הפקולטה להנדסה

המעבדה לעיבוד אותות

# זיהוי כיוון הדובר ע"י למידה עמוקה עבור מערכי חיישנים במיקום לא ידוע

שגיא דלה טורה

דביר דוידוביץ'

פרויקט שנה ד' לקראת תואר ראשון בהנדסה

מנחה: ד"ר עופר שוורץ

מנחה אקדמי: פרופ' שרון גנות

אוקטובר 2022

# תוכן עניינים


רקע לפרוייקט 3

מטרת הפרויקט 3

חלק א – שיערוך אנליטי של ה-DOA 4

הפקת האותות 4

שיטה ראשונה – חישוב באמצעות קרוס קורלציה 4

שיטה שניה – חישוב באמצעות מעבר לתדר וחישוב קרוס ספקטרום 7

חלק ב - עיבוד אנליטי מקדים לרשת 9

עיבוד מס' 1 – האות בזמן 9

עיבוד מס' 2 – קרוס הקורלציה בין האותות מחושבת בכל אחד מהכיוונים 9

עיבוד מס' 3 – תמונה תדרית עבור כל אחד מהכיוונים 10

חלק ג - עיבוד ברשת נוירונים 12

יצירת הdata base 12

שימוש ב-RIR (Room Impulse Response) generator 12

הוספת רעש 12

יצירת הfeatures 13

הlabels ופונקציית הVAD 14

מבנה התיקיות 14

הרשת 15

ארכיטקטורת הרשת 15

פונקציית ה-loss 17

התכנסות הרשת 18

ניסויים – הקלטות אמת 19

תוצאות 20

תוצאות הרשת על הקלטת אמת והקלטות סימולציה - זיהוי הזווית ב pre-process מספר 3 20

דוגמא לחישוב הזווית בצורה אנליטית בעזרת קרוס קורלציה 23

תוצאות עבור חיזוי אנליטי בעזרת מקסימום של הקרוס קורלציה 23

זיהוי VAD 25

רעיונות להמשך 28

סיכום ומסקנות 29

ביבליוגרפיה 29

## רקע לפרוייקט

לעיתים קרובות משתמשים בכיוון הגעה של דובר על מנת לנטרל רעשי סביבה, להתמקד באות מכיוון מסוים, ולבצע עיבודים נוספים שמבוססים על כיוון הדובר - DOA.

שיטות קלאסיות לזיהוי כיוון דובר עובדות בצורה טובה בסביבה שאינה רועשת ומהדהדת, אך בסביבות פחות אידאליות מתקשות להציג ביצועים טובים.

לכן, ניתן להשתמש בשיטות של למידה עמוקה, ורשתות נוירונים. שיטות אלו נותנות ביצועים טובים יותר גם בסביבות מהדהדות ורועשות.

הבעיה היא שעל מנת ללמד את הרשת, על מערך חיישנים מסוים נדרש מיקום של כל אחד מהחיישנים והמרחקים בניהם ונדרשות הקלטות רבות על מערך ספציפי - במטרה להגיע לכיול אופטימלי. שיטה זו דורשת עבודה רבה על כל מערך חיישנים חדש.

בפרויקט זה אנו נשתמש בשיטות של למידה עמוקה על מנת לגלות את מיקום הדובר ביחס למערך חיישנים ללא תלות במיקום החיישנים, וכך ניתן ללמד את הרשת פעם אחת בלבד על מערך כללי, ולאחר מכן היא תוכל לתת תוצאות לכל מערך אפשרי.

נעשה זאת על ידי עיבוד מידע ראשוני לפני הכניסה לרשת. את עיבוד המידע הראשוני שנעשה לאותות נבסס על שיטות אנליטיות וקלאסיות למציאת DOA כגון SRP-HT וSRP-PHAT.

## מטרת הפרויקט

בפרויקט נממש מערכת לזיהוי כיוון של דובר ע"י למידה עמוקה. המערכת אמורה להתאמן על מערך בעל מבנה גנרי כך שיתקבלו תוצאות טובות גם עבור מערך בעל מבנה בלתי צפוי. המערכת תיבדק בתנאים מסובכים של רעש והדהוד בהקלטות שונות.

התוצר הסופי יהיה מערכת למידה עמוקה שמזהה את כיוון הגעת הדובר עבור מערכי חיישנים במיקום לא ידוע.

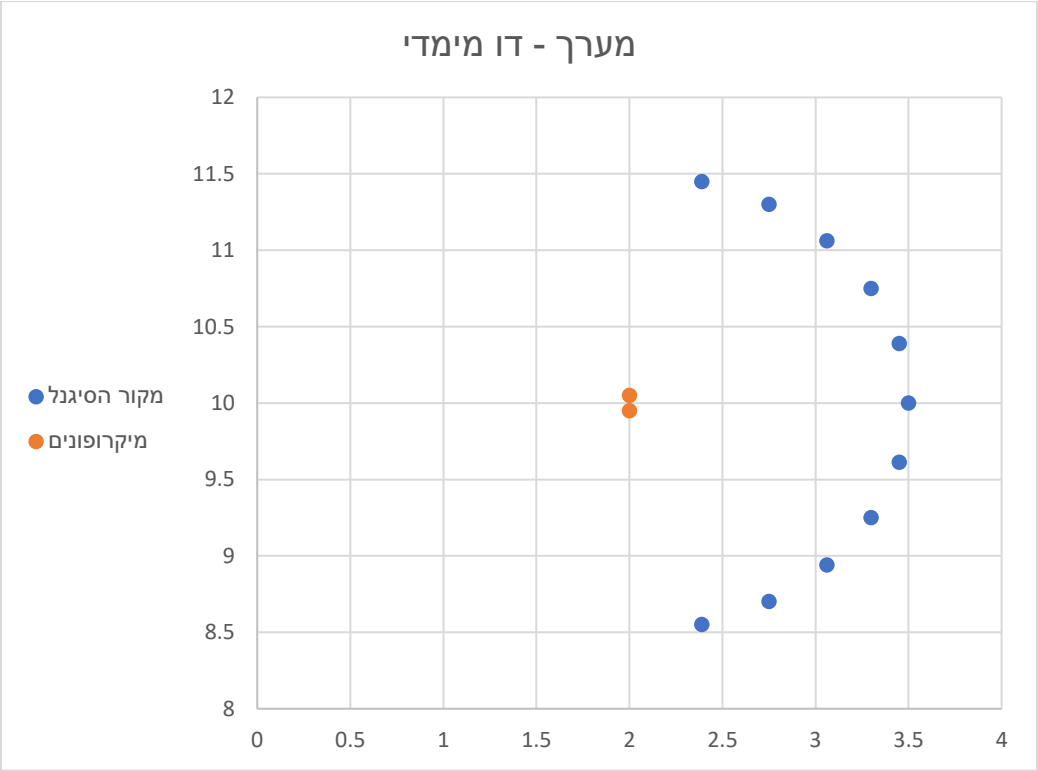
# חלק א – שיערוך אנליטי של ה-DOA

בשלב הראשון של הפרויקט, השתמשנו בשיטות אנליטיות ע"מ לשערך מתוך מערך מיקרופונים את כיוון ההגעה.

## הפקת האותות

יצרנו תגובת Room Input Response (RIR) עבור שני מיקרופונים שמונחים אחד ליד השני במרחק cm10 אחד מהשני. ויצרנו תגובת הלם עבור מספר זוויות שונות. לתגובת ההלם האלו עשינו קונבולציה עם הקלטה נקיה וכך קיבלנו סימולציית של הקלטות מזוויות שונות.

תמונת החדר והמיקומים השונים של ההקלטות והמיקרופונים שיצרנו:




כאשר ההקלטות נמצאות בזוויות:

$$[15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 175°]$$

## שיטה ראשונה – חישוב באמצעות קרוס קורלציה

על מנת לחשב את כיוון ההגעה, חישבנו את הפרש זמן ההגעה בין שתי המיקרופונים. ע"מ לעשות זאת בצורה מיטבית חישבנו את הקרוס קורלציה בין שתי האותות משני המיקרופונים, ובדקנו את מיקום הפיק. למשל אם הפיק הוא במרכז הקרוס קורלציה, משמע שאין הפרש בזמני ההגעה, ולכן האות הגיע מזווית של 90 מעלות.

האלגוריתם שמימשנו בפועל:

קיבלנו שני אותות. חילקנו אותם לפריימים של 1024 דגימות כל אחד, עם חפיפה של 512 דגימות בין פריים לפריים, כיוון שנרצה לבצע את החישוב של הכיוון בזמן אמת, וגם לאפשר לכיוון ההגעה להשתנות לאורך ציר הזמן.

על כל פריים, ביצענו קרוס קורלציה בין שתי האותות, וקיבלנו קרוס קורלציה באורך 2047.

בנוסף, ביצענו החלקה על הקרוס קורלציה, על מנת לקבל תוצאות יותר אמינות, ופחות קפיצות. כלומר עבור כל פריים קיבלנו קרוס קורלציה, והחלקנו אותה יחד עם הקרוס קורלציה הממוצעת שנוצרה לנו מהפריים הקודם. ביצענו זאת על פי הנוסחה הבאה:

$$avarge_cross_correlation = \alpha * avarge_cross_correlation + (1-\alpha) * new_cross_correlation$$

כאשר בחרנו: $\alpha = 0.95$. בעצם יוצא שכל קרוס קורלציה שמגיע משפיעה על הממוצע בצורה חזקה בפעם הראשונה, ואח"כ ההשפעה שלה על הממוצע הנע דועך בצורה אקספוננציאלית, ולאחר מספר פריימים ההשפעה שלה כבר תהיה זניחה.

בשלב הבא, חישבנו את המיקום של הפיק (נקודת המקסימום) של הקרוס קורלציה הממוצעת, ואת ההפרש בין מיקום זה לבין המרכז וקיבלנו הפרש דגימות.

לאחר מכן מתוך הפרש הדגימות חישבנו מתוך זה את זווית ההגעה. נניח שהפרש זמן ההגעה הוא x דגימות. כדי לעבור מדגימות להפרש זמנים נבצע:

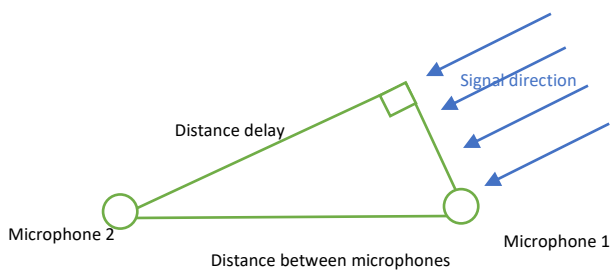
$$\Delta T = \frac{x}{f_s}[s]$$

כאשר $f_s$ הוא תדר הדגימה של האות.

