Image	SENET on Mel Image	DENSNET on Mel Image	VGG2D on Mel Image	VGG1D on Mel Image	VGG1D on Raw Data	Grade	Time
(%)	(%)	(%)	(%)	(%)	(%)	(%)	(sec)
10	10	10	10	10	50	92.35	4.917
10	10	10	10	50	10	91.82	4.745
10	10	10	50	10	10	92.42	4.821
10	10	50	10	10	10	92.47	4.947
10	50	10	10	10	10	91.65	4.948
50	10	10	10	10	10	91.54	4.937
15	15	25	25	5	5	93.5	5.027
15	15	25	5	25	5	93.63	5.067
15	15	25	5	5	25	93.9	4.936
15	15	5	25	25	5	93.57	4.79
15	15	5	25	5	25	93.83	5.008
15	15	5	5	25	25	93.8	4.853
15	25	15	25	5	5	93.44	4.919
15	25	15	5	25	5	93.47	5.391
15	25	15	5	5	25	93.62	4.951
15	25	25	15	5	5	93.42	4.983
15	25	25	5	15	5	93.45	5
15	25	25	5	5	15	93.49	4.959
15	25	5	15	25	5	93.49	5.054
15	25	5	15	5	25	93.68	4.772
15	25	5	25	15	5	93.46	5.193
15	25	5	25	5	15	93.54	4.784
15	25	5	5	15	25	93.68	4.832

15	25	5	5	25	15	93.54	4.851
15	5	15	25	25	5	93.67	4.823
15	5	15	25	5	25	93.92	4.96
15	5	15	5	25	25	93.9	5.177
15	5	25	15	25	5	93.75	4.842
15	5	25	15	5	25	94.03	4.741
15	5	25	25	15	5	93.61	4.841
15	5	25	25	5	15	93.64	4.81
15	5	25	5	15	25	94.02	4.809
15	5	25	5	25	15	93.81	4.834
15	5	5	15	25	25	93.92	5.287
15	5	5	25	15	25	93.94	4.742
15	5	5	25	25	15	93.78	4.818
25	15	15	25	5	5	93.58	4.909
25	15	15	5	25	5	93.61	5.134
25	15	15	5	5	25	93.84	4.967
25	15	25	15	5	5	93.56	5.163
25	15	25	5	15	5	93.59	4.926
25	15	25	5	5	15	93.6	4.949
25	15	5	15	25	5	93.58	4.775
25	15	5	15	5	25	93.8	4.969
25	15	5	25	15	5	93.62	4.786
25	15	5	25	5	15	93.64	4.834
25	15	5	5	15	25	93.87	5.229
25	15	5	5	25	15	93.6	4.752
25	25	15	15	5	5	93.42	4.935
25	25	15	5	15	5	93.43	4.918
25	25	15	5	5	15	93.45	4.995
25	25	5	15	15	5	93.41	4.928

25	25	5	15	5	15	93.49	5.28
25	25	5	5	15	15	93.49	4.975
25	5	15	15	25	5	93.72	4.794
25	5	15	15	5	25	93.94	4.755
25	5	15	25	15	5	93.69	4.76
25	5	15	25	5	15	93.72	4.763
25	5	15	5	15	25	94.03	4.779
25	5	15	5	25	15	93.73	4.918
25	5	25	15	15	5	93.72	4.947
25	5	25	15	5	15	93.75	4.937
25	5	25	5	15	15	93.82	4.946
25	5	5	15	15	25	93.98	4.817
25	5	5	15	25	15	93.75	4.842
25	5	5	25	15	15	93.8	5.054
5	15	15	25	25	5	93.59	5.116
5	15	15	25	5	25	93.8	4.997
5	15	15	5	25	25	93.77	4.821
5	15	25	15	25	5	93.62	4.795
5	15	25	15	5	25	93.84	5.105
5	15	25	25	15	5	93.52	4.779
5	15	25	25	5	15	93.54	4.941
5	15	25	5	15	25	93.88	4.82
5	15	25	5	25	15	93.63	4.832
5	15	5	15	25	25	93.73	4.893
5	15	5	25	15	25	93.8	4.768
5	15	5	25	25	15	93.6	4.771
5	25	15	15	25	5	93.46	4.808
5	25	15	15	5	25	93.63	5.496
5	25	15	25	15	5	93.47	4.793
5	25	15	25	5	15	93.52	4.806

5	25	15	5	15	25	93.64	4.811
5	25	15	5	25	15	93.54	4.836

RESNET on Mel Image (%)	SENET on Mel Image (%)	DENSNET on Mel Image (%)	VGG2D on Mel Image (%)	VGG1D on Mel Image (%)	VGG1D on Raw Data (%)	Grade (%)	Time(sec)
5	25	25	15	15	5	93.41	4.934
5	25	25	15	5	15	93.5	5.156
5	25	25	5	15	15	93.52	5.016
5	25	5	15	15	25	93.72	4.79
5	25	5	15	25	15	93.55	4.799
5	25	5	25	15	15	93.54	4.893
5	5	15	15	25	25	93.82	4.788
5	5	15	25	15	25	93.89	4.749
5	5	15	25	25	15	93.73	5.2
5	5	25	15	15	25	93.97	4.781
5	5	25	15	25	15	93.76	4.802
5	5	25	25	15	15	93.68	4.805
10	10	20	20	20	20	93.9	4.767
10	20	10	20	20	20	93.8	4.805
10	20	20	10	20	20	93.8	4.765
10	20	20	20	10	20	93.79	5.253
10	20	20	20	20	10	93.69	4.933
20	10	10	20	20	20	93.97	4.766
20	10	20	10	20	20	93.97	4.759
20	10	20	20	10	20	93.91	4.889
20	10	20	20	20	10	93.8	4.956
20	20	10	10	20	20	93.83	5.364
20	20	10	20	10	20	93.78	4.722
20	20	10	20	20	10	93.7	4.931
20	20	20	10	10	20	93.8	4.892
20	20	20	10	20	10	93.73	4.755
20	20	20	20	10	10	93.68	5.145

30	10	20	20	10	10	93.67	5.411
30	10	20	10	20	10	93.7	5.097
30	10	20	10	10	20	93.8	4.82
30	10	10	20	20	10	93.73	4.937
30	10	10	20	10	20	93.8	4.783
30	10	10	10	20	20	93.78	4.765
30	20	10	20	10	10	93.58	4.907
30	20	10	10	20	10	93.57	5.573
30	20	10	10	10	20	93.65	5.019
30	20	20	10	10	10	93.53	4.988
10	30	20	20	10	10	93.45	5.085
10	30	20	10	20	10	93.46	4.996
10	30	20	10	10	20	93.5	4.832
10	30	10	20	20	10	93.45	4.925
10	30	10	20	10	20	93.5	5.183
10	30	10	10	20	20	93.53	4.821
10	20	30	20	10	10	93.54	4.942
10	20	30	10	20	10	93.6	4.794
10	20	30	10	10	20	93.71	4.789
10	20	20	30	10	10	93.54	4.964
10	20	20	10	30	10	93.61	5.192
10	20	20	10	10	30	93.8	4.748
10	20	10	30	20	10	93.55	4.933
10	20	10	30	10	20	93.65	4.754
10	20	10	20	30	10	93.59	5.008
10	20	10	20	10	30	93.78	4.971
10	20	10	10	30	20	93.65	4.812
10	20	10	10	20	30	93.75	5.443
10	10	30	20	20	10	93.66	5.008

10	10	30	20	10	20	93.78	4.802
10	10	30	10	20	20	93.79	4.835
10	10	20	30	20	10	93.63	5.192
10	10	20	30	10	20	93.72	4.798
10	10	20	20	30	10	93.69	4.994
10	10	20	20	10	30	93.95	4.813
10	10	20	10	30	20	93.73	4.933
10	10	20	10	20	30	93.91	4.754
10	10	10	30	20	20	93.74	4.824
10	10	10	20	30	20	93.72	5.13
10	10	10	20	20	30	93.84	4.942
20	30	10	20	10	10	93.42	5.087
20	30	10	10	20	10	93.41	4.745
20	30	10	10	10	20	93.47	4.747
20	30	20	10	10	10	93.42	4.907
20	10	30	20	10	10	93.63	5.288
20	10	30	10	20	10	93.69	4.79
20	10	30	10	10	20	93.8	4.821
20	10	20	30	10	10	93.64	4.991
20	10	20	10	30	10	93.74	4.81
20	10	20	10	10	30	93.99	4.83
20	10	10	30	20	10	93.66	4.932
20	10	10	30	10	20	93.77	5.222
20	10	10	20	30	10	93.71	4.96
20	10	10	20	10	30	93.94	4.756
20	10	10	10	30	20	93.71	4.865
20	10	10	10	20	30	93.89	4.764
20	20	30	10	10	10	93.59	4.963
20	20	10	30	10	10	93.58	5.259
20	20	10	10	30	10	93.62	4.741

20	20	10	10	10	30	93.8	4.779
30	10	15	15	15	15	93.79	4.828
30	15	10	15	15	15	93.77	4.927
30	15	15	10	15	15	93.75	4.76
30	15	15	15	10	15	93.74	4.776
30	15	15	15	15	10	93.67	4.989
10	30	15	15	15	15	93.52	4.734
10	15	30	15	15	15	93.72	4.755
10	15	15	30	15	15	93.71	4.794

RESNET on Mel Image (%)	SENET on Mel Image (%)	DENSNET on Mel Image (%)	VGG2D on Mel Image (%)	VGG1D on Mel Image (%)	VGG1D on Raw Data (%)	Grade (%)	Time(sec)
10	15	15	15	30	15	93.78	4.81
10	15	15	15	15	30	94.07	4.8
15	30	10	15	15	15	93.52	5.396
15	30	15	10	15	15	93.49	4.93
15	30	15	15	10	15	93.48	4.835
15	30	15	15	15	10	93.43	4.763
15	10	30	15	15	15	93.75	4.756
15	10	15	30	15	15	93.71	4.791
15	10	15	15	30	15	93.79	5.151
15	10	15	15	15	30	94.09	4.826
15	15	30	10	15	15	93.72	4.769
15	15	30	15	10	15	93.7	4.802
15	15	30	15	15	10	93.65	4.768
15	15	10	30	15	15	93.71	4.808
15	15	10	15	30	15	93.77	4.804
15	15	10	15	15	30	94.05	4.932
15	15	15	30	10	15	93.68	5.01
15	15	15	30	15	10	93.64	4.947
15	15	15	10	30	15	93.75	4.747
15	15	15	10	15	30	94.04	4.878
15	15	15	15	30	10	93.71	4.818
15	15	15	15	10	30	94.05	4.977
25	25	25	15	5	5	93.19	4.764
25	25	25	5	15	5	93.41	4.804
25	25	25	5	5	15	93.45	4.998
25	25	15	25	5	5	93.37	4.836
25	25	15	5	25	5	93.43	4.801

25	25	15	5	5	25	93.55	4.778
25	25	5	25	15	5	93.32	4.871
25	25	5	25	5	15	93.45	4.838
25	25	5	15	25	5	93.22	4.974
25	25	5	15	5	25	93.56	4.93
25	25	5	5	25	15	93.45	4.954
25	25	5	5	15	25	93.43	4.742
25	15	25	25	5	5	93.49	4.987
25	15	25	5	25	5	93.61	4.748
25	15	25	5	5	25	93.85	4.839
25	15	5	25	25	5	93.55	4.74
25	15	5	25	5	25	93.8	4.769
25	15	5	5	25	25	93.73	4.775
25	5	25	25	15	5	93.5	4.761
25	5	25	25	5	15	93.64	5.066
25	5	25	15	25	5	93.57	5.112
25	5	25	15	5	25	93.9	4.745
25	5	25	5	25	15	93.72	4.951
25	5	25	5	15	25	93.89	4.954
25	5	15	25	25	5	93.63	4.802

