הפקולטה להנדסה

המעבדה לעיבוד אותות

Audio Segments Alignment

אילת ארנסט

שקד מזרחי

פרויקט שנה ד' לקראת תואר ראשון בהנדסה

מנחה: אביעד אייזנברג

מנחה אקדמי: פרופ' שרון גנות

אוקטובר 2024

תוכן עניינים

הצגת הבעיה 2
רקע תאורטי 4
ייצור הדאטה 4
ניתוח ספקטרלי של אות דיבור 6
מאפייני למידה עמוקה 9
שכבות במודלי למידה עמוקה 14
כלים ופתרונות לניהול ופיתוח בלמידה עמוקה 17
מדדי ביצועים 18
דרכי פתרון 21
גישות קלאסיות לעיבוד אותות 21
מודלים סטטיסטיים 21
רשתות עצביות עמוקות 21
הגישה הנבחרת - CNN&SA 22
מימוש המודל 23
ייצור הדאטה 23
ייצור RIR 23
ייצור הסיגנל 24
ארכיטקטורת המודל 26
אימון המודל 28
תוצאות 29
מסקנות וסיכום 33

הצגת הבעיה

voice activity detection - VAD

VAD - זיהוי של נוכחות או היעד דיבור אנושי.

VAD הינה טכנולוגיה חשובה אשר ממלאת תפקיד מרכזי התחום עיבוד אותות דיגיטליים. רבים מיישומים מבוססי דיבור מבוססים על אלגוריתם VAD כך שהוא נמצא בשימוש נרחב. ה-VAD משמש בדרך כלל כטריגר לאלגוריתמים שונים של עיבוד אודיו, כגון מערכות תקשורת דיבור, זיהוי דיבור אוטומטי (ASR) וביטול הד. במתחם רועש ו/או מהדהד, האתגר של VAD נהפך קשה עוד יותר, כך שיעילות אלגוריתם הVAD הולכת ופוחתת ככל שיחס האות לרעש גדל. על כן נדרש לשפר את האלגוריתם לתוצאה הטובה ביותר כתלות במשתנים שונים.

במקרה של אות כניסה מהודהד של דיבור בתוספת רעש, נרצה להשתמש בVAD על מנת לזהות את החלקים בהם היה דיבור. אות הדיבור מושפע מהסביבה האקוסטית שבה הוא נמצא. נוצר הדהוד של האות לפי ה – RIR – room impulse response שלו. הRIR המתאר את התגובה של הסביבה (החדר) לאות. בנוסף לתגובת החדר לאות ישנם רעשים נוספים הנקלטים ברמקול - רעשים מהסביבה / של הרמקול עצמו.

בעיית VAD נתונה הינה זיהוי נוכחות של דיבור עבור קלט של דיבור מהודהד בתוספת רעש. כלומר בהינתן אות דיבור מהודהד בתוספת רעש - $x(t)$, נדרש לדעת מהם החלקים בהם יש דיבור.

הבעיה בצורה מתמטית

נניח אות כניסה למיקרופון $x(t)$, אשר הינו אות דיבור מהודהד בליווי רעש.

ממודל באופן מתמטי לפי -

$$x(t) = s(t) * h(t) + n(t)$$

אות הדיבור המהודהד נוצר ע"י דיבור שעובר בRIR מהודהדת ונקלט במיקרופון עם רעשים.

כאשר: $s(t)$ - אות הדיבור הלא מהודהד.

$h(t)$ - התגובה להלם של החדר בין הדובר למיקרופון. ידועה כRIR

$n(t)$ - הרעש שמתווסף לאות.

סימן * הינו קונבולוציה בין אות הדיבור לתגובה להלם של החדר.

במימד הSTFT הקונבולוציה הופכת לפעולת כפל כך שאות הכניסה הינו כדלהלן –

$$x(l,f) = I(l) \cdot s(l,f) \cdot h(l,f) + n(l,f)$$

כאשר: l הוא מספר הפריים של ההתמרה ונע בטווח $0 \leq l \leq L-1$.

f הוא האינדקס של התדירות ונע בטווח $0 \leq f \leq F-1$.

$I(l)$ הינו הנעלם שנרצה לגלות. הוא מציין את הפריימים בהם יש דיבור. נקבע לפי הסכום של s(l,f) עבור תדירויות שונות. אם סכום זה גדול מסף TH כלשהו שאותו נגדיר, I(l) יהיה שווה 1, וייצג נוכחות של אות דיבור באותו הפריים, ואחרת I(l) יהיה שווה ל0, וייצג היעדר דיבור.

באופן מתמטי –

$$I(l) = \begin{cases} 0 & \sum_{f=0}^{F-1} |s(l,f)| < Th \\ 1 & \sum_{f=0}^{F-1} |s(l,f)| \geq Th \end{cases}$$

מטרת ה-VAD -

לזהות במדויק פריימים שבהם הדיבור פעיל. אצלנו בפרויקט, נבחן אות מהודהד רועש ונמצא את הפריימים בהם היה דיבור.

בצורה פורמלית. המטרה הינה למצוא את -

$$V(l) = p(I(l) = 1|x, r).$$

רקע תאורטי

ייצור הדאטה

מאגר נתונים LibriSpeech

מאגר נתונים גדול של הקלטות אודיו של דיבור, אשר נאספו מכמה מאות שעות של ספרי שמע ברשות הציבור. המאגר מכיל דוגמאות מגוונות של דיבור, כולל דוברים שונים, סגנונות דיבור, קצב דיבור ורמות רעש רקע. המאגר מכיל כמות גדולה של נתונים ,מה שהופך אותו אידיאלי לאימון מודלים עמוקים הדורשים כמויות גדולות של נתונים לביצועים טובים .המגוון הרחב של הדוגמאות מאפשר למודל ללמוד לזהות דיבור בתנאים שונים. בנוסף, הקלטות ב LibriSpeech-הן באיכות גבוהה יחסית ,מה שמפשט את תהליך הקדם-עיבוד של הנתונים.

בפרויקט שלנו,

הסיגנלים 'הנקיים' של הדוברים השונים - $s(t)$ הינם מתוך מאגר הנתונים של LibriSpeech.

מאגר נתונים WHAM

WHAM (Washoe Hotel Acoustic Meeting) הוא מערך נתונים אקוסטי ייחודי שנועד לאמן ולעריך מערכות של הפרדה של מקורות קול (speech separation) בתנאי חדר אמיתיים ומורכבים .בניגוד למערכות נתונים סינתטיות רבות, WHAMמציע הקלטות אותנטיות שנאספו בסביבת חדר ישיבות מה שהופך אותו למתאים במיוחד לפיתוח טכנולוגיות שמע שיוכלו לפעול בסביבות כאלה.

<u>מאפיינים בולטים של WHAM :</u>

- סביבה אקוסטית מגוונת :ההקלטות ב WHAM-נאספו במספר חדרים שונים במלון ,כל אחד עם אקוסטיקה ייחודית .זה מאפשר למודלים ללמוד להתמודד עם השפעות של הד ,החזר והפרעות אחרות.
- מקורות קול מרובים :מערך הנתונים כולל הקלטות של מספר דוברים בו זמנית ,מה שמאפשר לאמן מערכות להפריד בין קולות של דוברים שונים.
- רעשי רקע :בנוסף לקולות דיבור ,מערך הנתונים כולל גם רעשי רקע שונים כמו צעדים ,דלתות שנפתחות וסגירות ,ורעשי רקע כלליים של מלון.
- מיקרופונים מרובים :ההקלטות ב WHAM-נעשו באמצעות מערך של מיקרופונים ,מה שמספק מידע מרחבי על מקורות הקול.

בפרויקט שלנו,

הרעשים - $n(t)$ המוספים לסיגנלים הנקיים הינם ההקלטות מתוך מאגר WHAM.

RIR – room impulse response.

ייצוג של האופן שבו חדר או כל חלל סגור מגיבים לפרץ קצר של צליל, הנקרא לעתים קרובות דחף (impulse). הדחף הזה יכול להיות מחיאת כפיים חזקה, פיצוץ בלון או כל צליל פתאומי וקצר אחר. זוהי בעצם תמונת מצב של האופן שבו קול מתנהג בחדר לאורך זמן.

כאשר קול פוגע במשטח בחדר ,חלק ממנו מוחזר ,חלק נבלע וחלק עובר דרך המשטח .החלק המוחזר ממשיך להתפשט בחדר ,פוגע במשטחים נוספים ומוחזר שוב ושוב .התוצאה היא סדרה של השתקפויות המגיעות למיקרופון במרווחי זמן שונים ועוצמות שונות. RIR הוא למעשה תיאור מתמטי של תהליך זה ,והוא מורכב מסדרה של דחפים (impulses) המייצגים את ההשתקפויות הללו.

בפרויקט שלנו,

ה-RIRים - $h(t)$ יוצרו ע"י שימוש בספריית rir_generator בפייתון ושומשו על מנת ליצור אות מהודהד

Reverberation Time (RT60)

RT60הוא מדד אקוסטי המייצג את הזמן הנדרש להיחלשות עוצמת הקול ב-60 דציבלים לאחר הפסקת מקור הקול .במילים פשוטות ,זהו הזמן שלוקח להדהוד בחדר לדעוך עד לרמה שקשה לשמוע אותה.

בפרויקט,

הוגדרו פרמטרים לקביעת מדד זה.

SNR

הוא מדד המשמש להשוואה בין האות הרצוי לבין רעשי הרקע. ה-SNR מוגדר כיחס בין הספק האות להספק הרעש, לרוב הוא מבוטא בdB.

מחושב לפי –

$$SNR(dB) = 10\log_{10}(\frac{P_{signal}}{P_{noise}})$$

SNR הוא פרמטר חשוב המשפיע על הביצועים והאיכות של מערכות רבות.

