



פרויקט בחלופה למידת מכונה
במסגרת לימודי התמחות בהנדסת תוכנה 883589

שם הפרויקט: סיווג ז'אנר בשירים

שם התלמיד: ליאב אביטן

תעודת זהות: 327850061

בית ספר: תיכון ליאו באק

שם המנחה: אולגה בוטמן

שנת לימודים: תשפ"ג

שם החלופה: למידת מכונה

תאריך ההגשה: אפריל, 2023



הפונוגרף, המכשיר הראשון שאיפשר להקליט צלילים, לשמר אותם על גבי מדיה מכנית ולהשמיעם שוב(נוצר ע"י ה-AI - Midjourney)

תוכן עניינים

| | |
|--|----|
| מבוא: | 4 |
| הרקע לפרויקט | 4 |
| תהליך המחקר | 5 |
| אתגרים מרכזיים | 6 |
| הצגת פתרונות לבעיה (הפתרונות שנבחנו במסגרת המחקר המקדים) | 7 |
| רקע תיאורטי | 7 |
| מבנה / ארכיטקטורה של הפרויקט | 16 |
| מימוש: | 16 |
| שלב איסוף הכנה וניתוח הנתונים | 16 |
| רקע תיאורטי 2 | 19 |
| שלב בנייה ואימון המודל: | 23 |
| שלב היישום | 33 |
| מדריך למשתמש | 34 |
| מדריך למפתח | 36 |
| סיכום אישי / רפלקציה | 37 |
| ביבליוגרפיה | 38 |

מבוא:הרקע לפרויקט

מוזיקה כיום היא חלק אינטגרלי מחיינו, והיא מגיעה בז'אנרים רבים ושונים. בניתוח מוזיקה, סיווג ז'אנר היא משימה חיונית וחשובה, ויש לה שימושים רבים כגון:

- המלצת מוזיקה - שירותי הזרמת מוזיקה כגון Spotify, Apple Music ועוד משתמשים בסיווג ז'אנר מוזיקה כדי להמליץ למשתמשים על שירים ופלייליסטים חדשים. על ידי ניתוח הרגלי ההאזנה וההעדפות של המשתמש, שירותים אלו יכולים להמליץ על שירים מז'אנרים דומים שהמשתמש עשוי ליהנות מהם.
- חיפוש מוזיקה - סיווג ז'אנר מוזיקה משמש גם במנועי חיפוש מוזיקה. לדוגמה, אם משתמש מחפש ז'אנר מסוים של מוזיקה, כגון קלאסית או ג'אז, התוכנה יכולה להשתמש בתכונות סיווג הז'אנר כדי לסנן את תוצאות החיפוש ולמצוא את המוזיקה שהוא מחפש מהר יותר.
- הפקה מוזיקלית - מפיקים מוזיקליים יכולים להשתמש בסיווג ז'אנר כדי לנתח את המאפיינים של הז'אנרים השונים שעליהם הם רוצים לדעת, וליצור מוזיקה חדשה שמתאימה לאותם ז'אנרים. לדוגמה, מפיק שרוצה ליצור שיר חדש של היפ הופ או מוזיקה אלקטרונית, יכול להשתמש בסיווג ז'אנר כדי לנתח את המאפיינים של אותו הז'אנר וליצור שיר שיתאים לקריטריונים של הז'אנר.

מטרת הפרויקט - לבנות מודל למידת מכונה שיכול לסווג במדויק ז'אנרים של מוזיקה (מתוך 10 קטגוריות), באמצעות מערך הנתונים של ¹ GTZAN.

קהל היעד של הפרויקט - כל מי שמתעניין בניתוח מוזיקה, למידת מכונה ומדעי הנתונים. הפרויקט נועד לספק הבנה טובה יותר של טכניקות סיווג ז'אנר במוזיקה והיישומים המעשיים שלהן.

אופן הפעולה - הפרויקט יכלול מספר שלבים, הכולל עיבוד מוקדם של נתונים, מיצוי תכונות, בחירת מודל ושיפורו והערכת תוצאות המודל ושלב היישום. הפרויקט ישתמש באלגוריתמים מתקדמים של למידת מכונה, כגון CNN, מניפולציה מתמטית, ועוד כדי להשיג דיוק סיווג גבוה.

הסיבות לבחירת הנושא - אני מוזיקאי ורציתי להתעסק בפרויקט שישלב גם למידת מכונה, וגם הקשור למוזיקה, פיזיקה ומתמטיקה ובכלל לעסוק בפרויקט מאתגר. לכן, כאשר ראיתי שיש

¹ קישור: <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>

אפשרות לבחור בפרוייקט סיווג ז'אנרים במוזיקה שמשלב את כל אלו, החלטתי שזה יהיה הפרוייקט שאבצע.

תהליך המחקר

כיום, ישנן מאות של ז'אנרים שונים מסביב לעולם, ביניהם - פופ, רוק, מוזיקה אלקטרונית, היפ הופ, מוזיקה קלאסית ועוד.

סיווג הז'אנר כיום חשוב ממספר סיבות, ביניהן -

- ארגון מוזיקה: סיווג ז'אנר עוזר בארגון וסיווג ז'אנרים שונים של מוזיקה, ומקל על המשתמשים למצוא ולגלות מוזיקה שהם אוהבים.
- המלצות מותאמות אישית: פלטפורמות הזרמת מוזיקה ואפליקציות אחרות הקשורות למוזיקה משתמשות בסיווג ז'אנר כדי לספק המלצות מוזיקה מותאמות אישית למשתמשים על סמך היסטוריית ההאזנה וההעדפות שלהם.
- שיווק וקידום: חברות תקליטים ומפיקים מוזיקליים משתמשים בסיווג ז'אנר כדי למקד לקהלים ספציפיים ולקדם את המוזיקה שלהם בהתאם.
- מחקר מוזיקלי: מוזיקולוגים וחוקרים משתמשים בסיווג ז'אנר כדי לחקור את התפתחות המוזיקה לאורך זמן, לזהות מגמות ולחקור את המשמעות התרבותית של ז'אנרים שונים.
- זכויות יוצרים ורישוי: סיווג ז'אנר משמש גם בזכויות יוצרים ורישוי של מוזיקה, שכן לז'אנרים שונים עשויים להיות חוקי זכויות יוצרים ודרישות רישוי שונות.

לכן, יש צורך מתמיד של חברות הזרמת מוזיקה, מוזיקולוגים, חברות הפקת מוזיקה ועוד, לסיווג ז'אנר על מנת לגרום למשתמש חברת הזרמה (Spotify לדוגמה) להישאר יותר זמן באפליקציה, או לשימוש בזכויות יוצרים ורישוי ראויים שיכולים לחסוך טעויות רבות ביצירת זכויות היוצרים.

השימוש כיום בסיווג ז'אנרים מובנה בתוך יישומים ואתרים, ואין אפשרות לגשת אליו. לכן, בניתי קוד פתוח² ב-Github שכל אדם יוכל להשתמש בו בעזרת הוראות ההדרכה שכתבתי. נוסף על כך, בניתי

² קישור: <https://github.com/L33TAv/GenreClassificationGTZAN-with-CNN>

גם יישום מותאם (שעליו יפורט בהמשך) כך שבזמן אמת ניתן לשים שיר ולקבל חיזוי לז'אנר השיר שהועלה באמצעות ספריית Gradio³.

במהלך הפרוייקט, נעזרתי בפרוייקט ב-Kaggle של סיווג ז'אנר ב-CNN⁴ שבו היתה פונקצית המרה ל-MFCC, ופונקציות נוספות שהשתמשתי והתאמתי למטרת הפרוייקט שלי (פונקצית ה-MFCC השארתי כמעט בשלמותה).

במהלך הפרוייקט השתמשתי ולמדתי על ספריית Librosa⁵, השתמשתי ובניתי באמצעות Gradio אתר שפועל בצורה חיה, שבו אוכל ליישם את סיווג הז'אנר - שבו ניתן לשים שיר לסיווג וחיזוי ז'אנר/ז'אנרים אפשריים, יחד עם נתונים נוספים על השיר.

אתגרים מרכזיים

בעיות איתם צפוי הלומד להתמודד במהלך פיתוח הפרוייקט - במהלך תקופת הלימוד במגמה, למדנו רבות על סיווג תמונות וטבלאות ונתונים, ובעיה מרכזית שאני עלול להתמודד במהלך הפרוייקט, היא העבודה עם סוג נתונים חדש לחלוטין. כאמור, כאן מדובר בסיווג קבצי מוזיקה ואצטרך ללמוד איך לעבד את הנתונים, איך להשתמש בקבצי המוזיקה וליצור מודל שיוכל לסווג ז'אנרים ולכן בשלב עיבוד הנתונים תהייה טרנספורמציות מתמטיות, חישובים ועיבודים עם ידע מתמטי שעוד לא רכשתי (כגון התמרת פוריה) שאצטרך ליישם תוך כדי עבודה עם קובץ מוזיקה. נוסף על כך, החלק הארי של הספריות של הפרוייקט בו אני משתמש משתנות ומתעדכנות כל הזמן - ולכן עלול להיווצר שינוי בזמן עבודה על הפרוייקט מבחינת הקוד שנכתב.

על איזה צורך הפרוייקט עונה? איזה פתרון הפרוייקט הזה בא לתת? - פרויקט זה עוסק בסיווג ז'אנר במוזיקה ונועד לתת מענה לצורך בסיווג מוזיקה, לז'אנר מתוך מכלול ז'אנרים שונים ונפרדים. צורך זה נובע מהכמות העצומה של מוזיקה הזמינה כיום, מה שעלול להקשות על המשתמשים לזהות ולהאזין למוזיקה התואמת את העדפותיהם. על ידי סיווג אוטומטי של מוזיקה לז'אנרים, משתמשים יכולים, בקלות רבה יותר, לגלות מוזיקה חדשה שהם עשויים ליהנות ממנה. נוסף על כך, לסיווג ז'אנר במוזיקה יש שימושים רבים בתעשייה (כגון הפקה מוזיקלית, המלצת מוזיקה, חיפוש מוזיקה ועוד) והצורך לשפר את האלגוריתמים של הסיווג תמיד קיים.

³ קישור: <https://gradio.app/>

⁴ קישור: <https://www.kaggle.com/code/vrushaliingle/music-genre-classification-using-cnn/notebook>

⁵ קישור: <https://librosa.org/doc/latest/index.html>

לסיכום, פרויקט העוסק בסיווג ז'אנר במוזיקה מספק פתרון רב ובעל ערך לבעיה של סיווג יעיל של מוזיקה לז'אנרים נפרדים, ומאפשר למשתמשים לגלות וליהנות בקלות רבה יותר ממוזיקה התואמת את העדפותיהם.

הצגת פתרונות לבעיה (הפתרונות שנבחנו במסגרת המחקר המקדים)

במסגרת המחקר הבנתי כי הדרך המרכזית והסטנדרטית למיצוי תכונות ולייצוגן על מנת לסווג את הז'אנרים שנמצאים בכל שיר, היא באמצעות מטריצה שתייצג את המאפיינים של הקול כגון עוצמתו, או דומיננטיות של תדרים שונים (MFCC). בהמשך, אשתמש במטריצות הללו, ואבנה רשת סיווג (ניתן להשתמש במספר של רשתות סיווג מסוגים שונים, אך בהתאם למסגרת תוכנית הלימוד החלטתי להשתמש ברשת בשם - CNN) שתלמד לקבל שיר ולחזות את הז'אנר המתאים שלו.