RESNET on Mel Image (%)	SENET on Mel Image (%)	DENSNET on Mel Image (%)	VGG2D on Mel Image (%)	VGG1D on Mel Image (%)	VGG1D on Raw Data (%)	Grade (%)	Time(sec)
25	5	15	25	5	25	93.88	4.746
25	5	15	5	25	25	93.86	4.954
25	5	5	25	25	15	93.73	4.789
25	5	5	25	15	25	93.89	4.751
25	5	5	15	25	25	93.85	4.781
15	25	25	25	5	5	93.36	4.812
15	25	25	5	25	5	93.44	4.769
15	25	25	5	5	25	93.59	5.011
15	25	5	25	25	5	93.38	4.805
15	25	5	25	5	25	93.56	4.787
15	25	5	5	25	25	93.58	4.732
15	5	25	25	25	5	93.61	4.829
15	5	25	25	5	25	93.83	4.783
15	5	25	5	25	25	93.83	4.766
15	5	5	25	25	25	93.76	5.001
5	25	25	25	15	5	93.26	4.767
5	25	25	25	5	15	93.43	4.772
5	25	25	15	25	5	93.24	4.789
5	25	25	15	5	25	93.58	4.761
5	25	25	5	25	15	93.51	4.828
5	25	25	5	15	25	93.56	4.767
5	25	15	25	25	5	93.37	4.836
5	25	15	25	5	25	93.54	4.736
5	25	15	5	25	25	93.58	4.879
5	25	5	25	25	15	93.44	4.773
5	25	5	25	15	25	93.56	4.857

5	25	5	15	25	25	93.59	5.214
5	15	25	25	25	5	93.51	4.809
5	15	25	25	5	25	93.75	4.799
5	15	25	5	25	25	93.72	4.81
5	15	5	25	25	25	93.73	4.761
5	5	25	25	25	15	93.65	4.738
5	5	25	25	15	25	93.8	5.013
5	5	25	15	25	25	93.8	4.77
5	5	15	25	25	25	93.74	4.748
35	15	20	10	10	10	93.38	4.742
35	15	10	20	10	10	93.4	4.868
35	15	10	10	20	10	93.34	5.278
35	15	10	10	10	20	93.35	4.785
35	20	15	10	10	10	93.31	4.935
35	20	10	15	10	10	93.34	4.959
35	20	10	10	15	10	93.3	4.948
35	20	10	10	10	15	93.3	4.779
35	10	15	20	10	10	93.46	4.964
35	10	15	10	20	10	93.39	5.001
35	10	15	10	10	20	93.4	5.048
35	10	20	15	10	10	93.46	4.724

RESNET on Mel Image (%)	SENET on Mel Image (%)	DENSNET on Mel Image (%)	VGG2D on Mel Image (%)	VGG1D on Mel Image (%)	VGG1D on Raw Data (%)	Grade (%)	Time(sec)
35	10	20	10	15	10	93.43	4.78
35	10	20	10	10	15	93.45	4.933
35	10	10	15	20	10	93.41	5.137
35	10	10	15	10	20	93.42	4.835
35	10	10	20	15	10	93.45	4.91
35	10	10	20	10	15	93.47	5.221
35	10	10	10	15	20	93.39	4.788
35	10	10	10	20	15	93.41	4.817
15	35	20	10	10	10	93.07	5.17
15	35	10	20	10	10	93.1	4.961
15	35	10	10	20	10	93.08	4.817
15	35	10	10	10	20	93.08	4.847
15	20	35	10	10	10	93.36	4.952
15	20	10	35	10	10	93.4	5.25
15	20	10	10	35	10	93.4	4.766
15	20	10	10	10	35	93.81	4.806
15	10	35	20	10	10	93.37	4.92
15	10	35	10	20	10	93.48	4.827
15	10	35	10	10	20	93.53	4.846
15	10	20	35	10	10	93.4	5.296
15	10	20	10	35	10	93.51	4.824
15	10	20	10	10	35	93.96	5.033
15	10	10	35	20	10	93.44	4.972
15	10	10	35	10	20	93.5	4.808
15	10	10	20	35	10	93.46	4.944
15	10	10	20	10	35	93.92	4.841
15	10	10	10	35	20	93.35	5.278

15	10	10	10	20	35	93.85	5.041
20	35	15	10	10	10	93.09	4.977
20	35	10	15	10	10	93.09	5.146
20	35	10	10	15	10	93.08	4.971
20	35	10	10	10	15	93.1	4.831
20	15	35	10	10	10	93.45	4.873
20	15	10	35	10	10	93.45	4.947
20	15	10	10	35	10	93.42	4.773
20	15	10	10	10	35	93.94	4.856
20	10	35	15	10	10	93.44	4.739
20	10	35	10	15	10	93.46	4.8
20	10	35	10	10	15	93.51	5.291
20	10	15	35	10	10	93.44	5.286
20	10	15	10	35	10	93.44	4.767
20	10	15	10	10	35	94.01	4.896
20	10	10	35	15	10	93.46	4.976
20	10	10	35	10	15	93.51	5.107
20	10	10	15	35	10	93.42	4.773
20	10	10	15	10	35	94.01	4.764
20	10	10	10	35	15	93.39	4.806
20	10	10	10	15	35	93.92	4.802
10	35	15	20	10	10	93.16	4.881
10	35	15	10	20	10	93.13	4.816
10	35	15	10	10	20	93.14	5.216
10	35	20	15	10	10	93.11	5.233
10	35	20	10	15	10	93.11	4.943
10	35	20	10	10	15	93.12	4.794
10	35	10	15	20	10	93.11	4.804
10	35	10	15	10	20	93.12	4.824
10	35	10	20	15	10	93.14	5.113

10	35	10	20	10	15	93.18	4.831
10	35	10	10	15	20	93.12	5.034
10	35	10	10	20	15	93.15	5.123
10	15	35	20	10	10	93.37	5.002
10	15	35	10	20	10	93.48	4.811
10	15	35	10	10	20	93.53	4.829
10	15	20	35	10	10	93.36	5.006
10	15	20	10	35	10	93.51	4.849
10	15	20	10	10	35	93.93	4.937
10	15	10	35	20	10	93.39	4.976
10	15	10	35	10	20	93.46	4.954
10	15	10	20	35	10	93.46	4.924
10	15	10	20	10	35	93.9	4.867
10	15	10	10	35	20	93.35	4.805
10	15	10	10	20	35	93.82	4.799
10	20	35	15	10	10	93.35	4.895
10	20	35	10	15	10	93.38	5.071
10	20	35	10	10	15	93.41	4.863
10	20	15	35	10	10	93.36	5.3
10	20	15	10	35	10	93.42	5.055
10	20	15	10	10	35	93.85	4.802
10	20	10	35	15	10	93.38	4.89
10	20	10	35	10	15	93.42	4.783
10	20	10	15	35	10	93.4	4.854
10	20	10	15	10	35	93.84	5.176
10	20	10	10	35	15	93.37	4.834
10	20	10	10	15	35	93.76	4.731
10	10	35	15	20	10	93.48	4.726
10	10	35	15	10	20	93.51	4.801
10	10	35	20	15	10	93.39	4.899

10	10	35	20	10	15	93.43	4.793
10	10	35	10	15	20	93.54	5.167
10	10	35	10	20	15	93.55	4.909
10	10	15	35	20	10	93.4	4.967
10	10	15	35	10	20	93.46	4.841
10	10	15	20	35	10	93.48	5.004
10	10	15	20	10	35	93.96	4.793
10	10	15	10	35	20	93.37	5.28
10	10	15	10	20	35	93.88	4.804
10	10	20	35	15	10	93.37	4.982
10	10	20	35	10	15	93.41	4.823
10	10	20	15	35	10	93.51	4.803
10	10	20	15	10	35	93.99	4.806
10	10	20	10	35	15	93.49	4.812
10	10	20	10	15	35	93.92	5.006
10	10	10	35	15	20	93.47	4.969
10	10	10	35	20	15	93.47	4.797
10	10	10	15	35	20	93.35	4.891
10	10	10	15	20	35	93.87	4.796
10	10	10	20	35	15	93.44	4.776
10	10	10	20	15	35	93.87	5.344

בממוצע עבור כל 381 הרשומות שנאספו למטרת סטטיסטיקה :

זמן חישוב ציון המערכת הממוצע : 4.9 שניות

הציון הממוצע במערכת : 93.57

הציון המקסימלי במערכת : 94.09

נספח ג' – קוד
דיסק עם קוד המקור של הפרויקט
מצורף לספר הפרויקט.

עמוד	כותרת
8	והמחקר בהם נעשה שימוש בלמידה עמוקה
8	צורת מהמילה Yes מתוך מאגר הנתונים לאימון
8	צורת מהמילה Down מתוך מאגר הנתונים לאימון
10	
12	
13	
18	
23	
24	
26	
27	
29	Python להמרת קבצי אודיו בשפת
30	Flask השימוש ב
32	
35	למול איכות הזיהוי של הרשת תוך כדי האימון
35	למול איכות הזיהוי של רשתות DenseNet ו ResNet תוך כדי
37	
37	
38	
39	
40	
41	י הפלט
41	
42	
44	
45	
46	
40	
50	
51	Majority Voting וקבלת הציון הסופי של המערכת
52	Resnet אשר נלמד באמצעות Mel Spectrograms Images
52	משתמש
52	הציון והזיהויים הנכונים של המערכת עבור ערכי המשקלים
54	ערכי ברירת המחדל
56	ינם לשלוח ערכים שאינם שווים ל100 דרך ממשק המשתמש
60	
61	
66	ותקדמים שלו מתוך האתר של NVIDIA DGX

נספח ד' – התקנת המערכת

דרישות המערכת

1. מערכת הפעלה התומכת בpython3, עדיפות לubuntu linux
2. ספריות הpython הבאות :

1. csv
2. pathlib
3. math
4. torch
5. random
6. pandas
7. numpy
8. skimage
9. glob
10. librosa
11. operator
12. sys
13. os
14. json
15. flask
16. flask_socketio

שלבי התקנת המערכת

התקנת המערכת דרך הterminal של מערכת ההפעלה ubuntu מסוג linux :

sudo apt-get install python3
sudo apt-get install pip3
sudo pip3 install csv
sudo pip3 install pathlib
sudo pip3 install math
sudo pip3 install torch
sudo pip3 install random
sudo pip3 install pandas
sudo pip3 install numpy
sudo pip3 install skimage
sudo pip3 install glob
sudo pip3 install Librosa
sudo pip3 install operator
sudo pip3 install sys
sudo pip3 install os
sudo pip3 install json
sudo pip3 install flask