SNR חיובי מציין שיש יותר אות מאשר רעש ובמצב זה האות ברור וקל לזיהוי.

SNR שלילי מציין שיש יותר רעש מאשר אות דיבור, במצב זה האות פגום או מעורפל על ידי רעש ועשוי להיות קשה לזיהוי ולכן בעיית הVAD נהיית מסובכת יותר במצב זה.

בפרויקט,

הרעשים הוספו לסיגנלים הנקיים בעוצמות שנקבעו לפי פרמטר הSNR שהוגדר.

ניתוח ספקטרלי של אות דיבור

אות דיבור במימד התדר

קול הינו סוג של גל ומעצם כך הינו בעל מאפיינים של גל ומורכב מתדרים שונים. בניתוח ועיבוד האות בתדר נוכל להתבסס על מאפיינים תדריים שונים של האות שלא נראה בניתוח בזמן.

ניתוח תדרים מאפשר לבצע פעולות עיבוד אותות שונות ,כמו הסרת רעשים ,שינוי גובה הקול ,או דחיסת האות.

המעבר ממישור הזמן למישור התדר -

מבוצע ע"י התמרת פורייה. אך כיוון שאותות דיבור הם דינמיים ,כלומר התדרים שלהם משתנים כפונקציה של הזמן, התמרת פורייה הרגילה אינה מתאימה לניתוח אותות דיבור כיוון שהיא מניחה שהאות הוא סטטי.

לכן, המעבר לתדר יבוצע ע"י התמרת פורייה לזמן קצר (STFT) - חלוקת האות לקטעים קצרים (פריימים) וחישוב התמרת פורייה לכל פריים בנפרד. כך ,אנו מקבלים מידע על התדרים בכל נקודת זמן באות.

התמרת פורייה לזמן קצר (STFT)

משמשת לקביעת התדר הסינוסואידאלי ותכולת הפאזה של קטעים מקומיים של האות כפי שהוא משתנה לאורך זמן.

התמרת STFT היא כלי חיוני בניתוח אותות דיבור מכיוון שהיא מאפשרת לנו לנתח את האות הן בזמן והן בתדר. ולהבין כיצד התדרים משתנים לאורך זמן .זאת בשונה מהתמרת פורייה הרגילה ,אשר מספקת מידע רק על התכולה התדרית של האות כולו ,ללא התייחסות לשינויים בזמן.

אותות דיבור הם דינמיים כך שהתכונות שלהם משתנות עם הזמן. ביניהן, גובה הקול, מבטא צלילים שונה וכדו'.

חישוב הSTFT מבוצע על ידי חילוק האות הנתון למקטעים קצרים יותר באורך שווה ולאחר מכן חישוב התמרת פורייה בנפרד על כל מקטע.

בצורה מתמטית -

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-i\omega n}$$

כאשר –

$\omega(t)$ – פונקציית חלון

$x(t)$ – האות שיש להמיר

בחירת פרמטרים עבור ההתמרה:

בחירת הפרמטרים הנכונים עבור התמרת פורייה לזמן קצר (STFT) היא שלב חשוב בניתוח ספקטרלי.

הפרמטרים העיקריים הם:

- אורך החלון (window length) : קובע את מספר המדידות בכל פריים (חלון) .חלון ארוך יותר ייתן רזולוציה טובה יותר בתדר אך רזולוציה טובה פחות בזמן ,ולהיפך.
- הזזה של החלון (hop length) : קובע את כמות החפיפה בין פריימים עוקבים. הזזה קטנה יותר תגדיל את כמות החפיפה ותשפר את הרזולוציה בזמן, אך תגדיל גם את זמן החישוב.
- פונקציית החלון : קובעת את צורת החלון שיושם על האות לפני חישוב הFFT. פונקציות חלון שונות משמשות להפחתת דליפה ספקטרלית.

הפרמטרים של ההתמרה מגדירים את מספר התדרים ומספר הפריימים של האות המותמר:

- מספר נקודות התדר – F נקבע לפי הנוסחא –

$$F = \frac{n_{fft}}{2} + 1$$

- מספר הפריימים T מוגדר בפונקציית torch.stft בפייתון לפי הנוסחא –

$$T = 1 + \frac{L}{hop\ length}$$

- כמו כן, בפייתון ניתן להגדיר כי ההתמרה תחזיר שני ערוצים שונים של החלק הממשי והמדומה בנפרד.

ספקטוגרמה

ייצוג חזותי של ספקטרום התדרים של האות והשתנותו עם הזמן. מדובר בכלי מרכזי בניתוח אותות דיבור ואותות כלליים, מכיוון שהוא מאפשר לראות כיצד התדרים המרכיבים את האות משתנים עם הזמן.

הספקטרוגרמה מתקבלת על ידי חישוב התמרת פורייה לזמן קצר (STFT) עבור קטעים עוקבים של האות. עבור כל קטע, מחושב ספקטרום התדרים באמצעות התמרת פורייה, והמידע המתקבל נפרש על פני ציר הזמן כדי להציג את ההשתנות הדינמית של התדרים.

תבנית שכיחה לספקטוגרמה הינה גרף תלת מימדי שבו שני ממדים גאומטריים המייצגים זמן ותדר, ומימד שלישי שמציין את המשרעת של תדר מסוים בנקודת זמן מסוימת. המשרעת מיוצגת על ידי הבהירות או הצבע של הנקודה.

נהוג להפעיל פונקציית לוג על התדרים המתקבלים:

זוהי טכניקה נפוצה בעיבוד אותות דיבור כאשר הסיבה העיקרית לכך היא הטווח הדינמי הרחב של עוצמות האות. המשרעת של האות בתחום התדר עשויה להשתנות בסדרי גודל רבים. אם מציגים את הנתונים בקנה מידה לינארי, יהיה קשה להבחין בתדרים חלשים יותר ביחס לתדרים חזקים מאוד. כדי לשפר את הנראות של תדרים חלשים ובינוניים, משתמשים בקנה מידה לוגריתמי.

השימוש בלוגריתם עוזר "לדחוס" את המשרעות ולטפל בטווח הדינמי הרחב של האותות הקוליים, כך שגם תדרים חלשים יחסית, כמו רעשי רקע או תהודה עדינה, נראים בצורה ברורה יותר על הגרף. זה חשוב במיוחד באותות דיבור שבהם יש תדרים חזקים יותר ותדרים חלשים יותר שמתרחשים בו-זמנית. קנה מידה לוגריתמי מיישר את ההבדלים האלו ומספק נראות טובה יותר לניתוח ספקטרלי.

מאפייני למידה עמוקה

רשת נוירונים

רשתות נוירונים, הידועות גם בשם רשתות עצביות מלאכותיות ,(ANN) הן מודל מתמטי חישובי המבוסס על המבנה והתפקוד של המוח האנושי. הן מורכבות ממספר רב של יחידות מידע קטנות הנקראות נוירונים מלאכותיים, אשר מקושרים ביניהם בקשרים חזקים. בדומה לנוירונים במוח, כל נוירון מלאכותי מקבל קלט מנוירונים אחרים, מעבד אותו ומפיק פלט. קשרים אלו בין הנוירונים מוגדרים על ידי משקלות, המשפיעים על חוזק האות המועבר

רשתות נוירונים מאומנות על ידי חשיפה לכמות גדולה של נתונים, תוך כדי התאמת משקלות הקשרים בין הנוירונים. תהליך אימון זה מאפשר לרשת ללמוד את הקשרים הסטטיסטיים בין נתוני הקלט לנתוני הפלט, וכך לפתח יכולת לחזות תוצאות חדשות שלא נחשפה אליהן בעבר.

למידה עמוקה

למידה עמוקה, הידועה גם בשם Deep Learning או Deep Structured Learning, היא תת-תחום של למידת מכונה המשתמשת ברשתות נוירונים מלאכותיות בעלות מספר רב של שכבות נסתרות. שכבות אלו מאפשרות לרשת ללמוד תכונות מורכבות מהנתונים, ולשפר משמעותית את הביצועים במגוון רחב של משימות.

עקרונות פעולה:

- **רשתות נוירונים עמוקות :**למידה עמוקה משתמשת ברשתות נוירונים בעלות מספר רב של שכבות נסתרות. כל שכבה מורכבת ממספר רב של נוירונים מלאכותיים, המקושרים זה לזה בקשרים חזקים.
- **פונקציית הפעלה :**לכל נוירון יש פונקציית הפעלה, שמעבדת את הקלטים מהנוירונים הקודמים ומפיקה פלט.
- **פונקציית אובדן :**פונקציית אובדן מודדת את ההפרש בין הפלט הרצוי לפלט בפועל של הרשת.
- **אלגוריתם אופטימיזציה :**אלגוריתם אופטימיזציה משמש לכוונון משקלות הקשרים בין הנוירונים, במטרה למזער את פונקציית האובדן.
- **אימון :**תהליך האימון כרוך בחשיפה של הרשת לכמות גדולה של נתונים, תוך כדי התאמת משקלות הקשרים על מנת לשפר את הביצועים.

סוגי הDATA השונים

בלמידה עמוקה ולמידת מכונה, הדאטה משחק תפקיד מרכזי וחיוני. ניתן לראות בו את "החומר הגלם" ממנו לומדים האלגוריתמים ומשפרים את ביצועיהם.