מתוך $\Delta T$ חישבנו את הפרש ההגעה במטרים, ע"פ הנוסחא הבאה:

$$Distance\ delay = \Delta T * V_{sound} = \Delta T[s] * 343[\frac{m}{s}]$$

וכעת ניתן לחשב את הזווית בעזרת arccos. ניתן לראות זאת במשולש הבא:




והמשוואה היא:

$$\alpha = arccos\ arccos\left(\frac{Distance\ delay}{Distance\ between\ microphones}\right)\ [rad]$$

ע"מ להדגים את תוצאות החישוב, יצרנו קובץ אקסל, שמחשב את היחס בין הפרש ההגעה במספר דגימות, לבין הזווית (חישוב לשני הכיונים). הפרמטרים שנדרשים כinput הם: המרחק בין המקרופונים, ותדר הדגימה. לדוגמא, עבור תדר דגימה של 16kHz, ומרחק בין המירופונים של 0.1m, נקבל את שתי הטבלאות הבאות:

| by num of samples | time delay (s) | distance delay | angle |
|---|---|---|---|
| 16 | 0.001 | 0.343 | FAIL |
| 15 | 0.0009375 | 0.3215625 | FAIL |
| 14 | 0.000875 | 0.300125 | FAIL |
| 13 | 0.0008125 | 0.2786875 | FAIL |
| 12 | 0.00075 | 0.25725 | FAIL |
| 11 | 0.0006875 | 0.2358125 | FAIL |
| 10 | 0.000625 | 0.214375 | FAIL |
| 9 | 0.0005625 | 0.1929375 | FAIL |
| 8 | 0.0005 | 0.1715 | FAIL |
| 7 | 0.0004375 | 0.1500625 | FAIL |
| 6 | 0.000375 | 0.128625 | FAIL |
| 5 | 0.0003125 | 0.1071875 | FAIL |
| 4 | 0.00025 | 0.08575 | 30.96 |
| 3 | 0.0001875 | 0.0643125 | 49.97 |
| 2 | 0.000125 | 0.042875 | 64.61 |
| 1 | 0.0000625 | 0.0214375 | 77.62 |
| 0 | 0 | 0 | 90.00 |
| -1 | -0.000062 | -0.0214375 | 102.38 |
| -2 | -0.000125 | -0.042875 | 115.39 |
| -3 | -0.000187 | -0.0643125 | 130.03 |
| -4 | -0.00025 | -0.08575 | 149.04 |
| -5 | -0.000312 | -0.1071875 | FAIL |
| -6 | -0.000375 | -0.128625 | FAIL |
| -7 | -0.000437 | -0.1500625 | FAIL |
| -8 | -0.0005 | -0.1715 | FAIL |
| -9 | -0.000562 | -0.1929375 | FAIL |
| -10 | -0.000625 | -0.214375 | FAIL |
| -11 | -0.000687 | -0.2358125 | FAIL |
| -12 | -0.00075 | -0.25725 | FAIL |
| -13 | -0.000812 | -0.2786875 | FAIL |
| -14 | -0.000875 | -0.300125 | FAIL |
| -15 | -0.000937 | -0.3215625 | FAIL |
| -16 | -0.001 | -0.343 | FAIL |

| by angle (degrees) | distance delay | time delay (s) | num of samples |
|---|---|---|---|
| 5 | 0.0996 | 0.0002904 | 4.647 |
| 10 | 0.0984 | 0.0002871 | 4.594 |
| 15 | 0.0965 | 0.0002816 | 4.506 |
| 20 | 0.0939 | 0.0002739 | 4.383 |
| 25 | 0.09063 | 0.000264 | 4.228 |
| 30 | 0.0866 | 0.0002524 | 4.040 |
| 35 | 0.08191 | 0.000238 | 3.821 |
| 40 | 0.07660 | 0.0002233 | 3.573 |
| 45 | 0.07071 | 0.00020615 | 3.298 |
| 50 | 0.06427 | 0.0001874 | 2.998 |
| 55 | 0.05735 | 0.0001672 | 2.676 |
| 60 | 0.05 | 0.0001457 | 2.332 |
| 65 | 0.04226 | 0.0001232 | 1.971 |
| 70 | 0.03420 | 9.971E-05 | 1.595 |
| 75 | 0.02588 | 7.545E-05 | 1.207 |
| 80 | 0.01736 | 5.062E-05 | 0.810 |
| 85 | 0.00871 | 2.540E-05 | 0.407 |
| 90 | 6.125E-18 | 1.785E-20 | 0.000 |
| 95 | -0.00871 | -2.540E-05 | -0.407 |
| 100 | -0.01736 | -5.062E-05 | -0.810 |
| 105 | -0.02588 | -7.545E-05 | -1.207 |
| 110 | -0.03420 | -9.971E-05 | -1.595 |
| 115 | -0.04226 | -0.0001232 | -1.971 |
| 120 | -0.05 | -0.0001457 | -2.332 |
| 125 | -0.05735 | -0.0001672 | -2.676 |
| 130 | -0.06427 | -0.0001874 | -2.998 |
| 135 | -0.07071 | -0.0002061 | -3.298 |
| 140 | -0.07660 | -0.0002233 | -3.573 |
| 145 | -0.08191 | -0.000238 | -3.821 |
| 150 | -0.0866 | -0.0002524 | -4.040 |
| 155 | -0.09063 | -0.000264 | -4.228 |
| 160 | -0.09396 | -0.0002739 | -4.383 |
| 165 | -0.09659 | -0.0002816 | -4.506 |
| 170 | -0.09848 | -0.0002871 | -4.594 |
| 175 | -0.0996 | -0.0002904 | -4.647 |
| 180 | -0.1 | -0.0002915 | -4.665 |

אם כן, ראינו איך ניתן להגיע מתוך הפרש הדגימות בין הפיק לבין מרכז הקרוס קורלציה, לזווית הגעה של האות. הבעיה עם שיטה זו, היא שנוכל למצוא כיוונים רק של זוויות שמגיעות מתוך מספר דגימות שלם. למשל אם ניקח אות דגום ב16kHz ומרחק של m0.1 בין מיקרופונים, נוכל לקבל רק 9 זוויות אפשריות. ולא רק שנוכל לקבל רק 9 זוויות, אלא גם רוב הזוויות יהיו סביב המרכז (90 מעלות), ובצדדים נקבל רזולוציה מאד נמוכה (הדבר נובע מהאי ליניאריות של הcos, שמשתנה מהר סביב $\frac{\pi}{2}$ אך כמעט קבוע סביב 0 ו $\pi$).

לכן, נרצה לשפר את השיטה ע"מ לקבל יכולת לחשב זוויות שאינם נמצאות בהפרש דגימות שלם. ע"מ לעשות זאת, נעשה איטרפולציה לקרוס קורלציה, וכך נוכל לחשב את הקרוס קורלציה גם בין דגימות. ביצענו איטרפולציה פולינומיאלית ממעלה שלישית לקרוס קורלציה, ואכן ראינו שיפור ניכר בתוצאות, כאשר קיבלנו רזולוציה טובה יותר לזוויות ההגעה.

כדי לבדוק את עצמנו היינו צריכים שני אותות שיש ביניהם הפרש של מספר לא שלם של דגימות. ע"מ ליצור זאת השתמשנו בקוד מטלב של פרופ' שרון גנות, שיודע לייצר שני אותות עם הפרש של מספר לא שלם של דגימות.

## שיטה שניה – חישוב באמצעות מעבר לתדר וחישוב קרוס ספקטרום

אפשרות שניה לחישוב זווית ההגעה, היא ע"י מעבר לתדר, בצורה הבאה:

נתמיר כל אחד מהאותות ע"י FFT, לאחר מכן נכפיל אותם את ההתמרה של הראשון בהתמרה הצמודה של השני, ונקבל את הקרוס ספקטרום.

בשלב הבא נוכל לעשות התמרה חזרה לזמן ולקבל את הקרוס קורלציה. אך אם נעשה זאת בעצם נחזור שוב לשיטה הראשונה, ולא עשינו כלום. לכן יש שלושה יתרונות שנוכל להשיג במעבר לתדר:

1. כאשר קיבלנו את הקרוס ספקטרום נוכל להכפיל בפונקציות משקל שונות (למשל עבור שיטת PHAT נחלק בערך המוחלט של הקרוס ספקטרום), וכך בעצם לסנן את הקרוס ספקטרום, ולקבל תוצאות טובות יותר (מבוסס על המאמר של GCC של שנות ה-70).
2. כאשר נעבור חזרה לזמן נוכל להציב זמנים לא שלמים באקספוננט. וכך נקבל את הקרוס קורלציה ברזולוציה של הזמנים הרצויים לנו, בהתאם לזוויות שנבחר. כלומר בעצם נעשה התמרה הופכית לזמן ונציב זמנים שמתאימים לזוויות שנרצה לבדוק, במרווחים שווים.
3. בהמשך כאשר נעבור לחלק של הלמידה עמוקה, נוכל להזין את הרשת בתדר, ולא רק בזמן. זה יכול לעזור לרשת, כאשר היא תקבל מידע מהתדר, ולא רק מידע בזמן. נרחיב על כך בהמשך.

נציג את השיטה בצורה מתמטית:

האותות המקוריים שלנו:

$$\mathrm{x}_1(n),\ x_2(n)$$

נזכור כי אנו עוסקים באותות בדידים – במקרה שלנו אות דיבור רציף שדגום בתדר 16kHz.

התמרת פוריה לשני האותות:

$$X_{1,2}(k) = \mathcal{F}\left(\mathrm{x}_{1,2}(n)\right) = \sum_{n=0}^{N-1} x_{1,2}(n) \cdot e^{-\frac{2\pi i}{N}kn}$$

כעת נכפיל, ונקבל את הקרוס ספקטרום:

$$S_{X_1X_2}(k) = X_1(k) \cdot \overline{X_2(k)}$$

כעת, נרצה לחזור חזרה ממישור התדר למישור הזמן. אם נעשה פשוט התמרה הופכית של הקרוס ספקטרום נקבל את הקרוס קורלציה. אך אנו נעשה זאת בצורה מתוחכמת יותר. נעשה התמרה הופכית בנקודות מסוימות בזמן. כלומר נחשב את הקרוס קורלציה רק בנקודות הרצויות. נעשה זאת בצורה הבאה:

$$R_{X_1X_2}(\tau) = \mathcal{F}^{-1}\left(S_{X_1X_2}(k)\right) = \frac{1}{N}\sum_{k=0}^{N-1} S_{X_1X_2}(k) \cdot e^{\frac{2\pi i}{N}k\tau}$$

כעת נוכל להציב עבור הפרשי זמן, $\tau$, שונים. ולקבל את הקרוס קורלציה במקומות שונים. למשל עבור $\tau = 0$, נקבל את הקרוס קורלציה בהפרש זמן 0.