הוראות הפעלה

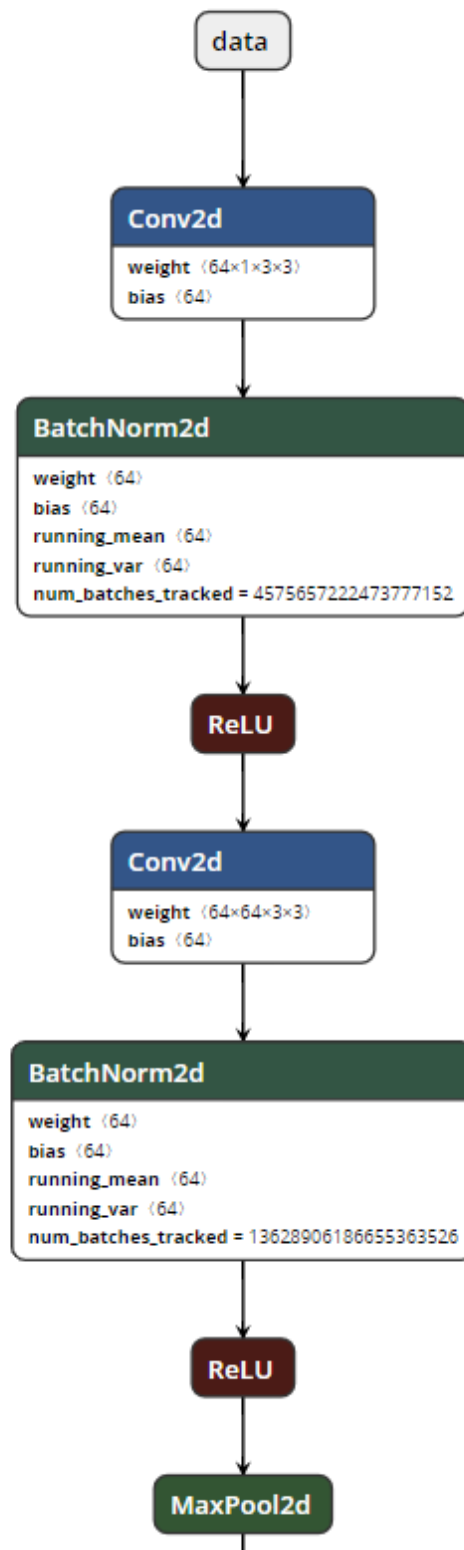
פקודה להפעלת ממשק המשתמש מתוך terminal של לינוקס :

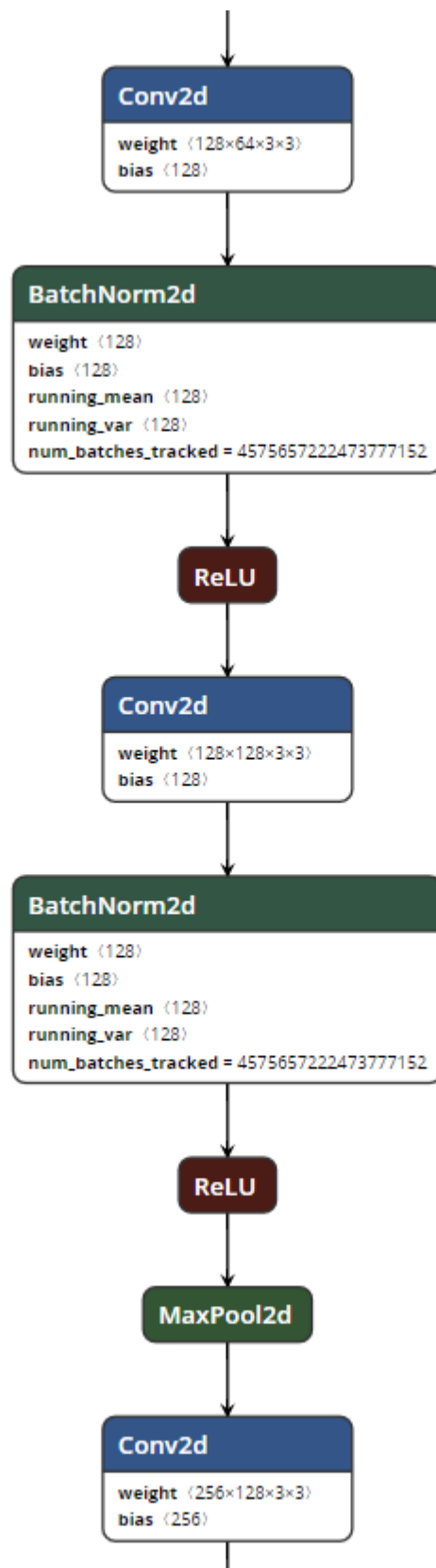
python3 main.py

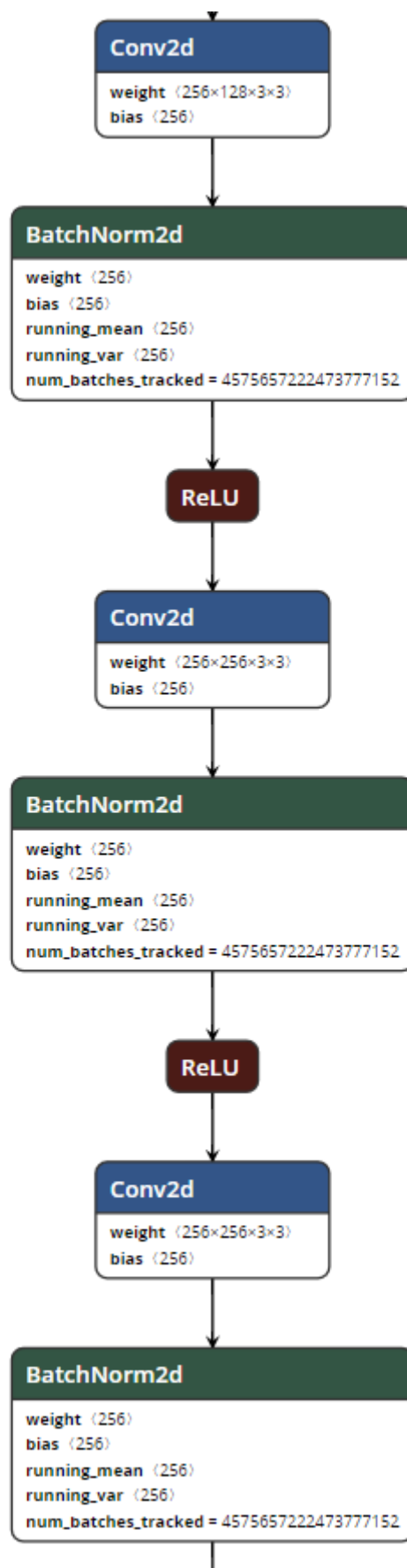
לאחר מכן על המשתמש לפתוח דפדפן אינטרנט התומך בHTML5 (בעדיפות לגוגל כרום או מוזילה פיירפוקס) ולהיכנס לכתובת הבאה :

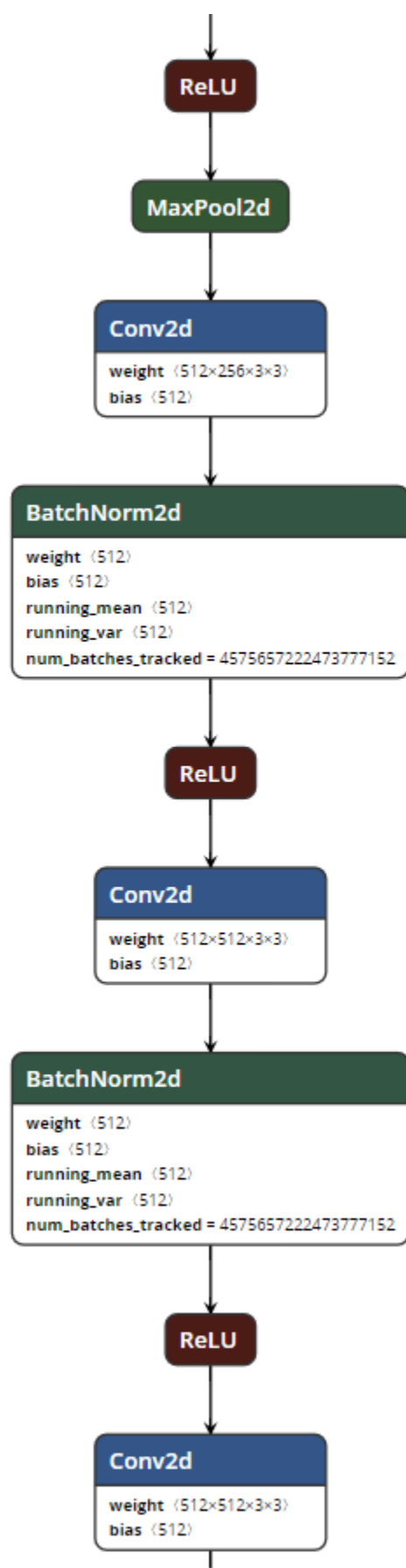
<http://localhost:5000>

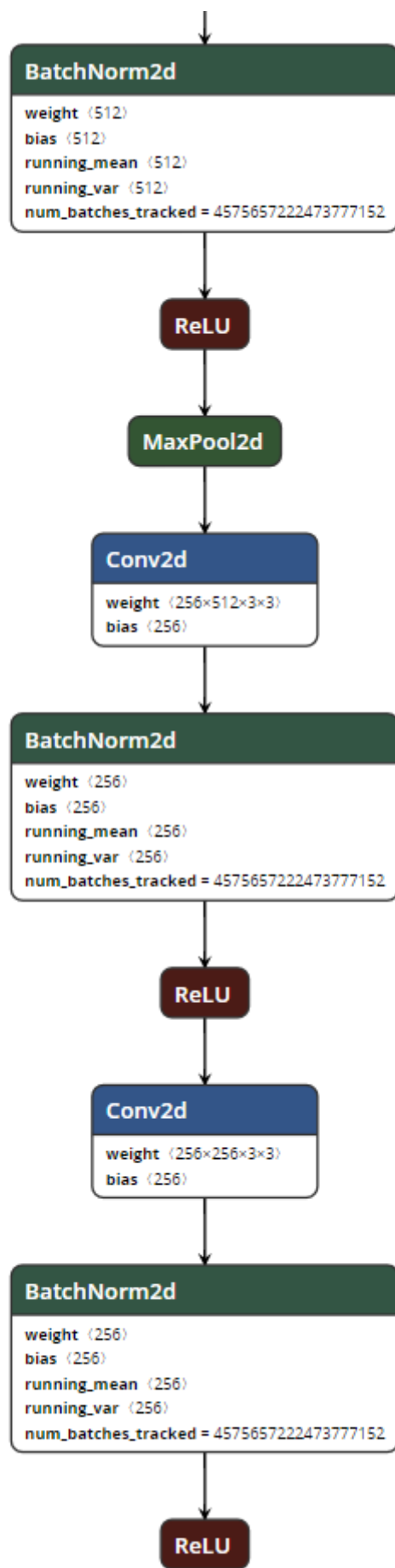
הגרף הנספח עבור רשת VGG-2D נוצר בעזרת כלי הוויזואליזציה Netron [22]