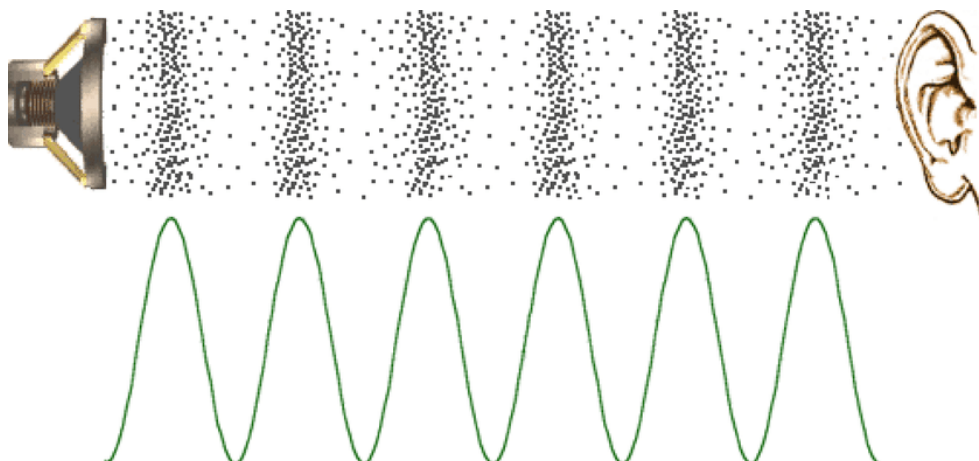
רקע תיאורטי

פרויקט זה עוסק בעולמות הפיזיקה והמוזיקה בפרט, לכן יש צורך לדעת ידע תאורטי על מושגים הנוגעים בנושא תורת הקול, למידת מכונה וסיווג ז'אנרים על מנת להבין את הפרוייקט לעומק:

כיצד נוצר גל קול?

גלי קול (sound waves) נוצרים כתוצאה מרטט של משהו. כשמשהו רוטט, הוא מייצר גלי קול באוויר ואז יש צליל. אנשים אוהבים לדמות את גלי הקול לאדוות, גלים זעירים, שנוצרים בבריכה כשמשליכים לתוכה אבן.

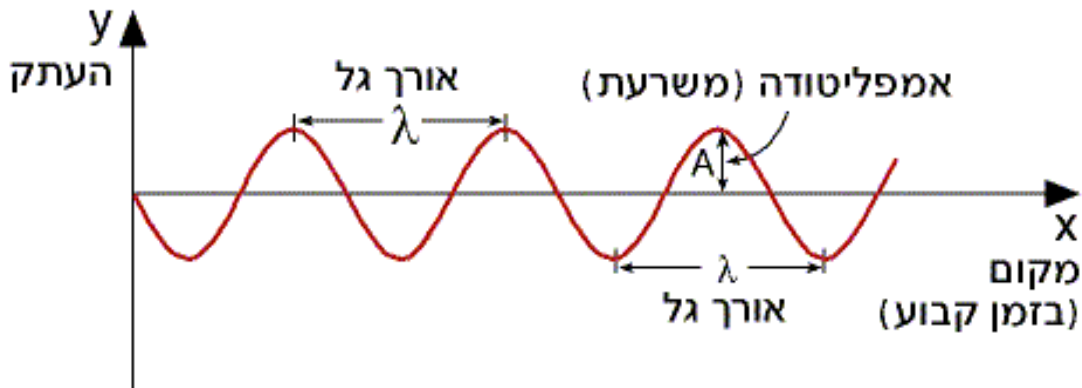
ומבחינה מדעית - גלי הקול הם תנודות מחזוריות בלחץ האוויר. התנודות הללו דוחפות את המולקולות של האוויר ואלו זזות הלך וחזור, ממצב של יציבות ובחזרה, שוב ושוב ושוב - כך נוצר הצליל. המולקולות מתפשטות באוויר ויוצרות גלים - גלי קול. ככל שהתנודות הללו יהיו מהירות יותר, יהיה צליל גבוה יותר, ככל שהתנודות יהיו חזקות יותר - תגדל העוצמה של הצליל שנוצר מגלי הקול הללו.



איור: התפשטות גל קול ממקור קול לאוזן האדם⁶

מהי משרעת (אמפליטודה)?

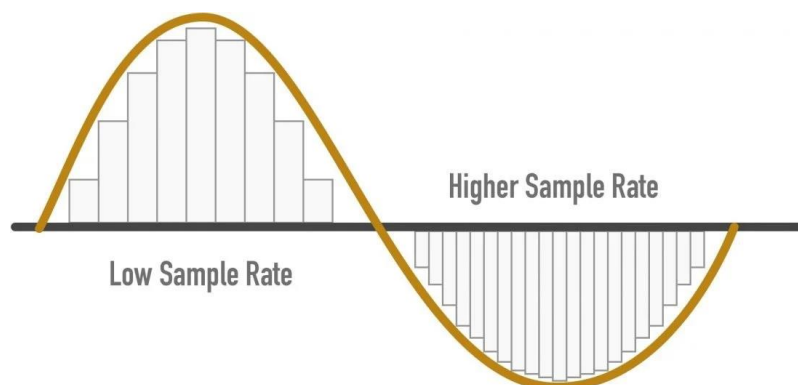
משרעת במוזיקה מתייחסת לעוצמתו או לגודלו של גל קול, שקובע את עוצמת הקול או העוצמה של צליל. בהקשר של מוזיקה, משרעת נמדדת בדרך כלל בדציבלים (dB) ויכולה לנוע בין צלילים שקטים מאוד עם משרעת נמוכה לצלילים חזקים מאוד עם משרעת גבוהה.



גרף המתאר ההעתק של הגל, כפונקציה של המקום במרחב⁷

מה זה Sample rate? (תדירות הדגימה)

קצב הדגימה מתייחס למספר הדגימות של אות רציף שנלקחות בשנייה על מנת לייצג את האות בצורה דיגיטלית. קצב הדגימה מבוסס בהרץ (Hz), שהוא מספר הדגימות בשנייה. ככל שתדירות הדגימה תהיה גבוהה יותר, כך הצליל יהיה קרוב יותר למקור.



איור: ההבדל בין קצב דגימה נמוך לקצב דגימה גבוה⁸

מה זה מונו?

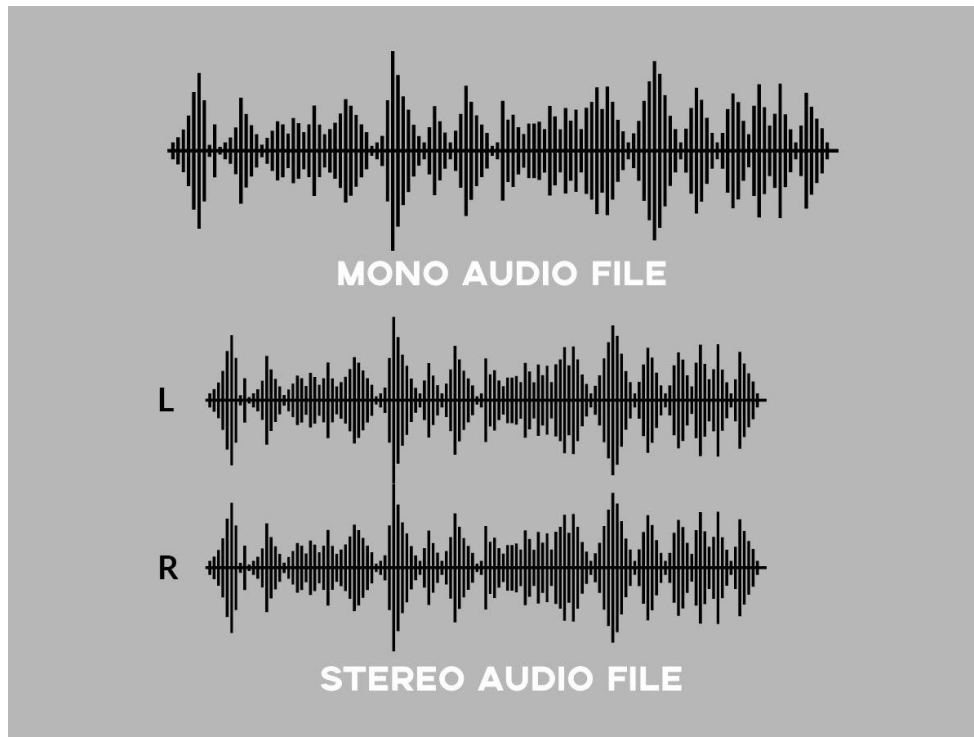
מונו היא שיטת עיבוד צלילים בעלת ערוץ אחד בלבד. קטע צליל מוגדר כ'מונו' כאשר מקור הצליל (כפי שהוקלט על ידי מיקרופון, למשל) הוא בודד. גם אם קיימים מספר מיקרופונים בעת ההקלטה, ובסופו של התהליך כל הערוצים מתאחדים אל תוך ערוץ אחד - ההקלטה הסופית נחשבת כהקלטת

⁶ קישור: https://www.researchgate.net/figure/Propagation-of-sound-wave-from-sound-source-to-ear_fig7_330560014

⁷ קישור: https://stwww1.weizmann.ac.il/communication/?page_id=529

⁸ קישור: <https://www.headphonesty.com/2019/07/sample-rate-bit-depth-bit-rate>

'מונו' - כלומר, כאשר קול נמצא בערוץ אחד. לדוגמא, כשנשמע מוזיקה באוזניות (שמכילות 2 צדדים, ערוצים - ימין ושמאל) או במספר רמקולים (עם 2 או יותר ערוצים), הקול ישתכפל בשני הצדדים ולא יהיה שלב שבו נשמע רק בצד אחד לדוגמא קטע מוזיקה ובצד השני קטע אחר(שזהו למעשה סטריאו).



תרשים השוואה בין קובץ מונו לעומת קובץ סטריאו⁹

כפי שניתן לראות באיור, במצב Mono יהיה לנו חלק אחד שמייצג את כל השיר, לעומת זאת ב-Stereo האודיו יהיה מחולק לצד ימין ולצד שמאל.

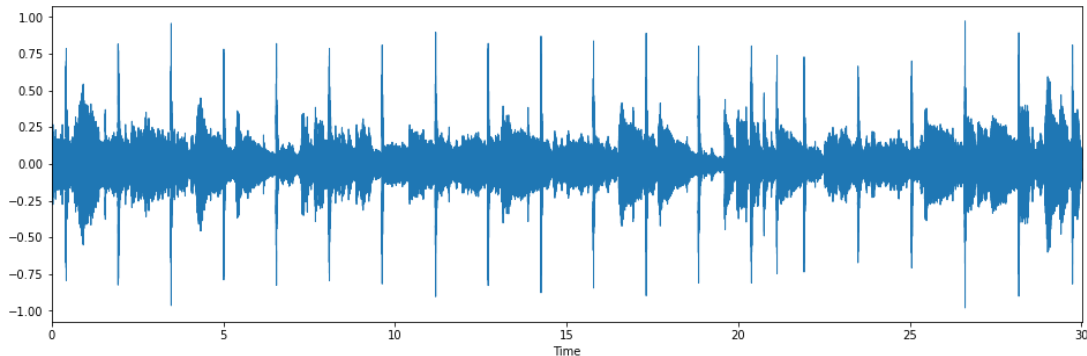
מהו waveform / waveplot?

waveplot מציגה את משרעת (אמפליטודה) של אות השמע לאורך זמן, כאשר ציר ה-x מייצג זמן, וציר ה-y מייצג משרעת (שמייצגת את העוצמה). ניתן להשתמש ב-waveplot כדי להמחיש את המבנה והמאפיינים הבסיסיים של אות אודיו, כולל תוכן התדר שלו, משך הזמן והצורה הכללית שלו.

ערכי גלים מנורמלים בדרך כלל בין 1 ל-1- מכיוון שגלי קול מיוצגים כתנודות בלחץ האוויר, ויכולים להיות חיוביים או שליליים. הלחץ החיובי המרבי מיוצג כ-1, והלחץ השלילי המרבי מיוצג כ-1-. לכן, כל ערך בין 1 ל-1- מייצג שינוי לחץ האוויר באופן פרופורציונלי ל-0. הערך של 0 מייצג מצב ללא שינוי לחץ.