היתרון הגדול בשיטה זו, כפי שהזכרנו למעלה, הוא שנוכל להציב עבור $\tau$, גם מספרים לא שלמים, וכך לקבל את הקרוס קורלציה בזמנים לא שלמים. אנו מעוניינים בקרוס קורלציה בזמנים לא שלמים, כי עבור רוב כיווני ההגעה הפרש הזמנים בין האותות יהיה לא שלם. אז בעצם נוכל לבחור את הזוויות שבהם נרצה לחשב, ומתוכם לחשב את הפרש הזמנים מתוך הטבלה שהראינו לעיל. ולאחר מכן לחשב את הקרוס קורלציה בזמנים לא שלמים אלו.

בהמשך, בחלק של התוצאות, נראה דוגמא לגרף של חישוב הקרוס קורלציה בכיוונים שונים, ונראה כיצד הפיק מתקבל בכיוון שממנו מגיע הדיבור.

# חלק ב - עיבוד אנליטי מקדים לרשת

כעת, נעבור לדבר על שיטת העבודה שלנו, בעיבוד המקדים, שאותו נזין בהמשך לרשת הנוירונים. אנו נציג כאן מספר אפשרויות שונות לעיבוד מקדים, בהמשך נזין כל אחד מהם לרשת, ונבחן את התוצאות השונות שיתקבלו.

## עיבוד מס' 1 – האות בזמן

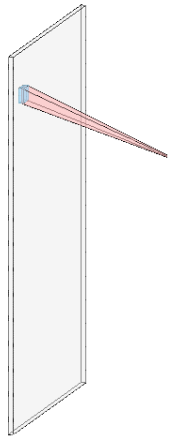
אפשרות הראשונה היא להזין לרשת את האות בזמן. כלומר הרשת תקבל מטריצה של 1024 דגימות (גודל הFrame) כפול N מיקרופונים.

בעצם כאן אנחנו מאכילים את הרשת באותות בלי לעבד אותם כלל.

החיסרון בשיטה זו, היא שהרשת תצטרך ללמוד מחדש עבור כל מערך מיקרופונים שונה, כיוון שמספר המיקרופונים, והמיקום שלהם מהותי עבור הרשת. עבור מערך חדש, או מספר מיקרופונים שונה נצטרך לאמן את הרשת מחדש. זה דורש מאיתנו המון עבודה של איסוף דאטה ואימון הרשת.

חיסרון נוסף שיש בשיטה זו, הוא העובדה שאנחנו יכולים בעצם לעזור לרשת ע"י עיבוד מקדים, וכך לשפר את התוצאות שלה.

לכן בשיטות הבאות ננסה לעשות עיבוד מקדים שאותו נזין לרשת.



1024xNx1

## עיבוד מס' 2 – קרוס הקורלציה בין האותות מחושבת בכל אחד מהכיוונים

כעת אנו נבצע עיבוד מקדים על האות, במטרה להגיע לקרוס קורלציה של האות. כפי שהצגנו למעלה, היחס בין הזווית לבין ההזזה במספר הדגימות אינו לינארי. אנו נרצה לחשב את ערך הקרוס קורלציה אך רק בנקודות שמעיינות אותנו, כלומר בהתאם למספר הכיוונים שנרצה לבדוק.

לדוגמא, אם נרצה רזולוציה של 24 כיוונים מתוך 360 מעלות (כלומר כל 15 מעלות), נחשב את פונקציית הקרוס קורלציה אך ורק ב24 נקודות המתאימות. אך כיוון שנקודות אלו לאו דווקא נוחות על מספר שלם של דגימות, נצטרך להשתמש באחת השיטות שתיארנו למעלה כדי להגיע לקרוס קורלציה בנקודות אלו.

אפשרות אחת תהיה לעשות אינטרפולציה פי 100, ואז לבחור רק את הנקודות שמעניינות אותנו.

אפשרות שניה היא לעבור לתדר (קרוס ספקטרום), ואז לחזור לזמן רק בנקודות שמעיינות אותנו. כלומר להציב מספר לא שלם במשתנה הזמן באקספוננט של פונקציית הIFFT.

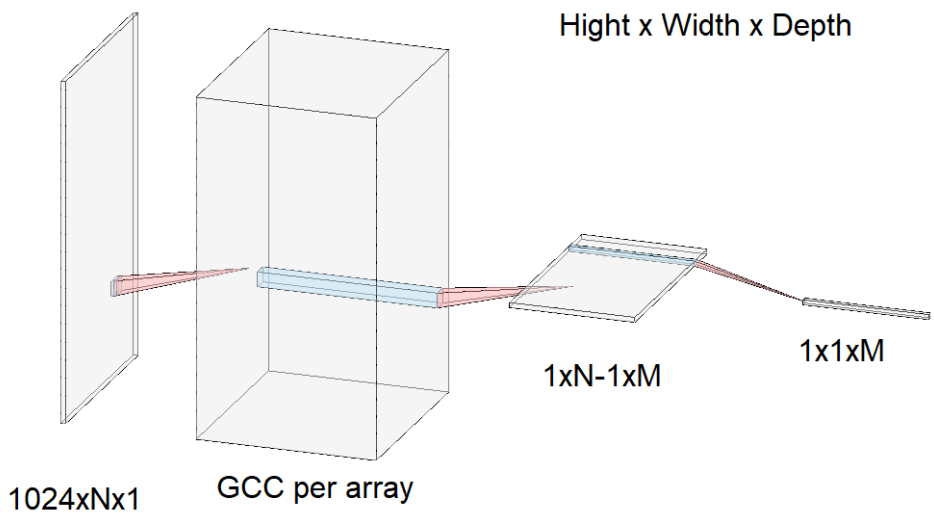
אם כן, קיבלנו $M$ דגימות של קרוס קורלציה (למשל בדוגמא לעיל $M = 24$), ואותם אנו נגיש לרשת. וקטור במימדים $(1, M)$.

כיוון שאנו מדברים על מערך של יותר משני מיקרופונים, ניתן לחשב וקטור כזה לכל אחד מהזוגות, או לחלופין, לבחור מיקרופון אחד שישמש כרפרנס, וליצור וקטור כזה למיקרופון הרפרנס יחד עם כל אחד מהמיקרופונים. כלומר אם יש לנו $N$ מיקרופונים נקבל $N - 1$ וקטורים כאלה.

לאחר מכן נסכום את כל הוקטורים האלו לוקטור אחד, ואותו נגיש לרשת.

נשים לב שבשיטה זאת, זה בכלל לא משנה לרשת כמה מיקרופונים היו, ומה היה הצורה של המערך, היא מקבלת מערך אחד של $M$ נקודות קרוס קורלציה, בכל אחד מהכיוונים, ולומדת אותו. כך בעצם פתרנו את הבעיה של מערך משתנה.

לסיכום, בשיטה זו אנחנו מזינים לרשת מטריצה בגודל $(1, M)$. כאשר 1 הוא הגודל של הקרוס קורלציה עבור כיוון מסוים, וM הוא מספר הכיוונים.




## עיבוד מס' 3 – תמונה תדרית עבור כל אחד מהכיוונים

בשיטה זו, ננסה להזין לרשת נתונים מעט יותר מעובדים, על מנת לעזור לה ללמוד ולהתאמן.

בדומה לשיטה הקודמת, נחשב FFT לכל אחד מהאותות (על פי גודל הפריים $F$), נכפיל ונקבל את הקרוס ספקטרום שוב בF נקודות תדר. למשל אם נבחר $F = 1024$ נקבל 1024 נקודות תדר. אך כעת במקום לעבור חזרה לקרוס קורלציה בזמן, ע"י התמרה הופכית, ננסה להשתמש במידע התדרי שלנו כדי לעזור לרשת.

נחשב וקטור של אקספוננטים בתדר עבור כל אחד מהזוויות הרצויות (כלומר נציב במשתנה הזמן של כל אקספוננט את מספר הדגימות שמתאים לזווית הזו), וכך נקבל $M$ וקטורים של $F$ אקספוננטים בתדר. נכפיל כל אחד מ$M$ הוקטורים בוקטור של הקרוס ספקטרום שלנו, וכך נקבל $M$ וקטורים של $F$ תדרים. את הוקטורים האלה נזין לרשת.

נשים לב, שאם נסכום את כל אחד מהוקטורים האלה, בעצם נקבל שוב את הקרוס ספקטרום שחישבנו בשיטה הקודמת. בעצם אנחנו עוצרים שלב אחד לפני הסכימה של הIFFT.

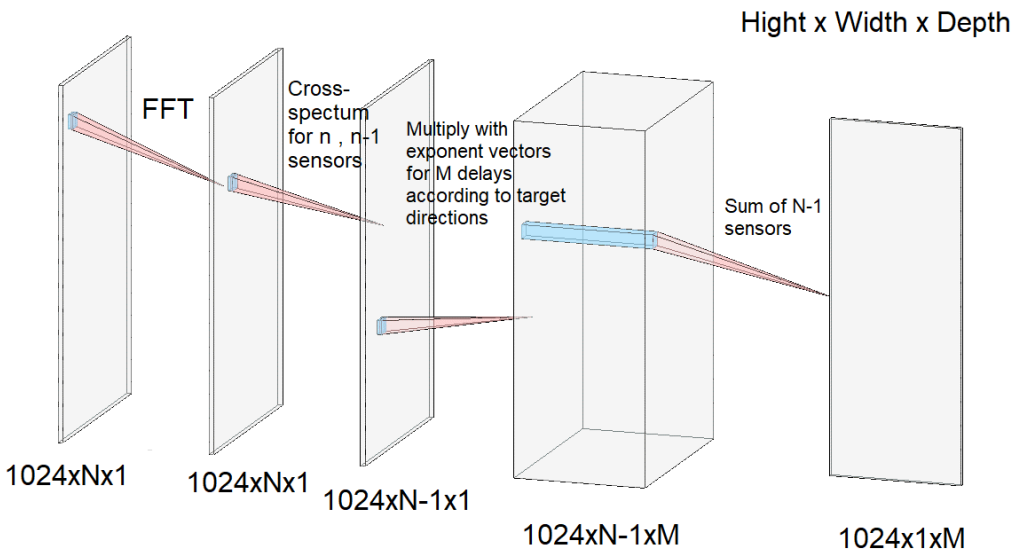
המטרה בשיטה זו, היא לקבל הרכב תדרי של כל אחד מהכיוונים, וכך להזין לרשת תמונת יותר מלאה של כל אחד מ$M$ הכיוונים. הרשת תוכל לזהות למשל תדרים שבהם יש אות דיבור, ותדרים שהם רועשים, וכך להשתמש בתדרי הדיבור, ולהתעלם מתדרי הרעש.

במידה ויש לנו מערך של יותר משני מיקרופונים, נעשה פעולה זהה למה שעשינו בשיטה הקודמת – נבחר וקטור רפרנס, ונחשב את M הוקטורים שלנו עבור $N-1$ זוגות. לבסוף נסכום על כל הזוגות, כך שנרד שוב לM וקטורים. כך אנו שוב לא מוגבלים למספר המיקרופונים או לצורתם, ובכל מצב אנו מקבלים מטריצה אחת עבור כל המערכת.