סוגי דאטה עיקריים:

- **Training Data**: נתונים אלו משמשים לאימון רשת הנוירונים או מודל למידת המכונה. הם צריכים להיות איכותיים ומייצגים את המשימה שאותה רוצים שהמודל יבצע. כמות נתוני האימון משפיעה רבות על איכות המודל ועל יכולת ההכללה שלו.
- **Test Data :** נתונים אלו משמשים להערכת הביצועים של המודל לאחר האימון. הם לא צריכים להיות חשופים למודל במהלך האימון, על מנת לאפשר הערכה אובייקטיבית. נתוני הבדיקה מאפשרים לוודא שהמודל לומד את המאפיינים הכלליים של הנתונים ולא רק את נתוני האימון הספציפיים.
- **Validation Data :** נתונים אלו משמשים לכיוון היפרא-פרמטרים של המודל במהלך האימון. הם לא צריכים להיות חשופים למודל במהלך האימון, אך הם דומים לנתוני הבדיקה מבחינת המאפיינים שלהם. נתוני הוולידציה מאפשרים למודל לבחור את ההיפרא-פרמטרים המיטביים עבור המשימה הספציפית.

פיצ'רים

פיצ'רים (Features) הם ייצוגים מוחשיים של נתונים, המשמשים כקלט לאלגוריתמי למידת מכונה. הם מאפשרים למכונה להבין את המאפיינים החשובים בנתונים וללמוד לבצע משימות שונות, כגון זיהוי עצמים, סיווג תמונות, תרגום שפות ועוד.

דוגמאות לפיצ'רים:

- **בזיהוי תמונות :**פיצ'רים יכולים להיות פיקסלים, קצוות או צורות.
- **בזיהוי דיבור :**פיצ'רים יכולים להיות תכונות אקוסטיות של האות.
- **בעיבוד שפה טבעית :**פיצ'רים יכולים להיות מילים, משפטים או חלקי דיבור.

היפרא-פרמטרים

היפרא-פרמטרים (Hyperparameters)הם הגדרות המגדרות את אופן פעולת אלגוריתם למידת מכונה. ניתן לראות בהם את "בורגי הכוונון" של האלגוריתם, המאפשרים להתאים את הביצועים שלו למשימה הספציפית שאותה רוצים לבצע.

דוגמאות להיפרא-פרמטרים:

מספר שכבות ברשת הנוירונים, מספר נוירונים בכל שכבה, קצב הלמידה ,פונקציית הפעלה, גודל חלון ההקשר, מספר עידנים, גודל אצווה

חשיבות בחירת היפרא-פרמטרים נכונה:

בחירת היפרא-פרמטרים נכונה היא קריטית להשגת ביצועים מיטביים של אלגוריתם למידת מכונה. בחירה שגויה עלולה להוביל לתוצאות גרועות, כגון תת-התאמה או התאמה יתר.

תת-התאמה (Underfitting)מתרחשת כאשר האלגוריתם לא לומד את המאפיינים החשובים בנתונים, ולכן הוא לא מצליח לבצע את המשימה בצורה נכונה.

התאמה יתר (Overfitting)מתרחשת כאשר האלגוריתם לומד את נתוני האימון בצורה מדויקת מדי, כולל את הרעש והשגיאות, ולכן הוא לא מצליח להכליל ולשפר את הביצועים שלו על נתונים חדשים.

שלב האימון

האימון (Training)הוא תהליך מרכזי בלמידת עמוקה ולמידת מכונה, שבו אלגוריתם לומד לבצע משימה ספציפית על ידי התאמה לנתונים.

השלבים העיקריים בתהליך האימון:

1. **הגדרת המודל :**בשלב זה, מגדירים את ארכיטקטורת המודל, הכוללת את מספר השכבות, מספר הנוירונים בכל שכבה, סוגי הפונקציות, ועוד .
2. **איסוף נתונים :**בשלב זה, אוספים נתונים איכותיים ומייצגים עבור המשימה הספציפית שאותה רוצים שהמודל יבצע. נתונים אלו מחולקים לסט אימון, סט ולידציה וסט בדיקה .
3. **הגדרת פונקציית ההפסד :**פונקציה זו מודדת את ההפרש בין הפלט החזוי של המודל לבין הפלט הרצוי .
4. **בחירת אלגוריתם אופטימיזציה :**אלגוריתם זה משמש לכוונון משקלות הקשרים בין הנוירונים במודל, במטרה למזער את פונקציית ההפסד .
5. **הזנת נתוני האימון למודל** :בשלב זה, נתוני האימון מוזנים למודל, והמודל לומד להתאים את משקלותיו על מנת להפיק מהנתונים את התובנות הרצויות .
6. **הערכה :**לאחר מספר עידנים של אימון, מעריכים את ביצועי המודל על סט ולידציה ועל סט בדיקה .
7. **כיוון היפרא-פרמטרים :**במידת הצורך, ניתן לכוון את היפרא-פרמטרים של המודל, כגון קצב הלמידה או מספר העידנים, על מנת לשפר את הביצועים .
8. **שמירת המודל** :לאחר שהמודל משיג ביצועים מספקים, ניתן לשמור אותו לשימוש עתידי.

טכניקות למניעת התאמה יתר:

- **הכללה מוקדמת :(Regularization)** טכניקות אלו מוסיפות עונש על מורכבות המודל, על מנת למנוע ממנו ללמוד את הרעש בנתוני האימון.
- **נטישת נתונים :(Dropout)** טכניקה זו משמיטה באופן אקראי נוירונים מהרשת במהלך האימון, על מנת למנוע מהמודל להסתמך על נוירונים ספציפיים יתר על המידה.
- **אימון מוקדם :(Pretraining)** טכניקה זו מאמנת את המודל תחילה על סט נתונים גדול ולאחר מכן מעדכנת את משקלותיו עבור המשימה הספציפית.

סט ולידציה:

סט הוולידציה משמש לכיוון היפרא-פרמטרים של המודל במהלך האימון. כיוון היפרא-פרמטרים כולל בחירת ערכים מיטביים עבור גורמים כמו קצב הלמידה, מספר שכבות הרשת וגודל אצווה. סט הוולידציה נפרד מנתוני האימון, על מנת למנוע התאמה יתר של המודל לנתונים ספציפיים. על ידי ניטור ביצועי המודל על סט הוולידציה, ניתן לבחור את הערכים המיטביים עבור היפרא-פרמטרים ולשפר את הביצועים הכלליים של המודל.

סט טסט:

סט הבדיקה משמש להערכה אובייקטיבית של הביצועים של המודל לאחר שהאימון הסתיים. סט הבדיקה נפרד מנתוני האימון ומנתוני הוולידציה, על מנת להבטיח שהמודל לא נחשף לנתונים אלו לפני ההערכה.
מדדי דיוק שונים, כגון דיוק (Accuracy), זיהוי (Recall) ודיוק (Precision), משמשים להערכת ביצועי המודל על סט הבדיקה. תוצאות סט הבדיקה נותנות אינדיקציה אובייקטיבית ליכולתו של המודל לבצע את המשימה המיועדת לו.

פונקציות LOSS

הן חלק חיוני מרשתות נוירונים עמוקות. הן משמשות למדידת הטעות שמודל עושה בהשוואה לפלט הרצוי.

מטרת פונקציית LOSS :

- **להעריך את הביצועים של המודל :**פונקציית LOSS מספקת ערך מספרי המתאר עד כמה הפלט של המודל רחוק מהפלט האמיתי.
- **לכוון את תהליך האימון :**במהלך אימון רשת נוירונים, פונקציית LOSS משמשת לחישוב הגרדיאנטים של הפרמטרים של המודל. גרדיאנטים אלו משמשים לעדכון הפרמטרים, תוך ניסיון למזער את ערך פונקציית.LOSS

סוגים עיקריים של פונקציות LOSS:

- **פונקציות LOSS ריבועיות:**
  - **Mean Squared Error (MSE):** פונקציה זו מחשבת את הממוצע הריבועי של ההפרשים בין הפלט החזוי של המודל לבין הפלט האמיתי. היא משמשת למשימות רגרסיה, שם המטרה היא לחזות ערך מספרי (כמו מחיר מניה).
  - **Mean Absolute Error (MAE):** פונקציה זו מחשבת את הממוצע המוחלט של ההפרשים בין הפלט החזוי של המודל לבין הפלט האמיתי. היא משמשת למשימות רגרסיה, שם חשוב יותר להתמקד בשגיאות גדולות יותר.
- **פונקציות LOSS לוגיסטיות:**

- **Binary Cross-Entropy (BCE):** פונקציה זו משמשת למשימות סיווג בינארי, שם המטרה היא לסווג נתונים לשתי קטגוריות. היא מחשבת את הפסד עבור כל דוגמה בנפרד.
- **Categorical Cross-Entropy**: פונקציה זו משמשת למשימות סיווג רב-קטגוריות, שם המטרה היא לסווג נתונים ליותר משתי קטגוריות. היא מחשבת את הפסד עבור כל דוגמה בנפרד.

בפרויקט,

פונקציית הLOSS בה המודל עשה שימוש הינה BCEWithLogitsLoss, אשר מחשבת את הפלט של הפונקציה BCE עם Sigmoid.

אופטומיזטורים

אלגוריתמים המיועדים לשפר את הביצועים של מודלים בלמידת מכונה על ידי עדכון המשקלים שלהם במהלך האימון. הם אחראים על מציאת הפתרון הטוב ביותר עבור פונקציית הפסד (loss function) על ידי חישוב גרדיאנטים של הפונקציה והזזת המשקלים בכיוון ההפוך ל gradient,-במטרה לצמצם את הפסד המודל.

אופטומיזטור AdamW:

מדובר בשיפור על האופטימיזטור Adam הקלאסי, המשלב את היתרונות של SGD (Stochastic Gradient Descent) יחד עם תהליכים של ממוצעים נעים ותחזוקת גרדיאנטים.