⁹ קישור: <https://www.headphonesty.com/2022/01/what-is-the-difference-between-mono-and-stereo>

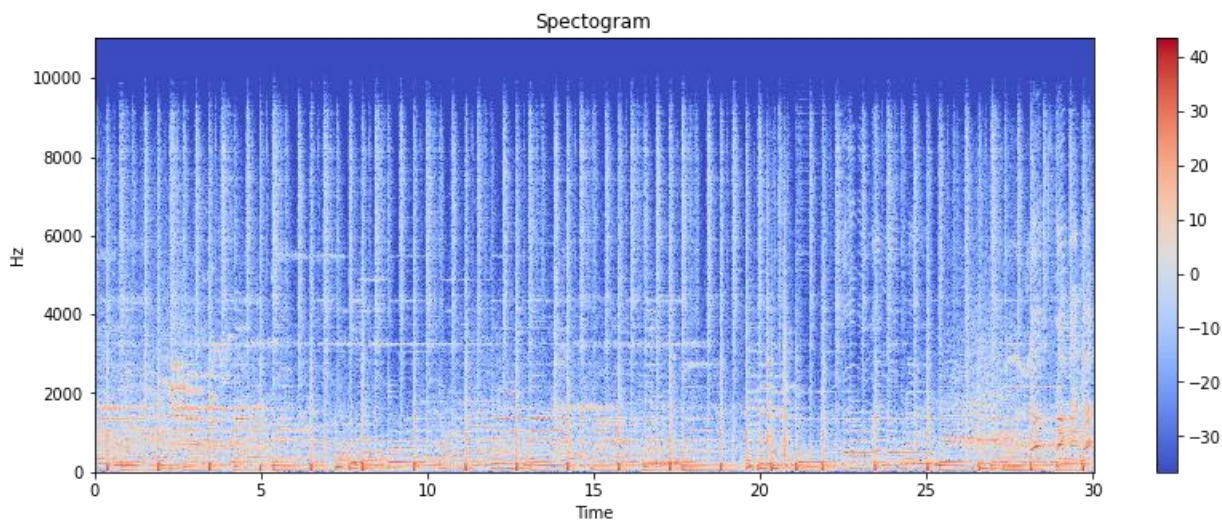
בעזרת השיטה הזו ניתן להשוות בין צורות גל שונות בקלות יותר, מכיוון שאנו משנים את צורות הגל לאותו טווח של ערכים.



תרשים waveplot שיצרתי בעזרת ספריית Librosa

מהי ספקטוגרמה (Spectrogram)?

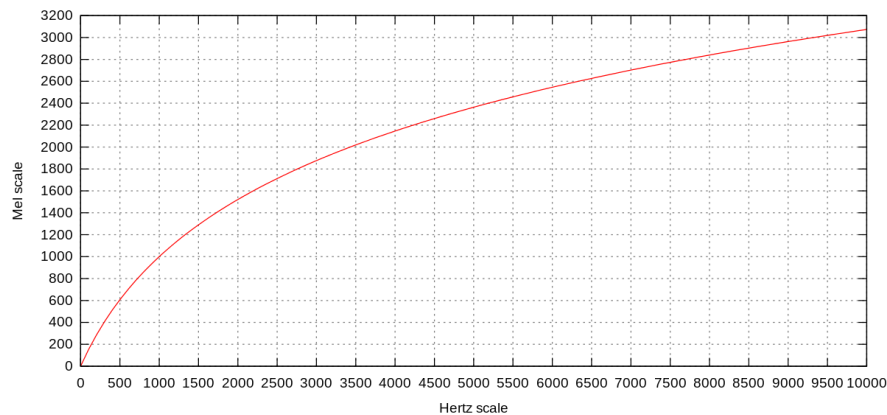
ספקטוגרמה בהגדרתה, היא ייצוג חזותי של ספקטרום התדרים של האות (קול) והשתנותו עם הזמן. כלומר, ספקטוגרמה היא ייצוג חזותי של 3 מימדים - זמן, תדירות הקול ועוצמתו.



ספקטוגרמה שבניתי בעזרת ספריית Librosa

כפי שניתן לראות בגרף, יש כאן 3 מימדים - הזמן בציר האופקי, התדר בציר האנכי, והמקרא מצד ימין מראה כיצד הצבע משתנה עם העוצמה.

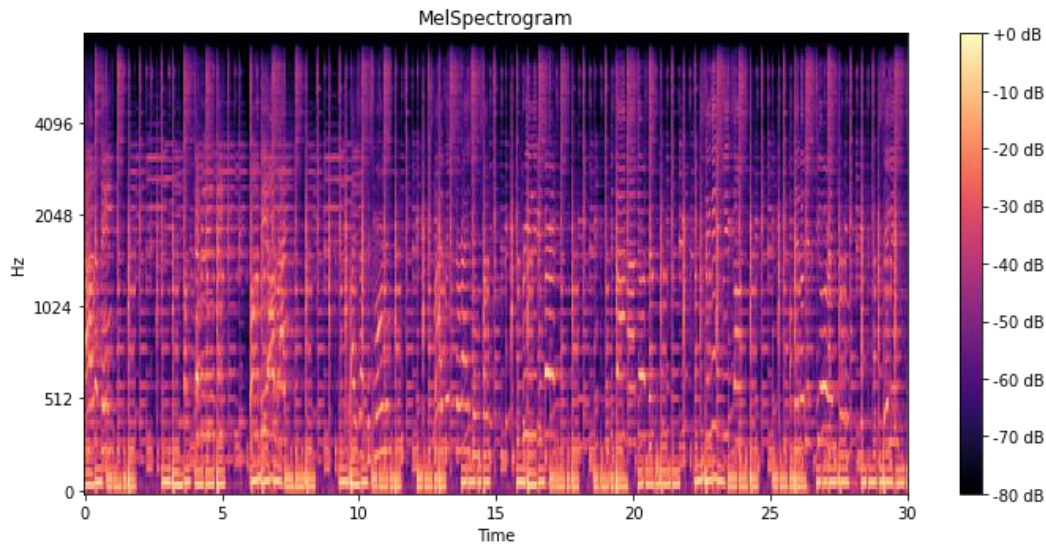
מהי מל ספקטוגרמה (Mel spectrogram)? ואיך היא שונה מן ספקטוגרמה (+ מהו סולם מל) ראשית, קצת על סולם מל - סולם מל הוא סולם המבוסס על הדרך שבה בני אדם תופסים שמיעתית את גובה הצליל. בסולם מל התדר עולה באופן לוגריתמי, בניגוד לסולם הרץ הסטנדרטי (שעולה באופן ליניארי). כפי שניתן לראות באיור שנמצא למטה, אנו תופסים את הזמן באופן שאינו ליניארי, וככל שהתדירות עולה כך השינוי פחות "נתפס" בשמיעה.



גרף המראה את היחס בגובה בין סולם מל לסולם הרץ סטנדרטי¹⁰

לצורך המחשה, הגרף (הנמצא למעלה) מראה את היחס בגובה של סולם מל בציר האנכי, וסולם הרץ סטנדרטי בציר האופקי. מל ספקטוגרמה, בדומה לספקטוגרמה מייצגת 3 מימדים - זמן תדירות הקול ועוצמתו, אך במקרה של מל ספקטוגרמה מימד תדירות הקול מיוחס באמצעות סולם מל - כלומר, בדומה לאיך שאוזן האדם שומעת. בספקטוגרמה נשתמש כאשר לכל התדרים יש "חשיבות" שווה. לעומת זאת, במל ספקטוגרמה נשתמש בייחודיות שלה כאשר היא תתאים ליישומים שבהם אנו ננסה להדגים את הדרך בה האוזן האנושית תתפוס את הצליל/התדר, ולכן לתדרים נמוכים יותר תהינה יותר "חשיבות" מתדרים גבוהים יותר.

¹⁰ קישור: https://he.wikipedia.org/wiki/%D7%A1%D7%95%D7%9C%D7%9D_%D7%9E%D7%9C

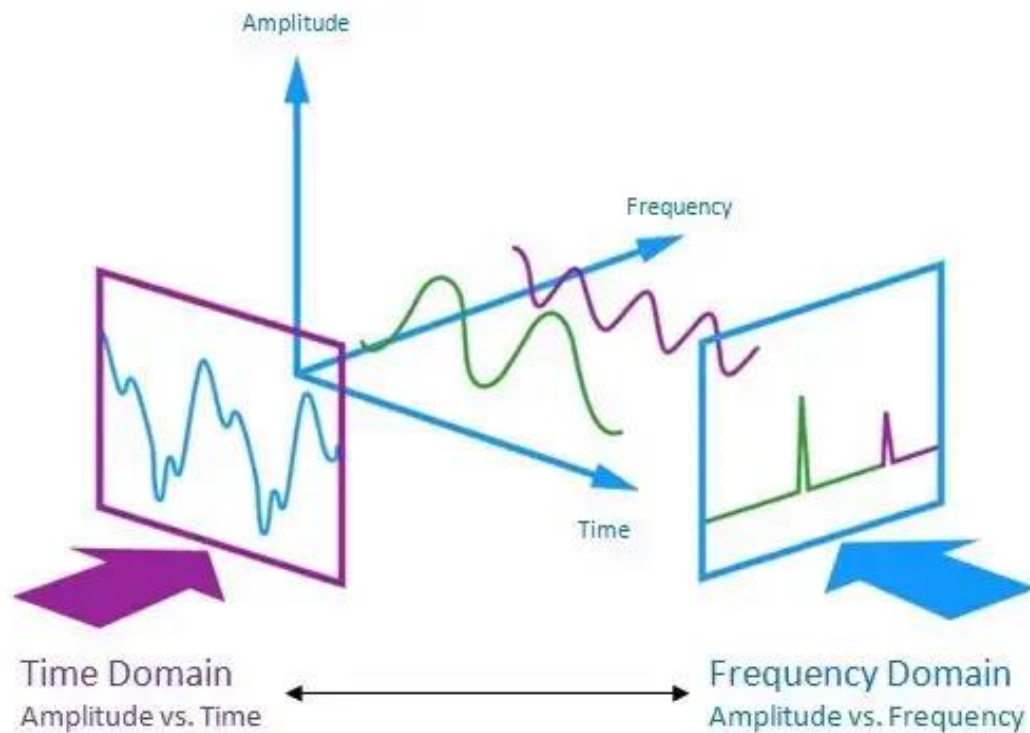


מל ספקטוגרמה שבנית באמצעות Librosa

מהו MFCC? איך הוא מחושב ומה הוא מבטא?

MFCC ראשי תיבות של Mel Frequency Cepstral Coefficients. זוהי דרך לייצג את המאפיינים הספקטראליים של אות אודיו, שהוא התפלגות האנרגיה של האות על פני תדרים שונים (ובכך לראות תדרים יותר/פחות דומיננטיים), ומיוצג בסולם מל.

MFCCs נמצאים בשימוש נרחב בעיבוד וניתוח אותות אודיו, במיוחד עבור זיהוי דיבור, זיהוי דובר ושליפה של מידע מהמוזיקה/קטע שיר. הם מחושבים באמצעות סדרה של טרנספורמציות מתמטיות הממירות אות אודיו של תחום זמן לייצוג של תחום תדר, ולאחר מכן לסט קומפקטי של מקדמים הלוכדים את התכונות החשובות ביותר של האות.