לסיכום, בשיטה זו אנחנו מזינים לרשת מטריצה בגודל $(F, M)$. כאשר F הוא מספר פסי התדר שלנו, וM הוא מספר הכיוונים.

כיוון שהאות הוא ממשי, יש סימטריה בתדר. ולכן במקום לקחת 1024 תדרים, ניקח רק 513 תדרים (התדרים החיוביים ותדר 0).

ולבסוף נקבל מטריצה בגודל: (513, M).



# חלק ג - עיבוד ברשת נוירונים

## יצירת הdata base

בשלב זה בחרנו שיטה שתאפשר לנו לייצר הקלטות עבור מערך מיקרופונים מסויים, כיוון דובר, מימדי חדר, הדהוד ורעש. מכיוון שמדובר בכמויות רבות של מידע שיקח זמן להקליט בהקלטות אמת, השתמשנו בהקלטות נקיות ובחרנו להשתמש ב-RIR generator כפי שנרחיב בהמשך.

### שימוש ב-RIR (Room Impulse Response) generator

למעשה, RIR הוא פונקציית התמסורת בין מקור הקול לבין מערך המיקרופונים. כך שקונבולוציה בין הקלטה נקיה לבין פונקצית תמסורת זו תתן לנו, באופן אידיאלי, את תוצאת ההקלטה כאילו הוקלטה בסביבה מסוימת שהגדרנו, על ידי מערך מיקרופונים שנבחר.

השתמשנו בפונקציה שתייצר לנו את הRIR, הפרמטרים שפונקציה זו מקבלת: מהירות הקול, תדר הדגימה של האות, מיקום חיישן, מיקום מקור, זמן ההדהוד וכמות הדגימות במוצא.

בנוסף, את ההקלטות ייצרנו באופן רנדומלי תוך קביעת פרמטרים בטווחים הבאים:

- זווית: 0 - 359 (פילוג אחיד בדיד)
- מספר חיישנים: 4 - 12 (פילוג אחיד בדיד)
- רדיוס (של מערך המיקורופונים המעגלי): מס' שלם 0 - 3 ס"מ (פילוג אחיד בדיד)
- הדהוד: 0.2-0.8 (פילוג גאוסי סביב 0.3)
- SNR - בין dB25 ל-dB0 (שיטת ההוספה מפורטת בהמשך)

בצורה זו, יצרנו 10,000 הקלטות שונות, וחילקנו אותם באופן הבא:

| Type | Samples |
|---|---|
| Training | 9000 |
| Validation | 900 |
| Testing | 100 |

### הוספת רעש

בחרנו בשתי שיטות להוספת רעש, רעש לבן (עבור הקלטות הסימולציה שייצרנו) ורעש חדר (עבור הקלטות אמת).

הוספת רעש לבן - עבור הקלטה signal ייצרנו וקטור עם רעש לבן (בעל תוחלת 0 ושונות 1) ונרמלנו אותו כך שהיחס אות לרעש יהיה לפי בחירתנו:

$$SNR = 10\ \log_{10}\frac{\left(std(signal)\right)^2}{\alpha^2}$$

$$\alpha = \frac{std(signal)}{\sqrt{10^{\frac{SNR}{10}}}}$$

הכפלת $\alpha$ ברעש הלבן תנרמל את הרעש כך שנקבל את ה-SNR הרצוי.

הוספת רעש חדר – באופן דומה עבור הקלטות אמת, הקלטנו את החדר ללא דיבור והגברנו את רעשי החדר (noise) על מנת להגיע ל-SNR רצוי, לפי החישוב הבא:

$$SNR = 10\ \log_{10}\frac{\left(std(signal)\right)^2}{\alpha^2 \cdot \left(std(noise)\right)^2}$$

$$\alpha = \frac{1}{\sqrt{10^{\frac{SNR}{10}}}} \cdot \frac{std(signal)}{std(noise)}$$

גם במקרה זה, הכפלת $\alpha$ ברעש החדר תנרמל את הרעש כך שנקבל את ה-SNR הרצוי.

קטע קוד למימוש:

```
def add_noise(signal, SNR_ratio ,mics_num, mic_locations, VAD_array,  Noise = 'WhiteNoise'):
    """
    add noise to the signals
    """

    if Noise == 'WhiteNoise': ## Add white noise by SNR

        # array_std = np.std(signal, axis = 0)
        array_std = np.std(signal[VAD_array], axis = 0)

        # normal noise std
        alpha = array_std / np.sqrt(10**(SNR_ratio/10))
        alpha = np.resize(alpha, (signal.shape))

        # matrix for normalize whiteNoise
        array_WhiteNoise = np.random.normal(0, 1,(signal.shape[0], mics_num))
        array_WhiteNoise_normalized = np.multiply(alpha, array_WhiteNoise)

        # add noise to signal
        array_signal_with_noise =  signal + array_WhiteNoise_normalized

    return array_signal_with_noise
```

## יצירת הfeatures

לאחר שיצרנו את ההקלטות, חישבנו מתוכם את הfeatures, כפי שהסברנו בחלק הקודם – חלק ב - עיבוד אנליטי מקדים לרשת. כפי שציינו, יצרנו שלוש סוגים שונים של features, על בסיס שלושה סוגים שונים של pre-processes. כל feature נוצר מתוך frame באורך של 1024 דגימות. כך שבעצם יש לנו זיהוי בזמן אמת של כיוון הדובר. כל 1024 דגימות שמגיעות – עוברות pre-process מסוים, ולאחר מכן נכנסות לרשת. והרשת מוציאה את הזווית.

נזכיר כאן את המימדים של ה features לפי סוגי העיבוד המקדים השונים:

| Type | Samples |
|---|---|
| Pre-process 1 | Nx1024 |
| Pre-process 2 | 1x36 |
| Pre-process 3 | 513x36 |

אלה מימדי הכניסה לרשת (N הכוונה למספר המיקרופונים במערך).

## הlabels ופונקציית הVAD

את הלייבלים יצרנו באופן הבא:

כפי שראינו למעלה, יצרנו features עבור כל frame, באורך של 1024 דגימות. עבור כל frame כזה, יצרנו שני labels, אחד שבו כתוב הזווית שממנה הגיע הדיבור, והשני זה VAD – האם יש דיבור בframe הנוכחי.

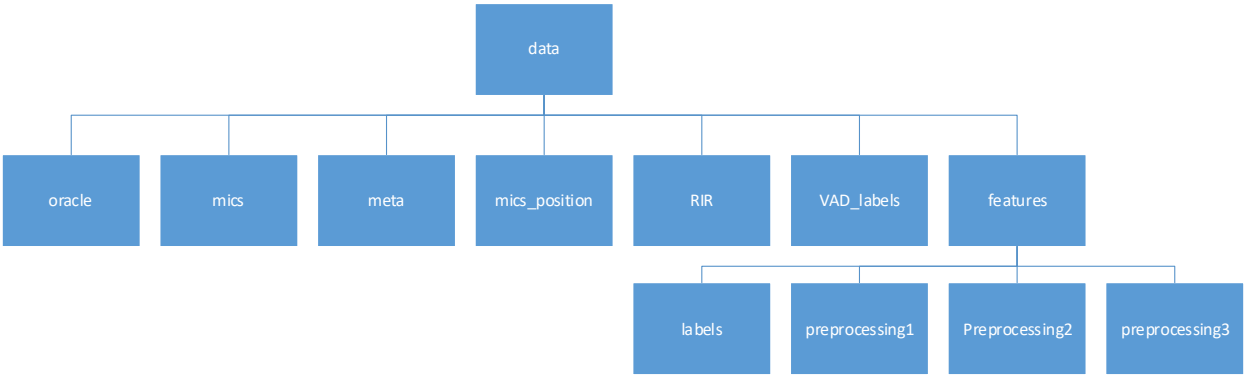
לVAD יש שני מטרות:

א. בframes שבהם אין דיבור, לא נרצה שהרשת תתאמן עליהם. במקרה כזה נראה שפונקציית הloss תהיה 0. כלומר הרשת לא אמורה לזהות כיוון של דיבור בframes שבהם אין דיבור או שהדיבור נמוך מעוצמת סף מסוימת. נסביר בהרחבה בהסבר על פונקציית הloss.

ב. יש אפשרות שהרשת תתאמן גם על הVAD – כלומר הרשת תלמד לזהות לבד האם יש דיבור בframe הנוכחי או לא. נציין בהמשך את התוצאות שקיבלנו עבור אימון של זיהוי VAD.

## מבנה התיקיות

על מנת ליצור את כל ההקלטות האלו, ניהלנו מבנה תיקיות מסודר, המתואר בגרף להלן:



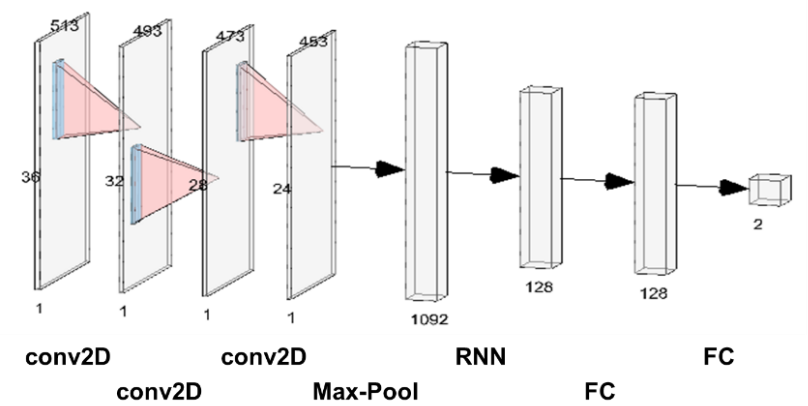

הסבר על התיקיות:

- Oracle: הקלטות נקיות ללא רעשים, שלקחנו מהאינטרנט (מתוך LibriSpeech).
- Mics: הקלטות לאחר שעברו בRIR. כאן כבר יש לנו סימולציה של הקלטה בחדר, מזווית משתנה, ע"י מספר משתנה של מיקרופונים, עם רעש שונה וכו' על פי כל הפרמטרים שכתבנו למעלה.
- Meta: מידע כללי על הפרמטרים של כל אחת מההקלטות (מאיזה הקלטה נקיה היא נלקחה, באיזה זווית, מרחק הדובר וכו').
- Mics_position: טבלה שמתארת את מיקומי כל אחד מהמיקרופונים במערך (x,y,z).
- RIR: כאן שמרנו את פילטר הRIR עצמו (למרות שלא עשינו בו שימוש חוזר).
- VAD_labels: כאן שמרנו מערך שבו כתוב על כל אחד מהדגימות האם יש דיבור או אין דיבור, ע"י זיהוי VAD.
- Features: כאן שמרנו את הfeatures עצמם. עם שלוש סוגים שונים של pre-process. בנוסף יש תיקייה של הlabels, כאשר עבור כל פריים הlabel מכיל שני מספרים – מה הזווית, והאם יש דיבור או אין דיבור בפריים הנוכחי.