לאופטומיזטור זה מספר יתרונות:

- שילוב של עקומות למידה AdamW :מאזן בין תהליך העדכון המהיר של Adam לבין הרגולציה של SGD, המאפשר יכולת הכללה טובה יותר של המודל.
- תיקון של בעיית הרגולציה AdamW :מבצע עדכונים של משקלים בשלב נפרד מהעדכון של הקצבות (weight decay), מה שמפחית את ההשפעה של הרגולציה על קצב הלמידה.
- אופטימיזציה של רמות שונות AdamW :מתעדכן באמצעות חישוב ממוצעים של גרדיאנטים מיידיים לאורך זמן, מה שמאפשר גמישות ואפקטיביות גבוהה יותר.

בפרויקט,

בוצע שימוש באופטומיזטור AdamW.

שכבות במודלי למידה עמוקה

פונקציות אקטיבציה

פונקציות אקטיבציה הן מרכיב חיוני ברשתות נוירונים עמוקות. הן קובעות את הפלט של נוירון כפונקציה של קלטיו. פונקציות אלו ממלאות תפקידים חשובים רבים, ביניהם:

- **הכנסת אי-ליניאריות למודל :**פונקציות אקטיבציה ליניאריות יובילו למודל לינארי פשוט, שאינו מסוגל ללמוד קשרים מורכבים בין נתונים. פונקציות אקטיבציה לא ליניאריות מאפשרות למודל ללמוד קשרים אלו, חיוניים לפתרון משימות רבות כמו זיהוי תמונות, עיבוד שפה טבעית ועוד.
- **שליטה בזרימת המידע :**פונקציות אקטיבציה יכולות לשלוט באיזה מידע עובר דרך הרשת. הן יכולות "לדכא" אקטיבציות חזקות מדי או "להגביר" אקטיבציות חלשות, תוך התמקדות במידע הרלוונטי ביותר למשימה.

פונקציות אקטיבציה נפוצות:

- **Sigmoid :** פונקציה חלקה הממפה ערכים בין 0 ל-1. פונקציה זו שימשה רבות בעבר, אך כיום פחות פופולרית בשל נטייתה להיעלם מגרדיאנטים, מקשה על אימון רשתות עמוקות.
- **Softmax:** פונקציות אלו משמשות בעיקר בשכבות פלט של רשתות נוירונים, ומשמשות לנורמליזציה של הפלטים כך שיסתכמו ב-1, ומאפשרות פרשנות כפרובאביליות.
- **ReLU (Rectified Linear Unit)** פונקציה פשוטה המוציאה את הקלט אם הוא חיובי, ו-0 אחרת. פונקציה זו פופולרית מאוד כיום בשל יעילותה החישובית וקלות אימונה.
- **PReLU:** היא פונקציית אקטיבציה שפותחה כדי להתגבר על בעיית "מוות נוירונים" הנפוצה בפונקציית ReLU. בפונקציה זו, משקלים חיוביים (α) נקבעים באופן פרמטרי, כך שניתן ללמוד אותם במהלך האימון.

<u>אלגוריתם הפונקציה:</u>

  - עבור ערכים חיוביים PReLU, מתנהגת כמו ReLU - $f(x) = x, \forall x > 0$
  - עבור ערכים שליליים, היא נותנת משקל חיובי קטן :- $f(x) = ax, \forall x > 0$, כאשר α הוא פרמטר נלמד.

<u>יתרונות הפונקציה:</u>

  - מאפשרת גמישות גבוהה יותר ויכולת להתמודד עם ערכים שליליים, דבר שמפחית את הסיכוי ל"מוות נוירונים ."
  - יכולת למידת משקלים שונים לערכים חיוביים ושליליים.

- **Max Pooling**: שכבת עיבוד שמשמשת להפחתת ממדי הקלט (downsampling) תוך שמירה על התכונות החשובות של הנתונים. הפונקציה מחלקת את הקלט לאזורים קטנים (אזורי פיצול) ומחזירה את הערך המקסימלי מכל אזור.

<u>יתרונות הפונקציה:</u>

- השכבה מפחיתה את מספר הפרמטרים ואת חישוב הזמן, דבר המהווה יתרון משמעותי ברשתות נוירונים עמוקות.
- שימוש בפונקציה זו עוזר במניעת overfitting על ידי הפחתת המידע שאותו צריך לרכוש המודל.

בפרויקט,

כחלק משכבת CNN embedder השתמשנו בפונקציות האקטיבציה PReLU ו Max Pooling.

CNN Embedder

מטרת השכבה של הCNN היא למצוא את הקשרים בין התדרים הסמוכים.

שכבה זו היא סוג של רשת נוירונים המשתמשת בפעולת הקונבלוציה במקום בכפל מטריצות כללי, מה שמאפשר לה לזהות דפוסים מקומיים ולבצע עיבוד יעיל של מידע בעל מבנה מרחבי או זמני, כמו תמונות, אותות קול, או נתונים אחרים שיש להם קורלציה בין תכונות סמוכות.

בשימוש בעיבוד אותות קול, ה CNN-מסוגל למצוא קשרים בין תדרים סמוכים או שינויים בתדר לאורך הזמן, דבר שעוזר בזיהוי דפוסים קוליים כמו נוכחות קול (VAD), זיהוי דובר או זיהוי דיבור.

תהליך הקונבולוציה:

השכבה מבוצעת על ידי הפעלת מסנן (filter) על הקלט. המסנן הוא מטריצה קטנה של משקלים הנלמדים במהלך האימון, והוא עובר על פני הקלט. בכל שלב, המסנן מבצע כפל נקודתי בין ערכי הקלט לבין המשקלים של המסנן ולאחר מכן מחבר את התוצאה לערך אחד. התהליך הזה יוצר מפה חדשה שמדגישה דפוסים מסוימים באות.

יתרונות שכבה זו:

- השימוש בקונבולוציה במקום בכפל מטריצות מלא חוסך משאבי חישוב בצורה משמעותית, כיוון שהוא מצמצם את כמות הפרמטרים שיש ללמוד.
- שכבות קונבולוציה הן מקומיות, כלומר מתמקדות בדפוסים במקטעים קטנים של הקלט בכל פעם, ומצליחות לזהות תכונות חשובות גם כאשר המיקום המדויק של התכונה בקלט משתנה.(invariance)

Self-Attention

Self-Attention הוא מנגנון המשמש למידת מכונה, במיוחד בעיבוד שפה טבעית (NLP) ומשימות ראייה ממוחשבת, כדי ללכוד תלות ויחסים בתוך רצפי קלט. הוא מאפשר למודל לזהות ולשקול את החשיבות של חלקים שונים ברצף הקלט על ידי טיפול בעצמו. האלגוריתם לוכד יחסים בין אלמנטים מרוחקים ברצף, ומאפשר להבין דפוסים ותלות מורכבים.

Self-Attention פועל בשלושה שלבים עיקריים:

1. **חישוב Query, Key ו:Value-**

הקלט של כל נוירון עובר דרך שלוש טרנספורמציות ליניאריות נפרדות, היוצרות שלושה וקטורים חדשים :

- **Query (שאילתה) :**וקטור זה מייצג את "השאלה" שנוירון שואל לגבי שאר הנוירונים.
- **Key(מפתח):** וקטור זה מייצג את המידע של כל נוירון, שיעזור לקבוע כמה תשומת לב הוא יקבל.
- **Value(ערך):** וקטור זה מכיל את המידע בפועל של כל נוירון.

2. **חישוב ציון התאמה:**

לאחר מכן, עבור כל זוג נוירונים, מחושבים ציוני התאמה בין וקטור ה Query-של נוירון אחד לבין וקטורי ה-Keyשל נוירונים אחרים. ציונים אלו מעריכים כמה רלוונטי המידע של נוירון מסוים לנושא שאותו בודק הנוירון הראשון.

3. **חישוב פלט משוקלל:**

לבסוף, עבור כל נוירון, מחושב פלט משוקלל על ידי הכפלת ציוני ההתאמה בכל וקטורי ה Value-של הנוירונים האחרים, וסכימה של התוצרים. פלט זה מייצג למעשה סיכום של המידע הרלוונטי ביותר לנוירון מתוך כל הקלט שלו, תוך התחשבות בחשיבות של כל חלק.

תהליך זה מאפשר למודל להתמקד במידע רלוונטי וללכוד תלות ארוכת טווח. על ידי התייחסות לחלקים שונים של רצף הקלט, Self-Attention עוזר למודל להבין את ההקשר ולהקצות משקלים מתאימים לכל אלמנט בהתבסס על הרלוונטיות שלו.

Fully Connected

שכבה זו מחברת כל נוירון של השכבה הקודמת לנוירון בשכבה הנוכחית, ומכאן מגיעה שמה. שכבה זו יוצרת חיזוי לצורך קבלת החלטה סופית על סמך התכונות שנלמדו בשכבות הקודמות לה.

השכבה לוקחת את הייצוג הנלמד מהשכבה הקודמת לה, ומשלבת אותם ליצירת חיזוי.

השכבה מאפשרת לקשר בין תכונות שונות, ובכך מאפשרת למודל ללמוד את ההקשרים המורכבים בין המידע שהוזן לו.

כלים ופתרונות לניהול ופיתוח בלמידה עמוקה

PyTorch

ספריית קוד פתוח עבור למידת מכונה ולמידה עמוקה בפיתון. היא מציעה גמישות רבה ,ביצועים גבוהים ותמיכה קהילתית חזקה.

יתרונות השימוש בספרייה:

- טנזורים דינמיים: מאפשרים לבנות ולשנות את מבנה המודל בזמן ריצה ,דבר אשר יכול להיות שימושי מאוד בפיתוח מודלים מורכבים.
- GPU: תמיכה מלאה בכרטיסי גרפיקה (GPU), מהירות חישוב משמעותית עבור מודלים גדולים ומסובכים.
- קהילת משתמשים גדולה וסביבת פיתוח פעילה מספקות תמיכה ,משאבים ומגוון רחב של כלים.
- הספרייה ניתנת לשימוש באינטגרציה עם כלים אחרים כמו NumPy, SciPy ו ,Scikit-learn- דבר המקל על בניית צינורות למידה מלאים.