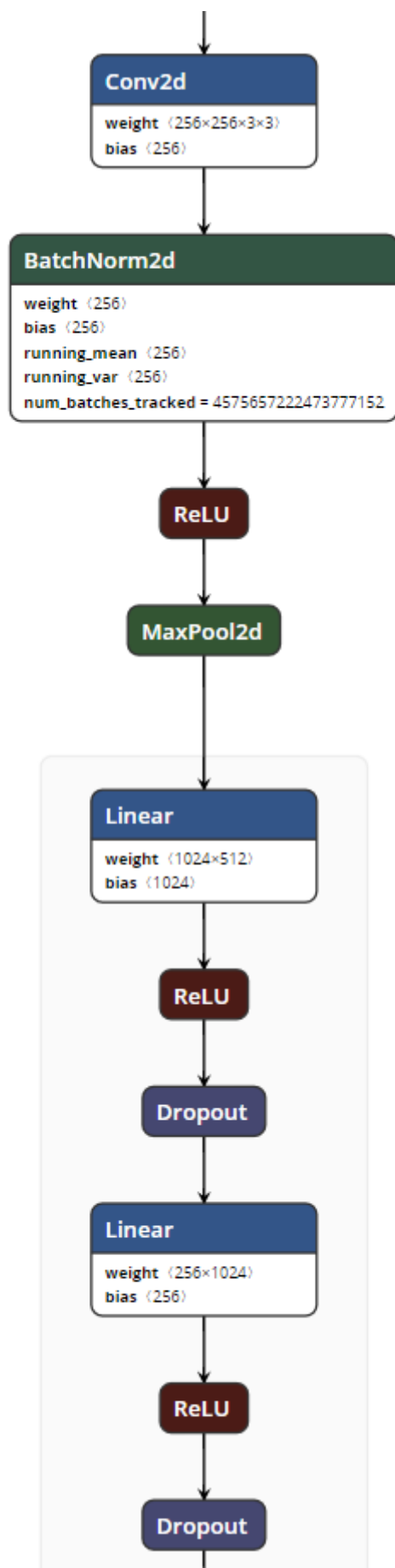


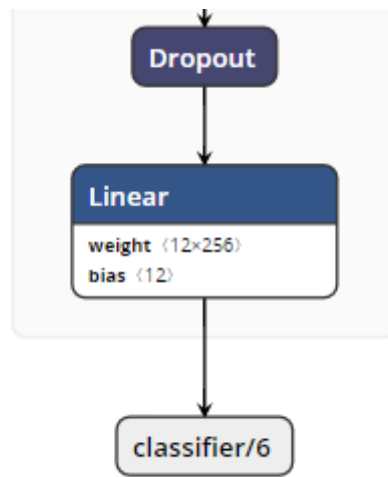






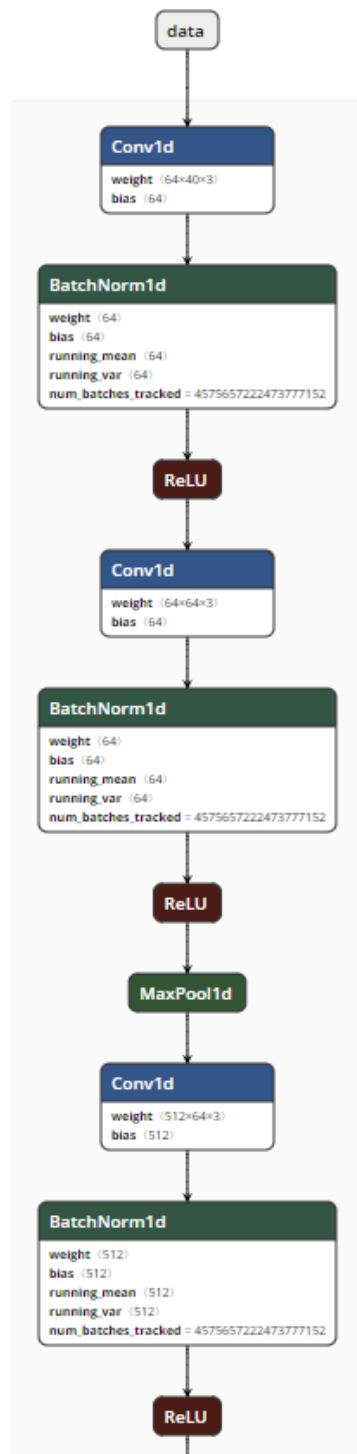


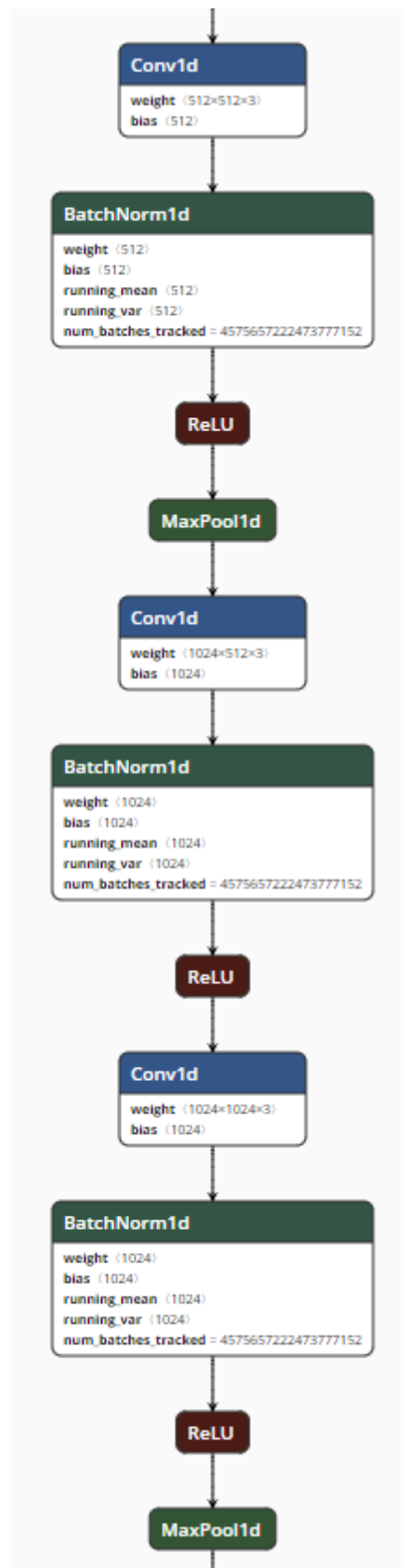


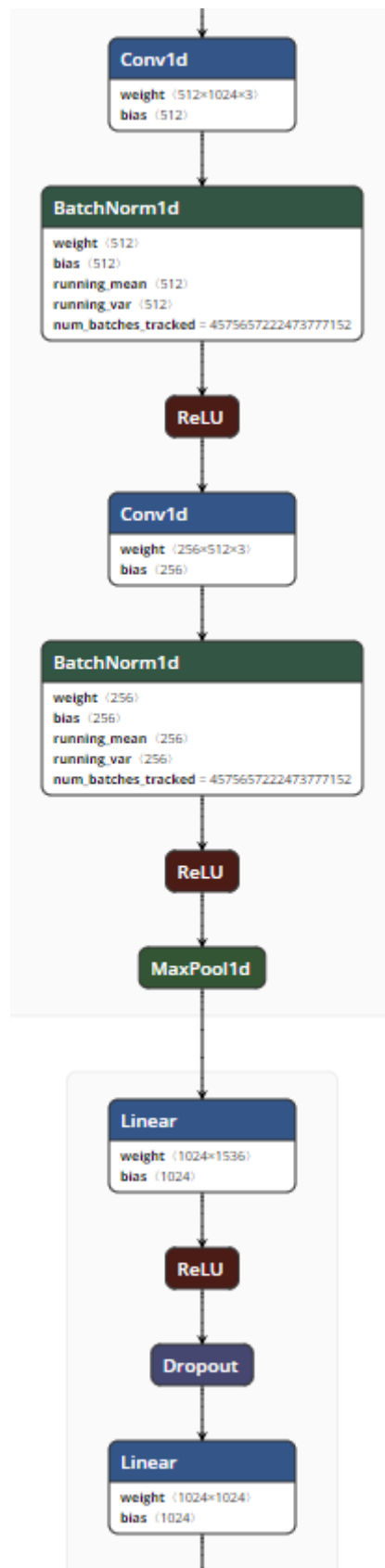


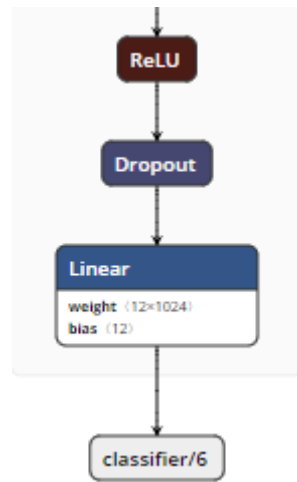
נספח ו' – מבנה רשת הנוירונים העמוקה VGG-1d עבור Mel Spectrograms

הגרף הנספח עבור רשת VGG-1D נוצר בעזרת כלי הוויזואליזציה Netron [22] ע"י טעינת המודל שנוצר בשלב האימון לממשק הweb של Netron.