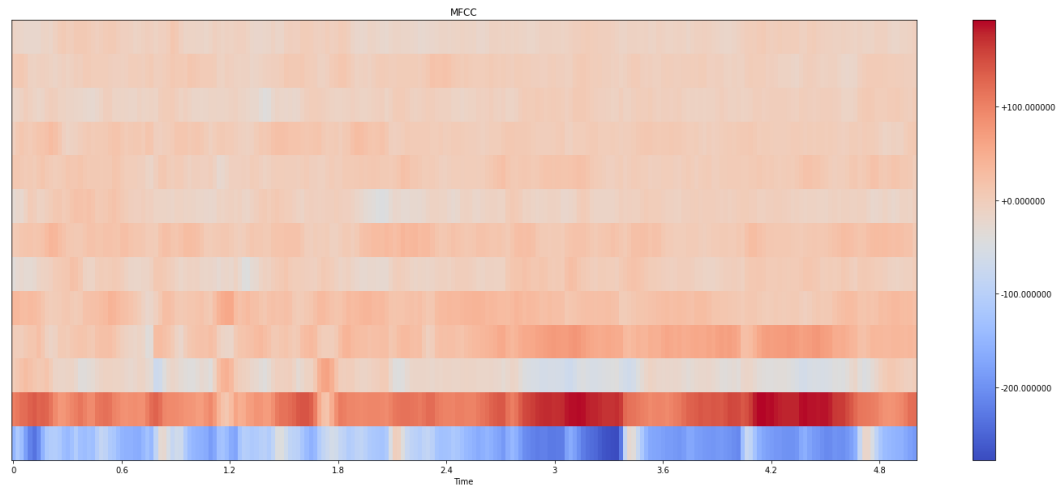


אות בתחום זמן ואות בתחום תדר¹¹

באיור למעלה ישנו ייצוג ויזואלי של ייצוג זמן-עוצמה לעומת ייצוג עוצמה-תדר ושל איך נראית המרה מייצוג תחום זמן לתחום תדר - ככל שהתדר יותר חזק, כך הגובה יהיה גבוה יותר.

במילים פשוטות - MFCCs הם דרך להמיר גלי קול לקבוצה של מספרים המייצגים את המאפיינים הייחודיים של הצליל, כגון גובה הצליל, הגוון והווליום שלו (והכל ביחס לשמיעת האדם, כלומר בסולם מל). לאחר מכן ניתן להשתמש במקדמים אלה כדי לסווג או להשוות צלילים שונים, כגון זיהוי מילים או זיהוי ז'אנר מוזיקלי.

¹¹ קישור: https://www.kindpng.com/imgv/hwiooiT_file-mel-hz-plot-svg-mel-scale-hd/



הצגת ויזואלית של MFCC לאורך זמן, באמצעות גרף שיצרת בעזרת ספריית Librosa

הגרף למעלה, מדגים הצגה ויזואלית של MFCC לאורך זמן. המימד האופקי - X מייצג את הזמן, והמימד האנכי - Y מייצג את קבוצת המקדמים שעליהם דיברנו לכל נקודת זמן שנקבעה מראש, כל מלבן מייצג מקדם מסדר גבוה או נמוך יותר (לפי הגובה). המקדמים מסדר נמוך נוטים לייצג את הצורה הכוללת של המעטפת הספקטרלית (כלומר, האנרגיה/הדומיננטיות של תדרים שונים מתוך ספקטרום התדרים), בעוד שהמקדמים מסדר גבוה יותר לוכדים מידע מפורט יותר על המעטפת הספקטרלית.

לדוגמה, מקדם ה-MFCC הראשון מכונה לעתים קרובות "הממוצע הספקטרי" (Cepstral mean) ומייצג את האנרגיה הממוצעת בכל פסי התדר. מקדם MFCC השני מייצג את השונות של התפלגות האנרגיה, שניתן לחשוב עליה כמדד ל"פיזור" האנרגיה על פני פסי התדרים השונים. מקדמים מסדר גבוה יותר לוכדים פרטים עדינים יותר ויותר על חלוקת האנרגיה.

באופן כללי, מקדמי הסדר הנמוך חשובים יותר עבור משימות כגון זיהוי דובר או זיהוי דיבור, כאשר הצורה הכוללת של המעטפת הספקטרלית היא תכונה מרכזית להבחנה בין דוברים או מילים שונות. מקדמי הסדר הגבוה יותר עשויים להיות שימושיים יותר עבור משימות כגון סיווג ז'אנר מוזיקה, כאשר מידע מפורט יותר על המעטפת הספקטרלית יכול להיות אינפורמטיבי להבחנה בין ז'אנרים שונים של מוזיקה.

הצבע של כל מלבן בגרף מייצג את הגודל או החוזק של המקדם המתאים במסגרת זמן מסוימת, טווח הצבעים המשמש בחלקות MFCC הוא בדרך כלל נמצא בטווח מכחול (בעוצמה נמוכה) לאדום (בעוצמה גבוהה), אם כי ניתן להשתמש בסכימות צבעים שונים.

נוסף על כך, ככל שאורך המלבן קטן יותר, כך חישבנו יותר מקדמים של MFCC (כמות המקדמים שבחרנו לכל אורך קטע זמן קטן יותר - X axis), וככל שגובה המלבן קטן יותר, חישבנו את העוצמה של יותר כמות של מקדמים לאורך קטע זמן מסויים (Y axis).

מהי התמרת פוריה?

התמרת פוריה הוא כלי מרכזי בניתוח הרמוניה/מוזיקה שאפשר לתארו כפירוק של פונקציה לרכיבים מחזוריים (סינוסים וקוסינוסים לדוגמא) וביצוע אנליזה מתמטית לפונקציה על ידי ניתוח רכיביה. שיטה זו פותחה על ידי המתמטיקאי ז'אן-בטיסט ז'וזף פורייה.

מה היא התמרת פוריה מהירה?

התמרה זו בפשטות, היא אלגוריתם יעיל לחישוב התמרת פורייה בדידה וההתמרה ההופכית שלה. באמצעות התמרה זו, נפרק את הרכיבים השונים בקול, בכדי שנוכל להמיר חלקים מהשיר מתחום זמן לתחום תדר (ראה עמוד 13), נוסף חישובים נוספים ונגיע ל-MFCCs.

מה זה אומר קובץ אודיו של 16 Bit?

קובץ שמע של 16 סיביות הוא סוג של קובץ שמע דיגיטלי המאחסן סאונד באמצעות 16 סיביות של מידע עבור כל דגימה של הסאונד (Sample). מספר הביטים קובע את כמות הפרטים והדיוק שניתן ללכוד בצליל, כאשר מספר סיביות גבוה יותר מאפשר בדרך כלל איכות ואמינות טובים יותר של השמע. קובץ אודיו של 16 סיביות משמש בדרך כלל עבור שמע באיכות שנקראת איכות CD, שהוא הסטנדרט עבור רוב התקני השמעת מוזיקה ואודיו.

מבנה / ארכיטקטורה של הפרויקט

מימוש:

- על מנת לממש את הפרויקט, השתמשתי במספר ספריות, ביניהן המרכזיות -
- **TensorFlow: TensorFlow** היא ספריית למידת מכונה בקוד פתוח שפותחה על ידי גוגל. זוהי אחת הספריות הנפוצות ביותר לבניית והדרכה של מודלים של למידה עמוקה, והיא מספקת מגוון רחב של כלים וממשקי API לעבודה עם רשתות עצביות (Neural networks). נשתמש בה בצורה רחבה בשלבי האימון.
 - **Librosa: Librosa** היא ספריית Python לניתוח ועיבוד אותות אודיו. הוא מספק מגוון כלים לטעינת קבצי אודיו, חילוץ תכונות מקבצי האודיו (כגון MFCCs), ותפעול הנתונים המתקבלים. Librosa היא ספרייה מרכזית בה נשתמש על מנת להכין את הנתונים לשלב האימון, וגם בחיזוי.
 - **Gradio: Gradio** היא ספריית Python המספקת דרך קלה לבנות ממשקי משתמש מבוססי אינטרנט (web-based UI) עבור מודלים של למידת מכונה. עם Gradio, אפשר ליצור במהירות ממשקים אינטראקטיביים המאפשרים למשתמשים להזין נתונים, להריץ מודל על נתונים אלה ולהציג את התוצאות. Gradio נשתמש בתהליך היישום.
 - **Resampy: Resampy** היא ספריית Python לדגימה מחדש של אותות אודיו. הוא מספק מגוון כלים לשינוי קצב הדגימה של אות אודיו, שיכול להיות שימושי לעיבוד מוקדם של נתוני אודיו לפני הזנתם למודל למידת מכונה. ב-Resampy נשתמש בתהליך היישום.

שלב איסוף הכנה וניתוח הנתונים

בשלב איסוף הנתונים השתמשתי ב-dataset של ז'אנרים שונים במוזיקה ושמו **GTZAN**. בשנת 2002, P. Cook ו-G. Tzanetakis חוקרים בתחום מדעי המחשב והמוזיקה, הציגו את המאמר הידוע שלהם על סיווג ז'אנר, "סיווג ז'אנר מוזיקלי של אותות אודיו", שפורסם ב-IEEE Transactions on Audio and Speech Processing¹². דאטא סט זה, מכיל 1,000 דגימות שירים המשתייכות לסך של 10 ז'אנרים מוזיקליים רגילים. הדגימות מסווגות לבלוז, קלאסי, קאנטרי, דיסקו, היפ הופ, ג'אז, מטאל, פופ, רגאי ורוק. כל ז'אנר מכיל 100 שירים. במערך הנתונים של GTZAN, כל שיר הוא באורך 30 שניות, עם קצב דגימה (Sample rate) של 22,050 הרץ, על מצב מונו, פורמט קובץ Wav וקובצי אודיו של 16 סיביות.

¹² IEEE היא אגודה מקצועית בינלאומית של העוסקים בהנדסת חשמל ואלקטרוניקה, הנדסת מחשבים והנדסת תוכנה.

מערך הנתונים נמצא בשימוש נרחב למחקר סיווג ז'אנר מוזיקה והיווה בסיס למספר רב של מאמרים אקדמיים ותחרויות למידת מכונה, ומערך זה הפך למערך סטנדרטי כמערך נתונים להערכת הביצועים של אלגוריתמים לסיווג ז'אנר מוזיקה.

הקבצים נאספו בשנים 2000-2001 ממגוון מקורות כולל דיסקים אישיים, רדיו, הקלטות מיקרופון, על מנת לייצג מגוון תנאי הקלטה.

תמונה ובה מוצגים אומנים מרכזיים בז'אנר, ולאחריה תמונה המציגה את הסאב ז'אנרים המרכזיים:¹³

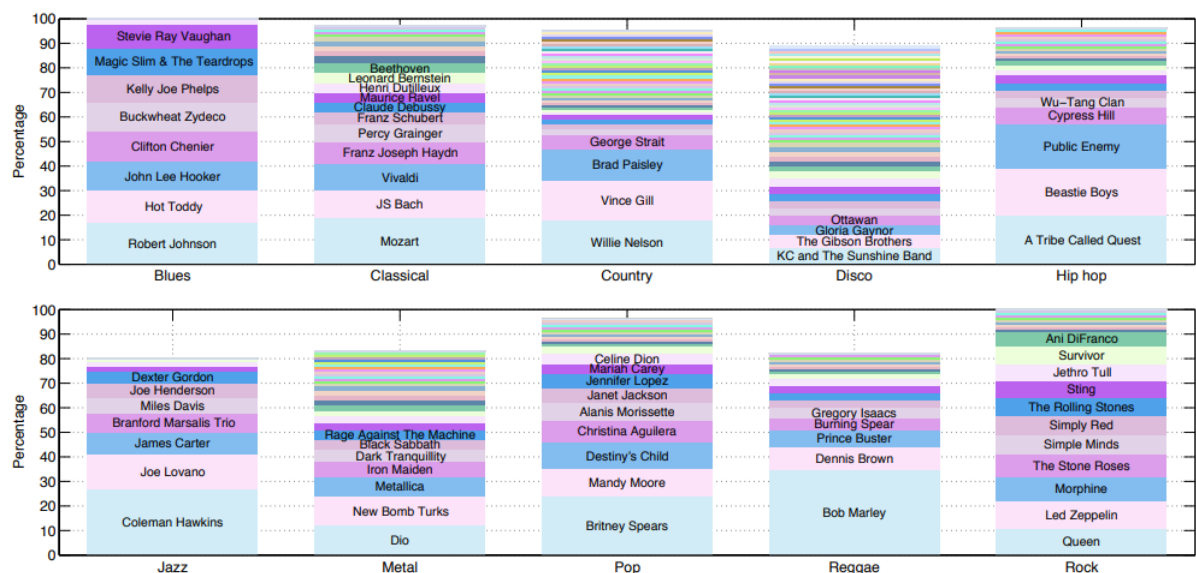


Figure 1: Artist composition of each GTZAN category. We do not include unidentified excerpts.

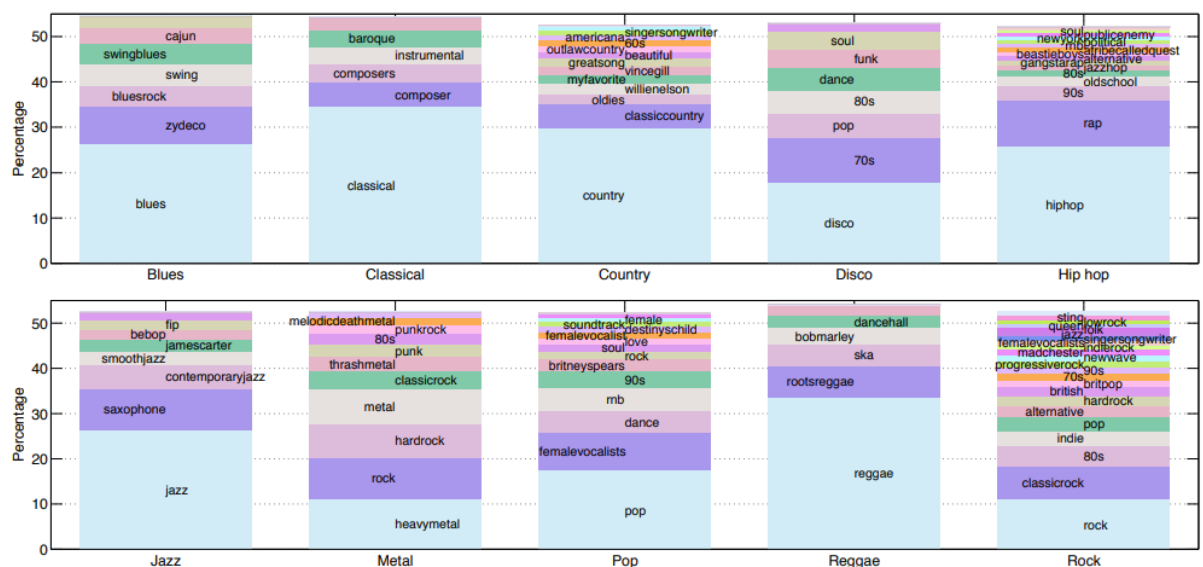


Figure 2: Top tags of each GTZAN category. We do not include unidentified excerpts.

- להכנת ה-Dataset שאיתו נעבוד ונאמן את המודל היינו צריכים להמיר אותו למבנה מסוג MFCC. בקוד, בנינו מבנה נתונים מסוג Dictionary מפתח-ערך, שיאחסן לנו 3 מרכיבים חשובים -
1. מערך ששומר את שמות 10 הז'אנרים - "mapping".
 2. מערך ששומר את הז'אנר "האמיתי" של הקטע מהשיר (ערך מ-0 עד 9) - "labels".
 3. מערך ששומר את ה-MFCC של הקטע שלעיל בהתאם - "mfcc".

בתהליך ההכנה: עברנו על הדאטא סט, ועברנו על כל ז'אנר לפי הסדר שלהם (את הז'אנר שמרנו בכדי שבסוף התהליך יהיו שמורים כל סוגי הז'אנרים), עברנו על כל שיר בז'אנר, וחילקנו את השיר ל-6 מקטעים, בכדי שיהיו יותר נתונים שנוכל להתאמן איתם ושיהיה לנו חישוב MFCC יותר יעיל (דהיינו, 5 שניות לכל מקטע). על כל מקטע בשיר מסויים, ביצענו חישוב מתמטי באמצעות התמרת פוריה מהירה על הקטע ועוד ויצרנו MFCC, ולכל מקטע שמנו את התגית המתארת את סוג הז'אנר של השיר במבנה הנתונים ב-"labels" וה-MFCC שלהם בהתאמה ב-"mfcc". סיימנו לעבור על כל הנתונים, ושמרנו הכל באמצעות קובץ JSON, מכיוון שמתאים לסוג מבנה הנתונים שבו השתמשנו (פורמט מפתח-ערך) ועל מנת לקבל קובץ אחד קומפקטי שאיתו נעבוד.

בנוסף, לפני הפעלת הפונקציה הממירה ל-MFCC, שמרנו במשתנים את הנתונים שידועים לנו, כגון אורך שיר, קצב דגימה, הנתביב לדאטאסט. ואת הDictionary נוציא ונשמור בגוגל דרייב כקובץ בנתיב שבחרנו.

- במהלך היצירה התעלמתי מהקובץ jazz.00054.wav, מכיוון שהקובץ ככל הנראה אינו נוצר כראוי ולכן מעלה שגיאה כשאנו מנסים לגשת אליו:

"fileError: Error opening

'/content/drive/MyDrive/ColabNotebooksNew/PROJECT/ProjectMusic/Data/genres_original/jazz/jazz.00054.wav': File contains data in an unknown format."

כאשר מתבצעת הפונקציה MFCC, היא לוקחת את כל החלקים (ישנם 1000 שירים, שמחולקים ל-6 חלקים שונים אך נתעלם מקטע שלם, כלומר 6 חלקים).

נוודא שגודל ה-MFCC של המקטע (segment) הוא כראוי. אך, נמצא שישנם שני מקטעים מתוך השירים שה-MFCC שלהם אינו בנוי כראוי והם:

- hiphop.00032.wav בשניות 25-30.
- country.00007.wav בשניות 25-30.

מספר הקטעים הסופי שהוכנסו, הם 5992 מכיוון שהתעלמנו משיר שלם (6 חלקים), ו-2 חלקים נוספים מתוך השירים. הגודל של כל MFCC יהיה (216,13) כלומר יש חישוב של 13 מקדמי MFCC, על כל דגימה, למשך 216 דגימות בקטע, ויהיו סך הכל 5992 מקטעים (Segments) כאלה.

רקע תיאורטי 2

מה זה קונבולוציה (convolution)?

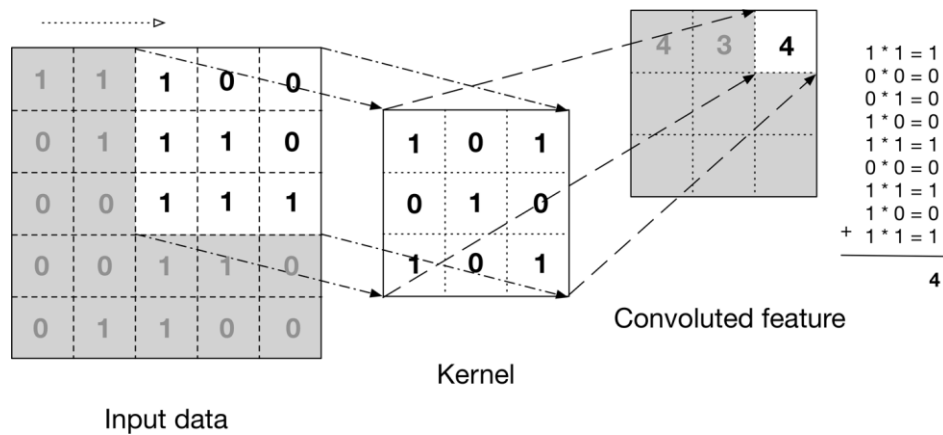
שכבת קונבולוציה היא אבן הבניין המרכזית של CNN. אנו יכולים לחשוב על תמונת הקלט כעל מטריצה, שבה כל ערך מייצג כל פיקסל, וערך בין 0 ל-255 המייצג את עוצמת הבהירות. כאשר מעבדים תמונה צבעונית, ולוקחים בחשבון את הצבעים, יהיו 3 ערוצים, מסוג RGB. במקרה שלי אני משתמש במימד אחד (שניתן להשוות לתמונה בשחור לבן - ערוץ 1), וערכי המספרים תלויים בערכי ה-MFCCs.

אחת הדרכים להבין את פעולת הקונבולוציה היא לדמיין הצבת Filter הקונבולוציה בחלק העליון של תמונת הקלט, הממוקם בצורה כך שה-Kernel והתמונה בפינות השמאליות העליונות חופפות, ולאחר מכן הכפלת הערכים של מטריצת תמונת הקלט בהתאם, עם הערכים המתאימים ב-Filter הקונבולוציה (כפל מסוג dot product).

כל הערכים המוכפלים מתווספים יחדיו וכתוצאה מכך נוצר סקלר יחיד, אשר ממוקם במיקום הראשון של מטריצת תוצאה.

לאחר מכן, ה-Kernel יזוז X פיקסלים ימינה, כאשר X מייצג את אורך הצעד (Stride) שאנו יכולים לקבוע, תהליך החישוב חוזר על עצמו, והתוצאה נרשמת בחלק השני של מטריצת התוצאה. לאחר שהסתיימה שורה, ה-Kernel יזוז X פיקסלים מטה בהתאם עד שיסתיים התהליך. לבסוף, יצא פלט המכונה Feature map.

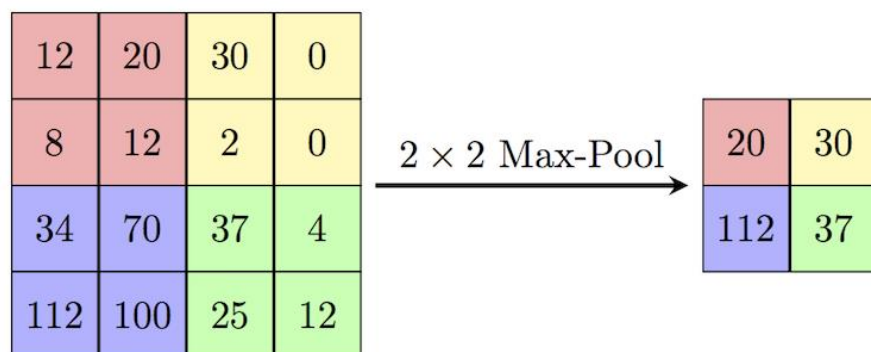
ככל שגודל ה-kernels גדול יותר ככה יהיה זיהוי מדויק יותר אבל השכבה תדרוש יותר כוח חישוב והמודל ירוץ לאט יותר.



איור: דוגמא לפעולת קונבולוציה¹⁴

מה עושה max pooling?

max pooling דומה לתהליך הקונבולוציה בכך שישנו גודל Kernel מסוים שנקבע, ויש אפשרות לבחור Stride מסוים. הפונקציה תקח מתוך ה-Kernel את המספר המקסימלי שנמצא בקלט שקיבלנו ותוציא אותו. היא תעשה את התהליך ותוציא פלט עם המספרים המקסימייליים בכל איזור שחישבה.



איור: דוגמא לפעולת max pooling¹⁵

¹⁴ קישור: <https://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets>

¹⁵ קישור: <https://computersciencewiki.org/index.php/File:MaxpoolSample2.png>

המשמעות של global average pooling?

global average pooling היא שיטה המשמשת למעבר מרשת CNN, לרשת DNN (תהליך שיטוח).

הקלט מורכב משלושה מימדים - אורך, רוחב ועומק - feature map. בשיטת global average pooling ייקח כל feature map וייחשב את הממוצע שלו, לבסוף יישאר מערך $(X, 1, 1)$ (כאשר X מייצג את הממוצעים מה-feature maps שנלקחו).

מהי פונקציית שגיאה sparse categorical cross-entropy?

פונקציית שגיאה, זו פונקציה המשמשת למדידה עד כמה מודל מסוים בלמידת מכונה מתפקד במשימה נתונה.

במהלך האימון, המודל מבצע תחזיות על קבוצה של קלטים, ומשווה אותן לסיווג בפועל (כלומר, משווה לדוגמא חיזוי ז'אנר שיר לעומת הז'אנר האמיתי שלו), פונקציית ההפסד מחשבת את ההפרש בין החיזויים, לסיווג בפועל, ומחזירה ערך בודד המייצג עד כמה תחזיות המודל היו "שגויות".

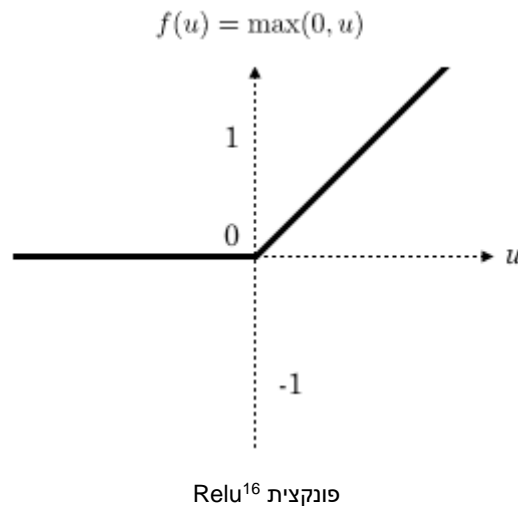
בזמן האימון, מטרתנו היא למזער כמה שניתן את הערך של פונקצת ההפסד - בכך, 'הפכו התחזיות ליותר מדויקות'.

פונקציות אקטיבציה:

relu:

היא פונקציית הפעלה המשמשת ברשתות נוירונים בשל הפשטות והיעילות שלה בשיפור ביצועי המודל. זוהי פונקציה מתמטית פשוטה שעוזרת לרשת לקבל החלטות לגבי איזה מידע לשמור ומה לזרוק.

אם ערך הקלט ל-ReLU חיובי, הוא ישמור על אותו ערך כמו הפלט. אבל אם ערך הקלט שלילי, הוא יקבע לאפס. במילים אחרות, ReLU "מופעל" עבור ערכים חיוביים ו"מכבה" ערכים שליליים. הגדרת ערכים שליליים לאפס עוזרת לרשת ללמוד תכונות חשובות מהנתונים ולהתעלם ממידע פחות רלוונטי.

softmax:

פונקציית ההפעלה של softmax משמשת בשכבת הפלט של מודל רשת הנוירונים עבור משימות סיווג, כאשר המטרה היא לחזות את ההסתברות של כל מחלקה בהינתן קלט. בקוד הספציפי הזה, המודל מיועד לסווג תמונות ל-10 מחלקות שונות, כך שלשכבת הפלט יש 10 נוירונים, אחד לכל מחלקה. פונקציית ההפעלה של softmax משמשת להמרת הפלטים של נוירונים אלה להתפלגות הסתברות על פני 10 המחלקות (כך שסכומן יהיה 1, קרי 100%), ולבצע תחזיות על סמך מחלקת ההסתברות הגבוהה ביותר.

מהי שכבת batch normalization?

שכבת batch normalization מחשבת את הממוצע ואת סטיית התקן של הקלט לשכבה, בתוך כל mini-batch (תת-קבוצה קטנה יותר של נתוני אימון המשמשים במהלך כל איטרציה של תהליך האימון) במהלך האימון. הנורמליזציה מוחלת על כל נוירון קלט בשכבה, ללא תלות בניירונים האחרים, תוך שימוש בממוצע וסטיית התקן המחושבים באמצעות mini-batch. שיטה זו, מפחיתה את ההשפעה של התפלגות ערכי הקלטים על ביצועי המודל (כלומר, נירמול ערכם), ועוזר לייצב את תהליך האימון ולשפר את הביצועים הכלליים וההתכנסות של הרשת.

¹⁶ קישור: https://www.researchgate.net/figure/ReLU-activation-function_fig3_319235847

שלב בנייה ואימון המודל:

תיאור המודל:

במודל השתמשתי ברשת נוירונים מסוג CNN, מכיוון שהיא מתאימה לסיווג ז'אנר באמצעות MFCCs. הסיבה לכך, שמבנה רשת נוירונים מסוג CNN, יכול ללמוד לזהות דפוסים ותכונות בייצוגי MFCC, מה שהופך אותו ליעיל בחילוף מידע שימושי לסיווג ז'אנר.

בניתי מודלים שונים ושיפרתי את המודל:

ראשית, לקחתי את המודל המקורי בו נעזרתי –

```
model = keras.Sequential()

# 1st conv layer
model.add(keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))
model.add(keras.layers.BatchNormalization())

# 2nd conv layer
model.add(keras.layers.Conv2D(32, (3, 3), activation='relu'))
model.add(keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))
model.add(keras.layers.BatchNormalization())

# 3rd conv layer
model.add(keras.layers.Conv2D(32, (2, 2), activation='relu'))
model.add(keras.layers.MaxPooling2D((2, 2), strides=(2, 2), padding='same'))
model.add(keras.layers.BatchNormalization())

# flatten output and feed it into dense layer
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(64, activation='relu'))
model.add(keras.layers.Dropout(0.3))

# output layer
model.add(keras.layers.Dense(10, activation='softmax'))

return model
```

```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 214, 11, 32)        320
max_pooling2d (MaxPooling2D) (None, 107, 6, 32)         0
batch_normalization (BatchN (None, 107, 6, 32)         128
ormalization)
conv2d_1 (Conv2D)            (None, 105, 4, 32)         9248
max_pooling2d_1 (MaxPooling (None, 53, 2, 32)          0
2D)
batch_normalization_1 (Batc (None, 53, 2, 32)         128
hNormalization)
conv2d_2 (Conv2D)            (None, 52, 1, 32)         4128
max_pooling2d_2 (MaxPooling (None, 26, 1, 32)          0
2D)
batch_normalization_2 (Batc (None, 26, 1, 32)         128
hNormalization)
flatten (Flatten)            (None, 832)                 0
dense (Dense)                 (None, 64)                  53312
dropout (Dropout)            (None, 64)                   0
dense_1 (Dense)               (None, 10)                   650
=====
Total params: 68,042
Trainable params: 67,850
Non-trainable params: 192

```

לאחר מכן, ניסיתי לשפר ולשנות את המודל ולראות את התוצאות (השינוי מודגש)-

| Optimizer | Learning rate | batch | epoch | Test Accuracy | More Information |
|----------------|----------------|-------|-------|---------------|--|
| Adam | 0.0001 | 32 | 30 | 68.95% | (Original model) |
| SGD | 0.0001 | 32 | 30 | 33.24% | |
| RMSprop | 0.0001 | 32 | 30 | 57.84% | |
| Adam | 0.0001 | 32 | 30 | 81.33% | 10% Test set, 9% Validation Set, 81% Training Set |
| Adam | 0.00015 | 32 | 30 | 83.66% | |

-
- ניסיתי לשנות את אחוזי האימון, שיפור ובדיקה וראיתי שהאחוזים אופטימליים בסביבות (10% Test set, 9% Validation set, 81% Training set)
 - לאחר מכן, ניסיתי אופטימיזרים שונים כגון SGD, RMSProp ועוד, אך ראיתי ש-Adam הביא את התוצאות המקסימליות.
 - שיניתי את הקצב הלמידה והוספתי פונקציה שתקטין את קצב הלמידה בזמן האימון עצמו - ששיפר את המודל.
 - הגדלתי את כמות ה-Epochs והוספתי Earlystopping - שראיתי כי שיפר את תוצאות המודל.
 - שיחקתי עם מבנה הקונבולוציה עד כי הגעתי למבנה האופטימלי(ניסיתי לשנות את מבנה הקונבולוציה, להוסיף גולריזציה ועוד).

```
# build network topology
model = keras.Sequential()

model.add(keras.layers.Conv2D(32, (3, 2), activation='relu', input_shape=input_shape, kernel_regularizer=tf.keras.regularizers.L1L2(0.01)))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv2D(64, (3, 2), activation='relu'))

model.add(keras.layers.MaxPooling2D((3, 2), strides=(2,1)))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv2D(64, (3, 2), activation='relu'))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv2D(128, (3, 2), activation='relu'))

model.add(keras.layers.BatchNormalization())

model.add(keras.layers.MaxPooling2D((3, 2)))
# flatten output and feed it into dense layer
model.add(keras.layers.GlobalAveragePooling2D())

model.add(keras.layers.Dropout(0.2))
# output layer
model.add(keras.layers.Dense(10, activation='softmax'))

return model
```

המודל נבנה באמצעות ספריית Keras שנמצאת בספריות הפתוחות של Tensorflow. מטרת מודל זה היא לסווג נתוני קלט לאחת מעשר קטגוריות אפשריות באמצעות למידה מפקחת.

צורת הקלט של הנתונים היא (1, 13, 216), מה שאומר שלנתוני הקלט יש 216 שורות, 13 עמודות וערוץ יחיד.

מבנה ה-CNN

1. שכבת קונבולוציה שמקבלת כגודל את הקלט, עם 32 מסננים בגודל (3,2) (הסיבה לגודל ה"מלבני" הוא גודל הקלט שאנו מקבלים) עם פונקציית הפעלה של פונקציית ReLU ורגולריזציה-L1L2 בגודל (0.01) שמונעת התאמת יתר ומשפרת את למידת המודל.
2. שכבת BatchNormalization.
3. שכבת קונבולוציה שנייה עם 64 מסננים בגודל (3,2) עם פונקציית הפעלה של Relu.
4. שכבת MaxPooling2D עם גודל (2, 3) ו Stride בגודל (2, 1), מה שמקטין את הממדים המרחביים של הפלט.
5. שכבת נוספת של BatchNormalization.

6. שכבת קונבולוציה שלישית עם 64 מסננים בגודל (3,2) עם פונקצית הפעלה של Relu
7. שכבת נוספת של BatchNormalization.
8. שכבת קונבולוציה רביעית עם 128 מסננים בגודל (3,2) עם פונקצית הפעלה של Relu
9. שכבת נוספת של BatchNormalization.
10. שכבת MaxPooling2D עם גודל (2,3), מה שמקטין את הממדים המרחביים של הפלט.
11. שכבת GlobalAveragePooling2D משטחת את הפלט ומזינה אותו לשכבה צפופה (DNN).

מבנה ה-DNN

1. מתווספת שכבת Dropout בשיעור של 0.2, כלומר במהלך האימון 20% מן הנירונים בשלב הלמידה יהיו מוגדרים לאפס בצורה רנדומלית (ובכך לא יהיה להם ייחוס) למניעת התאמת יתר (Overfitting).
2. לבסוף, שכבה צפופה עם 10 נירונים ופונקציית אקטיבציה של softmax מתווספת כשכבת הפלט כדי ליצור את הניבויים שבמחלקה.

```

1 # compile model
2 optimizer = keras.optimizers.Adam(learning_rate=0.00015)
3 model.compile(optimizer=optimizer,
4               loss='sparse_categorical_crossentropy',
5               metrics=['accuracy'])
6
7 model.summary()
8
9 earlystopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=8,verbose=1) #patience=stop after 8 epochs without improvement(on vall loss)
10
11 reduceLR = tf.keras.callbacks.ReduceLRonPlateau(monitor='val_loss', factor=0.9,patience=4, verbose=1, mode='auto')
12
13 # train model
14 history = model.fit(X_train, y_train, validation_data=(X_validation, y_validation), batch_size=64, epochs=150, verbose=1,callbacks=[earlystopping, reduceLR])
15
16 duration = datetime.now()
17
18 # evaluate model on test set
19 test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)
20 print('\nTest accuracy:', test_acc)
21 print("Training completed in time:", duration)

```

משתמשים באופטימיזר Adam, נגדיר את קצב הלמידה עבור האופטימיזציה ל-0.00015. תפקיד האופטימיזר, הוא עידכון המשקולות (weights) וה-bias תוך כדי האימון, על מנת לשפר את תוצאות האימון תוך כדי למידה.

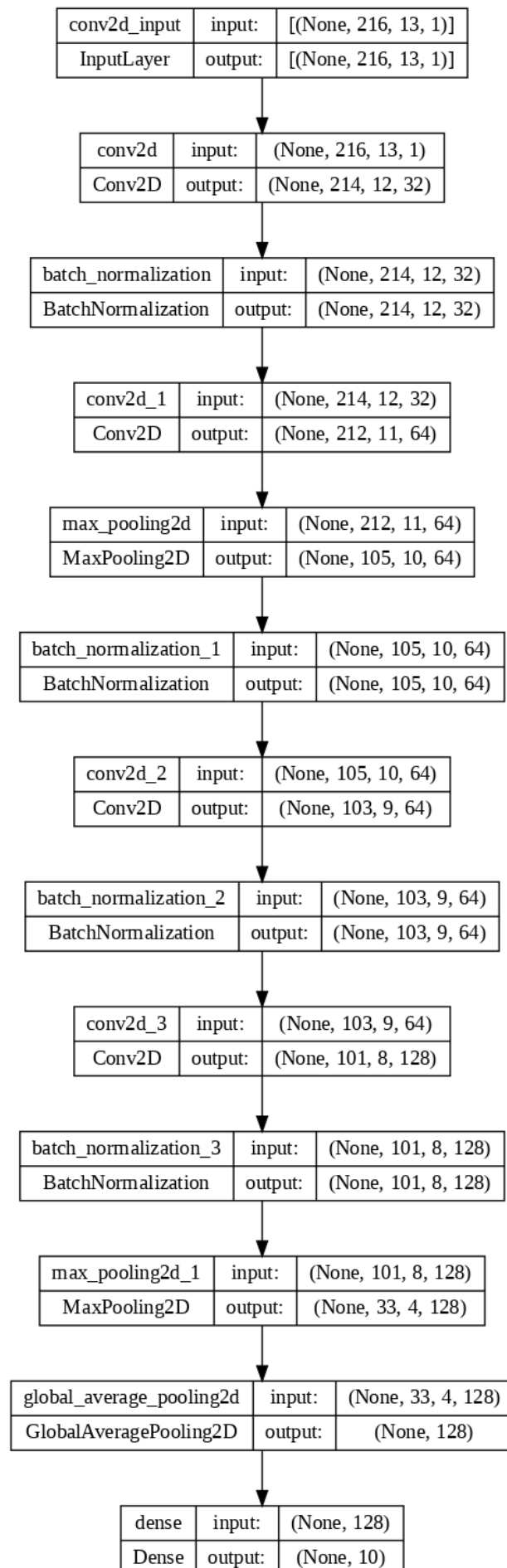
נבנה את המודל ונשתמש בפונקציית loss (שגיאה) בזמן האימון של sparse categorical cross-entropy, ונשתמש במדד הערכה באמצעות הדיק.

נשתמש ב-EarlyStopping על מנת במקרה ואין שיפור באחוזי המודל (באמצעות סט val) לאחר 8 אפוקים, המודל יפסיק להתאמן.

נשתמש ב-ReduceLROnPlateau, במידה ונראה לאחר 4 אפוקים כי אין שיפור, נקטין את ערך learning rate (קצב הלמידה), בכך ננסה להגיע לאחוזים מעט יותר מדויקים.

נבנה את המודל ונשתמש בפונקציה `model.fit`.

- batch size = 64
- epochs = 150



Model: "sequential_9"

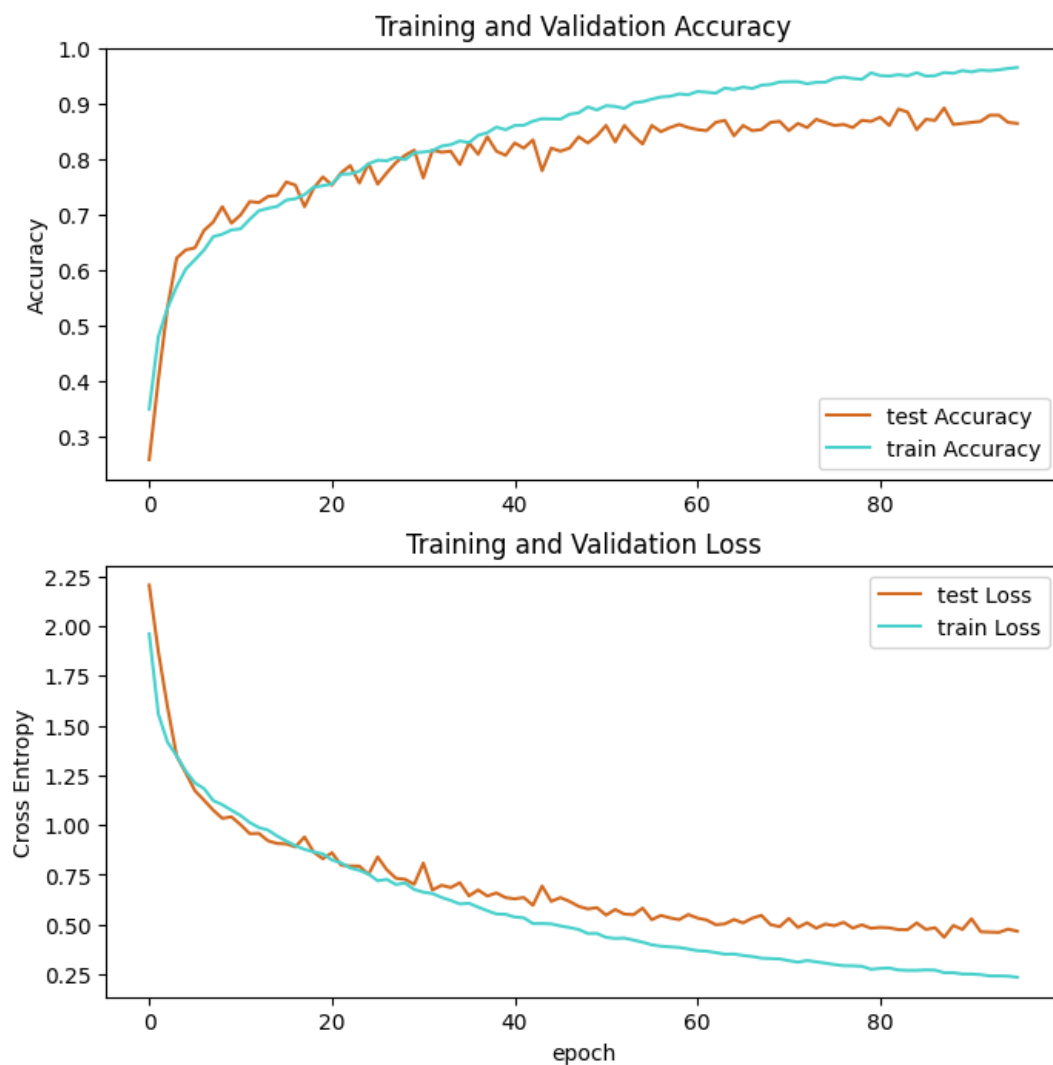
| Layer (type) | Output Shape | Param # |
|---|---------------------|---------|
| conv2d_24 (Conv2D) | (None, 214, 12, 32) | 224 |
| batch_normalization_24 (Batch Normalization) | (None, 214, 12, 32) | 128 |
| conv2d_25 (Conv2D) | (None, 212, 11, 64) | 12352 |
| max_pooling2d_12 (MaxPooling2D) | (None, 105, 10, 64) | 0 |
| batch_normalization_25 (Batch Normalization) | (None, 105, 10, 64) | 256 |
| conv2d_26 (Conv2D) | (None, 103, 9, 64) | 24640 |
| batch_normalization_26 (Batch Normalization) | (None, 103, 9, 64) | 256 |
| conv2d_27 (Conv2D) | (None, 101, 8, 128) | 49280 |
| batch_normalization_27 (Batch Normalization) | (None, 101, 8, 128) | 512 |
| max_pooling2d_13 (MaxPooling2D) | (None, 33, 4, 128) | 0 |
| global_average_pooling2d_6 (GlobalAveragePooling2D) | (None, 128) | 0 |
| dropout_3 (Dropout) | (None, 128) | 0 |
| dense_4 (Dense) | (None, 10) | 1290 |
| Total params: 88,938 | | |
| Trainable params: 88,362 | | |
| Non-trainable params: 576 | | |

תוצאות והערכת ביצועים:

מודל סופי:

- אחוזי דיוק: 91%
- Optimizer: Adam
- Learning rate: 0.00015 (עם תהליך הפחתה במהלך הלמידה)
- batch size: 64
- epoch size: 150 (עם EarlyStopping)
- טבלאות ונתונים נוספים -

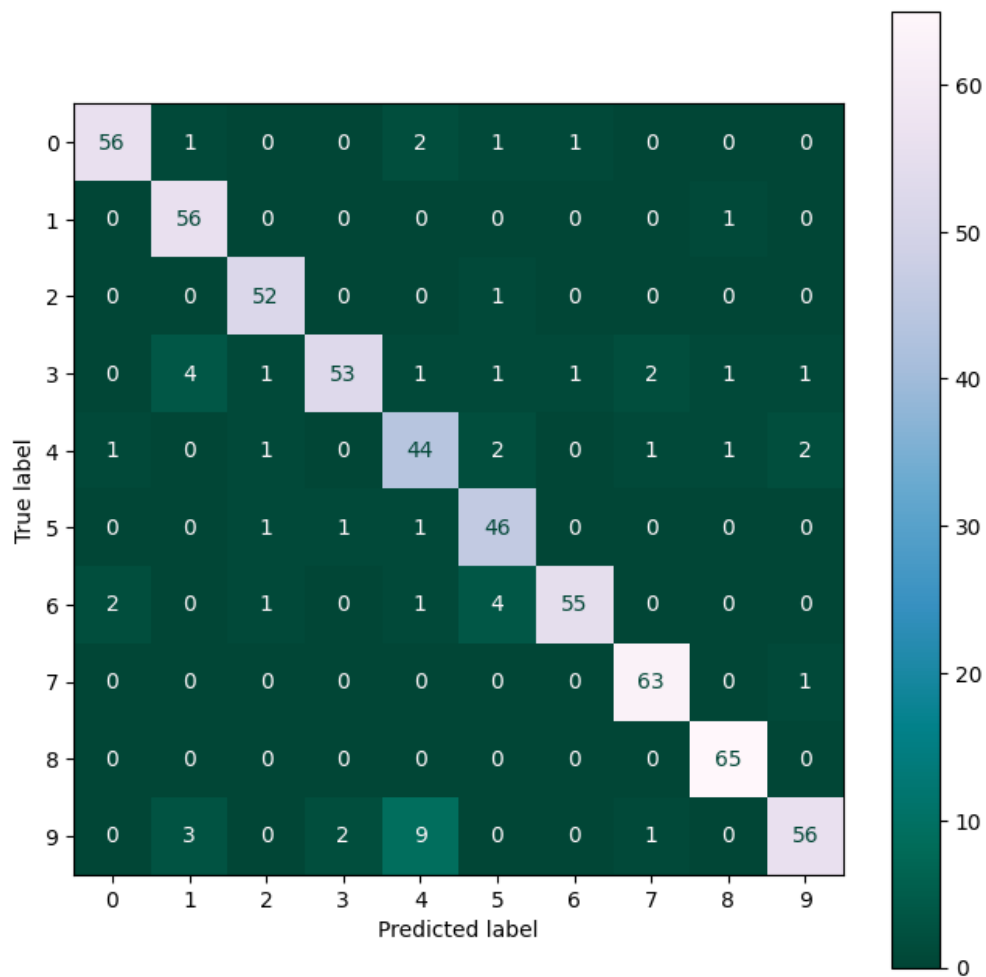
תיעוד גרף של אחוזי הדיוק וה-Loss באימון, ובמבחן (ניתן לראות שיש מעט אוברפיטינג שניסיתי להתמודד איתו במהלך הלמידה והקטנתי אותו בתהליך)