Hydra

מסגרת קונפיגורציה שנועדה לפשט את ניהול ההגדרות בפרויקטים מורכבים. מאפשרת להגדיר את כל הפרמטרים של המודל במקום אחד, ומספקת דרך נוחה לשנות אותם ולנסות עם הגדרות שונות.

מאפייני הספרייה:

1. קונפיגורציה היררכית: מאפשרת ליצור קונפיגורציות מורכבות בצורה מודולרית וברורה.
2. התאמה אישית: ניתן להתאים את הקונפיגורציה בקלות באמצעות קבצי תצורה או שורת הפקודה.
3. ניסויים מרובים: מקלה על ביצוע ניסויים מרובים עם פרמטרים שונים, דבר המאפשר למצוא את ההגדרות הטובות ביותר עבור המודל.
4. אינטגרציה עם PyTorch: Hydra משולבת בצורה חלקה עם PyTorch, דבר המאפשר לנהל את כל ההגדרות של המודל במקום אחד.

מדדי ביצועים

מדדי ביצוע הם קריטיים להערכה של מודלים בלמידה עמוקה, שכן הם מאפשרים להבין עד כמה המודל מצליח במשימות שהוגדרו לו. במדדים אלו משתמשים לאחר שלב האימון והטסט כדי לנתח את הביצועים של המודל על נתוני הבדיקה.

מטריצת בלבול - Confusion Matrix

מטריצה זו מציגה את התפלגות הניבויים של המודל על פני הקטגוריות השונות.

מרכיבים:

- True Positives (TP) - מספר הניבויים החיוביים הנכונים.
- True Negatives (TN) - מספר הניבויים השליליים הנכונים.
- False Positives (FP) - מספר הניבויים החיוביים השגויים.
- False Negatives (FN) - מספר הניבויים השליליים השגויים.

מבנה המטריצה:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

נכונות - Accuracy

אחוז הניבויים הנכונים של המודל מתוך כלל הניבויים.

מוגדר לפי הנוסחה –

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

דיוק הוא מדד פשוט להבנה, אך הוא עלול להיות מטעה כאשר יש חוסר איזון בין הקטגוריות (לדוגמה, אם רוב הנתונים שייכים לקטגוריה אחת).

דיוק - Precision

סודיות מודדת את אחוז הניבויים החיוביים הנכונים מתוך כלל הניבויים החיוביים.

מוגדרת לפי הנוסחה –

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

סודיות חשובה במיוחד כאשר יש צורך להימנע מניבויים חיוביים שגויים (למשל, בזיהוי מחלות).

זכירה - Recall

זכירה מודדת את אחוז הניבויים החיוביים הנכונים מתוך כלל הדוגמאות החיוביות.

מוגדרת לפי הנוסחה –

$$Recall = \frac{True\ positives}{True\ positives + False\ Negatives}$$

זכירה היא מדד חשוב כאשר יש צורך לא לפספס דוגמאות חיוביות, כמו בזיהוי פשעים או מקרים של מחלה.

F1 Score

מדד המאזן בין סודיות לזכירה, והוא נועד לספק תמונה כוללת על הביצועים של המודל.

מוגדר לפי הנוסחה –

$$Recall = \frac{Precision \cdot Recall}{Precision + Recall}$$

הוא מדד שימושי במקרים שבהם יש חוסר איזון בין הקטגוריות, והוא מסייע להבין את האיזון בין שני המדדים הקודמים.

AUC-ROC

מדד המעריך את הביצועים של מודל סיווג על פני מגוון ספים (thresholds).

(Receiver Operating Characteristic) ROC Curve

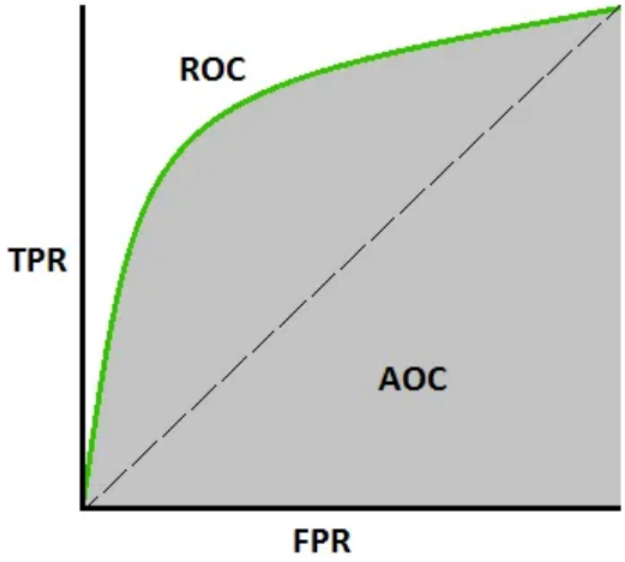
עקומת ROC מציגה את יחס הזכירה (True Positive Rate - TPR) מול שיעור השגיאות החיוביות (False Positive Rate - FPR) עבור כל סף אפשרי.

- **TPR** (Recall) - היחס של ניבויים חיוביים נכונים מתוך כלל הדוגמאות החיוביות.
- **FPR** - היחס של ניבויים חיוביים שגויים מתוך כלל הדוגמאות השליליות.

AUC (Area Under the Curve)

השטח שמתחת לעקומת Area Under the Curve - ROC, הוא מדד המייצג את יכולת המודל להבחין בין קטגוריות AUC . יכול לנוע בין 0 ל-1, כאשר:

- AUC של 1.0 מציין מודל מושלם.
- AUC של 0.5 מציין מודל שמנבא באקראי (כמו הטלת מטבע).
- AUC מתחת ל-0.5 מעיד על מודל גרוע יותר מאקראי.



דרכי פתרון

זיהוי מדויק של נוכחות או היעדר דיבור בהקלטה חד-ערוצית רועשת הוא אתגר מתמשך בעולם עיבוד האותות. היכולת להבחין בין אותות דיבור לרעש רקע מהווה נדבך קריטי במגוון רחב של יישומים כגון, מערכות תקשורת, זיהוי דיבור אוטומטי, וביטול הד.

לאורך השנים פותחו מגוון גישות לפתרון בעיה זו, כאשר כל אחת מהן מציעה יתרונות וחסרונות משלה.

גישות קלאסיות לעיבוד אותות

גישות קלאסיות לעיבוד אותות התמקדו בעיקר באנליזה של מאפיינים אקוסטיים של אות השמע. שיטות אלו מבוססות על חישוב תכונות כגון קצב חציית אפס, זיהוי גובה ואנרגיה, ולאחר מכן קביעת סף מתאים להפרדה בין אותות דיבור לרעש. למרות פשטותן היחסית, גישות אלו סובלות ממספר חסרונות משמעותיים. ראשית ,הסף המוגדר מראש עלול להיות רגיש מאוד לתנאי הסביבה המשתנים מה שמוביל לביצועים ירודים בסביבות רועשות או בתנאי הד. שנית ,שיטות אלו מתקשות להתמודד עם אותות דיבור בעלי מאפיינים משתנים, כגון דיבור ברמות אנרגיה נמוכות או דיבור עם מבטאים שונים.

מודלים סטטיסטיים

כדי להתמודד עם מגבלות אלו ,פותחו מודלים סטטיסטיים המנסים לתאר את ההתפלגות הסטטיסטית של אותות הדיבור והרעש .גישות אלו מבוססות על הנחה כי אותות הדיבור והרעש מקורם בהתפלגויות סטטיסטיות שונות ,ומאפשרות שימוש במבחני יחס הסבירות (Likelihood Ratio Test) כדי להחליט על ההשערה הסבירה ביותר .למרות שיפור בביצועים ביחס לגישות הקלאסיות ,מודלים סטטיסטיים עדיין רגישים מאוד לתנאי רעש משתנים ולשינויים בתכונות האות.

רשתות עצביות עמוקות - DNN

בשנים האחרונות, הוצגו פתרונות בשימוש של רשתות עצביות עמוקות שיטות אשר הראו תוצאות מבטיחות בהשוואה לאלגוריתמים הקלאסיים.

<u>CNNs</u>:

הוכחו כיעילות בחילוץ תכונות רלוונטיות מספקטרוגרמות של אותות שמע. עם זאת, הן דורשות שימוש בהשטחה ושימוש בשכבות FC אשר מוסיפות מספר רב של פרמטרים לאימון וכן מגבילות את אורך הקלט, מה שמגביל את האפשרות לעבד רצפים ארוכים של אותות.

<u>RNNs</u> :

רשתות עצביות חוזרות ,כגון ,LSTM שימשו ללכידת תלות ארוכת טווח באותות השמע.

<u>ACAM</u>:

מודל זה משלב LSTM וSA כדי להתמקד בחלקים הרלוונטיים של האות הקלט , אך מגבלותיו כוללות אופי סדרתי שמונע מהמודל לקשור בין חלקים שונים של הקלט בצורה יעילה.

SA:

מתמקד במידול הקשרים בין מסגרות הקלט, אך הוא מוגבל לשימוש על הקשר קצר בלבד ולא הצליח לנצל במלואו את המידע הספקטרלי של האות.

הגישה הנבחרת - שילוב של CNN ו-SA

בפרויקט, ביצענו VAD המורכב משילוב של שכבות CNN ושכבת SA.

הCNN משמש לחילוץ ושימור תכונות מכל פריים בנפרד לפי המידע הספקטרלי וה-SA מעבד את ההקשר הרחב של הפריימים השונים ובכך קובע את הרלוונטיות של כל פריים.