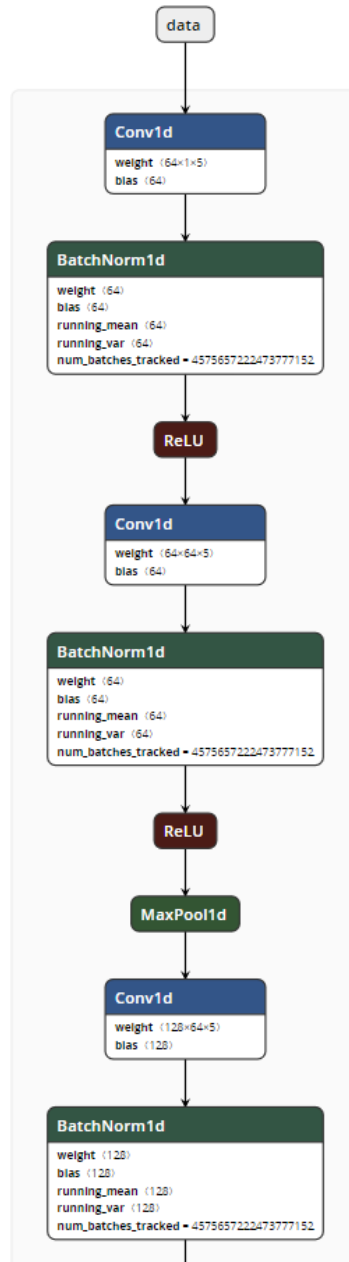


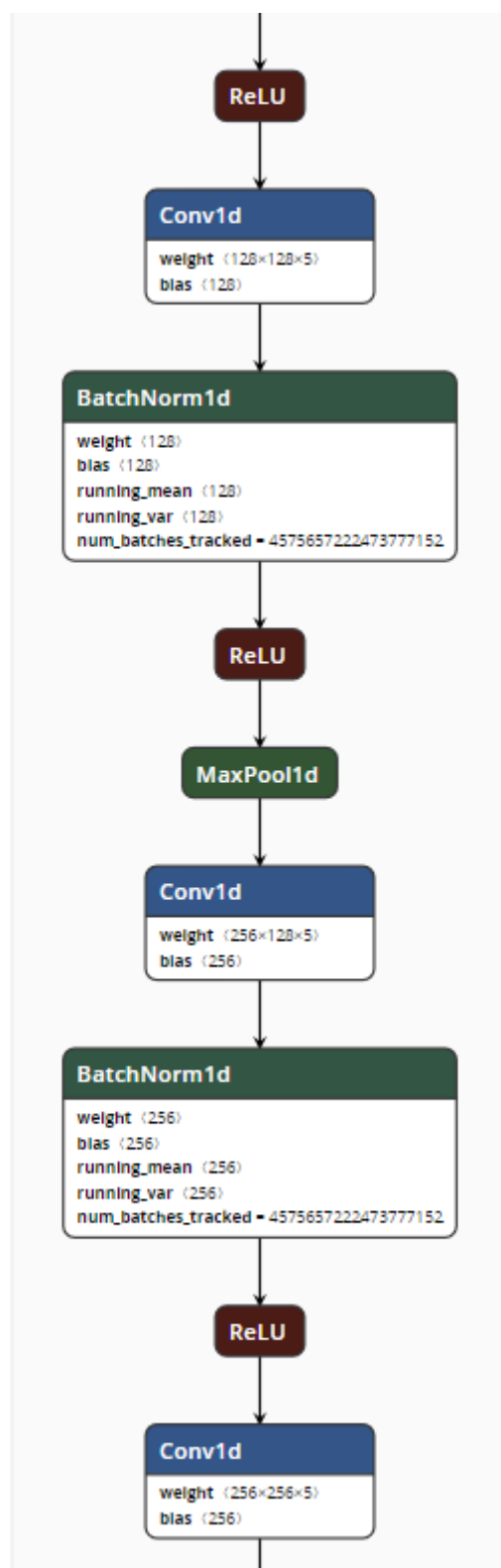


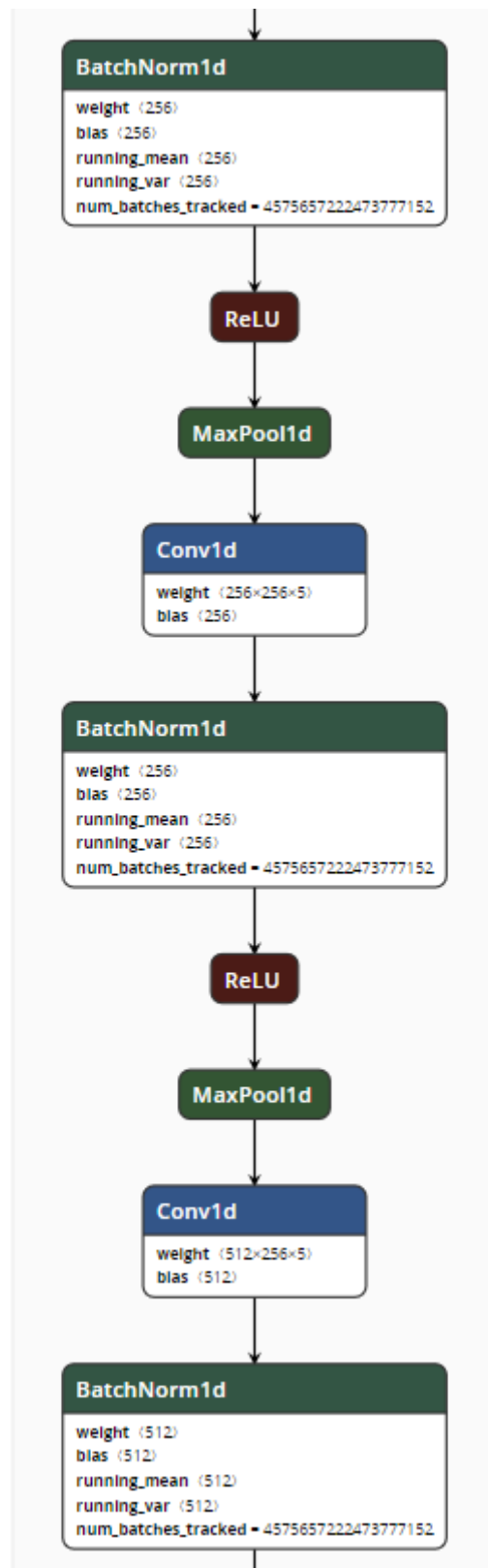


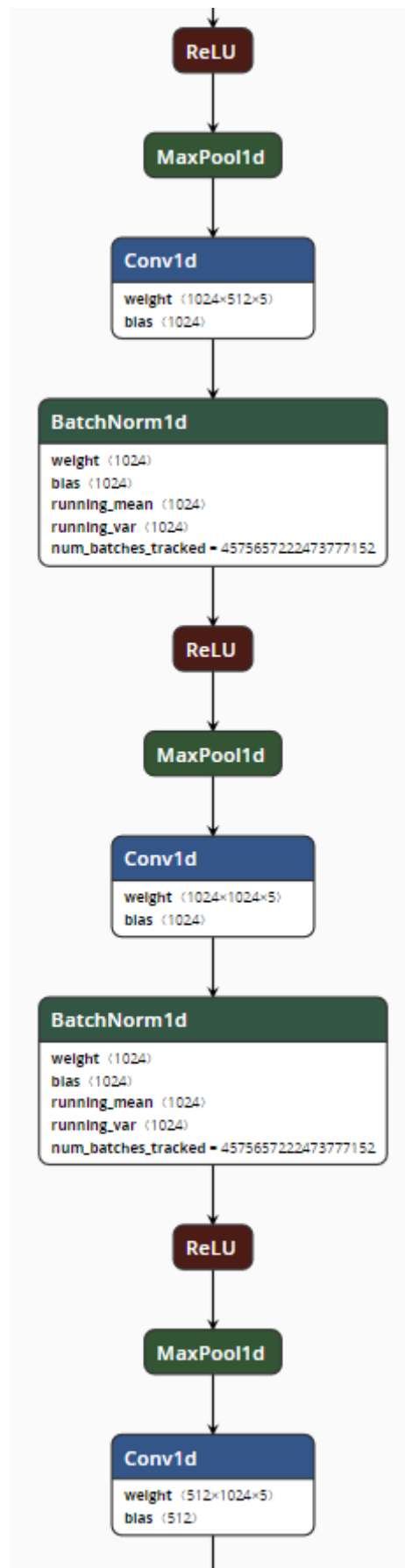
נספח ז' – מבנה רשת הנורונים העמוקה VGG-1d

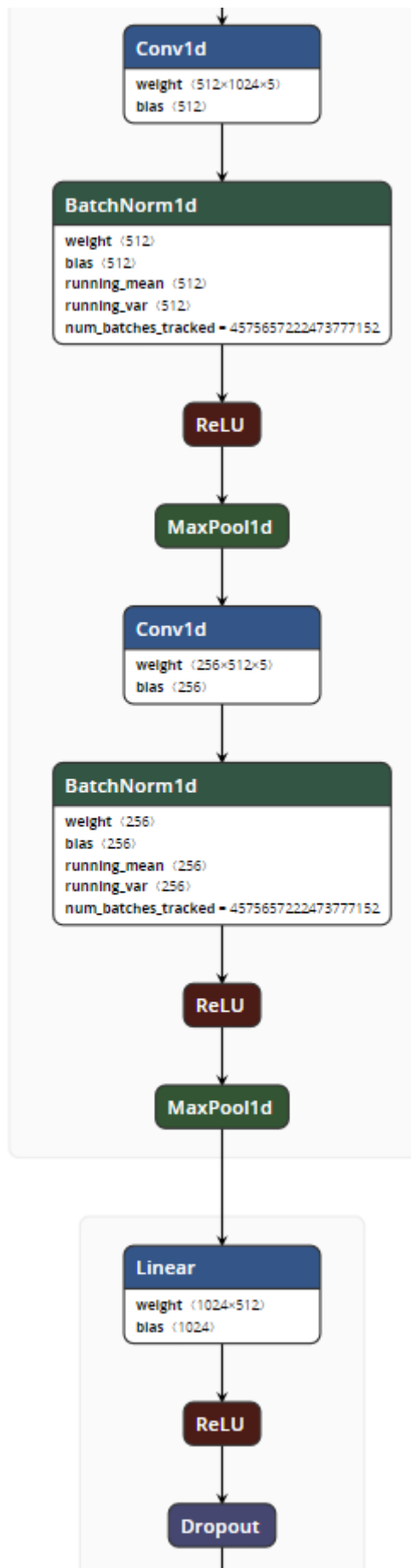
הגרף הנספח עבור רשת VGG-1D נוצר בעזרת כלי הוויזואליזציה Netron [22] ע"י טעינת המודל שנוצר בשלב האימון לממשק הweb של Netron.

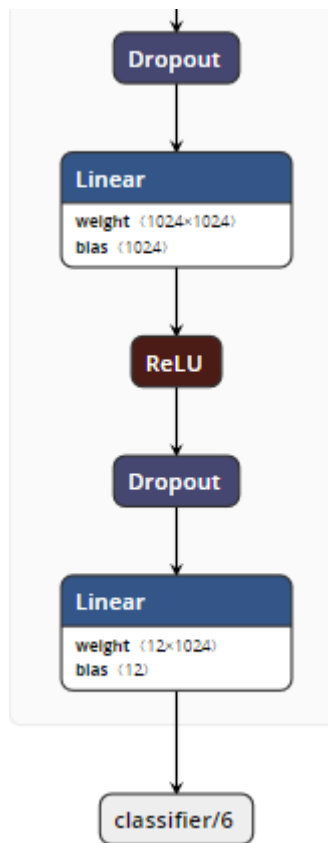




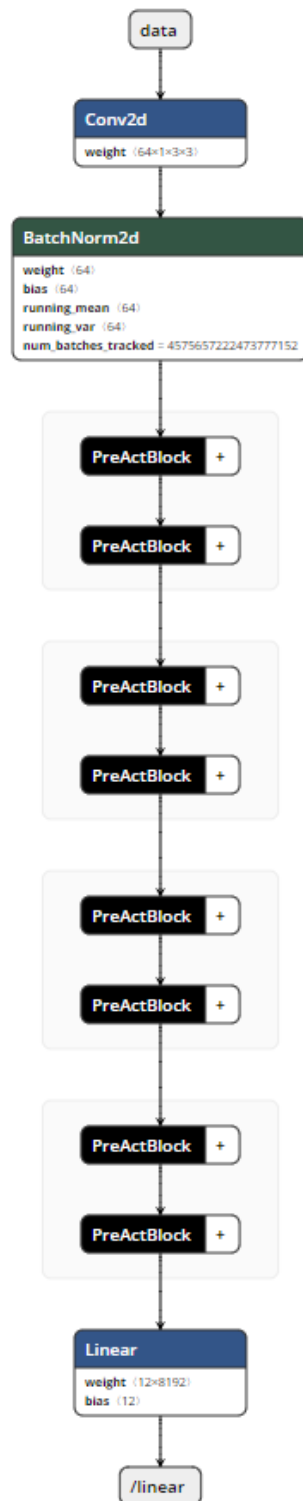






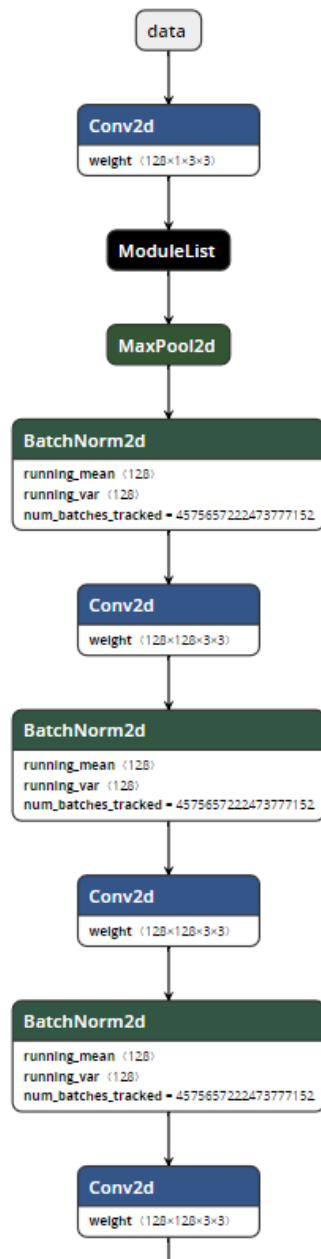


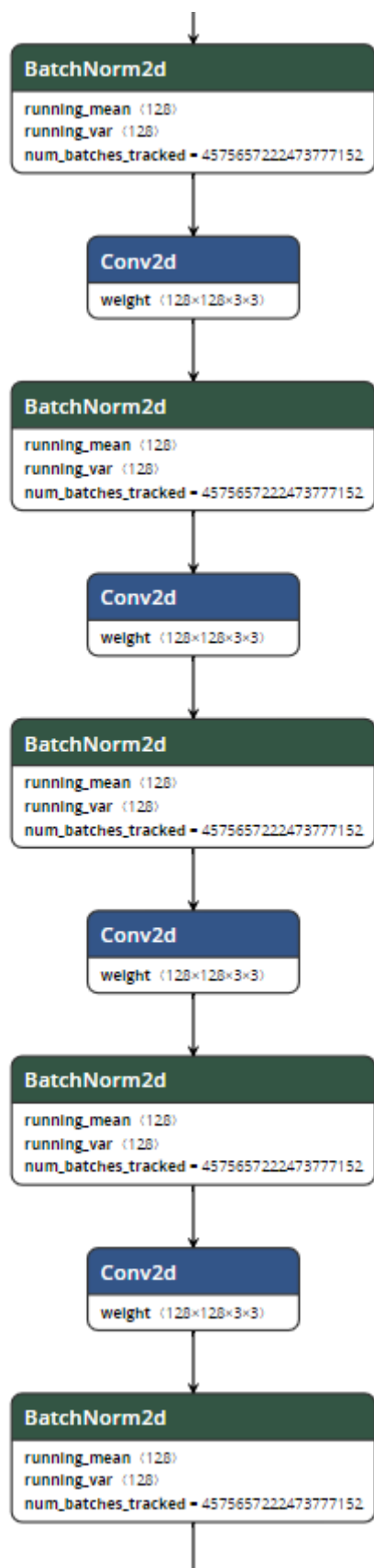
הגרף הנספח עבור רשת Senet נוצר בעזרת כלי הוויזואליזציה Netron [22] ע"י טעינת המודל שנוצר בשלב האימון לממשק הweb של Netron.