## הרשת

### ארכיטקטורת הרשת

עבור pre-process מס' 3, הרשת בנויה באופן הבא:




בשלב הראשון ישנם שלושה רשתות קונבולציה (CNN), על מנת לעבד את המידע של התמונה התדרית. הסיבה שרשת CNN מתאימה למקרה שלנו, היא שאנו רוצים להשתמש בעובדה שיש קשר בין התדרים השונים, ובעיקר בין הכיוונים השונים, והם אמורים להציג תמונה משלימה. כלומר אם למשל יש תבנית מסוימת בכיוון מספר 10 (כלומר בזווית 100), אנו נצפה לראות תבנית דומה בכיוון מספר 11 (זווית 110), כיוון שיש קרבה בין הזוויות. בעצם אנו מנצלים פה את המידע הדו מימדי (כיוון ותדר), כמו בתמונה.

ברשת הקונבולוציה, השתמשנו בגודל kernel של 21*5.

לאחר מכן שלושת רשתות הקונבולוציה ביצענו max-pool, כל מנת להוריד את המימדים לוקטור של 1092.

את הוקטור הזה הכנסנו לרשת GRU (סוג מסוים של RNN). הסיבה שהשתמנו בRNN, היא העובדה שאנו מדברים כאן על סדרה (אות דיבור בזמן), ויש קשר בין הframes השונים. לכן אנו רוצים להשתמש ברכיבי זיכרון ברשת.

לבסוף הוספנו שני שלבי Fully-Connected.

מוצא הרשת הוא וקטור באורך 2, כפי שהסברנו למעלה. מספר אחד מנסה לחזות את הזווית, והשני מנסה לזהות האם יש דיבור או אין דיבור.

נרמלנו את שני המספרים הם בין 0 ל1. כאשר עבור הזווית, המספר 0 מסמן זווית 0, ו1 מסמן זווית 360 (במובן שיש כאן ציקליות והתייחסנו אליה בפונקציית הloss, כפי שנסביר בהמשך). ועבור הVAD, מספר בין 0 ל1, כאשר 0 מסמן שאין דיבור, ו1 שיש דיבור.

עבור pre-process מס' 2, הרשת בנויה באופן זהה, אך נדלג על השלב של רשתות הקונבלוציה (כיוון שאין מידע מטריציוני ב pre-process מס' 2, אלא מתחילים מוקטור). מתחילים מוקטור של 32*1 שנכנס לתוך רשת הRNN. לאחר מכן הרשת בנויה באופן זהה.

קוד למימוש:

```python
class GRU(nn.Module):
    def __init__(self, args, hidden_size=128, num_layers=2, dropout=0., win_len=1024, number_of_direction=36):
        super(GRU,self).__init__()
        self.args = args
        freqs = (win_len // 2 + 1)
        self.num_layers = num_layers
        self.hidden_size = hidden_size
        self.number_of_direction = number_of_direction
        self.num_of_frames = 386
        gru_input_size = 36 if self.args.pre_process_type == "preprocessing2" else 1092
        hidden_size = 20 if self.args.pre_process_type == "preprocessing2" else 128

        if self.args.pre_process_type != "preprocessing2": # default - preprocessing3. model include CNN
            self.cnn1 = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=(21,5), stride=1, padding=0)
            self.cnn2 = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=(21,5), stride=1, padding=0)
            self.cnn3 = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=(21,5), stride=1, padding=0)
            self.maxpool = nn.MaxPool2d((5, 2), padding=(2, 0))

        self.time_rnn = nn.GRU(gru_input_size, hidden_size, num_layers, batch_first=True, dropout=dropout)

        self.time_fc1 = nn.Linear(hidden_size, hidden_size)
        self.time_fc2 = nn.Linear(hidden_size, 2)
        self.activation = nn.Sigmoid()

    def forward(self, x, states=None):
        """
        Args:
            x: [B, T, F, C] - mic & num of direction
            states (tuple): each [num_layers, B, hidden]
        """
        if self.args.pre_process_type != "preprocessing2": # default - preprocessing3

            B, T, F, C = x.shape
            if states is None:
                states = x.new_zeros((self.num_layers, B, self.hidden_size))

            x = x.reshape(B*T, 1, F, C)  # [B*T, F, C]
            x = self.cnn1(x)
            x = self.cnn2(x)
            x = self.cnn3(x)
            x = self.maxpool(x)

            x = x.reshape(B, T, -1)  # [B, T, F*C]

        time_rnn_out, states = self.time_rnn(x, states)  # [B, T, H]

        out = self.time_fc1(time_rnn_out)  # [B, T, F]
        out = torch.relu(out)
        out = self.time_fc2(out)
        mask = self.activation(out)

        return mask
```

## פונקציית ה-loss

אחד האתגרים בתכנון הפונקציה היה כיצד ניתן ללמוד את הזווית רק מזמנים בהם יש דיבור, מכיוון שלמידה מזמנים שאין בהם דיבור לא תורמת מידע אמיתי ולכן יכולה לגרום לביצועים גרועים יותר של הרשת. הפתרון לאתגר זה היה להגדיר וגם ללמוד VAD – voice activity detection ולהכפיל אותו בווקטור ה-loss של הזווית, כך שבמקרה שיש דיבור הרשת תלמד ובמקרה שאין דיבור, VAD יאפס את ה-loss וכך גם הרשת לא תלמד מאזורים אלו.

לכן, פונקציית ה-loss שהגדרנו תוכננה כך שתאפשר למידה של VAD ושל זווית הדובר, ובחרנו hyper parameter בשם "angle_loss_ratio" שמטרתו לתת משקל משמעותי יותר לאחד מהפרמטרים הנלמדים.

בנוסף, נרמלנו את גודל הזווית התיוג על ידי חילוק ב-360, על מנת לשמור על ערכי הVAD והזווית בין 0 ל-1, מכיוון שעל פי התאוריה עדיף לאמן את הרשת על ערכים בסדרי גודל זהים.

מכיוון שערך זווית הינו ציקלי, יש צורך להתחשב בכך באמצעות בחירת מינימום בין מרחק זווית הפרדיקציה לזווית התיוג, לבין מרחק זווית הפרדיקציה + 360 ובין זווית התיוג.

לאחר כל שיקולים אלו, הגדרנו את פונק' ה-loss באופן הבא:

$$angle\ pred = \min(|angle\ pred - angle\ label|, 1 - |angle\ pred - angle\ label|) + angle\ label$$

$$angle\ loss = criterion(VAD\ label \cdot angle\ pred, VAD\ label \cdot angle\ label)$$

$$VAD\ loss = criterion(VAD\ pred, VAD\ Label)$$

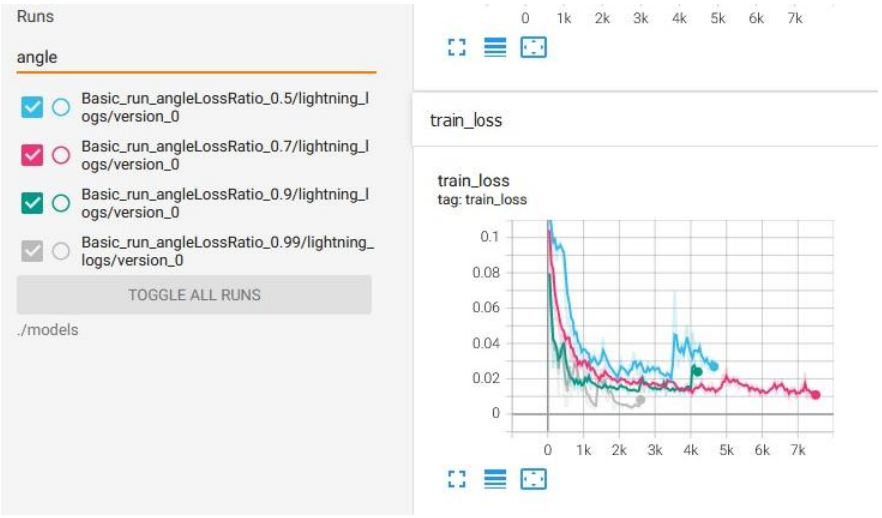
$$Loss = (angle\ loss\ ratio) \cdot angle\ loss + (1 - angle\ loss\ ratio) \cdot VAD\ loss$$

קוד למימוש:

```
def Loss(self,angleVAD,pred):

    angle_pred = pred[:,:386,0]
    VAD_pred = pred[:,:386,1]
    angle_label = (1/360)*angleVAD[:,:386,0]
    VAD_label = angleVAD[:,:386,1]

    # pred angle: take the minimum between the angle to (360 + angle). take the closet to the label.
    angle_pred = torch.minimum(abs(angle_pred-angle_label), 1 - abs(angle_pred-angle_label)) + angle_label
    angle_loss = self.criterion(VAD_label * angle_pred, VAD_label * angle_label)

    VAD_loss = self.criterion(VAD_pred, VAD_label)

    loss = self.args.angle_loss_ratio * angle_loss + (1 - self.args.angle_loss_ratio) * VAD_loss
    return loss
```

## התכנסות הרשת

נביא כאן תמונה של התכנסות הרשת, עבור מספר רשתות עם פרמטרים שונים:




ניתן לראות שכאן ניסינו לשחק עם ה hyper parameter שנקרא "angle_loss_ratio" שעליו דיברנו לעיל. כפי שנציין בהמשך, ניתן לראות כאן התכנסות של הרשת, ככל שאנו נותנים משקל יותר גדול לזיהוי הזווית, ופחות משקל לזיהוי הVAD. כלומר לרשת יותר קשה ללמוד את הזווית והVAD במקביל. ונרחיב על כך בהמשך.

# ניסויים – הקלטות אמת

נציין שהרשת לא התאמה על הקלטות אמת כלל, אלא רק על הקלטות ה RIR שיצרנו. אך את התוצאות ניתחנו על הקלטות אמת. ערכנו רשימה של הקלטות, על פי פרמטרים שונים. כל הקלטה באורך 30 שניות.

השתמשנו במערך מעגלי, בעל 6 מיקרופונים, ברדיוס 5 ס"מ, שקיבלנו מעופר. הרשת לא התאמנה על מערך מהסוג הזה, ולא התנסתה עליו לפני הtest שביצענו.

על מנת למדוד את ההדהוד בכל אחד מהמיקומים השתמשנו באפליקציה שמודדת הדהוד.

על מנת למדוד את הSNR, הלקטנו את הרעש בחדר ללא דיבור. ומדדנו את העוצמה של הדיבור ביחס לעוצמה של הרעש הבסיסי בחדר.

על מנת ליצור הקלטות רועשות – הקלטנו רעש של מזגן, והוספנו אותו להקלטה בעוצמות שונות, על מנת ליצור SNR שונים. הסברנו על כך בהרחבה לעיל.