לנ"ל יתרון כיוון שהוא משפר את הדיוק וכן את היעילות החישובית:

השילוב של CNN לניתוח מקומי של מסגרות תדר ושל SA לניתוח גלובלי של הקשר בין מסגרות, מוביל לשיפור בדיוק המודל בהשוואה לשימוש בכל אחד מהמרכיבים בנפרד. בנוסף, המודל משפר את היעילות החישובית בכך שהוא מעבד את האות כולו בפעולה אחת, ולא דורש חלוקה לקטעים קטנים.

מימוש המודל

מימוש המודל מחולק ל2 חלקים: הראשון, הינו ייצור הדאטה עבור המודל והשני הינו מימוש המודל והרצתו.

ייצור הדאטה

ייצור RIR

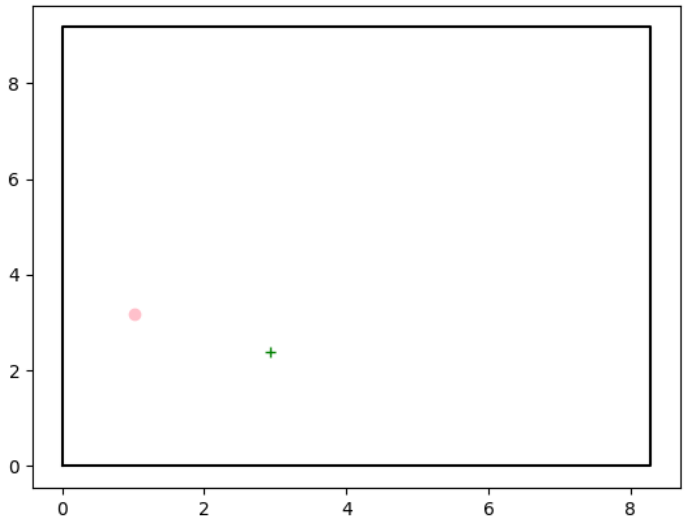
בשלב הראשון יצרנו תגובות הלם של חדרים שונים, כאמור – RIRים.

עבור כל RIR:

1. הגרלנו את גודל החדר, את מיקום המיקרופון בתוך החדר, את המרחק של הדובר מהמיקרופון ואת זמן ההדהוד (RT60).
2. עבור תנאים אלו חישבנו את כל המיקומים האפשריים של הדובר בחדר ומתוכם הגרלנו את מיקומו של הדובר.
3. עבור כל חדר כזה חישבנו את הRIR ע"י פונקציית rir_generator.generate

<u>להלן דוגמא לחדר שמידלנו עבור ייצרנו RIR:</u>

המיקרופון צבוע באדום והדובר צבוע בירוק.



<u>פרמטרים:</u>

$$Fs = 16k$$

$$Room\ size = W \sim U[8,10] \times L \sim U[8,10] \times H \sim U[2.5,5]$$

$$Microphon\ position = \ set\ to\ be\ at\ least\ 0.5\ away\ from\ the\ wall$$

$$Speaker\ position = \ positioned\ in\ a\ radios\ of\ \sim U[0.3,4]\ from\ \ the\ microphon$$

ייצור הסיגנל

בשלב השני בנינו את הקוד ליצירת הסיגנל הרועש, כאשר ייצור הדאטה בפועל התרחש תוך כדי אימון המודל.

הסיגנל אותו רצינו לבחון הינו אות מהודהד ורועש למטרות VAD.

- לכן בנינו את הסיגנלים ע"י לקיחת סיגנל נקי, העברתו בRIR והוספת רעש לאות.
- במקביל הגדרנו את הlabelים של הסיגנלים – התוויות הרצויות של הסיגנלים המייצגות נוכחות/היעדר דיבור עבור כל פריים. זאת ע"ב הסיגנל המהודהד, ללא תוספת הרעש.

את הסיגנל ייצרנו באופן הבא:

תחילה הגרלנו סיגנל נקי, RIR ורעש מתוך ספריות LibriSpeech, הRIRים שיצרנו וwhamr בהתאמה.

1. עבור הסיגנל הנקי:
   i. כיוון שאנו מתעסקים בVAD – זיהוי נוכחות או היעדר דיבור, רצינו לוודא כי ישנם מקטעים בהם אכן אין דיבור ועל כן הוספנו בהקלטת הדובר מקטעים קצרים של אפסים.
   ii. שנינו את אורך הסיגנל לפי הפרמטר שהוגדר בכדי לקבל אורך אחיד לסיגנלים.
2. מעבר בRIR:
   i. העברנו את האות בRIR ע"י הפעלת קונבולוציה ביניהם כך שקיבלנו אות מהודהד.
   ii. נרמלנו את האות ע"י חילוקו בסטיית התקן שלו.
3. בשלב זה חישבנו את הlabels:
   i. ביצענו התמרת STFT לאות.
   ii. חישבנו את הlabel של כל פריים באות ע"פ עוצמת האות:
      i. חישבנו את הלוג של הערך המוחלט של המגניטודה בתוספת אפסילון בכדי למנוע לוג 0.
      ii. חישבנו ממוצע של המגניטודה לאורך ציר התדרים עבור כל פריים.
      iii. עבור כל פריים קבענו נוכחות דיבור אם הממוצע מסעיף קודם היה גדול מערך הסף שהוגדר והיעדר דיבור במקרה ההפוך.
4. הוספת הרעש:
   i. שינינו את גודל הרעש לגודל האחיד שנקבע.
   ii. הוספנו את הרעש לסיגנל המהודהד בהגבר שנקבע לפי הSNR שהוגדר.
   iii. העברנו את האות למישור התדר ע"י התמרת STFT.
   iv. פיצלנו את החלק המדומה והממשי של האות לשני ערוצים שונים כך שהוספנו מימד נוסף לאות.
   v. נרמלנו את אות הSTFT לאורך תדריו השונים.

סה"כ קיבלנו אות מהודהד רועש ואת הlabelים של האות.

פרמטרים:

$$Fs = 16k$$

$$Snr = U[0,5]_{[dB]}$$

$$Threshold = -1.3_{[dB]}$$

$$STFT\ hop\ size = 256$$

$$STFT\ \ window = hann$$

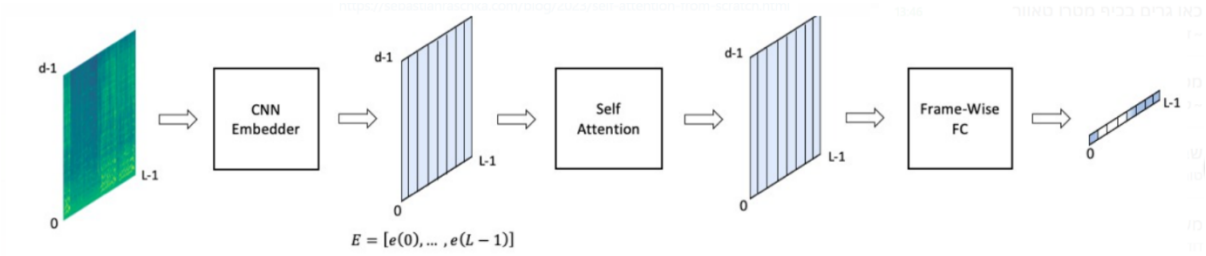
$$STFT\ window\ length = fft\ length = 512$$

ארכיטקטורת המודל

ארכיטקטורת המודל נבנתה בכדי לזהות פריימים בהם יש דיבור ע"ב ספקטוגרמת הסיגנל הרועש.

המודל בנוי משכבת CNN, המשמרת את מידע הפריים ומחלצת תכונותיו, לאחריה שכבת SA שמעבדת את המידע מרצף הפריימים - מנתחת את המידע של כל פריים אל מול שאר הפריימים, ולבסוף שכבת FC המשמשת להערכה סופית של נוכחות/היעד דיבור בכל פריים.

להלן תרשים של המודל –



נפרט על מבנה המודל:

CNN Embedder

לשכבה זו נכנס הקלט אשר הינו הספקטוגרמה של האות, שגודלו הינו [2,F,L].

כאשר – 2 - מספר הערוצים בכניסה, אחד עבור הערך הממשי והשני עבור הערך המדומה. F מספר התדרים, L מספר המסגרות/פריימים – מהווה את אורך הסיגנל.

מטרת השכבה של הCNN היא למצוא את הקשרים בין התדרים הסמוכים.

אנו משתמשים בארבע שכבות של CNN ובין כל שכבה לשכבה אנו מפעילים את פונקציות האקטיבציה: Batch Norm, PReLU non-linearly, Max-Polling.

הפרמטרים של שכבת הקונבולוציה –kernel size, padding נבחרו כך שגודל המוצא שווה לגודל הקלט כך שהמימד של אורך הקלט L נשמר ואיתו המידע המרחבי. במוצא כל מעבר בשכבת הקונבולוציה מתקבל אות בגודל [C,F,L], כאשר C – מספר הערוצים במוצא.

שכבת Batch norm: מנרמלת את המוצא המתקבל משכבת הקונבולוציה.

שכבת PReLU non-linearly: פונקציית אקטיבציה - הופכת את הרשת ללא לינארית.

שכבת max-polling: השכבת מצמצמת את המידע התדרי ע"י לקיחת התדר בעל האנרגיה הגבוהה ביותר מתוך מספר תדרים שהוגדר. ביצענו max-polling של [2,1], כך שמתוך כל 2 מקטעים של תדרים בחרנו את התדר בעל האנרגיה המקסימלית. לפיכך מימד התדר קטן פי 2 לאחר כל מעבר בשכבה זו. סה"כ לאחר 4 שכבות קונבולוציה מימד התדר קטן פי 16, את המימד החדש נסמן ב- F_0.

סה"כ לאחר הפעלת 4 שכבות CNN מתקבל אות שגודלו - $[C, F', L\ ]$.