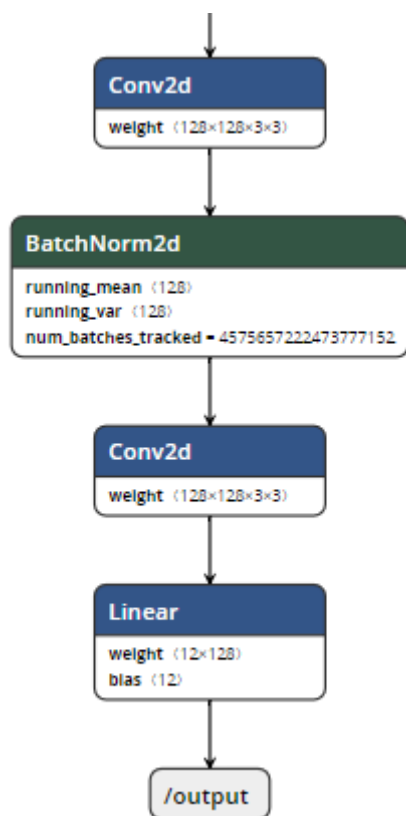


נספח ט' – מבנה רשת הניורונים העמוקה Resnet

הגרף הנספח עבור רשת Resnet נוצר בעזרת כלי הוויזואליזציה Netron [22] ע"י טעינת המודל שנוצר בשלב האימון לממשק הweb של Netron.

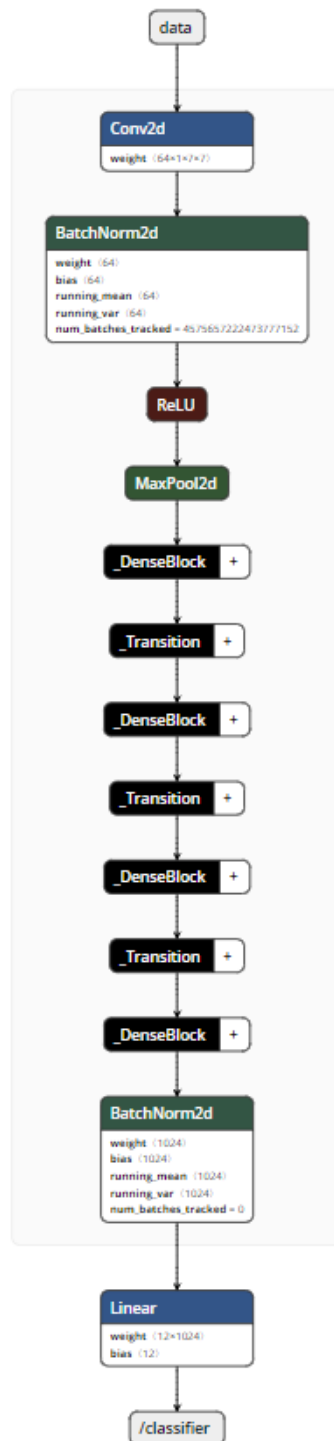






נספח י' – מבנה רשת הנוירונים העמוקה DenseNet

הגרף הנספח עבור רשת DenseNet נוצר בעזרת כלי הוויזואליזציה Netron [22] ע"י טעינת המודל שנוצר בשלב האימון לממשק הweb של Netron.



נספח י"א – מאפייני רשתות הנוירונים השונות

המאפיינים הבסיסיים המתקבלים מקובץ Pth שנוצר עבור כל רשת :

מאפיינים	רשת
MODEL PROPERTIES × format: PyTorch Graph type: vgg2d.VGG Inputs data: Id: data Outputs classifier/6: Id: classifier/6	VGG-2D
MODEL PROPERTIES × format: PyTorch Graph type: vgg1d.VGGMEL Inputs data: Id: data Outputs classifier/6: Id: classifier/6	VGG-1D For Mel Spectrograms
MODEL PROPERTIES × format: PyTorch Graph type: vgg1d.VGG1D Inputs data: Id: data Outputs classifier/6: Id: classifier/6	VGG-1D
MODEL PROPERTIES × format: PyTorch Graph type: senet.SENet Inputs data: Id: data Outputs /linear: Id: /linear	SENet

MODEL PROPERTIES format PyTorch Graph type resnet.ResModel Inputs data Id: data Outputs /output Id: /output	ResNet
MODEL PROPERTIES format PyTorch Graph type densenet.DenseNet Inputs data Id: data Outputs /classifier Id: /classifier	DenseNet

נספח י"ב – הצעת פרויקט הצעה לשיפור הביצועים

מציאת המשקלים האופטימליים בצורה אוטומטית

בממשק המשתמש הקיים, המשתמש מתבקש להזין את משקלי המודלים השונים על מנת לקבל את ההחלטה הסופית האופטימלית עבור קבצי הבדיקה.

ניתן לפתח אלגוריתמים שונים לחישוב המשקלים האידיאליים למודלים השונים ובכך לוותר על הצורך של הזנת משקלי המודלים ידנית ע"י המשתמש.

האלגוריתם הפשוט ביותר לחישוב כזה הינו אלגוריתם מסוג Brute Force אשר ינסה להריץ את כל האפשרויות השונות למשקלים עבור כל אחד מהמודלים השונים.

אלגוריתם כזה יכול להיות כבד ואיטי מידי עבור שלב הonline אך אם הוא יורץ פעם אחת בשלב הOffline (בהנחה שהמודלים השונים קבועים) ניתן יהיה למצוא את כל המשקלים האידיאליים אשר נותנים תוצאה מקסימלית ללא צורך בהתערבות המשתמש.

שיפור ביצועי המסווג

בבדיקה על מחשב ביתי – זמן החיזוי לוקח כ-5 שניות.

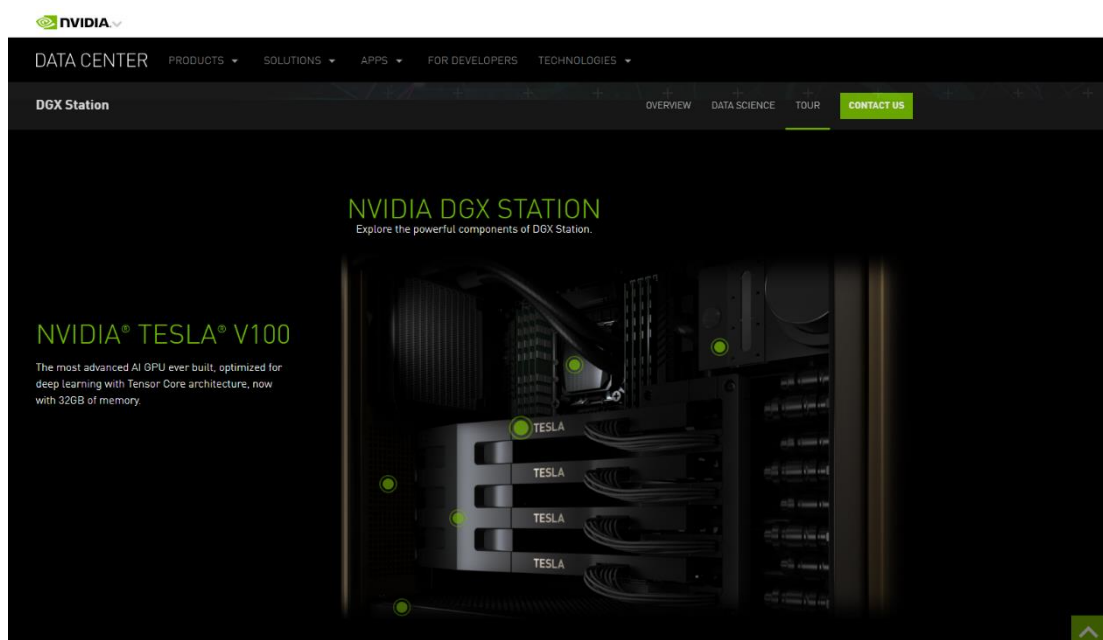
ניתן לשפר את זמן החיזוי יכול להתבצע ע"י:

- מקבול - שימוש בתהליכים מקבילים בעבודה עם מערכת הזיהוי וקבצי הCSV
- בLinux ניתן להחזיק את קבצי הCSV ישירות על זיכרון הRAM ובכך לחסוך את הזמן שלוקח למערכת לטעון אותם לזיכרון.

המחשב הביתי שלי מכיל GPU סטנדרטי של NVIDIA, זמני האימון בו היו ארוכים מאוד – זמן האימון עבור מודל בודד ארך כ-2 ימים, עבור אימון של 6 מודלים לפחות (ללא שלבי הניסוי והטעייה אשר במהלכם מודלים שאומנו בצורה לא נכונה נזרקו) נדרשה תקופה של כמעט שבועיים בהם המחשב הביתי מאמן מודלים ומושבת לצרכים ופעילויות אחרות.

על מנת לשפר את זמני אימון המודלים ניתן להשתמש במחשבים בעלי מספר רב של מעבדים גרפיים כמו למשל מחשב העל של NVIDIA מסוג NVIDIA DGX, אשר נכון לזמן כתיבת שורות אלו הוא נחשב כ state of the art בעולם הלמידה העמוקה.

זמני האימון של מודלי רשתות נוירונים במחשב על שכזה יכולים לרדת בצורה משמעותית כך שאימון של 6 מודלי רשתות נוירונים שונים בגישת למידה עמוקה יכול להסתיים ביומיים בודדים.



איור 38 – NVIDIA DGX עם 4 המעבדים הגרפיים המתקדמים שלו - מתוך האתר של NVIDIA

פרויקט זה מהווה בסיס לעבודה המסכמת שלי בנושא של זיהוי קולי באמצעות שיטות וגישות מעולם הלמידה העמוקה.

קורס: פרויקט מתקדם במדעי המחשב - 22997

מגיש: בן זגגי ת.ז: 300924859

שם הפרויקט: מערכת חכמה לזיהוי וסיווג פקודות קוליות

שם מנחה: ד"ר מיה הרמן

מטרת הפרויקט

מטרת הפרויקט הינה לבנות מערכת חכמה לזיהוי וסיווג פקודות קוליות אשר עומדת בתנאי התחרות "Tensorflow Speech Recognition Challenge" של Kaggle

באתגר זה נתונים שני מאגרי מידע משמעותיים:

מאגר קטעי הקול הנועד לאימון המערכת (Train Set) – מאגר קטעי קול המכיל 65,000 קבצים אשר הוקלטו ע"י אלפי אנשים שונים. קטעי הקול הוקלטו ע"י אלפי אנשים שונים, כאשר בכל קובץ כזה מופיעה פקודה בודדת.

מאגר קטעי הקול הנועד לבדיקת המערכת (Test Set) – מאגר זה מכיל כמעט 160,000 קבצים אשר בדיוק כמו Training Set הוקלטו ע"י אלפי אנשים שונים ומכילים בתוכם פקודות קוליות

שונות. אך בשונה ממאגר האימונים, קבצי הקול שבמאגר זה אינם מתויגים ומקוטלגים לפי תוכנם, שכן הזוכה בתחרות הוא זה שיצליח לקטלג אותם בצורה הכי נכונה מכל משתתפי התחרות.