רשימת ההקלטות שערכנו:

| File name | Mics num | Mics radius [cm] | Angle | Source distance [m] | Rev time | SNR ratio [dB] | Description | Location | Date |
|---|---|---|---|---|---|---|---|---|---|
| **file_0** | 6 | 5 | 0 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_1** | 6 | 5 | 30 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_2** | 6 | 5 | 60 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_3** | 6 | 5 | 90 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_4** | 6 | 5 | 120 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_5** | 6 | 5 | 150 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_6** | 6 | 5 | 180 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_7** | 6 | 5 | 210 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_8** | 6 | 5 | 240 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_9** | 6 | 5 | 270 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_10** | 6 | 5 | 300 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_11** | 6 | 5 | 330 | 1.5 | 0.49 | 18 | nominal | מעבדה ננו | יוני-2 |
| **file_12** | 6 | 5 | 90 | 2 | 0.49 | 18 | Source distance | מעבדה ננו | יוני-2 |
| **file_13** | 6 | 5 | 90 | 1 | 0.49 | 18 | Source distance | מעבדה ננו | יוני-2 |
| **file_14** | 6 | 5 | 90 | 1.5 | 0.83 | 18 | Reverberation | לובי הנדסה | יוני-23 |
| **file_15** | 6 | 5 | 90 | 1.5 | 0.16 | 18 | Reverberation | מחוץ לבניין הנדסה | יוני-23 |
| **file_16** | 6 | 5 | 90 | 1.5 | 0.49 | 0 | Noise | מעבדה ננו | יוני-2 |
| **file_17** | 6 | 5 | 90 | 1.5 | 0.49 | 3 | Noise | מעבדה ננו | יוני-2 |
| **file_18** | 6 | 5 | 90 | 1.5 | 0.49 | 6 | Noise | מעבדה ננו | יוני-2 |
| **file_19** | 6 | 5 | 90 | 1.5 | 0.49 | 10 | Noise | מעבדה ננו | יוני-2 |
| **file_20** | 6 | 5 | 90 | 1.5 | 0.49 | 15 | Noise | מעבדה ננו | יוני-2 |
| **file_21** | 6 | 5 | 90 | 1.5 | 0.49 | 18 | Noise | מעבדה ננו | יוני-2 |
| **Quiet room** | | | | | | | | מעבדה ננו | יוני-2 |
| **Air-condition noise room** | | | | | | | | מעבדה ננו | יוני-2 |

סימנו בצהוב את הפרמטר שרצינו לבדוק בכל הקלטה. את הרשת בדקנו על 21 הקלטות אלו.

# תוצאות

ראשית כל, נציין מספר הערות חשובות:

א. הרשת מעולם לא התאמנה על הקלטות אמת, אלא רק על הקלטות שיצרנו בעזרת RIR-generator. אך התוצאות שנציג כאן, הן עבור הקלטות אמת. כך שהמשימה אמורה להיות מאתגרת יותר (כיוון שהרשת לא התאמנה על הקלטות אמיתיות אנו יכולים לצפות שיהיה לה קושי לחזות זווית בהקלטות אלו). למרות זאת קיבלנו תוצאות טובות.

ב. כמובן שהרשת לא התאמנה על מערך המיקרופונים הספציפי שהקלטנו איתו, אלא על אינספור מערכים שונים שנוצרו באופן רנדומלי.

ג. כיוון שאלו הקלטות אמת, ברור לנו שיש לנו גם שגיאה כלשהי בlabels שרשמנו. כלומר כאשר הקלטנו הקלטה מזווית כלשהי, יתכן והזווית לא מדויקת ויש סטיה של כמה מעלות לכיוון מסוים.

ד. כיוון שאנו משתמשים ברשת מסוג GRU, יש לרשת זמן התכנסות. לכן, אנו מצפים שבפריימים הראשונים עד ההתכנסות תהיה לרשת שגיאה גדולה יותר. לעיתים הגרפים שנציג הם לאחר התכנסות הרשת, אך במקרים מסוימים הצגנו כאן גרפים שמתחילים עוד לפני התכנסות הרשת, כדי להדגיש את התופעה. ניתן לשים לב לשיפור שנעשה לאורך ציר הזמן.

## תוצאות הרשת על הקלטת אמת והקלטות סימולציה - זיהוי הזווית ב pre-process מספר 3

כאן השתמשנו בשיטת עיבוד השלישית שמופיעה בחלק ב': תמונה תדרית עבור כל אחד מהכיוונים.

ראשית כל נערוך השוואה של שגיאת mean absolute error על הקלטות שיצרנו ע"י הRIR לבין הקלטות אמת.

| סוג הקלטה | MAE [deg] |
|---|---|
| הקלטות סימולציה | 5.4 |
| הקלטות אמת | 10.8 |

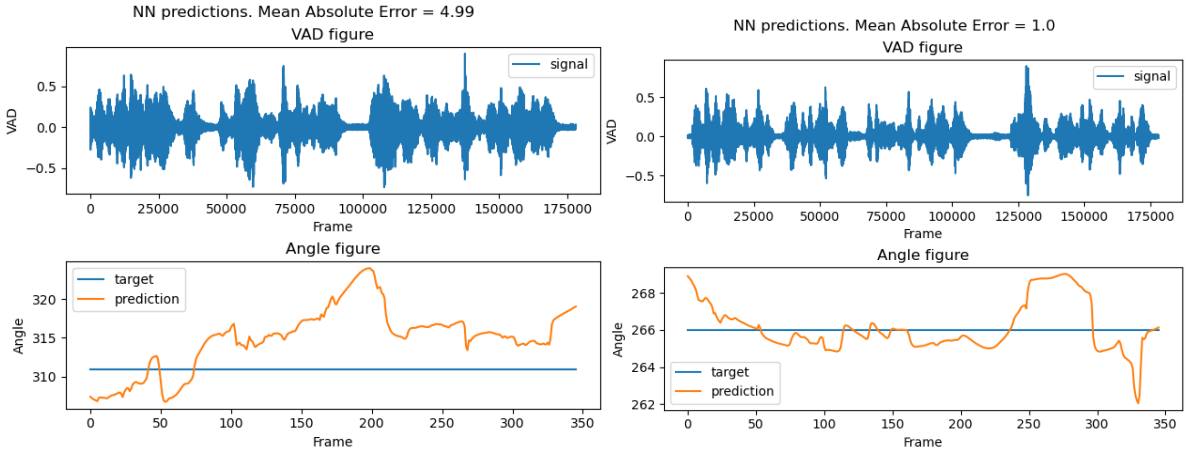
אנו יכולים להסיק מכאן, שהרשת אכן נותנת תוצאות טובות יותר (פי שניים), עבור הקלטות סימלוציה מאשר על הקלטות אמת. זוהי תוצאה הגיונית, ויש לה שתי סיבות מרכזיות:

א. הרשת התאמנה על הקלטות סימולציה, ולכן נותנת תוצאות טובות יותר עליהן.

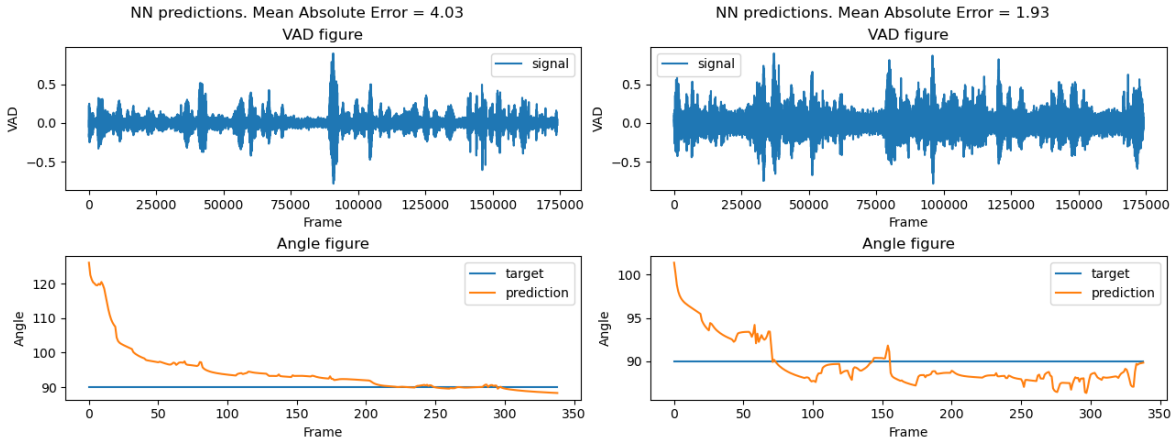
ב. בהקלטות אמת ייתכן שיש לנו שגיאת מדידה של כמה מעלות ימינה או שמאלה, ולכן יכול להיות שהlabel שאנחנו כתבנו הוא לא מדויק לגמרי.

נביא כאן כמה דוגמאות לגרפים שקיבלנו.

עבור הקלטות סימולציה:




עבור הקלטות אמת (מימין עם רעש של 6dB, ומשמאל עם רעש של 10dB):




הערה: בגרפים התחתונים ניתן לראות ביצד הרשת מתכנסת לאורך הזמן. כיוון שזאת רשת GRU, אנו מצפים שיהיה לה זמן התכנסות, ולכן זו תוצאה הגיונית.

כעת נערוך השוואה של שגיאת mean absolute error כפונקציה של SNR, הדהוד ומספר חיישנים. לנוחות ההשוואה, **ההקלטה הנומינלית** מודגשת.

הדהוד:

| Rev time [s] | MAE [deg] |
|---|---|
| **0.49** | **4.3** |
| 0.86 | 2.2 |
| 0.16 | 3.8 |

בסביבה של הדהוד גבוה ונמוך מהנומינלי ניתן לראות שהשגיאה נמוכה מאוד, לכן אנו מסיקים שהרשת נותנת תוצאות טובות גם תחת הדהוד.

מספר מיקרופונים:

| Mics num | MAE [deg] |
|---|---|
| **6** | **4.3** |
| 5 | 6.52 |
| 4 | 3.1 |
| 3 | 2.38 |

גם כשמפחיתים את כמות המיקרופונים ניתן לראות שהרשת מספקת שגיאה נמוכה, לכן אנחנו מסיקים שלרשת יש יכולת חיזוי טובה גם עבור שינוי מערך מיקרופונים.