Classification report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.9492 | 0.9180 | 0.9333 | 61 |
| 1 | 0.8750 | 0.9825 | 0.9256 | 57 |
| 2 | 0.9286 | 0.9811 | 0.9541 | 53 |
| 3 | 0.9464 | 0.8154 | 0.8760 | 65 |
| 4 | 0.7586 | 0.8462 | 0.8000 | 52 |
| 5 | 0.8364 | 0.9388 | 0.8846 | 49 |
| 6 | 0.9649 | 0.8730 | 0.9167 | 63 |
| 7 | 0.9403 | 0.9844 | 0.9618 | 64 |
| 8 | 0.9559 | 1.0000 | 0.9774 | 65 |
| 9 | 0.9333 | 0.7887 | 0.8550 | 71 |
| accuracy | | | 0.9100 | 600 |
| macro avg | 0.9089 | 0.9128 | 0.9085 | 600 |
| weighted avg | 0.9138 | 0.9100 | 0.9095 | 600 |

Confusion Matrix:



שלב היישום

את היישום כתבתי בעזרת הספרייה Gradio, שכפי שפורט עליה היא מאפשרת לכתוב ממשק משתמש נוח ופשוט, ומשומשת בעיקר לבניית Web עם פרוויינקטים של למידת מכונה. במהלך היישום, היישום מקבל קובץ שיר,

- ממיר אותו לקובץ מסוג Wav (כך שיתאים להמרת MFCC)
- משנה את Sample rate של השיר בכך שיהיה מתאים לפונקציה - שנמיר ל-MFCC
- משתמש בפונקציה Predict, שמנבאה הרבה חלקים מהשיר, ומציגה חיזויים בעזרת מספר עשרוני (של אחוז של חיזוי של הז'אנר הספציפי מתוך כל החיזויים של קטעי השיר)
- מחזיר את החיזויים (שממומרים ב-Gradio לאחוזים אוטומטית), ביחד עם זמן האימון שלקח, MelSpectrogram ו-Waveform של השיר.

מדריך למשתמש

ראשית נגיע לאתר הזה:

Classify

Hello there! Welcome to Classify made by Liav. Please add a song you would like classify, and wait until you receive the song genre-classification!

Source

Drop Audio Here
- or -
Click to Upload

Clear

Submit

Prediction

Prediction time

0

Waveform

MelSpectrogram

Use via API · Built with Gradio

נוסיף לSource קובץ 3mp שנרצה לסווג -

Classify

Hello there! Welcome to Classify made by Liav. Please add your song and wait to your song to be predicted which genre it is in!

Source

0:00 / 2:02

Clear

Submit

Prediction

Prediction time

0

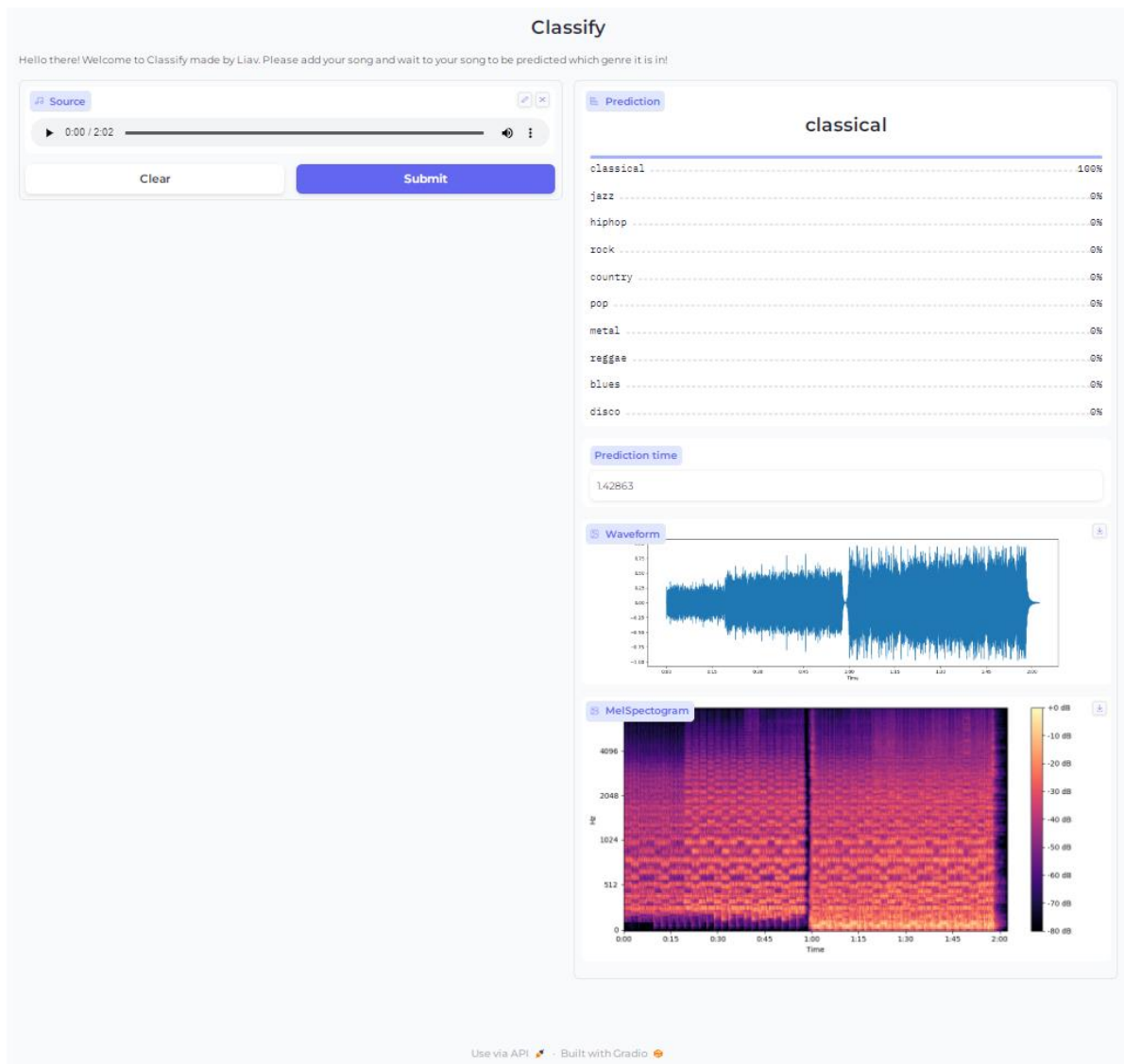
Waveform

MelSpectrogram

Use via API · Built with Gradio

נלחץ על כפתור Submit ונחכה.

כעת, כפי שניתן לראות נקבל סיווג משוער (באחוזים) ביחד עם זמן החיזוי שלקח, ביחד עם תיאור ה-Waveplot של השיר וה-MelSpectrogram. את התמונות ניתן להוריד למחשב. כעת, במידה ונרצה להוסיף שיר אחר, נלחץ על Clear, ונתחיל מהשלב הראשון.



מדריך למפתח

קישור ל-Github ובו נמצא הפרוייקט עצמו שנכתב בCollab, יחד עם קישורים לקבצים הנדרשים: <https://github.com/L33TA/GenreClassificationGTZAN-with-CNN>

סיכום אישי / רפלקציה

הפרוייקט היה משמעותי מאוד בשבילי, העבודה על הפרויקט הייתה מאוד מאתגרת וגם מאוד מעניינת, מכיוון שבחרתי נושא שקרוב לליבי - מוזיקה. קיבלתי המון ערך, בין אם מדובר על ידע בלמידת מכונה, התעסקות בספריות שאינן בחומר הלימוד, ויחד עם תמיכתה של אולגה, למדתי רבות באופן עצמאי מתמטיקה ופיזיקה בכל הנוגע למוזיקה וסיווג במוזיקה. נוסף על כך, למדתי רבות על עצמי, על העצמאות והיכולות שלי להתמודד עם אתגרים בדרך. אני שמח מאוד שהגעתי להישגים גבוהים באחוזי הסיווג, אך ישנם מקומות נוספים שניתנים לשיפור - ביניהם ייעול הזמן שלוקח ליישום, הוספת ושיפור התוכן ועיצוב היישום, וייעול המודל לאחוזים גבוהים עוד יותר.

להמשך אני לוקח איתי את הכלים, של עבודה מסודרת, עמידה בזמנים, לא לוותר גם אם משהו לא עובד כי בסוף הצלחתי ועמדתי בקשיים ובמכשולים וכמובן, הידע הרב שרכשתי.

בין היתר, קשיים שהייתי צריך להתמודד איתם הם -

- עמידה בזמני ההגשות.
- ידע מתמטי ופיזיקלי בכל הנוגע למוזיקה ועריכת מוזיקה, שלא היה לי לפני כן.
- למידה ותכנות בחומר שלא הכרתי לפני כן.
- שינוי ובאגים בקוד, בין אם בפונקציות מתמטיות שהיו לא פשוטות לתקן או בבאג קטן בגלל עדכון באחת הספריות שבהן השתמשתי.

המסקנות שלי, במיוחד לאור המצב כיום בתחום למידת המכונה, שזהו חלק שהולך להיות אבן יסוד בעתיד של מדעי המחשב ובכלל ביום יום שלנו. בנוסף לכך, הבנתי המון על עצמי במהלך הדרך.

אילו הייתי מתחיל כיום, הייתי עובד באופן פעולה יותר מסודר (למשל: לסדר לפני כן את כל הקוד), והייתי חוקר יותר לעומק על הספריות והאמצעים שבהם אני משתמש ומוודא שניתן לעבוד איתם למטרותיי.

אם העבודה הייתה יותר יעילה עבורי, הייתי לדעתי מסיים יותר מוקדם ואולי גם מספיק להוסיף עוד דברים לפרוייקט כגון שיפור של המודל, או שיפור היוזאוליות ביישום.

ביבליוגרפיה

- Bob, L. Sturm (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. <https://arxiv.org/pdf/1306.1461.pdf>
- Guangxiao, S., Zhijie W., Fang H., Shenyi D. (2017). Transfer Learning for Music Genre Classification. <https://hal.inria.fr/hal-01820925/document>
- VRUSHALI. (2021). Music Genre Classification Using CNN. <https://www.kaggle.com/code/vrushalinge/music-genre-classification-using-cnn/notebook>
- Harsh M. (2021). Terms you need to know to start Speech Processing with Deep Learning. <https://towardsdatascience.com/all-you-need-to-know-to-start-speech-processing-with-deep-learning-102c916edf62>
- Tanveer, S. (2019). MFCC's Made Easy <https://medium.com/@tanveer9812/mfccs-made-easy-7ef383006040>