מאגרי הנתונים הנ"ל פתוחים לציבור הרחב ונמצאים בכתובת:

<https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data>

האתגר העיקרי בתחרות זו הינו בניית אלגוריתם חזק מספיק כך שבעזרת שימוש במאגר האימונים בלבד נוכל לזהות ולסווג את קטעי הקול ממאגר הבדיקות.

על מנת לבדוק את איכות המערכת שאבנה בפרויקט, אשתמש בפלט אשר התקבל מהאלגוריתם המנצח בתחרות ואשווה אותו לפלט של המערכת שלי.

רוב האלגוריתמים הקיימים היום בעולם הלמידה העמוקה מכוונים לסיווג בין תמונות בלבד. רשתות למידה עמוקה המסווגות תמונות מקבלות מידע בצורה דו ממדית עם משקלים שונים לאורך צירי X ו Y . גישה זו מתבססת על כך שאובייקט בתמונה ייראה אותו הדבר ללא קשר במיקום שלו בתוך התמונה. עבור ייצוג של קטעי קול גישה זו יכולה להיות נכונה כאשר קטעי קול מוזזים לאורך ציר ה X המייצג את ציר הזמן, אך לא כאשר הם מוזזים לאורך ציר ה Y המייצג את התדירות. (9)

זיהוי קולי הוא תהליך המשמש לזהות דיבור שנאמר ע"י אדם, תחום זה נחקר כבר למעלה מחמישה עשורים מאז שנות ה-50. תקשורת קולית היא דרך התקשורת השימושית והאפקטיבית ביותר בין בני האדם.

השימוש בטכנולוגיה לזיהוי קולי הוא תחום רחב אשר מתקדם בצעדי ענק עם פוטנציאל ענק לא פחות.

השימוש באלגוריתמי למידה עמוקה עבור נתוני אותות ונתוני קול הינו תחום מחקרי המתפתח בעת הזו (5)

אחת ההחלטות המתעוררות בעת עיצוב רשת ללמידה עמוקה (לכל יישום) היא קביעת מבנה נתוני הקלט לרשת.

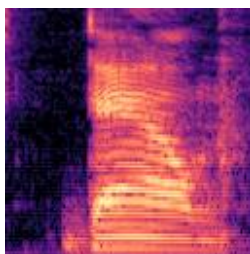
על מנת להשתמש באלגוריתמי למידה עמוקה מוכחים, יש להעביר את מרחב הבעיה מתחום האותות והקול לעולם התמונה. טרנספורמציה זו תאפשר שימוש באלגוריתמים חדשניים הקיימים כיום כ open-sources.

המרת קול לתמונה בייצוג ספקטרלי

קבצי קול יכולים להיות מיוצגים בדרכים שונות כתמונות. עיצוב ובחירת הייצוג הנכון הוא מרכיב קריטי בהצלחת יכולת הפרויקט לזהות קטעי קול.

ייצוגים ספקטראליים עשויים להתאים לשימוש ברשתות למידה עמוקה שכן הם מכילים יותר מידע מרוב צורות הייצוג המסורתיות למרחב הקול, כמו כן הם בעלי מממד נמוך יותר מקובץ קול גולמי (raw). (8)

בפרויקט זה אשתמש בהמרות למרחב הספקטראלי ואותן כאמור אזין למערכת ללמידה עמוקה.



דוגמא לתמונת Mel Spectrogram הנוצרת מהמילה Down מתוך מאגר הנתונים לאימון

שימוש ברשתות למידה עמוקה קיימות לצורך Transfer Learning

הדרך האפקטיבית ביותר לאימון מודל רשת נוירונים עמוקה על קלט חדש הינו אימון של רשת קיימת בעלת משקלים שונים על קבצי הקלט לאימון הקיימים, גישה זו נקראת Transfer Learning (7).

לאחר מעבר על פורום התחרות ועל פרויקטים שונים אשר זכו במקומות גבוהים בתחרות החלטתי להשתמש במודלים הבאים כבסיס לרשת הנוירונים שלי :

VGGNet

הוצגה לראשונה ב-2014 ע"י Karen Simonyan ו Andrew Zisserman, המאמר המתאר את הרשת הזו זכה ליותר מ-10,000 ציטוטים מאז שעלה לאוויר. המאמר הנ"ל מתייחס לחלק חשוב מאוד בארכיטקטורה של רשתות נוירונים ללמידה עמוקה. (15)

ResNet

הוצגה לראשונה ב-2014 ע"י Kaiming He ממיקרוסופט, המאמר המתאר את הרשת הזו זכה ליותר מ-8,000 ציטוטים מאז שעלה לאוויר. (14)

DenseNet

הוצגה לראשונה ב-2016 ע"י Gao Huang, Zhuang Liu, Kilian Q. Weinberger, המאמר המתאר את הרשת הזו זכה לקרוב ל-1000 ציטוטים מאז שעלה לאוויר, שמה של הרשת מתייחס ל-Densely Connected convolutional. (12)

SeNet

הוצגה לראשונה ב-2017 ע"י Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, המאמר המתאר את הרשת הזו זכה לקרוב ל-100 ציטוטים מאז שעלה לאוויר, רשת זו נבחרה למנצחת האתגר של ImageNet לשנת 2017. שמה של הרשת מתייחס ל Squeeze and Excitation. (13)

הפרויקט מורכב מארבעה שלבים עיקריים :

1. המרת נתוני הקלט למרחב התמונה.
2. אימון מודלים שונים מהתמונות המתקבלות בשלב 1 בגישת Transfer-Learning המתבססים על סוגי מודלים קיימים.
3. ביצוע Majority Voting על תוצאות המודלים שהתקבלו בשלב 2 מול מאגר קטעי הקול הנועד לבדיקת המערכת.
4. השוואת תוצאות שלב Majority Voting לתוצאות המקוריות של הזוכה באתגר, לאחר מכן ניתוח התוצאות אותן קיבלתי והסקת המסקנות המתאימות.

שלב 1 – המרת נתוני הקלט למרחב התמונה

כפי שציינתי בהקדמה זהו החלק החשוב ביותר, בחירה נכונה של מרחב התמונה הנכון ינפיק תוצאות טובות ובחירה לא נכונה ינפיק תוצאות לא שמישות.

בשלב זה אשתמש בספריית Librosa אשר כתובה בקוד פתוח (6), ספרייה זו נותנת אפשרות המרה טובה מקובץ קול ל Mel Spectrogram במרחב התמונה.

שלב 2 – אימון מודלים קיימים בגישת Transfer Learning עבור המידע המתקבל מהשלב הקודם

שלב זה הוא קריטי לא פחות מהשלב הקודם שכן המודלים השונים יהוו את לב המערכת שבפרויקט, אימון נכון ובחירת מודלים בסיסיים מתאימים ייתנו תוצאות טובות.

לאחר מעבר על פורום התחרות ועל פתרונות שונים מתוך התחרות אשר קיבלו ציון גבוה וזכו במקומות גבוהים בתחרות, נוכחתי לראות את אותם מודלים בסיסיים לרשתות נוירונים עמוקות חוזרים על עצמם כמעט עבור כמעט כל פתרון ומערכת שנבנתה, לכן אשתמש בהם גם כבסיס לרשתות הנוירונים העמוקות אותן אאמן לצורך הפרויקט, רשתות הנוירונים העמוקות הללו הן :

- **ResNet** – אתבסס על לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV
- **SeNet** – אתבסס על לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV
- **DenseNet** – אתבסס על לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV
- **VGG2d** – אתבסס על לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV
- **VGG1d** – אתבסס על לימוד MEL SPECTROGRAMS הנוצרות מקובץ WAV וכמו כן על קבצי ה WAV כ RAW Data ללא המרה

שלב 3 – ביצוע Majority Voting על תוצאות המודלים שהתקבלו בשלב 2 מול מאגר קטעי הקול

בשלב זה אחליט כמה "כוח" לתת לכל מודל מתוך ששת המודלים שיצרתי על מנת לבצע ממוצע משוקלל לבחירת החיזוי הנכון. כאמור, כחלק מהאתגר של Kaggle, מצורפים כ-158,536 קבצי אודיו לבדיקות ולדירוג המודלים. לאחר שלב האימונים, אפעיל את המודלים השונים על קבצים אלו ואשמור את התוצאות אותן הוא נותן בצורה אוטומטית לקובץ CSV.

עבור קובץ אודיו בדיקה מסוים אשתמש בסיווג מתוך 6 המודלים על מנת לקבוע את הציון הסופי של אותו קובץ מתוך המערכת.

מודל הניקוד הבסיסי בו אשתמש הינו :

- מודל Resnetn המבוסס על Mel Spectrograms : יקבל 15% מהניקוד עבור כל קובץ
 - מודל SeNetn המבוסס על Mel Spectrograms : יקבל 15% מהניקוד עבור כל קובץ
 - מודל DenseNetn המבוסס על Mel Spectrograms : יקבל 15% מהניקוד עבור כל קובץ
 - מודל VGG2Dn המבוסס על Mel Spectrograms : יקבל 15% מהניקוד עבור כל קובץ
 - מודל VGG1Dn המבוסס על Mel Spectrograms : יקבל 10% מהניקוד עבור כל קובץ
 - מודל VGG1Dn המבוסס על Raw WAV Data : יקבל 30% מהניקוד עבור כל קובץ
- במקרה של שוויון אבחר בסיווג מסוג Unknown.

דוגמא :

אם עבור קובץ אודיו כלשהו test.wav יתקבלו את הסיווגים הבאים מכל מודל :

מודל	סיווג
מודל Resnetn המבוסס על Mel Spectrograms	Unknown
מודל SeNetn המבוסס על Mel Spectrograms	No
מודל DenseNetn המבוסס על Mel Spectrograms	No
מודל VGG2Dn המבוסס על Mel Spectrograms	Unknown
מודל VGG1Dn המבוסס על Mel Spectrograms	Yes
מודל VGG1Dn המבוסס על Raw WAV Data	Yes

מערכת Majority Voting תסווג אותו כקובץ אודיו המכיל את המילה Yes, שכן הציונים עבורו יהיו :

סיווג	ציון לפי הדוגמא
Unknown	$15+15 = 30$
No	$15+15 = 30$
Yes	$10+30 = 40$

כמו כן אפרט בהמשך על ממשק משתמש אותו בכוונתי לחבר לפרויקט אשר יעזור למצוא את הניקוד המתאים לכל מודל וקבלת התוצאות עבורו.

שלב 4 – השוואת תוצאות שלב ה Majority Voting לתוצאות המקוריות של הזוכה באתגר

לאחר שלב ה Majority Voting אקבל תוצאת חיזוי עבור כל אחד מ-158,536 קבצי האודיו לבדיקת האלגוריתם, בשלב 4 אשווה את תוצאות החיזוי שהתקבלו לתוצאות החיזוי של הזוכה בתחרות אשר כאמור החיזוי שלו יהיה הכי נכון – כך אוכל לקבל תוצאה אמינה על איכות פתרון האלגוריתם שלי.

תיאור הממשק גרפי

אגדיר ממשק גרפי לאינטראקציה פשוטה עם המשתמש עבור שני השלבים האחרונים :

שלב 3 בו מתבצע Majority Voting ושלב 4 בו אני משווה את הסיווגים השונים לסיווגים של המנצח בתחרות.

הממשק הגרפי יתקשר עם ספריות הפרויקט הכתובות בPython3 בעזרת ספריית פיתון התומכת בממשקים ויזואליים.

Running Mel Spectrograms Images as Input for Model:

Resnet Model percentage: 15%

Senet Model percentage: 15%

Densenet Model percentage: 15%

VGG2d Model percentage: 15%

VGG1d Model percentage: 10%

Running Raw Data as Input for Model:

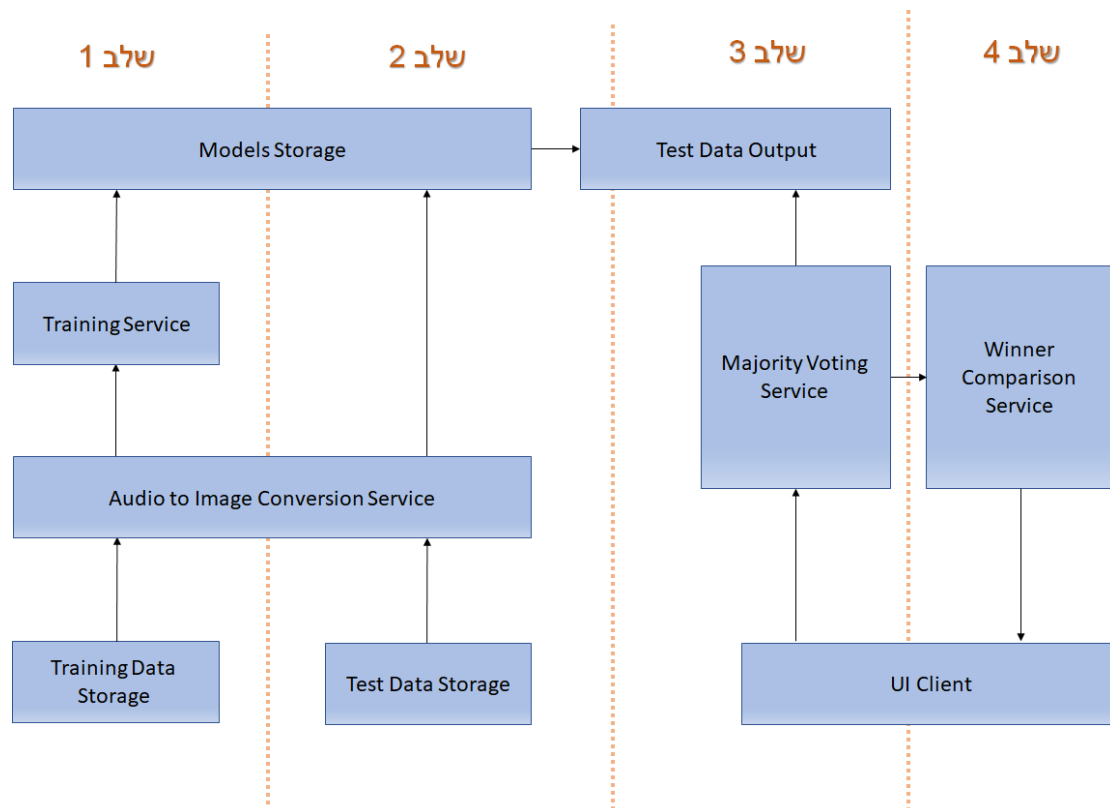
VGG1d Model percentage: 30%

Calculate Score

סקיצה לממשק הגרפי

לאחר לחיצה על כפתור Calculate Score , המערכת תריץ את המודלים השונים על כל קבצי test, הסיווג של כל קובץ יתקבל ע"י ממוצע משוכלל לפי הערכים אותם יזין המשתמש לגבי הסיווג מכל מודל והשוואת סיווג זה לסיווגים שהתקבלו מהפרויקט של המנצח בתחרות. לאחר החישוב יוצג למשתמש הציון המתקבל עבור הערכים אותם הזין למערכת לביצוע הממוצע המשוקלל.

Components Diagram



תיאור רכיבים עיקריים

• Training Data Storage

כאן יוחזק כל המידע הנדרש ל-Training, מדובר על 64,732 קבצי אודיו מסוג wav המקוטלים לפי תוכן המילה שנאמרת בהם לתיקיות שונות, המילים האפשריות ע"פ דרישות התחרות הן:

Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Silence, Unknown

• Test Data Storage

כאן יוחזק כל המידע הלא מקוטלג הנדרש על מנת לבדוק את איכות המערכת שלי, מדובר על 158,536 קבצי אודיו מסוג wav.

• Audio to Image Conversion Service

שירות זה יבצע המרה פשוטה של תוכן קובץ wav לייצוג של תמונה, כך שכל העבודה מול הרשתות השונות תתבצע במישור התמונה. כפי שצינתי על מנת לבצע את ההמרה הנ"ל

אשתמש בספריית Librosa אשר כתובה בקוד פתוח, ספרייה זו נותנת אפשרות המרה טובה מקובץ קול Mel Spectrogram במרחב התמונה.

- **Training Service**

שירות זה עוסק בעצם את שלבי האימון של רשת נוירונים. אשתמש בקוד של PyTorch שייצרו בפייסבוק אשר תומך ונוח לשימוש בסביבת python. זהו בעצם לב המערכת אשר ייצר מודלים המאומנים לפי הנתונים אותם אגב מקבצי Training.

- **Models Storage**

כאן אחזיק את התוצר הסופי של שלב Training, מודלים של PyTorch מסוגים שונים אותם פירטתי ואשר ניתן לתשאל כל אחד מהם לגבי זיהוי וסיווג של קובץ קול(לאחר שהומר למרחב התמונה).

- **Test Data Output**

כאן אחזיק את תוצאות הקטלוג של כל אחד מהמודלים שבניתי בצורת CSV עבור כל קובץ בדיקה, כך שבסופו של דבר יתקבלו 6 קבצי CSV (אחד לכל מודל שנבנה) אשר מכילים את הזיהוי והסיווג של כל מודל לכל אחד מ-158,536 קבצי test.

- **UI Client**

ממשק הUI נועד לטעינת האחוזים השונים לקראת Majority Voting והוא יחזיר את התשובה הסופית לגבי טיב המודל לאחר השוואת התוצאות לתוצאות של המנצח בתחרות של Kaggle.

- **Majority Voting Service**

שירות זה יקבל את האחוזים השונים אשר יילקחו בחשבון בעת סיווג קובץ test במערכת, לאחר מכן תתקבל ממנו תוצאה סופית לאחל ממוצע משוכלל לגבי הזיהוי והסיווג של המערכת לגבי כל אחד מקבצי test.

- **Verification Service**

שירות זה יקבל את תוצאות ה Majority Voting ויחזיר לממשק המשתמש תוצאה סופית באחוזים לגבי איכות תוצאות המערכת למול המערכת המנצחת בתחרות של Kaggle.

בדיקות

יבוצעו בדיקות יחידה ובדיקות אינטגרציה בין כל הרכיבים.

כלים

- פיתוח כלל המערכת יתבצע בpython בעזרת סביבת PyCharm.
- מימוש הCNN באמצעות Pytorch, אשר מותאמת לסביבת python.
- שימוש במאגר קבצי הקול של Tensorflow על מנת לאמן את המודלים השונים.
- הדפים הגרפיים ייכתבו בעזרת ספריית PyQT המותאמת לכתיבת UI לpython.
- המרה של קבצי wav למישור התמונה יתבצע בעזרת ספריית Librosa בpython.

תיאור תהליכים עיקריים

- תהליך מקדים – בניית רשתות נוירונים עמוקות לסיווג פקודות קוליות
- תהליך מקדים – אימון הרשתות השונות
- בניית 6 קבצי תוצאות (אחד לכל מודל) המכילים 158,536 רשומות כל אחד לפי תוצאות הסיווג של המודלים השונים למול קבצי הבדיקות.
- בניית ממשק לשינוי אחוזי Majority Voting השונים וקבלת תוצאת הסיווג של המערכת.
- השוואת קובץ התוצאות אשר מתקבל מהמערכת לקובץ התוצאות של הזוכה באתגר על מנת לעמוד את איכות המודל

חשיבות הפרויקט

כבר היום ישנו שימוש אדיר באלגוריתמים לזיהוי קולי.

אלגוריתמים אלו עוזרים ומשפרים את חוויות המשתמש בשלל רחב של תחומים אשר הולך וגדל בקצב מסחרר [5].

בין אם בסירי מבית אפל, במשחקי מחשב, בסימולטורים, בבית החכם ואפילו במטוסי קרב. השימוש העתידי באלגוריתמים לזיהוי קולי והצורך באלגוריתמים רובסטיים ואיכותיים מסוג זה הוא עצום.

שיטות לביצוע הפרויקט

- קריאת חומר תיאורטי אודות Deep Learning. [שגיאה! מקור ההפניה לא נמצא].
- קריאה וניתוח שיטות לעיבוד של קבצי קול למטרות למידה עמוקה.
- ביצוע סקירה אודות תשתיות רשתות נוירונים עמוקות דוגמת Tensorflow, Keras ו Pytorch.
- בניית UI של המערכת.
- ארגון ואינטגרציה של המודולים ובדיקות.
- בדיקת איכות התוצאות המתקבלות.

לוח זמנים לביצוע

פירוט משימה	משך ביצוע (חודשים)
תיאוריה, ניתוח ועיצוב פרויקט	3
בניית מאגר נתונים עבור הרשת	1
מימוש הפרויקט	4
בדיקת ביצועים וניתוח תוצאות	2
כתיבת תיק הפרויקט	2

- [1] Tensor Flow Speech Recognition Challenge - <https://www.kaggle.com/c/tensorflow-speech-recognition-challenge>
- [2] Training & Testing sets for the challenge - <https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data>
- [3] L. Wyse. 2017. Audio Spectrogram Representations for Processing with Convolutional Neural Networks. Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN.
- [4] Speech Recognition – Solution Outline <https://towardsdatascience.com/tensorflow-speech-recognition-challenge-solution-outline-9c42dbd219c9>
- [5] Nvidia inception program for Israeli startups - <https://www.nvidia.com/en-il/gtc/sessions/ai-startups>
- [6] Nvidia DGX AI Workstation Overview - <https://www.nvidia.com/en-us/data-center/dgx-station>
- [7] Kaggle Speech Recognition Challenge Winner Repository - https://github.com/see--/speech_recognition
- [8] Transfer Learning with Facebook PyTorch - https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
- [9] Librosa – python package for music and audio analysis - <https://librosa.github.io/librosa>
- [10] Pytorch – An open source deep learning platform by Facebook - <https://pytorch.org>
- [11] Flask – microframework for python for using built-in server-client platform - <http://flask.pocoo.org>
- [12] Image Classification Architectures review - <https://medium.com/@14prakash/image-classification-architectures-review-d8b95075998f>
- [13] Very Deep Convolutional Networks For Large Scale Image Recognition (VGG) - <https://arxiv.org/pdf/1409.1556.pdf>
- [14] Deep Residual Learning for Image Recognition (Resnet) - <https://arxiv.org/abs/1512.03385>
- [15] Densely Connected Convolutional Networks (Densenet) - <https://arxiv.org/pdf/1608.06993.pdf>

- [16] Squeeze-and-excitation Networks (Senet) - <https://arxiv.org/abs/1709.01507>
- [17] Boosting and Bagging : how to develop a Robust Machine Learning Algorithm - <https://hackernoon.com/how-to-develop-a-robust-algorithm-c38e08f32201>
- [18] Werkzeug: a WSGI utility library for Python. It's widely used and BSD licensed - <https://palletsprojects.com/p/werkzeug>
- [19] Tensorflow: an open source software library for high performance numerical computation – <https://www.tensorflow.org>
- [20] Tensorflow or Pytorch: The force is strong with which one - <https://medium.com/@UdacityINDIA/tensorflow-or-pytorch-the-force-is-strong-with-which-one-68226bb7dab4>

[21] CNN Features off the shelf: an astounding Baseline for recognition -
<https://arxiv.org/abs/1403.6382>

[22] Benchmark Analysis of Representative Deep Neural Network Architectures:
Simone Bianco, Remi Cadene, Luigi Celona and Paolo, University of Milano-Bicocca,
Department of Informatics, Systems and Communication, 2018 -
<https://arxiv.org/pdf/1810.00736.pdf>

[22] Netron – viewer for neural network, deep learning and machine learning models -
<https://github.com/lutzroeder/netron>