לאחר מכן התוצאה עוברת ב-**Frame-Wise flattening operation** אשר משטחת את המידע של התדרים F והערוצים C עבור כל פריים כך שבמוצא אנו מקבלים מידע שגודלו - $[C \cdot F', L\ ]$.

לאחר מכן, התוצאה עוברת בשכבת **Frame-Wise FC** אשר מורידה את המימד של כל וקטור מ- $C \cdot F'$ ל- d כך שהמימד לאחר מכן הינו- [d, L].

את המידע המתקבל במוצא השכבה נסמן ב- $E = [e(0), \dots, e(L-1)]$.

<u>פרמטרים:</u>

$$CNN: \quad Kernel\ size = 3 \times 3, \qquad padding = 1$$
$$F = 256, \qquad C = 32 \quad \rightarrow \quad F' = 16 \quad \rightarrow \quad C \cdot F' = 512$$
$$d = 256$$

Self Attention encoder

מטרת השכבה של הSA היא למצוא את הקשרים בין הפריימים השונים המיוצגים ע"י הוקטורים e(l).

בשכבה זו אנו מבצעים multi head SA ולאחריו שכבת נורמליזציה ולאחריה שכבת *Feed Forward*.

שכבת הmultiheadattention מורכבת מH ראשים. כמו כן, בשכבה מתבצע dropout בכדי למנוע overfitting.

בשכבת הSA למעשה משווים בין כל וקטור לשאר הוקטורים ולכן במוצא הSA לבדו מתקבל אות במימד [L,L].

שכבת ה-*Feed Forward*, מעלה את המימד של כל וקטור מ*L* ל*d_ff* ואז מורידות את המימד חזרה לגודל המקורי - *[L,d]*

<u>פרמטרים:</u>

$$H = 16, \qquad d_{ff} = 512, \qquad dropot = 0.1$$

Fully connected

בשכבה זו מתקבלת ההחלטה על נוכחות דיבור בכל פריים.

השכבה מקבלת אות המורכב מL וקטורים(פריימים) בגודל d כל אחד ועבור כל פריים מחזירה מספר המעיד על הסיכוי לנוכחות או אי נוכחות דיבור בפריים הנ"ל.

לפיכך הפלט המתקבל הינו בגודל של [L,1] ולאחר השטחה למימד אחד הינו [L].

בחלק זה העברנו את הקלט בשכבת FC. בכדי לקבל הסתברויות נדרשנו להעביר את המידע בפונקציית סיגמואיד. השימוש בפונקציה בוצע בחישוב הLOSS – בחישובו השתמשנו בפונקציה BCEWithLogitsLoss המעבירה ראשית את המידע בפונקציית סיגומאיד.

אימון המודל

<u>פרמטרים:</u>

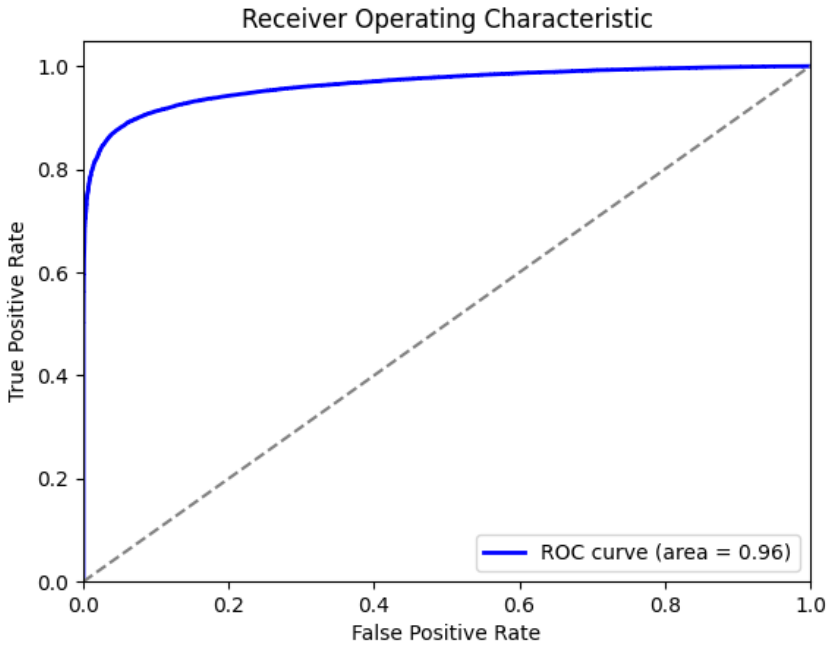
- פונקציית הloss של המודל הינה – BCEWithLogitsLoss המשלבת שכבת סיגמואיד ו- *BCELoss*.
- האופטומיזטור בו מבוצע שימוש במודל הינו – AdamW. אופטומיזטור זה משלב את היתרונות של SGD (Stochastic Gradient Descent) יחד עם תהליכים של ממוצעים נעים ותחזוקת גרדיאנטים.
- $Learning\ rate\ =\ 0.001,\ \ Batch\ size = 20$.

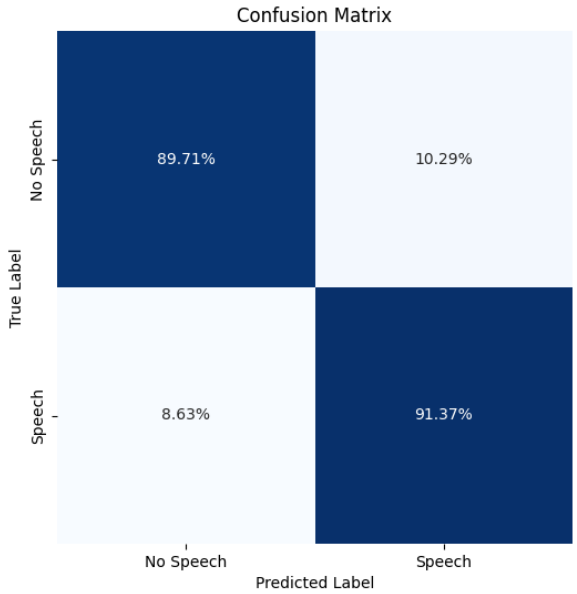
תוצאות

המודל מנסה לחזות את הפריימים בהם יש דיבור ע"ב מודל למידה עמוקה.

המודל הביא לתוצאות טובות בעלי ציונים גבוהים כפי שניתן לראות בגרף הROC וב'מטריצת הטעות':

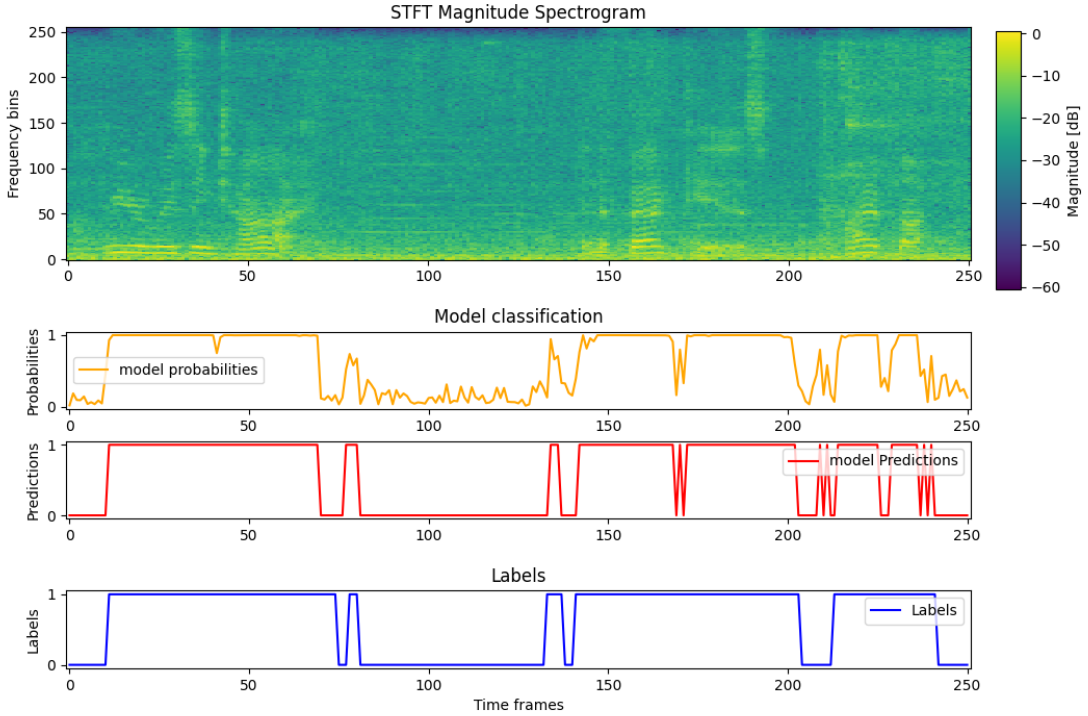
את הטסט הרצנו עם $Snr \sim U[0,5]_{[dB]}$ כפי שהרצנו את המודל עצמו.





כאשר התגיות שנחזו נקבעו ע"ב קביעת ערך סף להסתברויות שנחזו. הסתברות הגדולה מ- 0.5 נקבעה כדיבור.

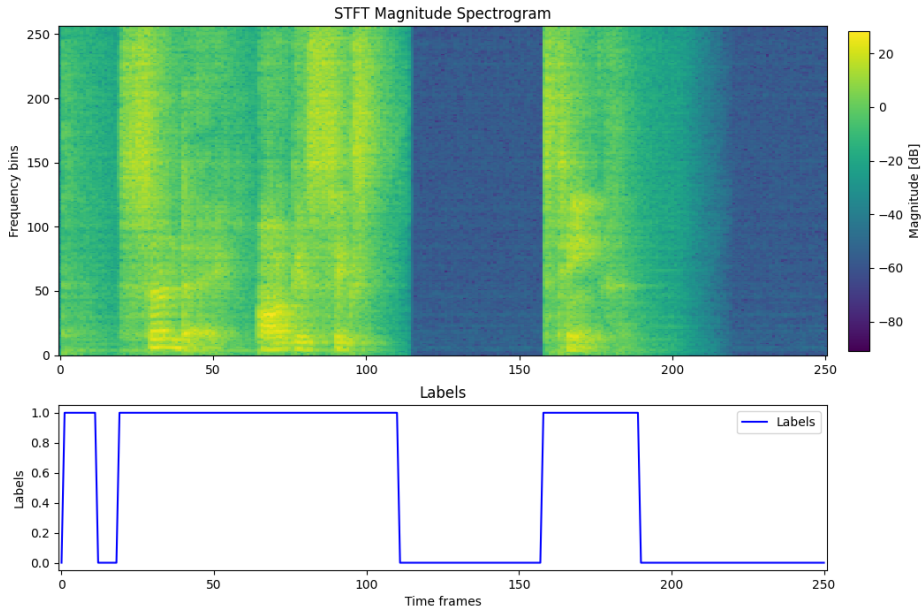
להלן דוגמא לתוצאות חיזוי של המודל עבור סיגנל כניסה:



הגרף הראשון מציג את הספקטוגרמה של האות, שני הגרפים אחריו מציגים את התוצאות שהתקבלו לפי המודל, האחד ההסתברויות שהתקבלו, והשני מציג את התגיות שהתקבלו לפי סף ההחלטה. לאחר מכן מוצגות התוויות שנקבעו בשלב ייצור הדאטה.

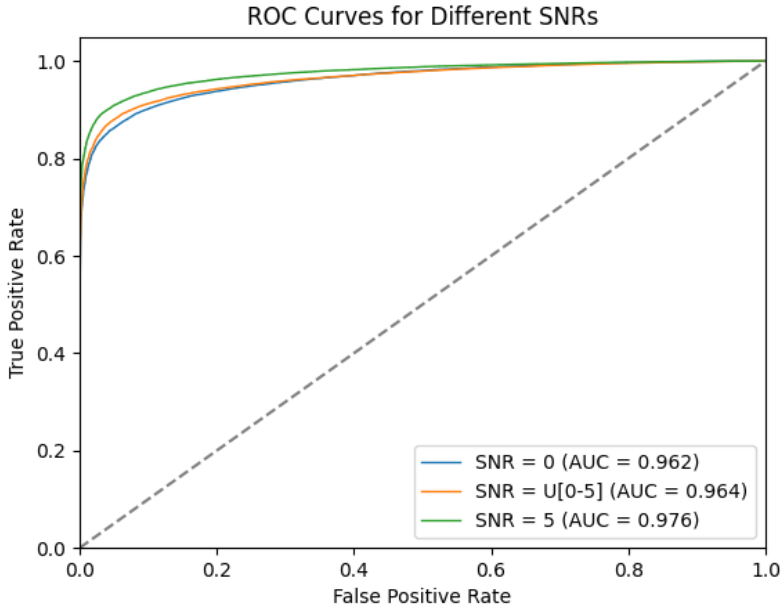
אנו רואים בגרף כי התגיות שנחזו קרובות מאוד לתגיות הנכונות שנקבעו כך שזוהי דוגמא לכך שהמודל פועל בהצלחה.

הlabels נקבעו ע"ב ספקטוגרמת האות המהודהד (ללא הרעש) כפי שניתן לראות בדוגמא להלן:



השווינו את תוצאות המודל המתקבל עבור סט טסט של $Snr \sim U[0,5]_{[dB]}$ לתוצאות המתקבלות עבור SNRים נמוכים/גבוהים יותר בסט הטסט:

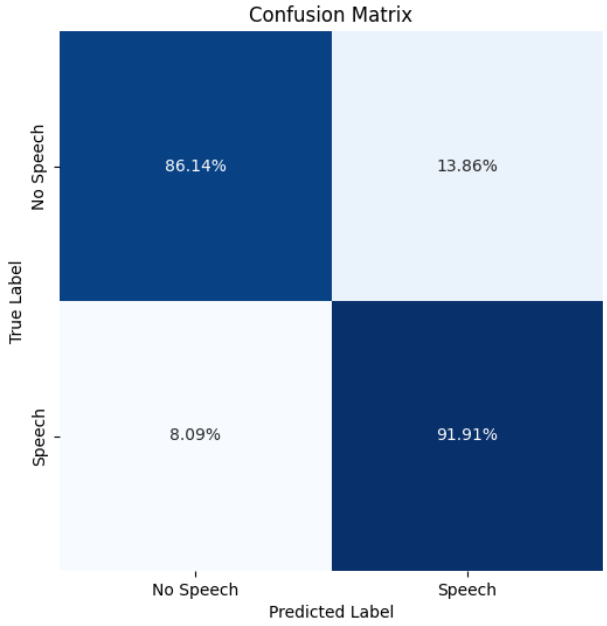
<u>להלן גרף הROC-AUC עבור SNRים שונים:</u>



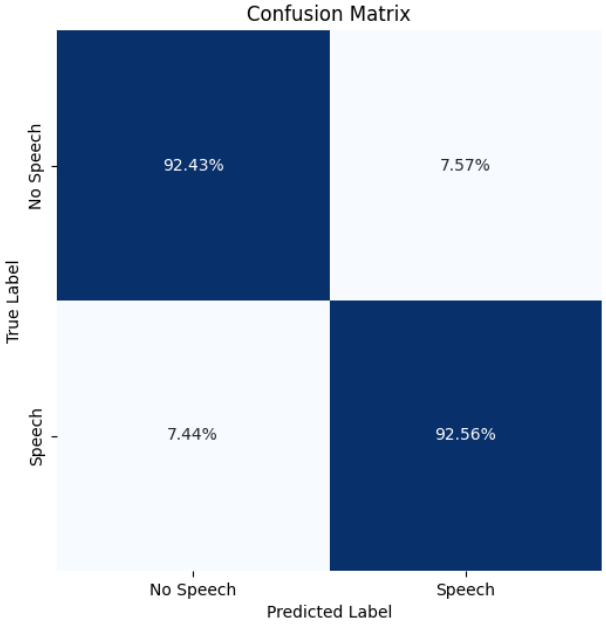
ניתן לראות כפי שצפוי כי ככל שיחס האות לרעש גבוה יותר כך המודל נותן תוצאות טובות יותר.

להלן מטריצות הטעות של המודלים:

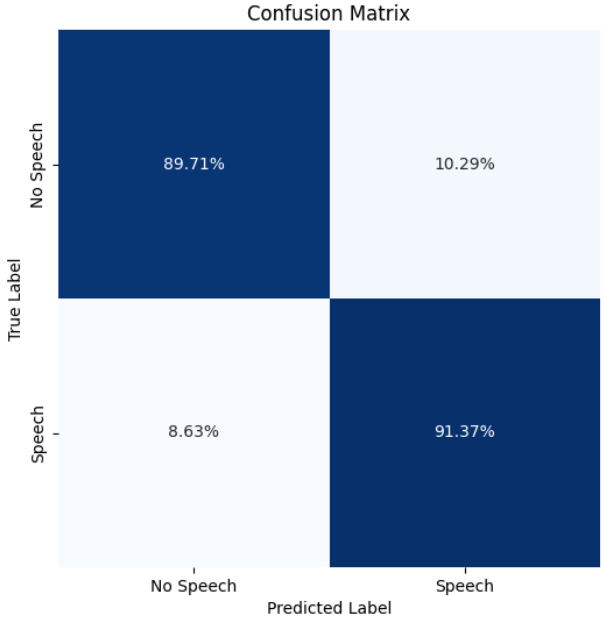
SNR 0:



SNR 5:



SNR ~U[0,5]:



גם במדדים של מטריצת הטעות אנו רואים כי ככל שיחס האות לרעש גבוה יותר כך התוצאות טובות יותר. נשים לב שכל שהSNR גבוה יותר כך היחס בין הFP לFN קטן. הנ"ל יכול להשפיע על בחירת ערך סף שונה לקביעת נוכחות דיבור ע"ב ההסתברויות שהתקבלו מהמודל, זאת אם קיים מידע מקדים על היחס של האות לרעש בחדר.

מסקנות וסיכום

בפרוייקט זה מוצג מודל למידה עמוקה החוזה בהצלחה מרובה נוכחות/אי נוכחות של דיבור בסיגנל.

הנ"ל מהווה פתרון לבעיית הVAD.

ארכיטקטורת המודל מבוססת על CNN וSA. דבר המאפשר למודל לחזות דיבור בצורה טובה מתוך אות מהודהד ורועש: שכבת הCNN משמשת לניצול המידע המרחבי מתוך ספקטרום התדרים, כך שהיא משמשת למציאת קשרים בין תדרים סמוכים.

ושכבת הSA משמשת בכדי למצוא מידע הקשרי. היא מוצאת את הקשרים בין הפריימים השונים.

שילוב השכבות הנ"ל ביחד עם שכבת FC מוביל לתוצאות טובות במיוחד ע"י כך שהוא מנצל הן את המידע המקומי והן את המידע הגלובאלי.

על אף הישגיו של מודל וזה, ואף בהמשך אליו, כמובן כי שיפורים נוספים רצויים ונדרשים בתחום הVAD על מנת לשפר את המודלים הקיימים כיום ולהביא לתוצאות טובות אף יותר.

לסיכום,

המודל המובא מציג ארכיטקטורה משולבת CNN וSA הפועלת באופן מיטבי ומביאה תוצאות עם אחוזי הצלחה גבוהים.