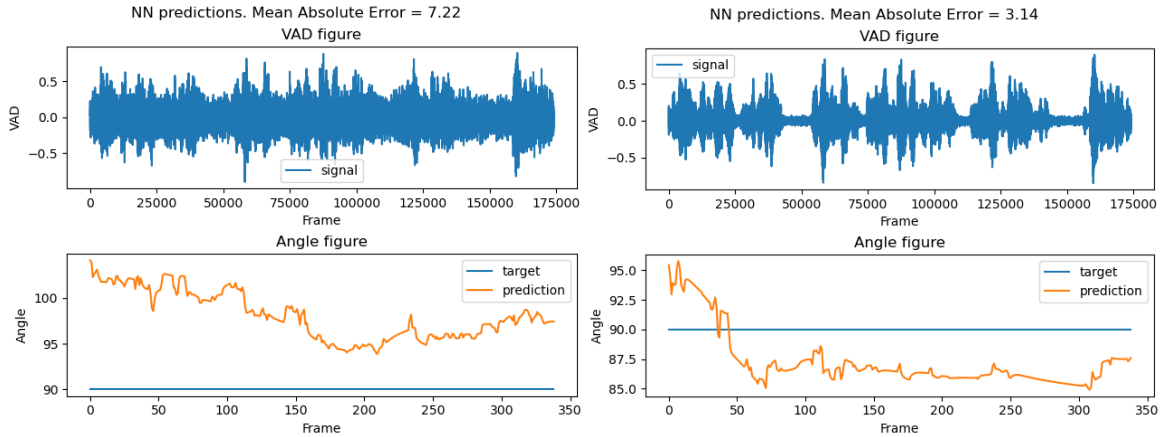
רעשים שונים:

| SNR [dB] | MAE [deg] |
|---|---|
| **24** | **4.3** |
| 20 | 3.52 |
| 15 | 3.14 |
| 10 | 2.22 |
| 6 | 1.4 |
| 3 | 2.35 |
| 0 | 7.22 |

בהוספת רעש אנו רואים שהרשת עדיין מספקת חיזוי מדויק, רק במקרה של 0dB הרשת נותנת תוצאה פחות טובה מההקלטה הנומינלית, אך עדיין תוצאה טובה. לכן אנו מסיקים שהרשת חסינה לרעש משמעותי.

בסופו של דבר, עבור 115 הקלטות אמת בSNR שונים (כמו בטבלה לעיל) קיבלנו MAE ממוצע של 10.8 מעלות. זוהי תוצאה טובה, במיוחד בהתחשב בעובדה, שיש גם שגיאת מדידה בזווית.

נציג שוב כמה גרפים נבחרים: הקלטה זהה בתוספת רעש בעוצמה שונה – מימין 15dB ומשמאל 0dB:



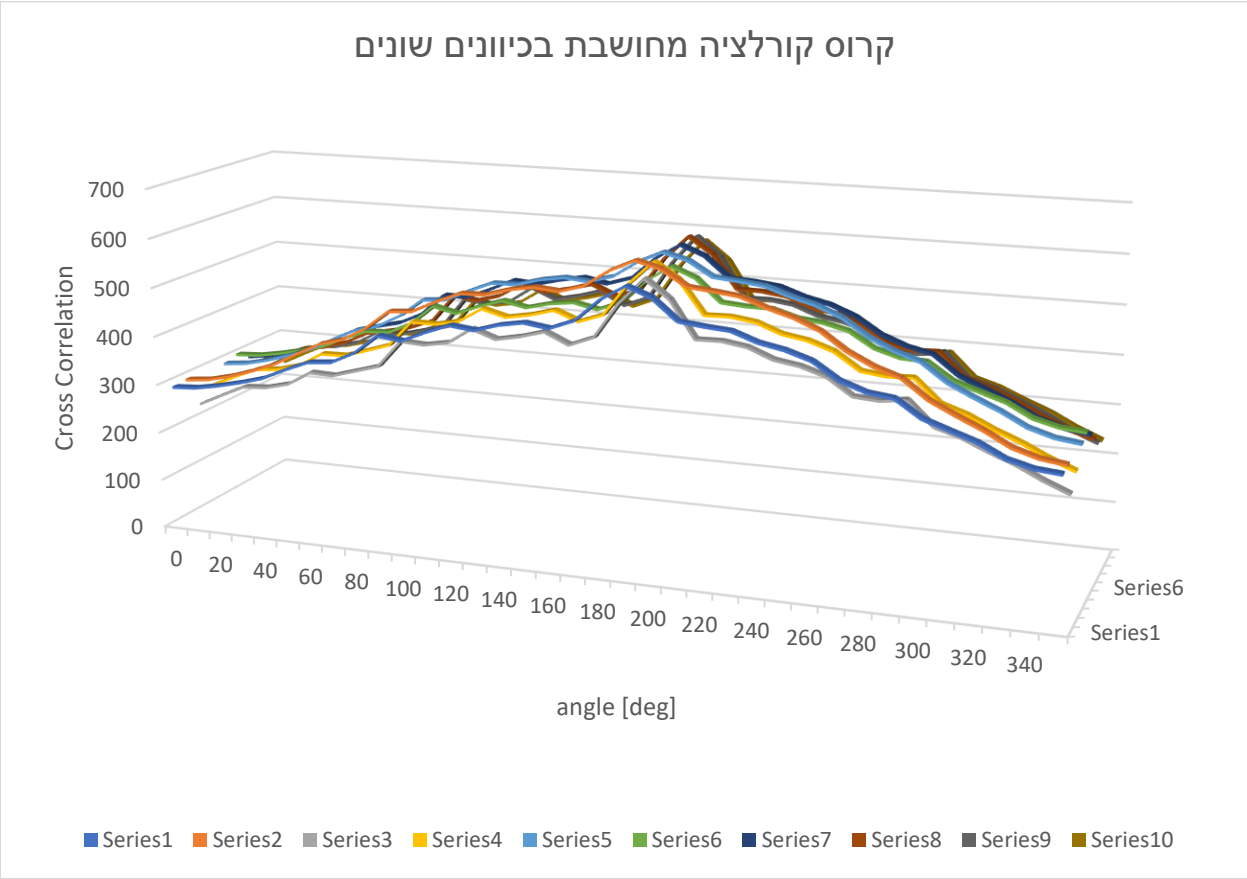
## דוגמא לחישוב הזווית בצורה אנליטית בעזרת קרוס קורלציה

נציג כאן דוגמא לחישוב אנליטי בעזרת הקרוס קורלציה. כאן השתמשנו בשיטה השניה שמופיעה בחלק א': חישוב קרוס ספקטרום באמצעות מעבר לתדר ואז להתמיר חזרה לזמן.

בדוגמא זו יש 11 מיקרופונים, כיוון הדיבור מגיע מזווית של 191 מעלות. רדיוס המערך: 0.07m. מרחק הדובר: 1m. זמן ההדהוד: 0.5. SNR: 18.35.

ניקח כאן 10 frames מתוך האות. כל frame הוא של 1024 דגימות. לכל פריים כזה, חישבנו קרוס קורלציה, ב36 נקודות (כל 10 מעלות), לפי השיטה שכתבנו למעלה (כמו בעיבוד המקדים מספר 2: עבור כל שתי מיקרופונים – מעבר לתדר, חישוב קרוס ספקטרום, מעבר חזרה לזמן בזוויות ספיצפיות. ולבסוף סכימה על כל המיקרופונים).

בגרף זה ניתן לראות בצורה מובהקת את הפיק שנוצר בקרוס קורלציה, בדיוק בכיוון שממנו מגיע האות. ואכן הפיק מתקבל באיזור הזווית 190.




## תוצאות עבור חיזוי אנליטי בעזרת מקסימום של הקרוס קורלציה

כעת, לאחר שראינו שהרשת מציגה ביצועים טובים על הקלטות האמת, למרות שהתאמנה על הקלטות מסימולציית RIR בלבד. בדקנו את יכולת הדיוק של החישוב האנליטי (מקסימום על הקרוס קורלציה) לעומת יכולת הדיוק של הרשת. על מנת לחשב את הקרוס ספקטרום השתמשנו שוב בשיטה השניה שמופיעה בחלק א': לקיחת מקסימום על הקרוס קורלציה, שמחושבת מתוך התמרה הופכית של הקרוס ספקטרום, בזמנים לא שלמים.

כמובן שאת הtest הזה ביצענו על אותם הקלטות שביצענו את הtest על הרשת.

ראשית כל ביצענו את החישוב האנליטי (מקסימום על הקרוס קורלציה) על הקלטות הסימולציה. קיבלנו תוצאה של ממוצע שגיאה של 8.26 מעלות שגיאה (לעומת 5.4 ע"י הרשת).

לאחר מכן בדקנו את החישוב האנליטי על הקלטות אמת, וקיבלנו תוצאה של 15.01 מעלות שגיאה (לעומת 10.8 ע"י הרשת).

סיכום התוצאות:

| סוג הקלטה | Neural Network MAE [deg] | Max Cross-Correlation MAE [deg] |
|---|---|---|
| הקלטות סימולציה | 5.4 | 8.26 |
| הקלטות אמת | 10.8 | 15.01 |

כלומר הרשת אכן מביאה תוצאות טובות יותר מתוצאות החישוב האנליטי. בנוסף נציין, ששמנו לב שעבור החישוב האנליטי, השגיאה גדלה ככל שההדהוד גדל, וככל שהSNR קטן. ואכן כך ציפינו – עבור SNR גבוה או עבור הדהוד נמוך, ציפינו שהחישוב האנליטי ייתן תוצאות טובות. אך כאשר ההקלטות יותר קשות מבחינת הדהוד או רעש, לחישוב האנליטי קשה לתת תוצאות טובות.

בכל אופן, השיפור לא מאד משמעותי (5 מעלות עבור הקלטות אמת), אך אנו משערים שאם היינו עורכים הרבה הקלטות אמת, עם הדהוד מאד גבוה, היינו מקבלים תוצאות אפילו יותר קיצוניות. והחישוב האנליטי היה נותן תוצאות הרבה פחות טובות. רוב הקלטות האמת שיצרנו היו עם הדהוד נמוך יחסית.

## זיהוי VAD

כפי שכתבנו לעיל, הרשת שיצרנו יכולה גם להתאמן על זיהוי VAD. אמנם VAD ניתן לזיהוי באופן פשוט יחסית בצורה אנליטית, אך בכל אופן ניסינו להיעזר ברשת עבור הדבר הזה.

כאשר ניסינו לזהות את הVAD, הפרמטר החשוב הוא היחס בפונקציית הloss, בין העונש שאנחנו נותנים על אי זיהוי VAD לבין העונש שאנחנו נותנים על שגיאה בזווית. שיחקנו עם הפרמטרים השונים, ונציג כאן כמה תוצאות שקיבלנו עבור פרמטרים שונים.

באופן כללי נאמר, שככל שביקשנו מהרשת לזהות VAD, כך היא התקשתה יותר בזיהוי הזווית. ולכן בתוצאות למעלה, הצגנו רשתות שהתאמנו על זיהוי זווית בלבד ללא זיהוי VAD. ובנוסף שמנו לב שיש צורך לתת משקל יותר גדול לזיהוי הזווית מאשר לזיהוי VAD, ההגיון בכך הוא לזווית יש 360 ערכים אפשריים ול-VAD רק שניים, לכן מלאכת הסיווג לזווית תדרוש יותר פרמטרים.

כעת נציג את ההשוואות השונות עבור רשת שלומדת גם את זיהוי הVAD ולא רק את הזווית. נשים לב להבדלים בין תוצאות אלו לבין הרשת שמשימתה ללמוד את הזווית בלבד ללא זיהוי הVAD.

נציג שגיאת mean absolute error כפונקציה של SNR, הדהוד ומספר חיישנים. לנוחות ההשוואה, **ההקלטה הנומינלית** מודגשת:

| Rev time [s] | MAE [deg] |
|---|---|
| **0.49** | **13.26** |
| 0.86 | 31.14 |
| 0.16 | 3.75 |

בסביבה של הדהוד גבוה ניתן לראות שהשגיאה גדלה מאוד, לכן אנו מסיקים שהרשת אינה חסינה להדהוד.

נציג את התוצאות עבור מספר מיקרופונים משתנה:

| Mics num | MAE [deg] |
|---|---|
| **6** | **13.26** |
| 5 | 32.53 |
| 4 | 67.02 |
| 3 | 94.62 |

כשמפחיתים את כמות המיקרופונים ניתן לראות ששגיאת הרשת גדלה, לכן אנחנו מסיקים שיכולת החיזוי נפגעת עבור שינוי מערך מיקרופונים.

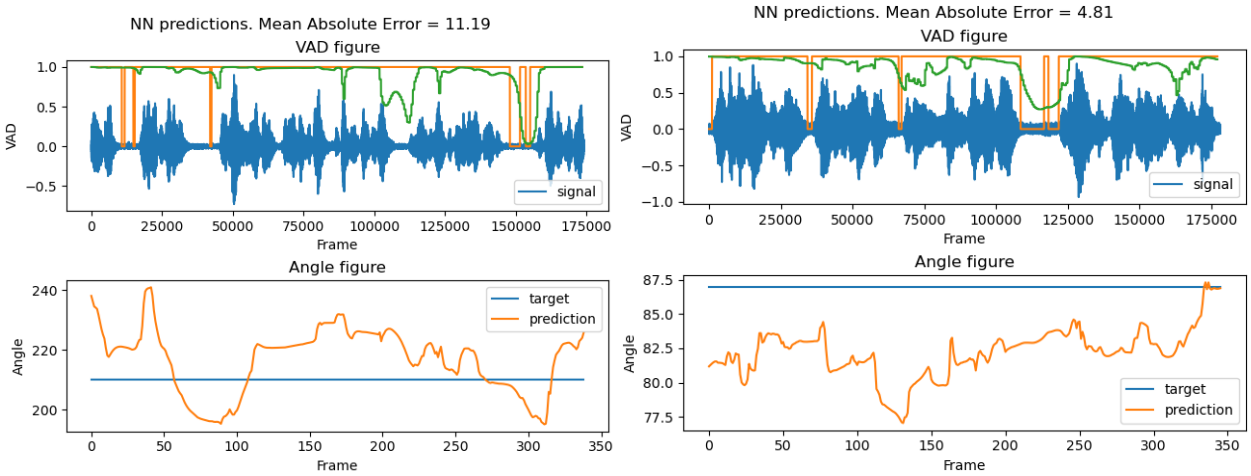
נציג את התוצאות עבור רעשים שונים:

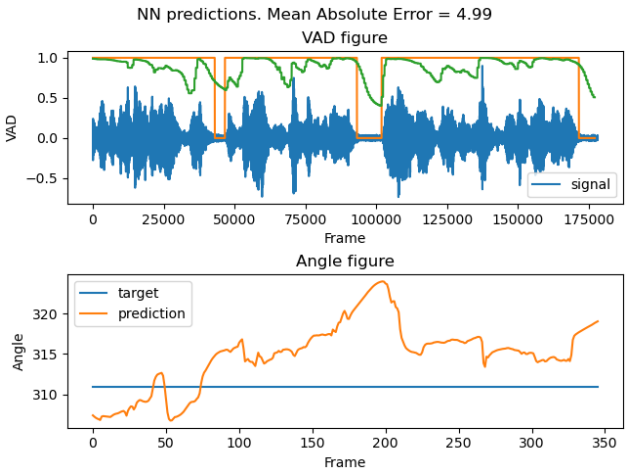
| SNR [dB] | MAE [deg] |
|---|---|
| **24** | **13.26** |
| 20 | 13.31 |
| 15 | 13.37 |
| 10 | 13.47 |
| 6 | 11.46 |
| 3 | 10.36 |
| 0 | 12.27 |

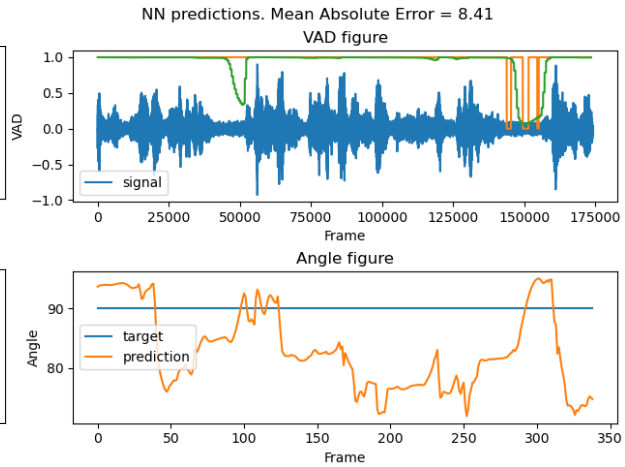
בהוספת רעש אנו רואים שהרשת עדיין מספקת חיזוי מדויק, רק מעל 6dB הרשת נותנת תוצאה פחות טובה מההקלטה הנומינלית, אך עדיין תוצאה טובה. לכן אנו מסיקים שהרשת חסינה לרעש משמעותי. אך עם זאת, תוצאות השגיאה גדלו ברשת שלומדת גם VAD.

עבור 115 הקלטות בSNR שונים (כמו בטבלה לעיל) קיבלנו MAE ממוצע של 18.3 מעלות, לעומת 10.8 מעלות כאשר לא לומדים את ה-VAD. בנוסף נציג שתוצאת הVAD אינה מדויקת מספיק ולכן בסך הכל קיבלנו פגיעה בשיערוך הזווית אך ללא תמורה גדולה מבחינת זיהוי VAD. לכן לדעתנו כדאי לקבוע את הVAD בצורה אנליטית, כדי לא לפגוע בביצועים של הרשת על זיהוי הזווית.

נציג גרפים נבחרים עבור רשתות שמזהות גם את VAD:






בחלק העליון של כל אחד מהגרפים ניתן לראות בכחול את אמפליטודת הקלט, בכתום את תווית ה-VAD (0 – אין דיבור , 1 – יש דיבור) ובירוק את חיזוי הרשת. מתוצאות אלו ניתן לראות שלעיתים הרשת מזהה כשאין דיבור ולעיתים לא, זיהינו תלות באורך הקטע השקט – ככל כשאורך הקטע גדול יותר הרשת מצליחה לזהות שאין דיבור. בנוסף ישנם מקרים שהרשת מזהה שאין דיבור למרות שיש דיבור.

# רעיונות להמשך

ניתן לחשוב על מספר שיפורים שניתן לעשות על מנת לשפר את ביצועי הרשת:

- השיפור המרכזי, שאותו ניתן לעשות: כפי שראינו במאמר, ניתן להוסיף עיבוד מקדים לפני הכניסה לרשת. לאחר התמרת הפוריה, ניתן לחלק בפונקציות משקל שונות (במאמר מוצעות מספר פונקציות).
  לדוגמא בשיטת PHAT נכפיל את התמרת הפורייה בפונקציית המשקל הבאה (פונקציה התלויה בתדר):
  $$\Psi_p(f) = \frac{1}{\left|G_{x_1x_2}(f)\right|}$$
  כאשר $G_{x_1x_2}(f)$ היא פונקציית הקרוס ספקטרום של שתי האותות (משני המיקרופונים).
- כאשר יצרנו את ההקלטות בעזרת הRIR, הוספנו להם סוג רעש אחד בלבד: רעש לבן. משימה נוספת שניתן לממש, על מנת לשפר את יכולת למידת הרשת, היא להוסיף סוגי רעשים נוספים:
  - רעש דפיוזי – רעש שמגיע מכל הכיוונים באופן שווה.
  - רעש כיווני – רעש שמגיע מכיוון מסוים.
- ניתן לחשוב על שיפורים שונים לרשת. להוסיף או לשנות את הרשתות הCNN, ולשנות את הפרמטרי הGRU, על מנת לשפר את התוצאה.
- בנוסף, כמו בכל לימוד של רשתות, ניתן להגדיל את הגיוון בdata, כל מנת שהרשת תוכל ללמוד טוב יותר, ולהיות מוכנה יותר על הקלטות אמת.
  כמו כן, אולי ניתן להקליט מספר כלשהו של הקלטות, ולאחר שהרשת מסיימת ללמוד, לתת לה כיוון קטן נוסף בעזרת ההקלטות האמת. זה יכול לשפר את ביצועי הרשת על הקלטות אמת.

# סיכום ומסקנות

הפרויקט שלנו התחלק לשלושה שלבים:

- שלב א': שערוך אנליטי של ה-DOA
- שלב ב': עיבוד אנליטי מקדים לרשת הנוירונים ב-3 שיטות
- שלב ג': מימוש ומיטוב ארכיטקטורת רשת נוירונים בהתבסס על מאמר

לסיכום, ראינו שהרשת שאימנו על הקלטות מסימולציית RIR, הצליחה להתמודד בצורה מאוד טובה עם הקלטות אמת, כאשר ההשוואה הייתה בפרמטרים של SNR, הדהוד ומספר חיישנים. בנוסף, ראינו ש- 3 preprocessing משלב ב' נתן את הביצועים הטובים ביותר כפי שציפינו. נזכיר שמדובר בעיבוד שמאפשר לרשת ללמוד מתוך תמונה תדרית עבור כל אחד מהכיוונים, ללא תלות במספר החיישנים.

לאחר מכן, בחנו את ביצועי הרשת לעומת חישוב אנליטי. ראינו בדומה לתוצאות המאמר, שלרשת יש יתרון לעומת חישוב אנליטי והיא חסינה יותר לאתגרים כגון SNR נמוך, סביבה מהדהדת ומספר חיישנים נמוך. מהשוואה זו, הצלחנו לשחזר את תוצאות המאמר ברשת שלנו.

מסקנה נוספת, היא שעדיף ללמוד את הזווית בלבד ולא לנסות לשלב זיהוי VAD וזווית באותה רשת, מכיוון שזיהוי הזווית וחסינות הרשת נפגעו משמעותית, אך לעומת זאת לא קיבלנו תוצאות טובות בזיהוי הדיבור (VAD).

# ביבליוגרפיה


[1] Charles H. Knapp and G. Clifford Carter, "The Generalized Correlation Method for Estimation of Time Delay", IEEE Transactions on Acoustics, Speech and Signal Processing, Volume:24, Issue : 4, August 1976

[2] Xueliang Zhang, Hao Li and Li Qinghlong, "Online Direction of Arrival Estimation Based on Deep Learning", IEEE ICASSP (International Conference on Acoustics, Speech and Signal Processing), April 2018

[3] https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator