אוניברסיטת בר־אילן
Bar-Ilan University

הפקולטה להנדסה

המעבדה לעיבוד אותות

תש"פ2020,

פרויקט גמר:

# Speech Separation with Utterance-Level Permutation Invariant Training (uPIT)


עמית פנחס Amit Pinhas

יוגב קליין Yogev Klein

מנחים: שלומי חזן.

פרופסור שרון גנות.

# תוכן עניינים


הקדמה ..... 3
מוטיבציה והצגת הבעיה ..... 3
מטרה ..... 4
רקע תיאורטי ..... 5
Short Time Fourier Transform (STFT) ..... 5
Deep Neural Network רשתות נוירונים עמוקות ..... 6
תהליך אימון של רשתות נוירונים ..... 7
Convolutional Neural Network רשתות קונבולוציה ..... 9
U-net ..... 10
פרמול הבעיה ..... 11
מודל הWDO ..... 12
מציאת המסיכות ..... 15
הכנת הData- Base ..... 17
בעיית הפרמוטציה ..... 18
הפתרון: שימוש במנגנון הPIT (Permutation Invariant Training) ..... 19
עבודה עם המערכת בסיום האימון ..... 20
מדד האיכות ..... 21
תוצאות ..... 22
מערכת רב – ערוצית ..... 27
פורמליזציה של הבעיה החדשה ..... 28
בניית הDatabase במודל החדש ..... 31
תוצאות במודל החדש ..... 33
סיכום ..... 34
ביבליוגרפיה ..... 35

# הקדמה

## מוטיבציה והצגת הבעיה

בהמון סיטואציות בחיינו אנו רוצים לשמוע רק דובר אחד (האדם שמדבר אלינו ישירות או בשיחת טלפון) אך בפועל בגלל הסביבה בה אנחנו נמצאים שומעים בפועל מספר דוברים המדברים בו זמנית יחד עם הדובר הרצוי.

אם היינו יכולים להפריד את הדיבור של הדוברים ובעצם "לנקות" את אותם דיבורי רקע , זה היה יכול להוביל לחוויית שמיעה הרבה יותר נעימה ומובנת ולשיחה עניינית וקולחת יותר.

בני אדם יודעים לעשות זאת בצורה מאוד טובה מכיוון שהם נתקלים בהמון סיטואציות כאלה ובאמצעות החושים שלהם כמו ראייה ושמיעה הם מצליחים להתרכז באדם אחד ובמהלך חייהם הם מפתחים את המיומנות הזאת. בנוסף, אנשים שומעים בשני האוזניים כלומר הם קולטים את הדיבור גם בעוצמה שלו וגם בכיוון שלו מה שמאוד מקל על ההפרדה.

אנו נרצה שתהיה מערכת שתוכל לבצע משימה כזאת. יש למשימה זו המון שימושים יום יומיים:

1) שני אנשים שרוצים לתת פקודה לAlexa, וכאשר הם מדברים בו זמנית נרצה שהמערכת תדע להפריד בין שתי הבקשות ולבצע אותם במקביל.

2) מערכת לשיחות ועידה (כמו Zoom) עם מספר משתתפים – נרצה שהמערכת תזהה את הדיבור של כמה אנשים ותוכל להשמיע את הדיבור של כל אחד מהם, כך ששאר האנשים בשיחה יוכלו לשמוע את כל המשתתפים בבירור.

עבור בעיה זו קיימים כל מיני פתרונות :

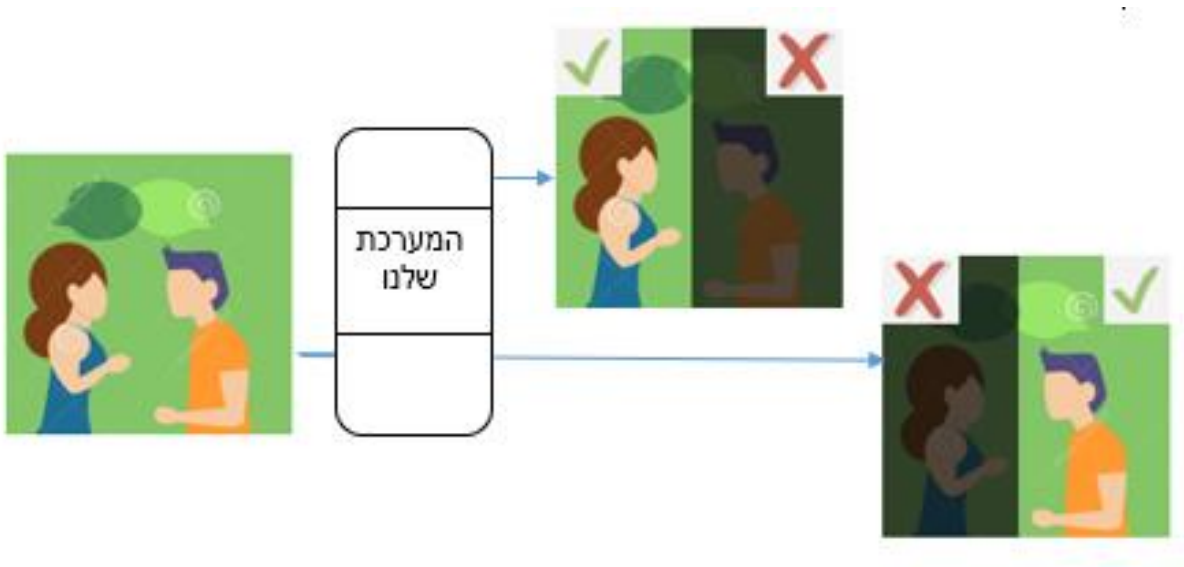
1) חלוקה לקבוצות תדר- זמן ושיוכם לכל דובר לפי פרמטרים של הדיבור (למשל עוצמה ומחזוריות).

2) מודלים הסתברותיים שממדלים את האינטרקציה המורכבת של היעד עם אותות דוברי הרקע.

אבל פתרונות אלו מאוד מוגבלים  וצריך לדעת בשבילם מידע מקדים. בשנים האחרונות נכנסו פתרונות מתוך עולמות המכונות למידה.

## מטרה

הפרויקט שלנו מתרכז בהפרדה של 2 דוברים. מטרתנו היא לתכנן מערכת המקבלת כקלט הקלטה עם שני דוברים חופפים בו זמנית שהוקלטו באמצעות מיקרופון יחיד (single channel) ומפרידה אותה לשתי הקלטות כך שישמעו כל דובר בנפרד.



# רקע תיאורטי

## Short Time Fourier Transform (STFT)

בהמון בעיות שקשורות לעולם עיבוד אותות נרצה להתמקד בהתמרת פוריה ודרכה לעבור ממישור הזמן למישור התדר מכיוון שבמישור התדר יש לא מעט תכונות של האותות שאיתן עובדים שלא באות לידי ביטוי במישור הזמן.

אנו עוסקים באותות דיבור, שאלו אותות שאינם סטציונריים בזמן, לכן לא ניתן לעשות להם התמרת פוריה עם חלון זמן גדול (כי אנו ממצעים על הרבה ערכים ובעצם אנחנו מפספסים שם שינויים קטנים הנגרמים עקב חוסר הסטציונריות ואז חלק מהמידע החשוב הולך לאיבוד), לכן נרצה להשתמש בהתמרת פוריה לזמן קצר (STFT).

$$X_{STFT}(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]v[n-m]e^{-j\omega m}$$

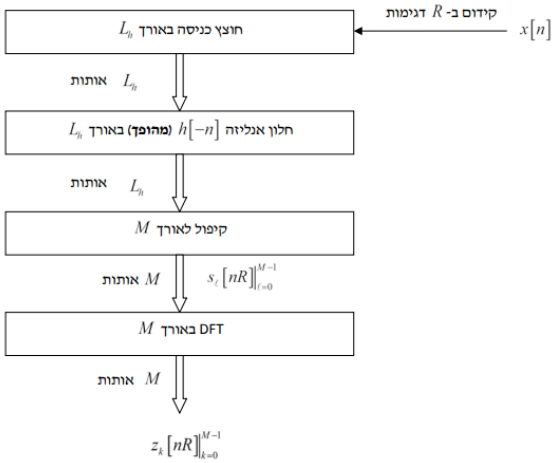
כאשר $x[n]$ הוא האות בזמן ו$v[n]$ הוא חלון בזמן.

כעת, בכל זמן, נבצע *DTFT* עם רזולוציית תדר *M*, על המקטע הנתון. כלומר, $\omega_k = \frac{2\pi k}{M}$ ונקבל בכל זמן את התפלגות העוצמה על-פני התדרים, בעזרת *M* נקודות תדר.

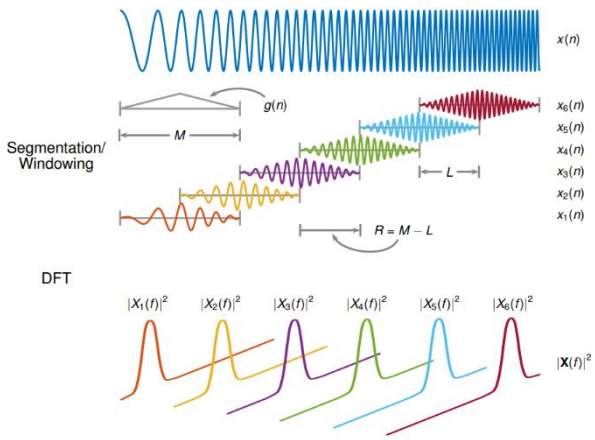
סה"כ נקבל:

$$X_{STFT}(n,k) = \sum_{m=-\infty}^{\infty} x[m]v[n-m]e^{-j\frac{2\pi}{M}nm}$$

בסופו של שלב זה נקבל מטריצה שבה הציר האופקי הוא ציר הזמן, והציר האנכי הוא ציר התדר, שבו כל עמודה היא פריסה של *M* נקודות תדר עבור נקודת זמן מסויימת וכל שורה היא תדר ספציפי שבה רואים את התנהגות התדר במסגרות נקודות הזמן השונות. בעצם $X_{STFT}(n,k)$ ייתן לי נקודת זמן-תדר שבו אראה את עוצמת התדר הספיציפי $\omega_k$ בזמן$n$.




איור 2.1: STFT




בנוסף, קיימת ההתמרה ההפוכה *ISTFT* שהיא מחזירה אותי ממישור התדר למישור הזמן.

## Deep Neural Network רשתות נוירונים עמוקות

רשתות נוירונים עמוקות הוא מודל למידה חישובי המורכב ממספר רב של שכבות, משקולות במעברים בין השכבות (פרמטרי הרשת) ופונקציות לא ליניאריות. כל שכבה מייצגת את המידע במרחב אחר.

בניגוד לאלגוריתמי למידת מכונה שהם משתמשים בפונקציות ליניאריות לייצוג הdata – בעקבות השימוש ברשתות נוירונים בפונקציות לא ליניאריות – הרשת יכולה להגיע להחלטות לא ליניאריות, מה שנותן תוצאות טובות יותר, כי רוב הבעיות המורכבות אינן ניתנות להחלטות ליניאריות, אלא החלטות שאינן ליניאריות.

נגדיר את וקטור הכניסה לרשת כ-$\boldsymbol{x}$ ואת וקטור המוצא כ-$\boldsymbol{y}$.

ישנם כמה היפר-פרמטרים שקובעים את מבנה הרשת : מספר השכבות הנסתרות ברשת (hidden layers) מספר הנוירונים בכל שכבה, סוג הפונקציות הלא ליניאריות במעברים בין כל שתי שכבות ועוד.

מספר השכבות ומספר התאים מגדירים את גודל הרשת.

במודל ישנם שלושה סוגי שכבות:

<u>1. שכבת הכניסה- input layer:</u>

לשכבה זו אנו מכניסים את וקטור הכניסה שלנו- $\boldsymbol{x}$. מספר האלמנטים בשכבה זו יהיה מספר הפיצ'רים שיש לנו עבור המידע הנכנס.

מוצא שכבה זו מועבר לכניסת השכבה הבאה.

<u>2. שכבות חבויות- hidden layers:</u>

כל שכבה כזו מקבלת את הקלט שלה מהשכבה הקודמת, מוכפלת במשקולות המעבר בין השכבות ומבצעת חישוב כלשהו בעזרת הפונקציה של הלא ליניארית של אותה שכבה. מוצא כל שכבה הינו כניסת השכבה הבאה אחריה.

מספר האלמנטים והפונקציה הלא לינארית בכל שכבה חבויה יכולים להשתנות כתלות בבעיה שנרצה לפתור. יש לבחור רשת גדולה מספיק, אך לא גדולה מדי. מצד אחד, רשת קטנה מדי לא תוכל לקרב בדיוק מספיק את המיפוי הנדרש, ואילו רשת גדולה מדי תמנע לימוד יעיל.

<u>3. שכבת היציאה- output layer:</u>

לשכבה זו אנו מכניסים את מוצא השכבה החבויה האחרונה.

שכבה זו מבצעת חישוב על הכניסות שלה ומוציאה את פלט הרשת. המוצא הרשת הוא מוצא השכבה הזאת.

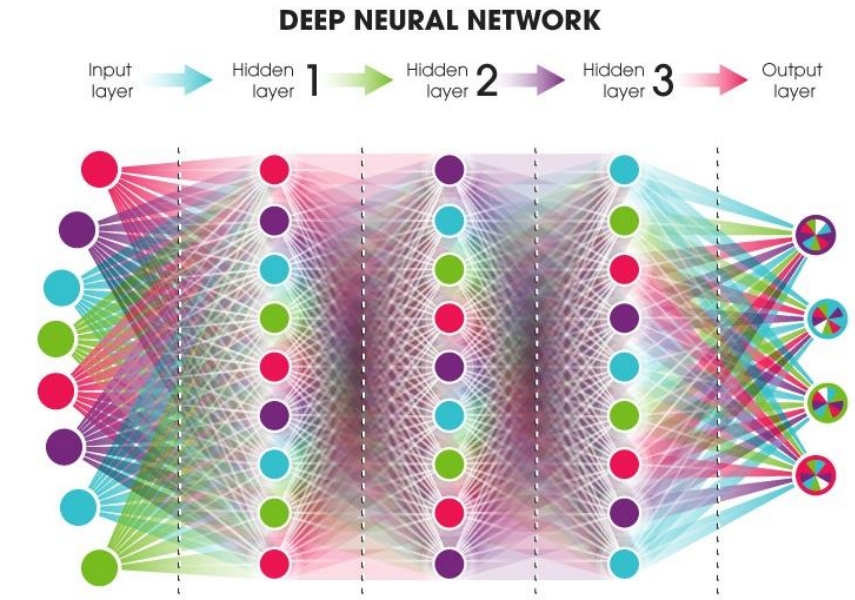
מספר האלמנטים בשכבה זו הוא כגודל הפלט הרצוי והוא משתנה כתלות בבעיה שמגדיר. מוצא הרשת יכול להיות הסתברות לסיווג (קלסיפיקציה) או לחזות ערך מספרי (רגרסיה).

המעבר מייצוג אחד לאחר ברשת מתבצע בשלושה שלבים:

1 .הכפלת הפלט של השכבה הקודמת במשקולות של הרשת.

2 .לכל אלמנט בשכבה, נחבר את כל הקלטים הממושקלים המקושרים אליו.

3 .נפעיל פונקציית אקטיבציה שהיא על פי הגדרה פונקציה לא ליניארית. פונקציה זו פועלת על כל הקלטים הממושקלים בכל אלמנט בשכבה.




neuralnetworksanddeeplearning.com - Michael Nielsen, Yoshua Bengio, Ian Goodfellow, and Aaron Courville, 2016.

## תהליך אימון של רשתות נוירונים

כמו שהזכרנו- הרשת מורכבת מסט פרמטרים $\theta$ ונרצה שהוא יביא אותנו לביצועים אופטימליים. את השגת האופטימליות של הפרמטרים נעשה באמצעות אימון על המון דוגמאות כך שבסופו של דבר סט הפרמטרים יתאים לבעיה.

כמו שאמרנו, כדי להתחיל לאמן אנחנו צריכים בסיס נתונים המכיל דוגמאות (training data) ותיוג (labels) שלהם שאומר לנו מה התוצאות הנכונות עבור כל דוגמא. כך שנכניס לרשת וקטור כניסה $\boldsymbol{x}$ והוא מחושב על ידה כאשר התוצאה הרצויה $\boldsymbol{y}$ גם כן ידועה לרשת. על ידי בסיס נתונים מספיק גדול הרשת תמצא לנו פונקציה המקשרת בין הכניסה והמוצא. תהליך ההעברה ברשת נקרא feed-forward.

כיצד נוכל להגיד מתי היא צודקת / טועה וכל היא תוכל לדעת להתכנס לפתרון נכון? על סמך פונקציית שגיאה הנקראת loss function. קיימות מספר פונקציות כאלה כמו: cross entropy, MSE. ומשתמשים בכל אחת כתלות בבעיה שאנחנו צריכים לפתור. באמצעות פונקציה זו נוכל לתת ציון כתלות בכמה הרשת צדקה/טעתה ובכך להתכנס לפתרון נכון.

איך זה עוזר לנו למצוא את סט הפרמטרים האופטימלי? בעצם סט הפרמטרים יהיה אופטימלי כאשר פונקציית השגיאה שלנו תהיה מינימלית. נשתמש כדי להוריד את השגיאה באלגוריתם SGD (Stochastic gradient descent).

הוא מתבסס על העיקרון שהגרדיאנט של פונקציה בנקודה מסוימת ייתן לנו את הוקטור בעל השיפוע הוא הגדול ביותר. ואם נלך צעד זה בכיוון ההפוך, נתקדם הכי מהר לעבר הנקודה שתיתן לי את השגיאה המינימלית. הגרדיאנט הינו אופרטור ליניארי, ולכן סכימה של וקטור הגרדיאנטים עבור כל כניסה, הוא שקול לגרדיאנט של סכום פונקציית המחיר עבור כל הכניסות. את וקטור הנגזרת נכפיל במקדם למידה כלשהו שהינו פרמטר של הרשת.

הנוסחה האיטרטיבית של הSGD היא: $\theta_{n+1} = \theta_n - \alpha\nabla\mathcal{L}(\theta_n)$. כאשר $\theta$ זה סט הפרמטרים של הרשת, $\alpha$ הוא יחס הלמידה, ו$\mathcal{L}(\theta_n)$ זו פונקציית השגיאה.

ערכו של מקדם הלמידה קובע אם תהיה התכנסות, ובאיזה קצב. אם הוא גדול מדי, אנו עלולים להתקדם אל נקודת המינימום בצעדים גדולים מדי ובעצם לדלג עליה מה שיגרום שלא נתכנס אליה. ואם הוא קטן מדי, אזי נעשה בכל שלב צעד קטן מאוד ואז יכול להיות שנתכנס אבל התכנסות האלגוריתם תהיה איטית מאוד.

כמובן שלא נרצה לעדכן את הפרמטרים (צעד של האלגוריתם) עבור כל דוגמא כי אז התהליך יהיה מאוד איטי- אז נרצה לעדכן אותו עבור מספר מסוים של דוגמאות (*mini- batch*), שתלוי בסוג הבעיה והמשתמש בוחר אותו.

וכמובן שמעבר אחד על כל הדוגמאות (*epoch*) הוא לא מספיק כדי שהמערכת תלמד טוב - אז נצטרך לעבור עליהם מספר מסוים של פעמים (בדר"כ עשרות פעמים).

מה שנעשה הוא שנאתחל את סט הפרמטרים בצורה רנדומלית ובעזרת אלגוריתם SGD נגיע למינימום של פונקציית השגיאה ובעצם לסט פרמטרים אופטימלי לבעיה שלנו $\theta_{opt}$.

סט הפרמטרים יהיה אופטימלי על הדוגמאות שלנו מהאימון. אנחנו לא יודעים אם הוא יעבוד טוב גם על דוגמאות אחרות וחדשות. כי להיות בטוחים שהם כן יעבדו טוב- נרצה שהדוגמאות באימון ישקפו טוב את סך הדוגמאות שנרצה לבחון בעתיד וזה דורש מגוון עשיר ורחב של דוגמאות שאנו מאמנים. ככל שיהיה מגוון כזה- הרשת תדע להתמודד עם מגוון רחב יותר של דוגמאות בעתיד.

לאחר האימון -נרצה לבחון את ביצועי הרשת (סט הפרמטרים שלה). בשלב זה נשתמש במאגר נתונים חדש, כאשר המוצא הרצוי עבור מאגר זה ידוע לנו אך לא נתון לרשת. אנו בוחנים כאן את ביצועי הרשת שכן הרשת לא התאמנה על מאגר זה ולא מכירה אותו. על מאגר זה נבצע רק את תהליך ה- forward Feed (חלחול ברשת) ונבחן את ביצועי הרשת ביחס לתוצאות הידועות. חלק זה נקרא האימות – Validation.

## Convolutional Neural Network רשתות קונבולוציה

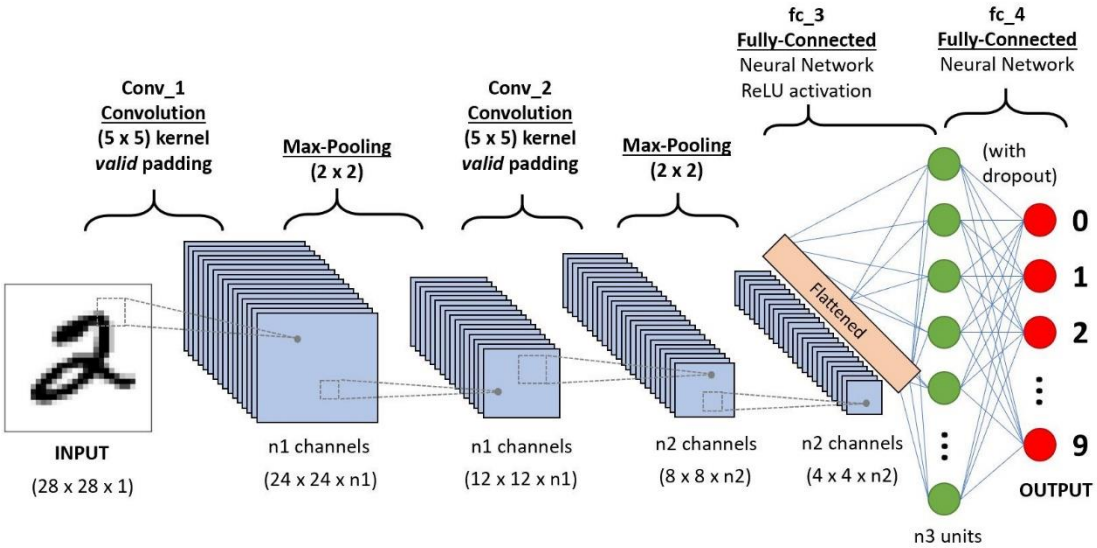
סוג מסוים של רשתות נוירונים שהוכחו כאפקטיביות עבור משימות כמו זיהוי וסיווג תמונות או כל מידע ויזואלי אחר.

הייחודיות שלו זה שבגלל המבנה שלו – לא כל נוירון מסתמך על כל הנוירונים שבשלב הקודם (Fully- Connected NN) אלא רק על הנוירונים הקרובים אליו. היתרון בזה הוא היעילות- מדובר על פחות מכפלות וסיכומים מהמודל הקודם.

במעבר בין שכבות אנו מבצעים שני שלבים:

שלב הקונבולוציה: אנו לוקחים פילטר שהוא בעצם סט המשקולות של הרשת באורך ורוחב שאנו קובעים אותו ועומק כמו בעומק השכבה הקודמת- ואנחנו כל פעם עושים מכפלה פנימית בין החלון לבין חלק מהשכבה הקודמת. כך שמכל מכפלה – יוצאת לנו תוצאה אחת.

שלב הpooling: זה שלב שבו אנו מורידים את ממדי התמונות שיוצאת מן המסנן. מתוך מטריצה בזאת- אנו נבחר את הערך הגדול ביותר מתוך מטריצה בגדול מסוים. המטרה של שלב זה היא להוריד מימד בצורה חכמה- כי ניקח את הערך הכי משמעותי בסביבה הקרוב ואת שאר הערכים אנו נוריד.



## U-net

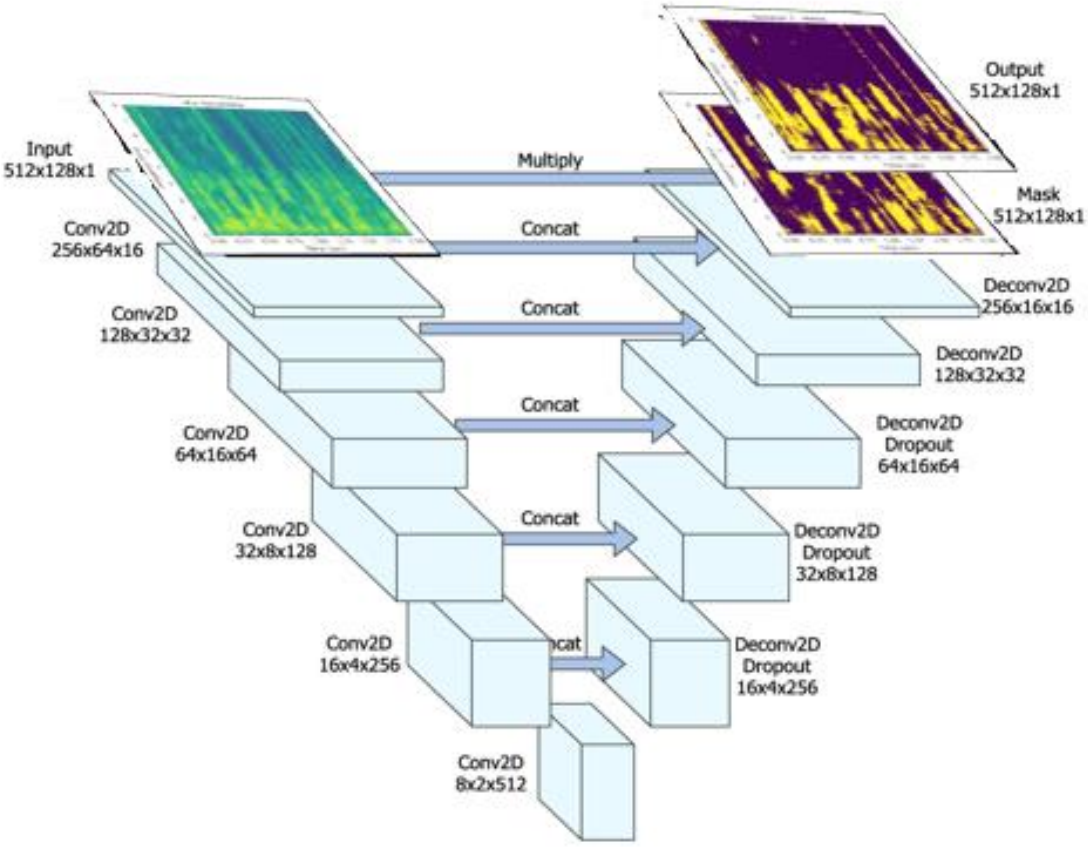
זו רשת קונבולוציה שפותחה לסגמנטציה של תמונות. הרשת בנויה משלב מכווץ כאשר החצי הראשון של השכבות מורידות כל פעם את מימדי התמונה הנכנסת ע"י פעולת ה-downsampling עד כדי תמונה מאוד קטנה ולאחר מכן, משלב המרחיב אותה כשחצי השני של השכבות בכל פעם מעלות את מימדי התמונה חזרה ע"י פעולת ה-upsampling. כך שהתמונה במוצא היא באותם המימדים כמו התמונה הנכנסת לרשת. בנוסף, הנתיבים המכווצים והמרחיבים הם פחות או יותר סימטריים ומתבצעים אחד אחרי השני- מה שמקנה לה את הארכיטקטורה בצורת U ומכאן שמה.

השלב המכווץ מורכב משכבות כאשר הורדת המימד נעשה באמצעות פעולת הקונבלוציה- פעולת ה-max pooling ומעבר בפונקציה לא לינארית- Relu. מטרת הכיווץ היא שהמידע המרחבי יצטמצם והמידע על פיצ'רים חשובים של התמונה יישמר.

השלב המרחיב משלב פיצ'רים ברזולוציה נמוכה על המידע כתוצאה משכבות הקונבולוציה המרחיבות את התמונה ו פיצ'רים ברזולוציה גבוהה כתוצאה מהמידע המרחבי מהשכבה המשוקפת בשלב המכווץ.

מוצא כל שכבה בשלב הכיווץ מועבר לשכבה המקבילה אליו בשלב ההרחבה בנוסף להליך הטבעי שעוברת התמונה ע"י שכבות ההרחבה. זה בעצם נותן מידע נוסף על פרטים נוספים בתמונה ששכבת ההרחבה לא נותנת לי ועוזר לחדד את הרזולוציה של התמונה שמוצא שכבה זו.

תמונה להמחשה של רשת נוירונים Unet:



# פרמול הבעיה

בבעיה שלנו יש 2 אותות דיבור המדברים באותו הזמן שהוקלטו במיקרופון יחיד.

נתונה הקלטה על ידי L דגימות של אות דיבור משולב $z[n] = x_1[n] + x_2[n],\ n = 0,1,\dots,L-1$

כאשר $x_1[n]$ זה אות הדיבור של דובר א' ו$x_2[n]$ זה אות הדיבור של דובר ב'. נרצה בסופו של דבר, באמצעות $z[n]$, לשחזר את $x_1[n]$ ואת $x_2[n]$ שבנו אותו – כלומר את $\widehat{x_1[n]}$ ואת $\widehat{x_2[n]}$. התמרת STFT של $z[n]$ היא סכום ההתמרות של כל אות דיבור. נבצע התמרת STFT שתיתן לנו אות מרוכב:

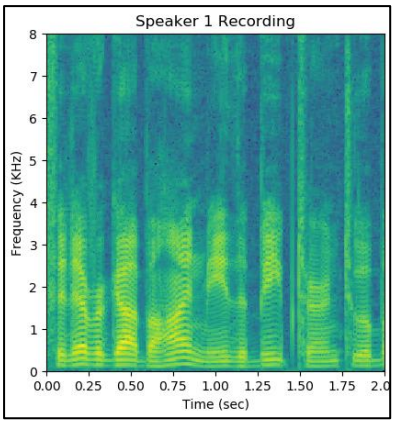
$$z[n] = x_1[n] + x_2[n] \xrightarrow{STFT} Z[l,k] = X_1[l,k] + X_2[l,k]$$

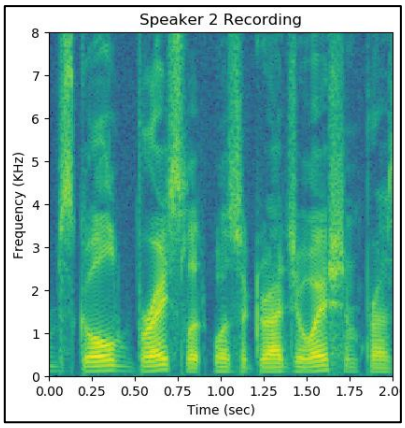
נפעיל ערך מוחלט שייתן לנו ערך המגניטודה (העוצמה) שלו ונקבל ערך ממשי, ובנוסף נוציא log משני הצדדים כדי שנוכל לקבל "מתיחה" בציר התדר (אנו עושים זאת כדי שבתדרים הנמוכים, ששם רוב תדרי הדיבור עם האינפורמציה שאנו צריכים לזהות ולהפריד, נוכל להתבונן טוב).

סה"כ נקבל עבור $k = 0,1,\dots,K-1$ (נזכור שבגלל שהאות שלנו הוא ממשי בזמן אז הוא סימטרי בתדר ולכן אנחנו משתמשים רק בחצי מהתדרים בסה"כ) ועבור כל $l = 0,1,\dots,L-1$:

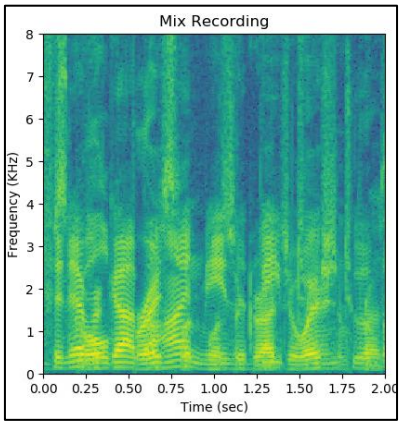
$$(1)\ X_{1_{l,k}} \triangleq \log_spectrum(x_1[n]) = \log(|STFT\{x_1[n]\}|) = \log(|X_1[l,k]|)$$

$$(2)\ X_{2_{l,k}} \triangleq \log_spectrum(x_2[n]) = \log(|STFT\{x_2[n]\}|) = \log(|X_2[l,k]|)$$




$$(3)\ Z_{l,k} \triangleq log_spectrum(z[n]) = log(|STFT\{z[n]\}|) = log(|Z[l,k]|)$$



## מודל הWDO

אנו מתבססים על העיקרון $W-disjoint\ orthogonality$ שאומר שבכל נקודת זמן-תדר במישור ה-$spectrum$ , פעיל רק דובר אחד, או ששני הדוברים אינם פעילים.

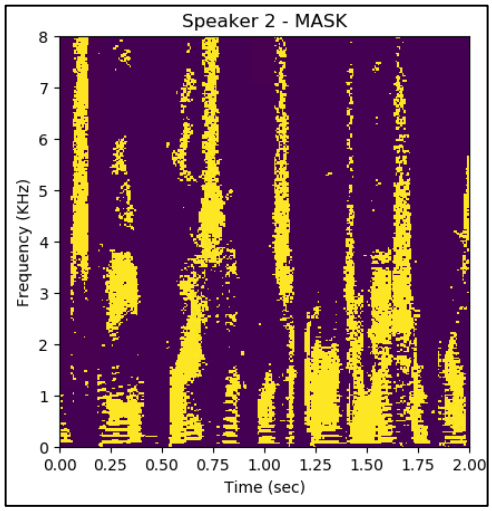
הנחה זאת מספיקה לנו בשביל מידול מספיק טוב של הבעיה: בצורה זו נוכל לייצר 2 מסיכות שתחליט בכל נקודת זמן-תדר האם לפנינו מידע של דובר א' או דובר ב', לפי מודל מקסימיזציה בין שני הספקטרומים.

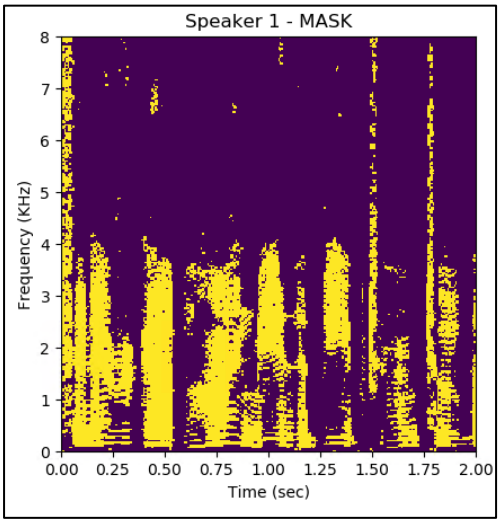
מעתה נוכל בקלות להנחית את נקודות הזמן-תדר של הדובר הלא פעיל (באותה נקודה) מבין השניים, ולהשאיר את נקודות הזמן-תדר של הדובר הפעיל (באותה נקודה) מבין השניים, ולמעשה כל מסכה תשאיר לי את ספקטרום המקורי של כל אחד מהדוברים.

הגדרת המסכות תהיה:

$$(4)\ binary_mask_1 = \begin{cases} 1, & |X_{1_{l,k}}| \geq |X_{2_{l.k}}| \\ 0, & |X_{1_{l,k}}| < |X_{2_{l,k}}| \end{cases}$$

$$(5)\ binary_mask_2 = \begin{cases} 1, & |X_{2_{l,k}}| \geq |X_{1_{l,k}}| \\ 0, & |X_{2_{l,k}}| < |X_{1_{l,k}}| \end{cases}$$




לאחר מכן, "נעביר" את הספקטרום המשולב במסיכות של שני הדוברים. נבצע זאת על ידי הכפלה של הספקטרום המשולב בכל אחת מהמסיכות:

$$\boxed{\widehat{X_{1_{l,k}}} = binary_mask_1 * Z_{l,k}}$$

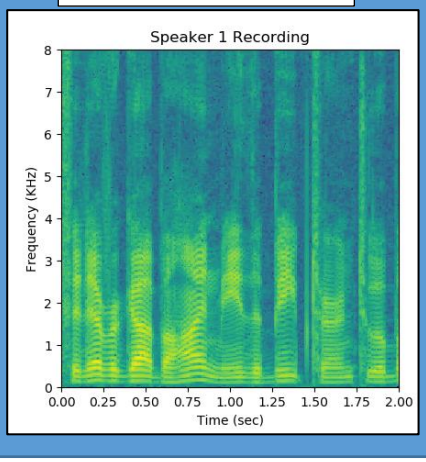
$$\boxed{\widehat{X_{2_{l,k}}} = binary_mask_2 * Z_{l,k}}$$

את שני הספקטרומים נחזיר כמובן למישור הזמן, לקבלת האודיו של כל אחד מהדוברים.

נדגים את התהליך באמצעות הגרפים הבאים:

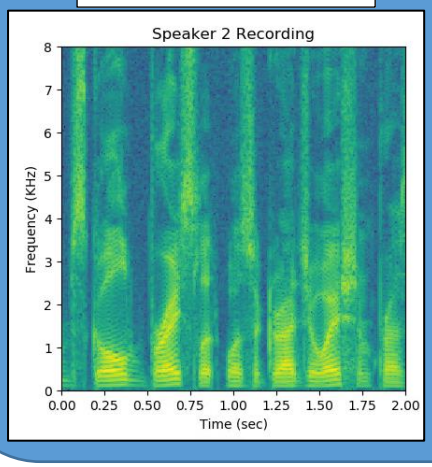
שלב א'

לפי משוואה (1) $X_{1_{l,k}}$




(a)

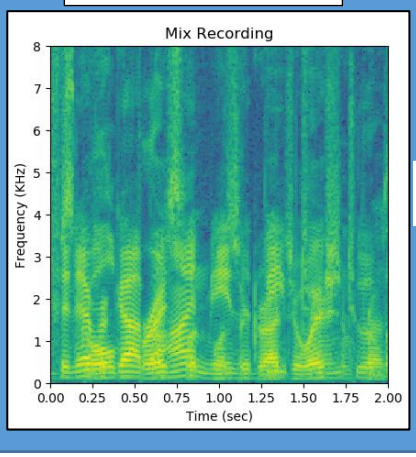
לפי משוואה (2) $X_{2_{l,k}}$




(b)

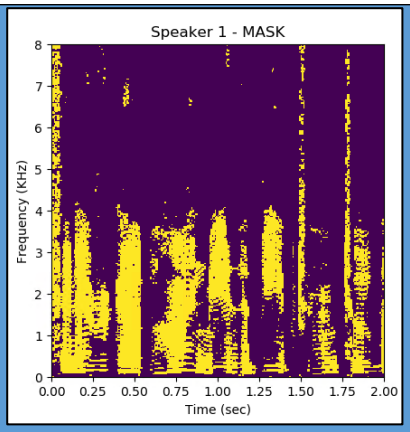
שלב ב'

לפי משוואה (3) $Z_{l,k}$



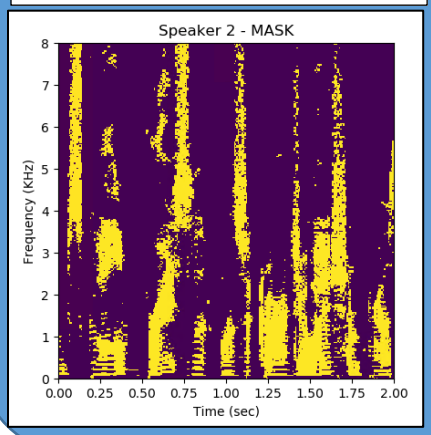

(c)

לפי משוואה (4) $binary_mask_1$




(d)

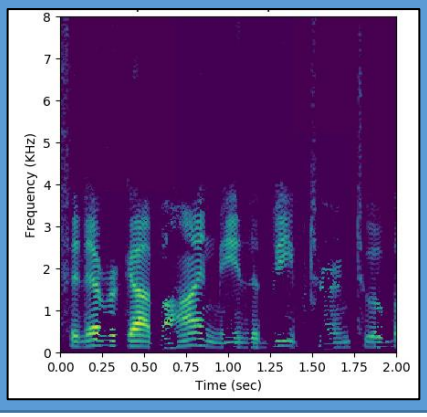
לפי משוואה (5) $binary_mask_2$




(e)

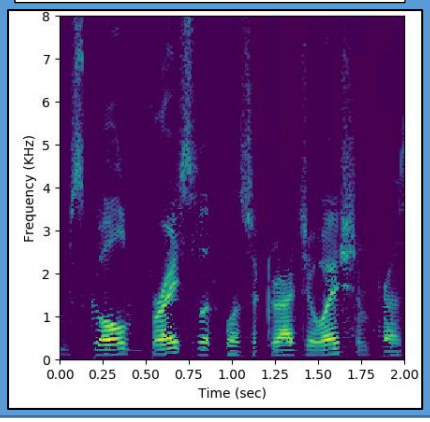
שלב ג'

$\widehat{X_{1_{l,k}}} = binary_mask_1 * Z_{l,k}$




(f)

$\widehat{X_{2_{l,k}}} = binary_mask_2 * Z_{l,k}$




(g)

(a) תצוגת $log_spectrum$ של אות הדיבור של דובר א'

(b) תצוגת $log_spectrum$ של אות הדיבור של דובר ב'

(c) תצוגת $log_spectrum$ של אות הדיבור של שני הדוברים בו זמנית

(d) תצוגת $binary_mask_1$ כאשר הערכים הצהובים הם הערך '1' (החלטה להשאיר את הערך הזה) והערכים הסגולים הם הערך '0' (החלטה לאפס את הערך הזה)

(e) תצוגת $binary_mask_2$ כאשר הערכים הצהובים הם הערך '1' (החלטה להשאיר את הערך הזה) והערכים הסגולים הם הערך '0' (החלטה לאפס את הערך הזה)

(f) תצוגת $log_spectrum$ עבור אות הדבור של דובר א' לאחר שעבר במסכה הבינארית שלו.

(g) תצוגת $log_spectrum$ עבור אות הדבור של דובר ב' לאחר שעבר במסכה הבינארית שלו.

חשוב לזכור שמה שהראנו פה זה הסבר למודל WDO שבו אנו יודעים איך נראה כל אות בנפרד (גרפים (a) , (b)). אבל, בזמן מבחן אין לנו אות של כל דובר אלא רק אות של שני הדוברים ביחד (גרף (c)).

לכן, האלגוריתם לפתרון הבעיה שלנו היא לבנות מערכת שיודעת למצוא בכל נקודת זמן-תדר מי הדובר הדומיננטי על סמך הלוג ספקטרום המשולב בלבד ובאמצעותו לבנות את המסכות.

משם, נכפיל את המסיכות בספקטרום המשולב לקבלת כל אחד מהספקטרומים המקוריים.

לבסוף, נחזיר את הספקטרומים המתקבלים חזרה למישור הזמן לקבלת האודיו המקורי של כל אחד מהדוברים.

## מציאת המסיכות

מטרתנו היא למצוא פונקציה עם סט פרמטרים $\theta$ - $f_\theta$, שבהינתן אות המעורבב במישור ה– $log_spectrum$ - $Z_{l,k}$ (ספקטוגרמה) היא תוציא שתי הסתברויות לאיזה דובר דומיננטי בכל נקודת זמן-תדר. לפי הסתברויות אלו, נבנה מסכות בינאריות $binary_mask_{1,2}$ שהם אפסים ואחדות בדיוק במקומות הרצויים בדיוק כמו שהגדרנו אותן במשוואות $(4), (5)$.

$$l = 0,1, \dots ,255 \qquad k = 0,1, \dots ,255$$

סה"כ נרצה פונקציה- $binary_mask_1[l,k], binary_mask_2[l,k] = f_\theta(Z_{l,k})$ ובאמצעות 2 המסכות נדע להפריד בין 2 הדוברים:

$$X_1[l,k] = binary_mask_1[l,k] * Z[l,k] \, , X_2[l,k] = binary_mask_2[l,k] * Z[l,k]$$

יש המון שיטות למציאת הפונקציה $f_\theta$.

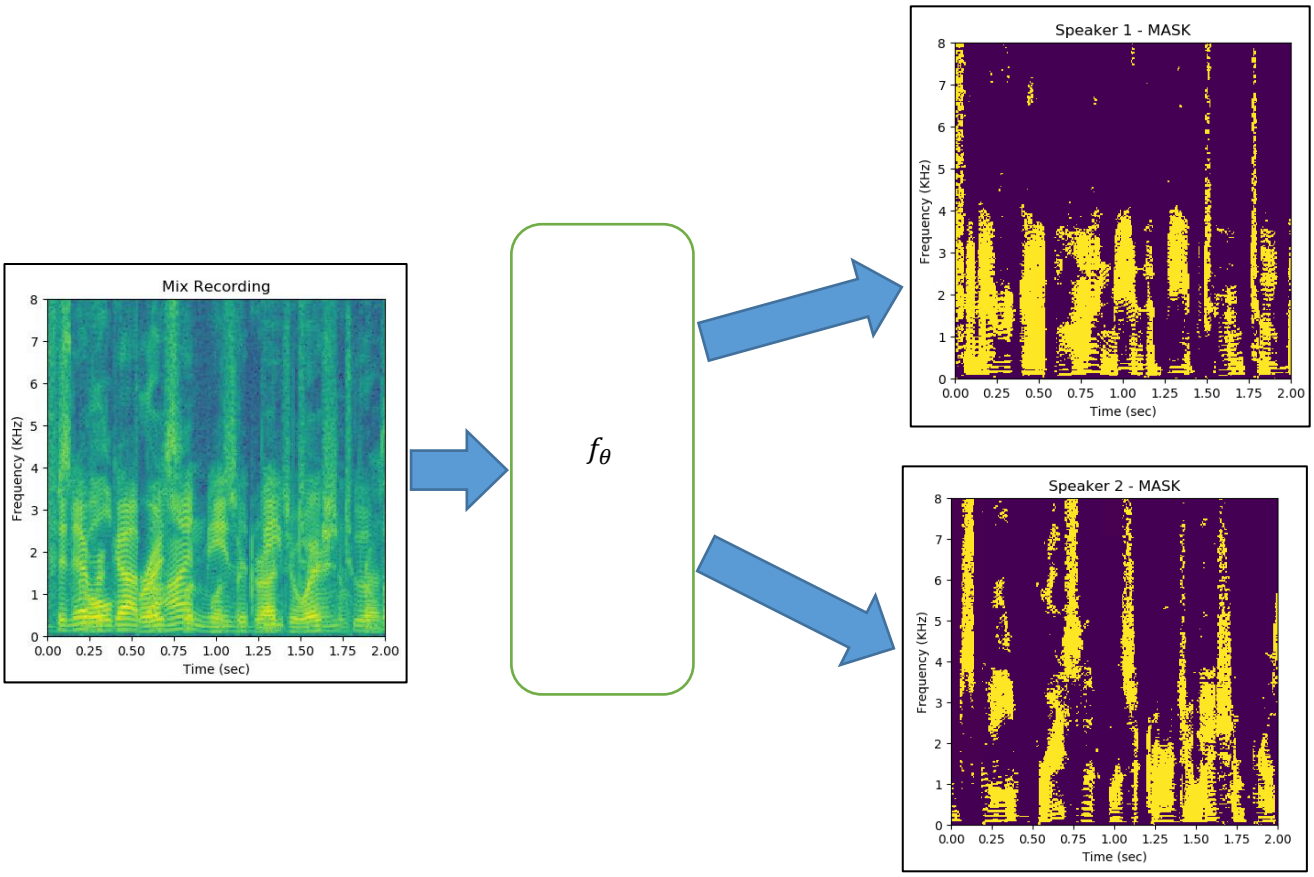
אנו מציעים שהפונקציה הזאת תהיה רשת נוירונים עמוקה (DNN)- בפרט רשת קונבולוציה (CNN) ובפרט רשת הנקראת Unet.

כדי לעזור לרשת, ניעזר בטכניקה הנקראת VAD (Voice activity detector) ברזולוציה גבוהה (זמן-תדר). זו טכניקה שעוזרת לנו לדעת מתי יש דיבור ומתי יש שקט וכך הרשת תלמד להבין איפה עיקר הדיבור מתבצע ותתאמן רק על אזורים משמעותיים ולא על כל הלוג ספקטרום.

במקרה שלנו, כלומר כשההקלטות מנורמלות ואין רעש רקע בהקלטה – אנו נותנים למערכת ערך סף שבעצם זו העוצמה של הדיבור שמתחתיה האזורים לא מוגדרים כדיבור ולכן לא נתחשב בהם ומעליה זה נחשב דיבור וכן נתחשב בהם. בסופו של דבר, אנחנו עוזרים לרשת מכיוון שאנו ממקדים אותה איפה אנחנו רוצים שהיא תלמד.

המודל שלנו הוא מודל דיסקרימינטיבי. זה מודל בתחום הלמידת מכונה שממדלים את התלות במשתנה לא ידוע במשתנה ידוע. נשתמש בפונקציית ההתפלגות המותנית $P(y|x)$ כך שלפיה נוכל למצוא את $y$ באמצעות ידיעת $x$.

אצלנו בפרויקט- באמצעות סט הפרמטרים – ננסה למצוא את המסכות הבינאריות באמצעות ידיעת האות במישור הלוג ספקטרום של האות המעורבב.



# הכנת הData- Base

כדי לאמן את רשת הנוירונים $f_\theta$ אנחנו צריכים סט של $N$ דוגמאות כך שכל דוגמא מכילה

1) $input - Z[l,k]$

2) $targets - binary_mask_1[l,k], binary_mask_2[l,k]$

כל דוגמא היא גם בשביל ה$train$ וגם בשביל ה$test$.

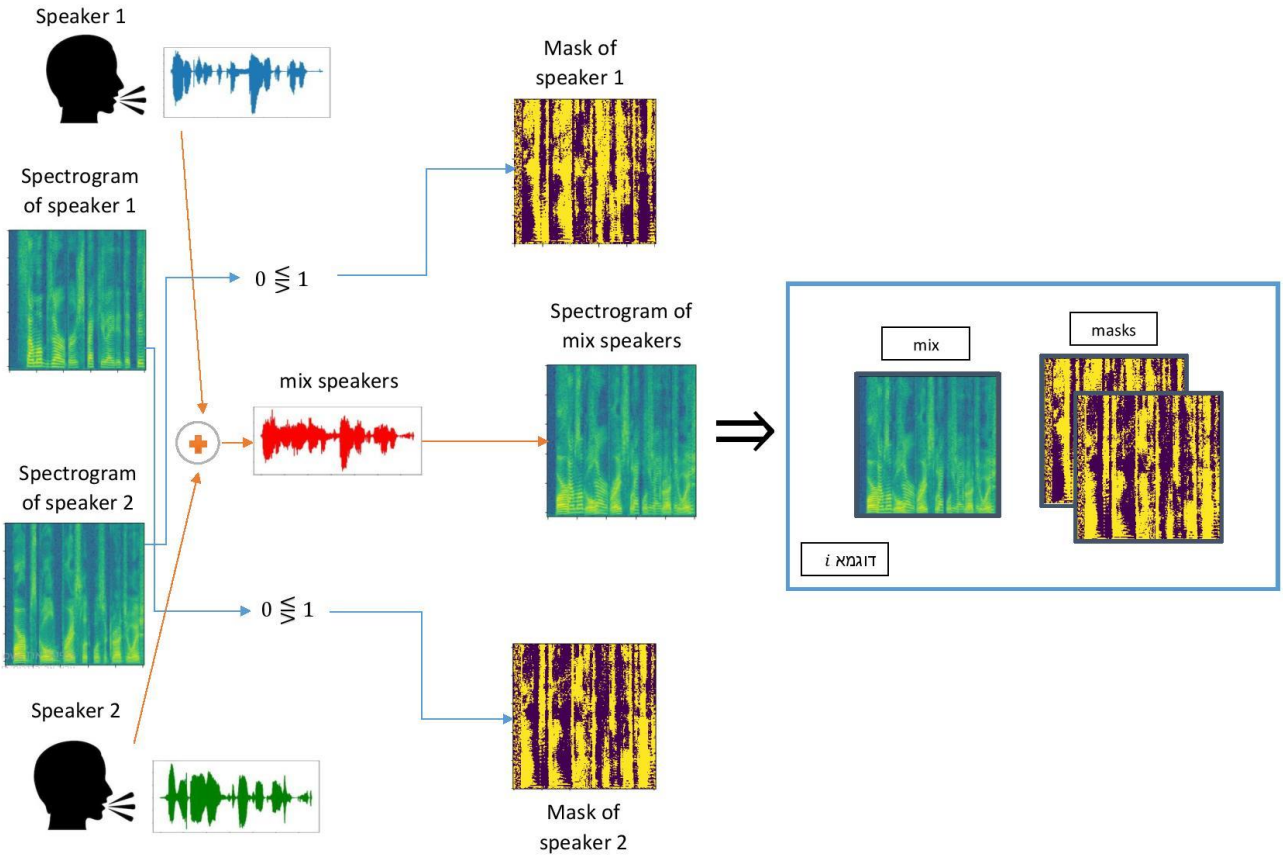
לקחנו Dataset של אותות דיבור נקיים הנקראת $TIMIT$ שהכילה הקלטות דוברים מאזורים שונים בארה"ב המקריאים משפטים שונים. זה מאגר מוכר, עשיר מבחינת קולות, הברות ומבטאים (accents) שמיועד לעזור בפתרון בעיות שונות בתחומים כמו NLP והפרדת דוברים.

עבור כל דוגמא בחרנו רנדומלית שתי הקלטות של דוברים שכל אחת מהן היא שתי שניות – חיברנו אותם להקלטה בה שומעים את שניהם מדברים יחד והמרנו למישור $log_spectrum$ לקבלת ה$input$ .

בנוסף, את האותות של כל אחד מהדוברים המרנו למישור ה-$log_spectrum$ והשווינו בכל נקודת זמן – תדר (פרט לנקודות בהם הVAD מאפס את האותות במישור ה-$log_spectrum$) מי מדבר יותר חזק כדי לייצר את המסכות הבינאריות שהם ה$targets$ שלנו.

סה"כ אנחנו בונים Data Base עבור הקלטות בעלות אורכים קבועים של שתי שניות בלבד ולאחר מכן נסביר איך נעבור מהקלטות באורך קבוע להקלטות באורך משתנה.

סכמה ליצירת דוגמא אחת של ה$DataBase$:



# בעיית הפרמוטציה

נסביר עכשיו על בעיית הפרמוטציה בה נתקלנו.

יש לציין כי בעיה זו קיימת גם ברשתות שפותרות בעיות דומות לבעיה שלנו.

באופן כללי, במהלך אימון של רשתות נוירונים, הרשת מקבלת המון דוגמאות של כניסות $inputs$ ($X_1$ ו-$X_2$) ויציאות <u>רצויות</u> $targets$ ( $\widehat{X_1}$ ו-$\widehat{X_2}$ בהתאמה), בודקת מה הביצועים שלה על אותן דוגמאות באמצעות פונקציית השגיאה שבמקרה של הרשת שלנו, U-net , מתוארת בפונקציה הבאה:

$$loss_function = MSE\left(X_1, \widehat{X_1}\right) + MSE\left(X_2, \widehat{X_2}\right)$$

כאשר:

(*Mean Square Error*) $MSE(X,Y) = \frac{1}{n}\sum_{k=0}^{n}(\mathrm{X[k]} - \mathrm{Y[k]})^2$ .

במקרה שלנו, אחרי כל מעבר של קבוצת מסגרות ברשת (batch) יוצאות לנו שתי תוצאות לפונקציית ה-Loss Function - נסמן אותם כפלט 1 וכפלט 2:

$$binary_mask_{1_{U-Net}}, binary_mask_{2_{U-Net}}.$$

ואותן אנו משווים למסיכות הרצויות- $binary_mask_{1_{Target}}, binary_mask_{2_{Target}}$.

$$\text{Loss Function} = MSE\left(binary_mask_{1_{U-Net}}, binary_mask_{1_{Target}}\right) + MSE\left(binary_mask_{2_{U-Net}}, binary_mask_{2_{Target}}\right)$$

קיימת לנו בעיה שהיא שיוך הפלטים לדוברים. קיימות 2 אפשרויות:

א) פלט 1 ($binary_mask_{1_{U-Net}}$) ← דובר א' ($binary_mask_{1_{Target}}$)
ופלט 2 ($binary_mask_{2_{U-Net}}$) ← דובר ב' ($binary_mask_{2_{Target}}$).

ב) פלט 2 ($binary_mask_{2_{U-Net}}$) ← דובר א' ($binary_mask_{1_{Target}}$)
ופלט 1 ($binary_mask_{1_{U-Net}}$) ← דובר ב' ($binary_mask_{2_{Target}}$).

המטרה שלנו היא להפריד את שני הדוברים והסדר שבו המערכת תוציא את הדוברים הוא לא משנה אבל מבחינת הרשת שלומדת לפי פונקציית שגיאה- אם אנחנו מחזירים את סט המסגרות בסדר ההפוך – פונקציית השגיאה תניב לי ערך שגיאה גבוה מאוד וכתוצאה מכך, הרשת לא תלמד כי היא כל הזמן תשגה (למרות שהיא מבצעת את ההפרדה בצורה טובה).

לכן נציע פתרון כדי להימנע ממצב כזה.

## הפתרון: שימוש במנגנון הPIT (Permutation Invariant Training)

נרצה שהרשת תלמד לשייך את הפלטים לדוברים הרצויים גם אם היא החזירה את המסכות בסדר נכון וגם אם היא החזירה את המסכות בסדר הפוך.

הרשת מבצעת את עבודתה כראוי כאשר פונקציית השגיאה תניב ערך נמוך. לכן נשנה את פונקציית השגיאה כך שהיא לא תהיה תלויה בסדר היציאות. נעשה זאת על ידי חישוב שתי האופציות בין הפלט לדובר (אצלנו מדובר בשתי אופציות בלבד, אך ניתן להכליל את זה ליותר פלטים) וניקח בסוף את השגיאה המינימלית.

במצב כזה, הרשת לא צריכה להחליט מה הסדר הנכון של הפלטים- כי כל סדר כזה ייתן את אותה שגיאה.

$$\text{Loss Function with PIT} = \min$$

$$\{\left(MSE\left(binary_mask_{1_{U-Net}}, binary_mask_{1_{Target}}\right) + MSE\left(binary_mask_{2_{U-Net}}, binary_mask_{2_{Target}}\right)\right),$$

$$\left(MSE\left(binary_mask_{2_{U-Net}}, binary_mask_{1_{Target}}\right) + MSE\left(binary_mask_{1_{U-Net}}, binary_mask_{2_{Target}}\right)\right)\}$$

# עבודה עם המערכת בסיום האימון

מה שהראנו עד כה זה איך להפריד הקלטה באורך קבוע (שתי שניות) של 2 אנשים המדברים בו זמנית.

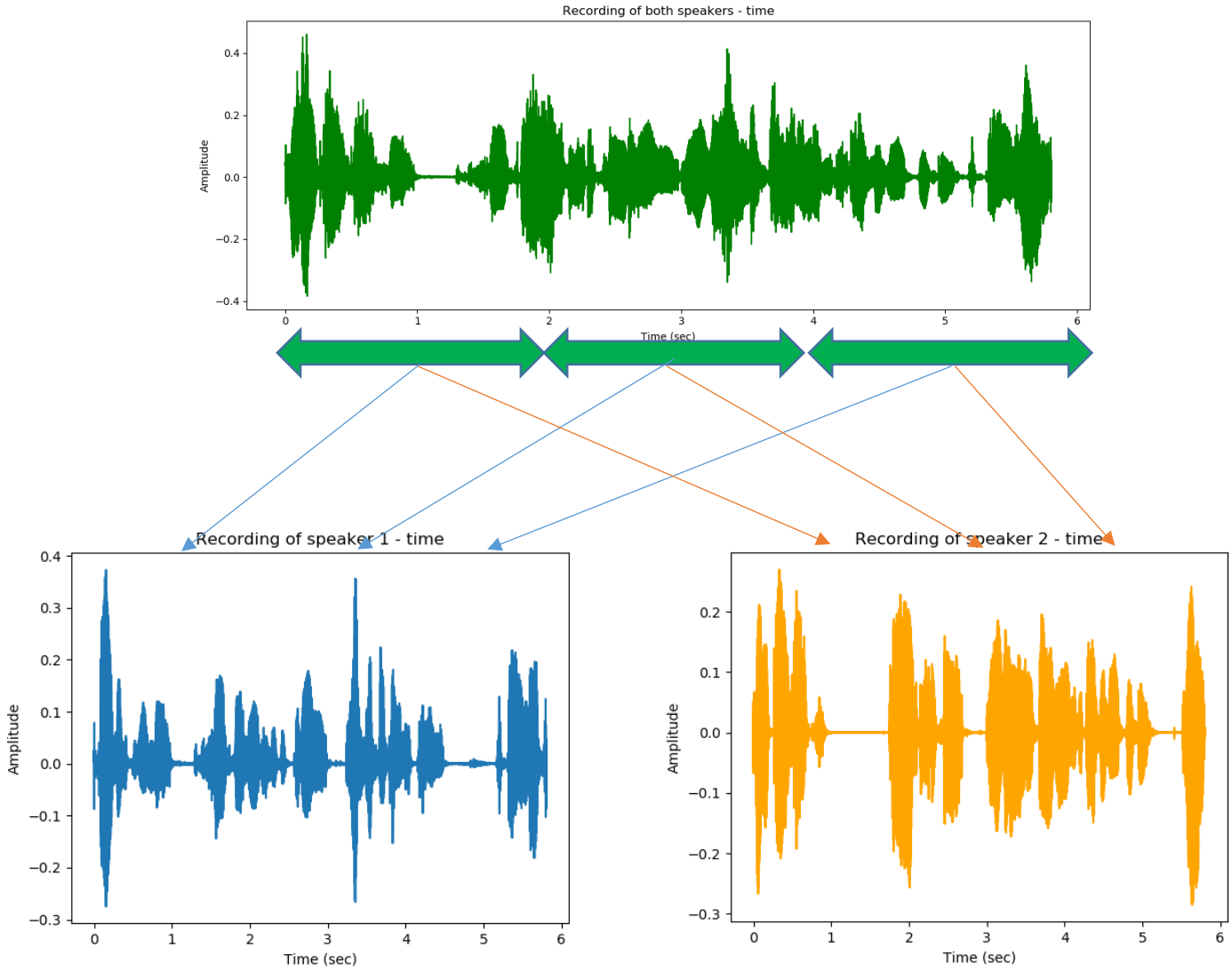
כעת נרצה להפריד כל הקלטה של שני דוברים ללא קשר לאורך ההקלטה. כלומר הרשת עובדת עם אורכים קבועים של שתי שניות ואנו נרצה שהמערכת תדע להפריד כל הקלטה- כלומר הקלטות באורך משתנה.

כדי לפתור בעיה זו - נחלק כל הקלטה לקטעים קבועים של שתי שניות ועבור כל קטע –נכניס לרשת ונקבל הפרדה.

לבסוף, נשרשר את הקטעים המופרדים לכדי שתי הקלטות באורך מלא.

אנחנו יכולים לעבוד בצורה כזאת מכיוון שהרשת שלנו עובדת רק על המקטע המסוים ולא על כל ההקלטות בו זמנית ולא תלויה בכל המקטעים הקודמים, בניגוד לרשתות רקורסיביות כמו B-LSTM שזו רשת שמשתמשת גם במקטעים הקודמים.

לכן, בגלל חוסר התלות שלנו- המערכת שלנו מאוד מהירה ונוכל גם להשתמש בה עבור הפרדה בזמן-אמת, כאשר ההשהיה היא קטנה מאוד בניגוד להפרדה המתבססת על רשתות רקורסיביות.



# מדד האיכות

בהקלטה המשולבת, קשה להבין מה אומר כל אחד מהדוברים. לכן המטרה היא שנדע להפריד את קטעי הקול כך שנוכל להבין מה אמר כל אחד מהדוברים.

המדד העיקרי הוא המובנות – כלומר, בהקלטה של כל דובר לאחר ההפרדה אכפת לנו אם מבינים את הדובר הראשון – מה שלא יכולנו להבין לפני ההפרדה.

עם זאת, נרצה למדוד את ביצועי המערכת בצורה פורמלית, גם כדי שנוכל להבין עד כמה ההפרדה שלנו הייתה טובה, ובמידה שלא - לנסות לשפר את פרמטרי הרשת כך שנגיע לתוצאות טובות יותר ושנוכל להשוות לפרויקטים אחרים בתחום.

מה שנעשה הוא שנבנה דוגמאות מלאכותיות של דיבור בו זמנית (חיבור של 2 הקלטות של דוברים נפרדים), נפריד אותם ונשווה אותם להקלטות המקוריות.

יהיה לנו מדד שנמדד בדציבלים והוא ייתן לנו הערכה כמה ההפרדה שלנו טובה:

**<u>SIR</u>** - היחס בדציבלים בין הסיגנל המשוחזר- $S_1$ לבין ההפרעה מהסיגנל השני - $S_2$ . ובצורה פורמלית:

$$SIR := 10\ \log_{10}\frac{\|S_1\|^2}{\|S_2\|^2}$$

כדי לקבל הערכה טובה של המערכת שלנו עבור המדד הזה, נשתמש בהמון דוגמאות ונמצע את הביצועים עבור המדד.

ההקלטות שמהם אנו מבצעים את ההערכה הן הקלטות שלא השתמשנו בהם באימון, אלא דוגמאות חדשות.

# תוצאות

את בדיקת המערכת שלנו ע"י מדדי האיכות הרצנו על 200 דוגמאות. אלו מספיק דוגמאות בשביל לתת הערכה משקפת לביצועים שלנו. הערך שקיבלנו (בדציבלים) למדד הוא:

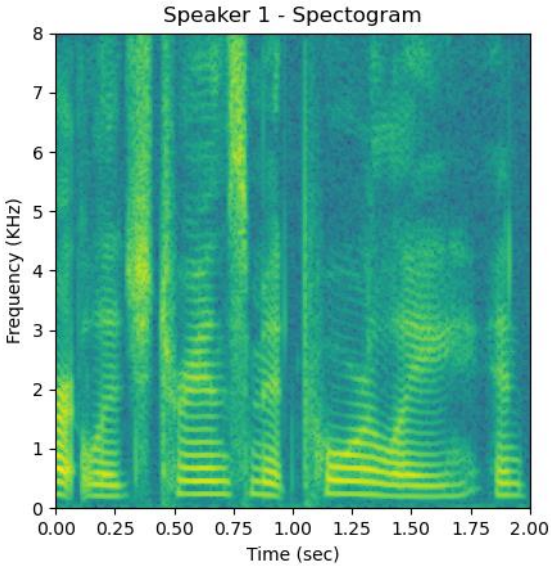
| | |
|---|---|
| 16.19703 | SIR |

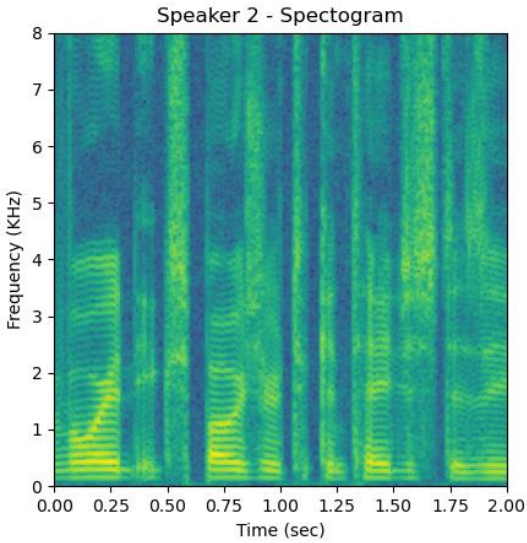
בData Base שהכנו ההקלטות היו באותה עוצמה או שהקלטה אחת הייתה בעוצמה של 2dB מהשנייה.

כעת, בהקלטות המשוערכות, הדובר שבהקלטה הוא (בממוצע) בעוצמה של 16dB ביחס לדובר השני כך שמדובר בשינוי משמעותי.

מדובר בתוצאות מהקלטות שונות, אבל תהליך יצירתם היה זהה לדוגמאות שבאימון כלומר מדובר בהקלטות נקיות – ללא רעשי רקע מסביב.

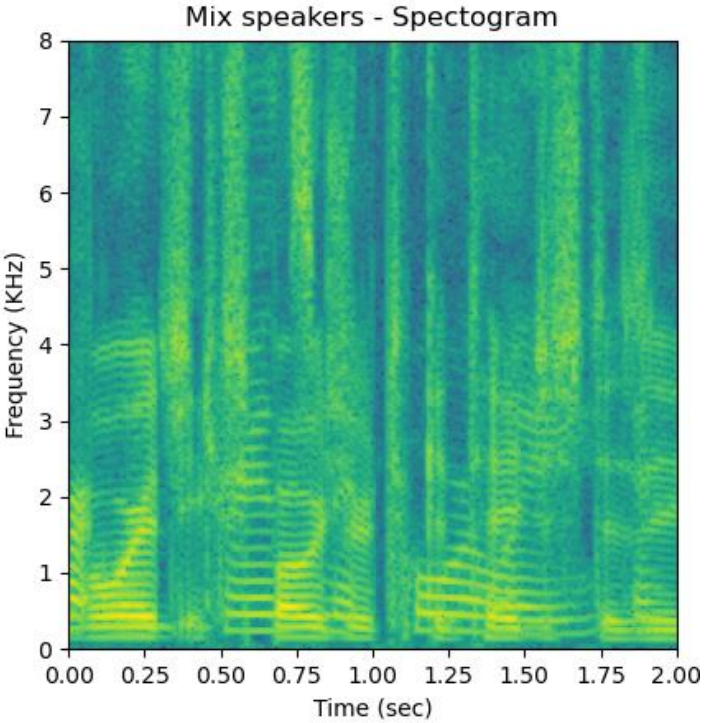
נדגים את תוצאות המערכת: תחילה ניקח שתי הקלטות (במקרה הזה גבר ואישה) ונציג את הספקטוגרמות שלהם:

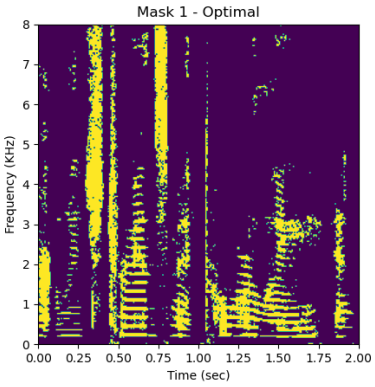


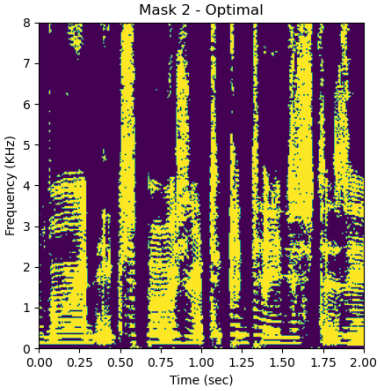



נחבר את ההקלטות לקבלת האות המעורבב וממנו ניצור את הספקטוגרמה שלו.

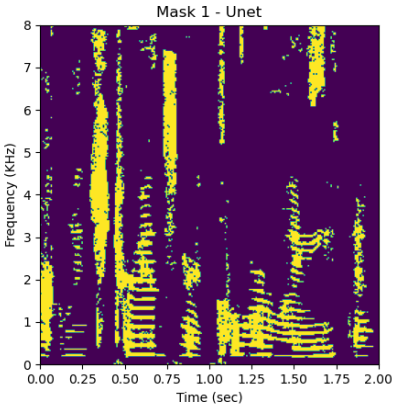
בנוסף, ניצור את המסיכות האופטימליות לפי משוואות (4) ו(5) (לשם השוואה בלבד):

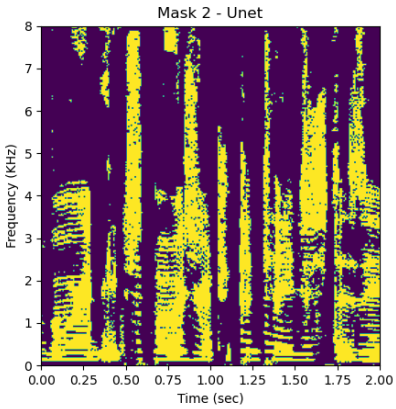



ואת הספקטוגרמה המשולבת נעביר במערכת לקבלת שערוך של אותן המסיכות:



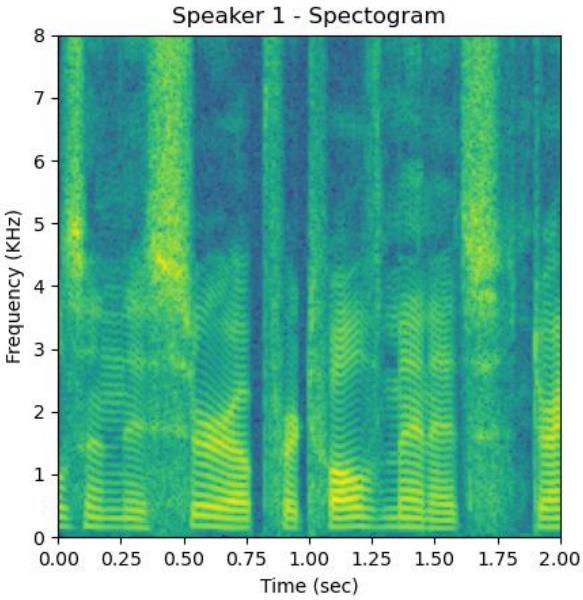
ניתן לראות דיוק רב של המסיכות, ועם זאת הפרדה טובה של שני הדוברים.

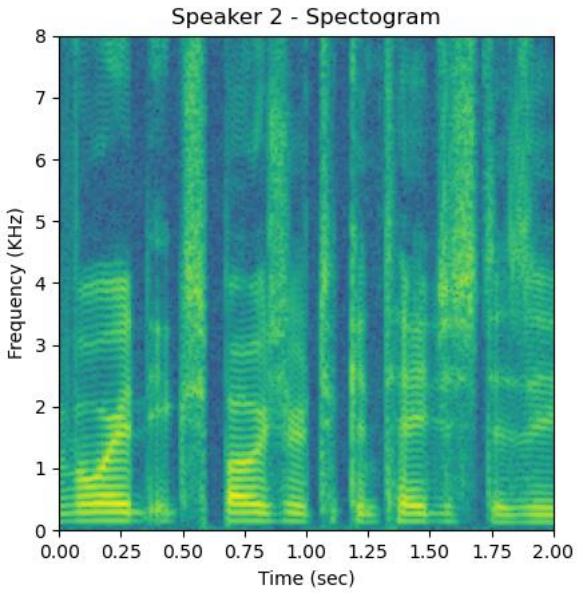
עם זאת - נתקלנו בשתי בעיות:

1) המערכת מתקשה יותר להפריד דוגמאות מהקלטות פחות נקיות, שאנו מקליטים בסביבה רועשת.

2) המערכת יודעת מאוד טוב להפריד אנשים בעלי צורת ספקטרום שונה אבל אנשים בעלי ספקטרום דומה (כמו הקלטה של שני גברים או שתי נשים) - המערכת מתקשה מאוד להפריד ביניהם ויש פעמים שהיא לא מצליחה כלל. נדגים זאת בתהליך דומה לזה שהוצג קודם.

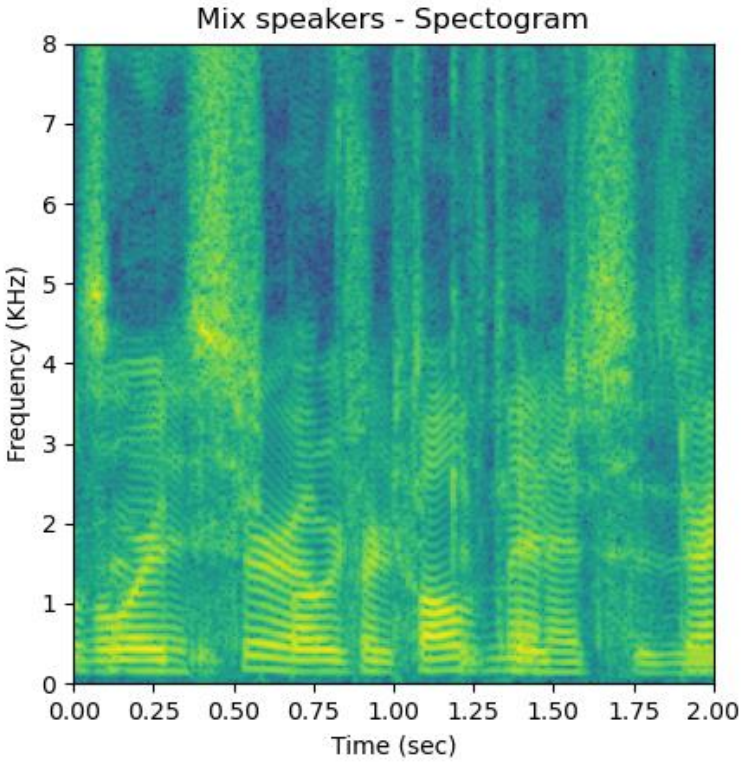
תחילה ניקח שתי הקלטות (במקרה הזה שני גברים):

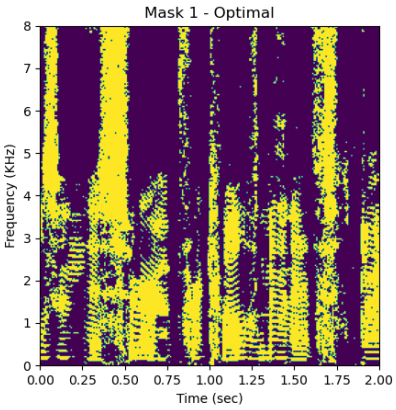


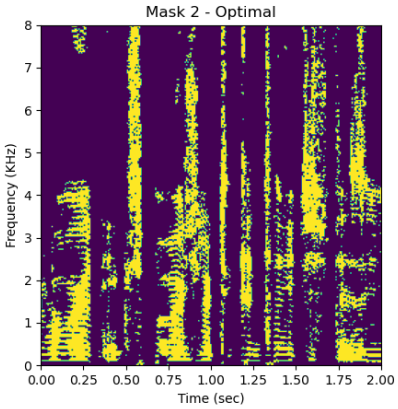



נחבר את ההקלטות לקבלת האות המעורבב וממנו ניצור את הספקטוגרמה שלו.

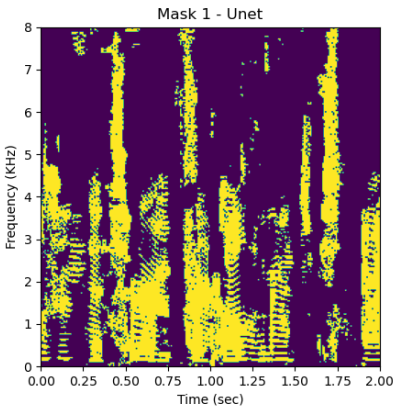
בנוסף, ניצור את המסיכות האופטימליות לפי משוואות (4) ו(5) (לשם השוואה בלבד):

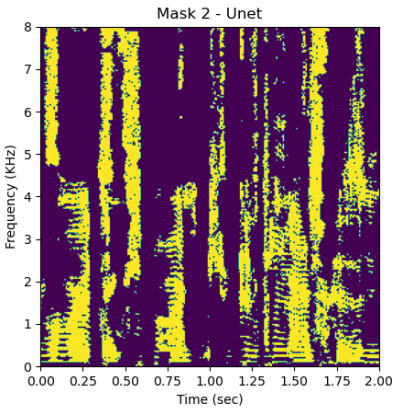



ואת הספקטוגרמה של האות המעורבב נעביר במערכת לקבלת שערוך של אותן המסיכות:



ניתן לראות קושי בקביעת המסיכות, ולכן ההפרדה אינה טובה.

ננסה להציע להן פתרונות:

1) הבעיה שלנו נובעת מזה שהבאנו למערכת רק דוגמאות מאוד נקיות ולא דוגמאות מורעשות. לכן כדי לנסות ולהתגבר על הבעיה הזאת בעצם בעת יצירת הDB שלנו – נבצע סימולציה בו הקלטה רועשת כלומר שני אנשים המדברים מעט רחוק מהמיקרופון (כלומר הקול שלהם עובר בתוך מערכת עד ההגעה למיקרופון). הסימולציה הזאת בעצם מדמה לנו מצב שהוא יותר אמיתי שמשקף הקלטות אמיתיות (תופעות כמו הדהוד וסביבה יותר רועשת) וכאשר נאמן את המערכת עם דוגמאות כאלה- נדע להפריד גם הקלטות מסביבה רועשת.

2) הבעיה היא במודל שלנו. מכיוון שאנו מסתמכים על מיקרופון אחד –אז הדבר היחיד שאנו יכולים להשתמש בו הוא הספקטרום של הדוברים. לכן , נרצה לשנות את המודל שלנו ובמקום מיקרופון אחד – נרצה להפריד באמצעות שני מיקרופונים. זה ייתן לנו מידע מרחבי על ההקלטה מה שלא היה לנו עם מיקרופון אחד. נוכל להיעזר בזה בכך שזה נותן יותר מידע למערכת שלנו ולכן היא תוכל להפריד גם אנשים בעלי צורת ספקטרום דומה כי והיא תוכל ללמוד מאפיינים אחרים שהם לאו דווקא הספקטרום, ובעזרתם להפריד.

בחרנו לשפר את ביצועי המערכת באמצעות אופציה 2.

# מערכת רב – ערוצית

ניתן לחשוב על חלק זה כמו על השמיעה האנושית.

לבני האדם אמנם קשה להבין מתוך קטע אודיו של שני דוברים שהוקלט על ידי מיקרופון אחד, להבין מה אמר כל אחד מהדוברים, אך לעומת זאת, אם נבקש משני אנשים שלידנו לדבר לכיוווננו משני כיוונים שונים, נוכל להתרכז ובעצם כן להבין מה אמר כל אחד מהדוברים.

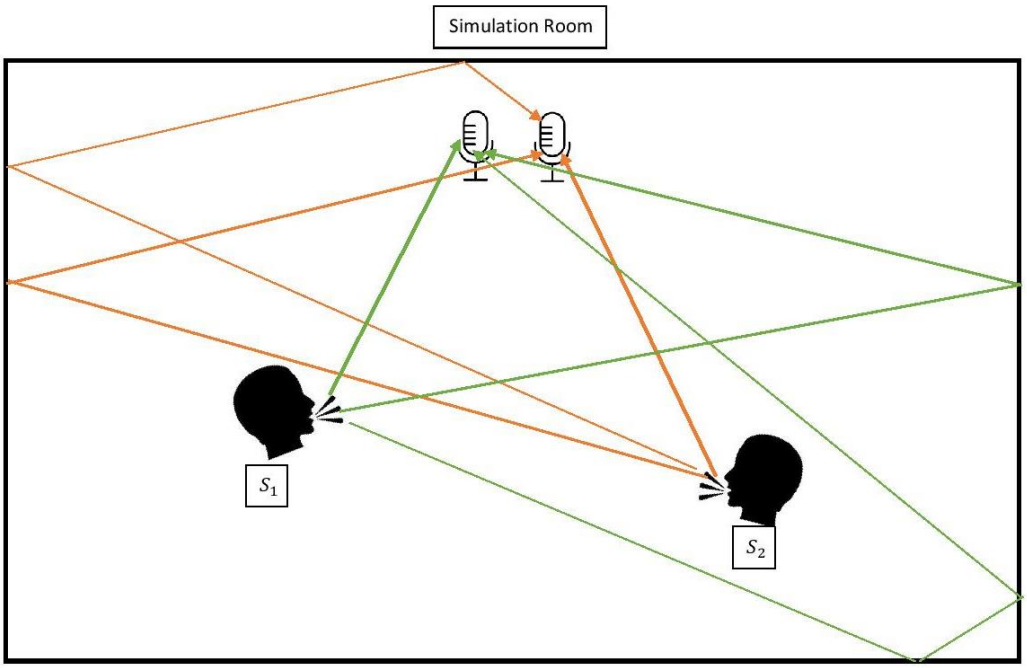
זאת בגלל שלבני אדם יש שמיעה כיוונית. יש לנו שתי אוזניים, שבעצם הן שני מיקרופונים, שכל הזמן מאזינות לסביבה.

לדוגמה, אם יש שני אנשים שמדברים לכיווני, האחד מצד ימין, והשני מכיוון שמאל, המוח ידע לזהות זאת (על ידי הפרשי זמני הגעת גלי הקול והפרשי העוצמות ועוד רמזים מרחביים) ומאיזה כיוון מדבר כל דובר, ויהיה אפילו מסוגל לבחור להאזין לאחד מהם.

אם בני אדם מסוגלים לעשות זאת, אין סיבה שלא ננסה לממש מנגנון כזה באמצעות מחשב.

לשם כך, נשנה מעט את הבעיה שנרצה לפתור.

נרצה מערכת שתקבל קלטים משני מיקרופונים במרחק קבוע (מדמה את האוזניים), ובאמצעותן תדע להפריד למה כל אחד אמר.

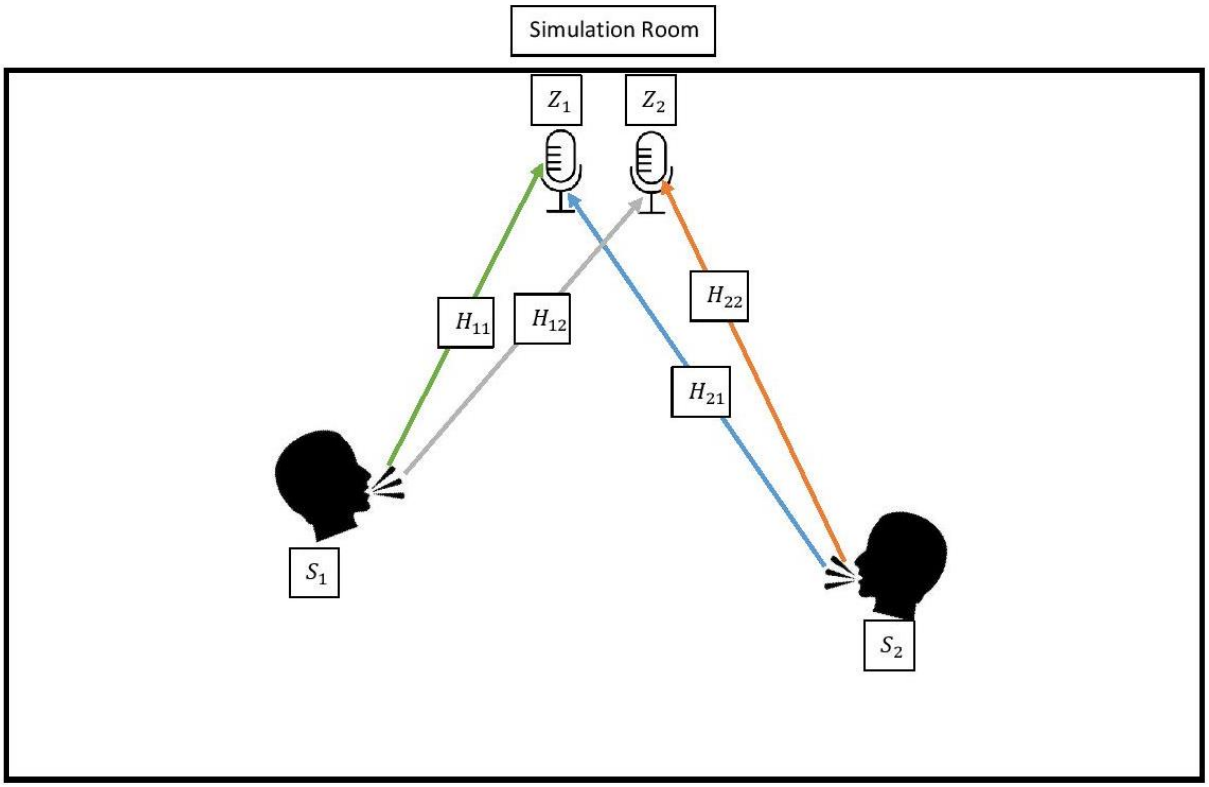



אפשרות ראשונה היא פשוט לבנות רשת שתקבל את שני הספקטרומים שמתקבלים בכל אחד מהמיקרופונים, ולבקש ממנה ליצור לנו מסיכות, בתהליך דומה למה שהוצג בחלקים הקודמים.

עם זאת, הבעיות שמהן ניסינו להימנע יכולות שוב להיווצר (הפרדה גרועה של דוברים בעלי ספקטרום דומה), ולכן נרצה לבנות מודל שלא תלוי בערכי הספקטרום, אלא במידע מרחבי בלבד.

## פורמליזציה של הבעיה החדשה

אנו עוסקים בדיבור של שני דוברים בחדר. אחת הבעיות שלנו כפי שהצגנו למעלה- היא שהדיבור היוצא מהפה שלנו הוא לא הדיבור הנכנס למיקרופונים הנמצאים בחדר מכיוון שהוא עובר בחדר ומתפשט לכל עבר עד ההגעה למיקרופון. נוכל להניח שהקול לנו עובר דרך מערכת $LTI$ עד שהוא מגיע למיקרופון והמערכת היא בעצם החדר שלנו.




באשר:

$s_i[n]$ זה קול הדיבור היוצא מדובר $i$.

$h_{ij}[n]$ זה המערכת שבא עובר קול הדיבור היוצא מדובר $i$ למיקרופון $j$.

$z_i[n]$ זה קו הדיבור המגיע למיקרופון $i$.

לפי הציור ניתן לתאר זאת ע"י המשוואות:

הדיבור המגיע למיקרופון 1 הוא: $z_1 = s_1 * h_{11} + s_2 * h_{21}$

הדיבור המגיע למיקרופון 2 הוא: $z_2 = s_1 * h_{12} + s_2 * h_{22}$

וכאשר נעבור למישור הSTFT נקבל:

הדיבור המגיע למיקרופון 1 הוא:

$$(6)\ Z_1 = STFT\{s_1 * h_{11} + s_2 * h_{21}\} = S_1 \cdot H_{11} + S_2 \cdot H_{21}$$

הדיבור המגיע למיקרופון 2 הוא:

$$(7)\ Z_2 = STFT\{s_1 * h_{12} + s_2 * h_{22}\} = S_1 \cdot H_{12} + S_2 \cdot H_{22}$$

נגדיר את ה- $RTF\ (Room\ Transfer\ Function)$ להיות היחס של האות המגיע למיקרופונים:

$$(8)\ RTF \triangleq \frac{Z_1}{Z_2} = \frac{S_1 \cdot H_{11} + S_2 \cdot H_{21}}{S_1 \cdot H_{12} + S_2 \cdot H_{22}}$$

לפי עיקרון $W - disjoint\ orthogonality$ שהסברנו אותו בתחילת הספר (בו בכל נקודה זמן תדר רק דובר אחד פעיל), נקבל תוצאה מאוד מעניינת:

אם דובר 1 פעיל ודובר 2 לא - כלומר $S_2 = 0$ ואז:

$$RTF|_{S_2=0} = \frac{S_1 \cdot H_{11} + 0 \cdot H_{21}}{S_1 \cdot H_{12} + 0 \cdot H_{22}} = \frac{S_1 \cdot H_{11}}{S_1 \cdot H_{12}} = \frac{H_{11}}{H_{12}}$$

מצד שני, אם דובר 2 פעיל ודובר 1 לא - כלומר $S_1 = 0$ ואז:

$$RTF|_{S_1=0} = \frac{0 \cdot H_{11} + S_2 \cdot H_{21}}{0 \cdot H_{12} + S_2 \cdot H_{22}} = \frac{S_2 \cdot H_{21}}{S_2 \cdot H_{22}} = \frac{H_{21}}{H_{22}}$$

בעצם , נקבל בזכות העיקרון הזה נקבל שהפונקציה הזאת אינה תלויה בספקטרום הדיבור עצמו אלא במאפייני החדר ומיקומי המיקרופונים/הדוברים בלבד.

אם ניזכר בבעיות שהצגנו לעיל – אחד הפתרונות שהצענו הוא הסתמכות על מודל שאינו תלוי בספקטרום ובדיוק הצענו זאת ע"י מודל שאינו תלוי בספקטרום הדוברים אלא במאפיינים מרחביים בלבד!

כעת, נכניס לרשת את הRTF שנוצר משני המיקרופונים, ונבקש מהרשת לייצר מסיכות שאותן נכפיל בספקטרום המקורי.

נשים לב שמדובר במודל חזק יותר, שכן הוא מקבל יותר מידע (קלט משני מיקרופונים) לעומת המודל הקודם (קלט ממיקרופון אחד).

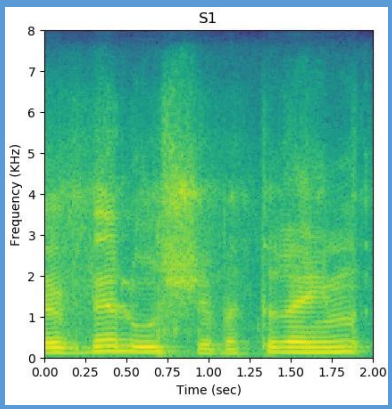
בנוסף, נשים לב שהפעם המסכות יוגדרו כהשוואה בין ההתמרות של קולות הדיבור היוצאים מהדוברים אלא כהשוואה בין קולות הדיבור המגיעים למיקרופון.

$$(9)\ binary_mask_1 = \begin{cases} 1, & |\mathrm{STFT}\{s_1 * h_{11}\}| \geq |\mathrm{STFT}\{s_2 * h_{21}\}| \\ 0, & |\mathrm{STFT}\{s_1 * h_{12}\}| < |\mathrm{STFT}\{s_2 * h_{22}\}| \end{cases}$$

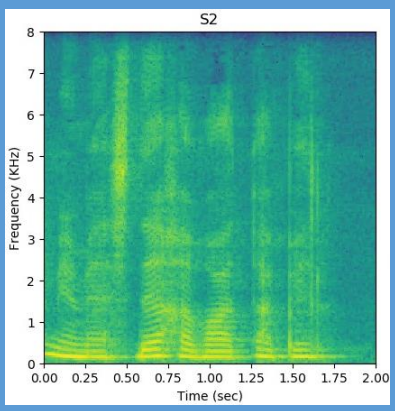
$$(10)\ binary_mask_2 = \begin{cases} 1, & |\mathrm{STFT}\{s_1 * h_{11}\}| \leq |\mathrm{STFT}\{s_2 * h_{21}\}| \\ 0, & |\mathrm{STFT}\{s_1 * h_{12}\}| > |\mathrm{STFT}\{s_2 * h_{22}\}| \end{cases}$$

שלב א'

$S_1$

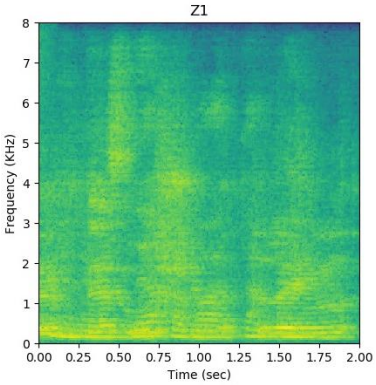



$S_2$

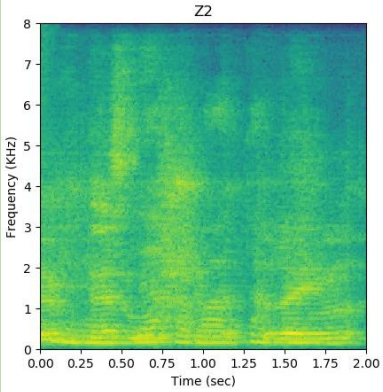



שלב ב'

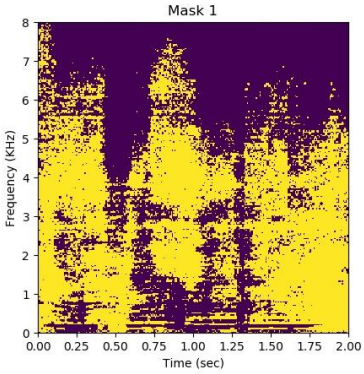
$Z_1$ לפי משוואה (6)



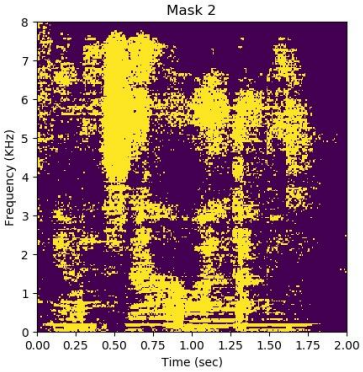

$Z_2$ לפי משוואה (7)




$binary_mask_1$ לפי משוואה (9)




$binary_mask_2$ לפי משוואה (10)



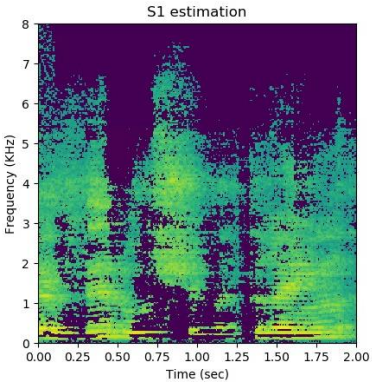

שלב ג'

$RTF$ לפי משוואה (8)

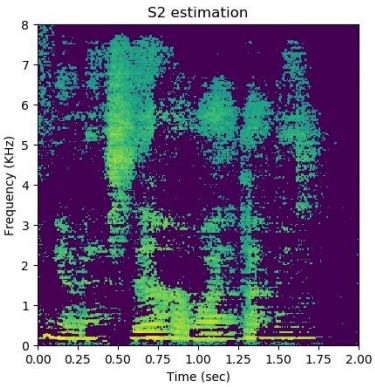
$RTF_{real}$

$RTF_{image}$

$\widehat{S_1} = binary_mask_1 * Z_1$




$\widehat{S_2} = binary_mask_2 * Z_1$



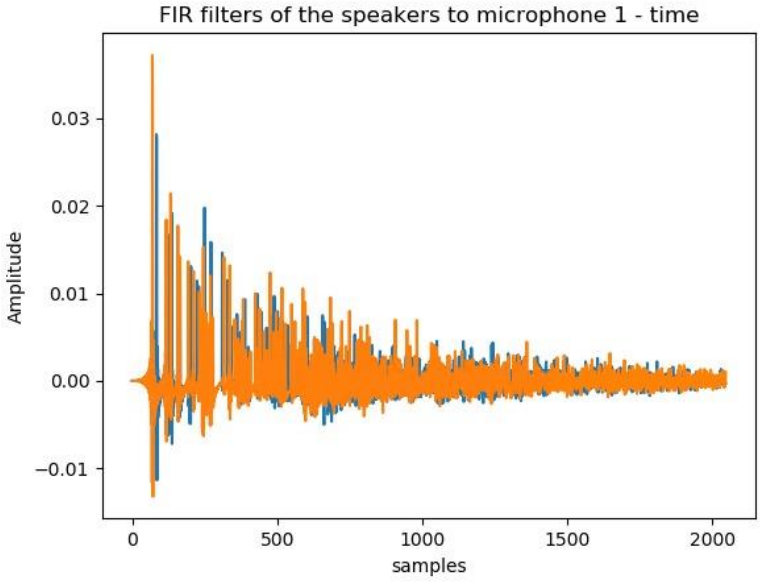
# בניית הDatabase במודל החדש

בכדי לבצע את האימון למערכת, אנחנו צריכים לבנות דאטאבייס שיכיל המון דוגמאות ואת התוצאות שלהן.

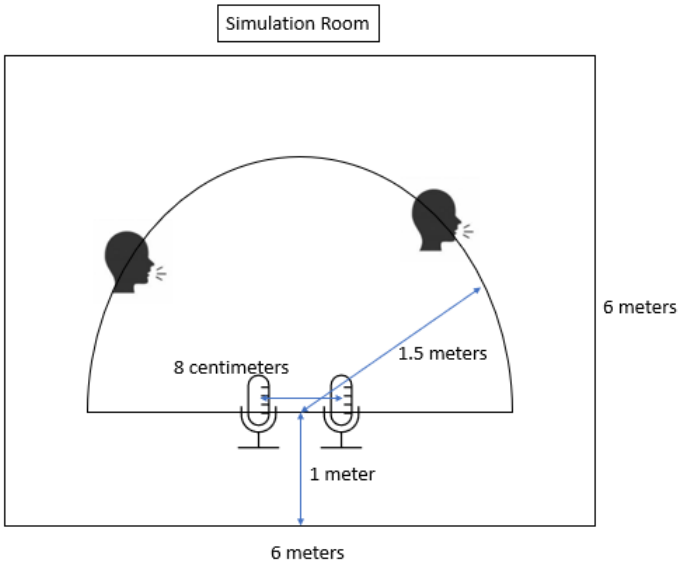
הבעיה נוצרת, כשבמאגר TIMIT אין דוגמאות של שני אנשים שהוקלטו בחדר עם שני מיקרופונים כמו בקונפיגורציה שהוצגה קודם, ולכן ניצור דוגמאות כאלה בעצמנו, תוך כדי שימוש בדוגמאות דיבור שכן קיימות במאגר TIMIT.

למעשה, קיימת ספרייה בפייתון בשם rirgen שמיועדת בדיוק בשביל זה. היא מקבלת קונפיגורציה של חדר, מיקרופונים ומיקומי דוברים, ומייצרת בעצמה את התגובה להלם של המערכת ממיקום הדובר ועד למיקרופון, בצורה שמשקפת די טוב את המערכות בעולם האמיתי.

דוגמה לשני מסננים שנוצרו על ידי הספרייה rirgen:




נשתמש בספרייה זו כדי לסמלץ שני דוברים בחדר כמתואר בציור.



מימדי החדר הם 6x6x2.5 (במטרים) והמיקרופונים מוצבים במרחק של 8 ס"מ אחד מהשני ובצורה אופקית, כשמרכזם בנקודה 3x1x1.5 (במטרים).

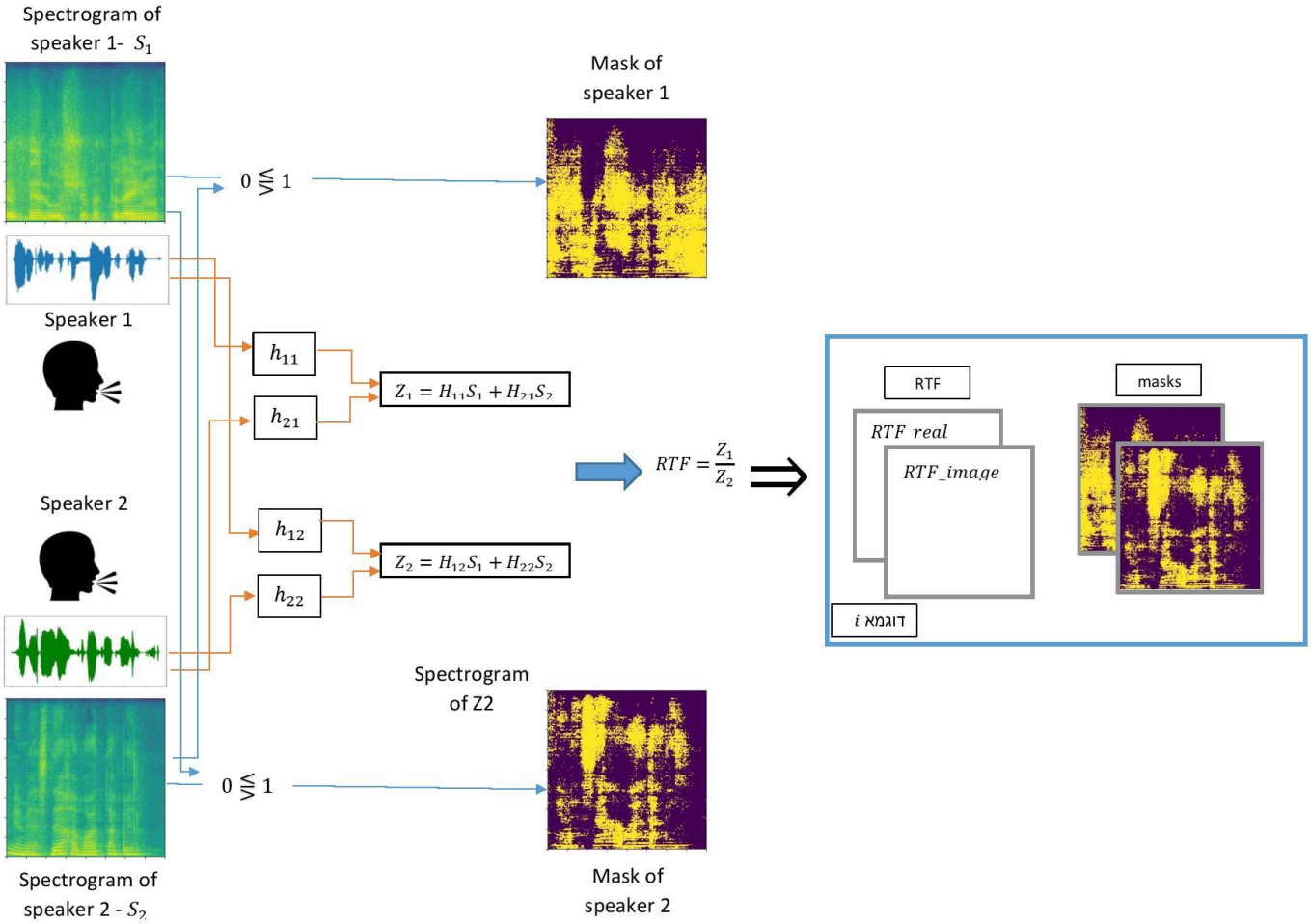
מיקום הדוברים מוגרל באופן רנדומלי, אך ברדיוס ממוצע של מטר וחצי ממיקום המיקרופונים.

בנוסף, וידאנו שזוויות הדוברים ביחס למיקום הרמקולים לא יהיו קרובות מדי, אלא שהדוברים יהיו רחוקים זה מזה.

כעת, נבקש מrirgen ליצור לנו תגובות להלם של המערכות הנדרשות, נייבא שתי דוגמאות דיבור רנדומליות ממאגר TIMIT ונבצע להן מעבר במערכות המתאימות, לקבלת האותות במיקרופונים.

נחשב את הRTF ואת המסיכות (שיהוו הכניסות והתיוגים לאימון המערכת) ונשמור זאת כדוגמה מוכנה לאימון.

נחזור על תהליך זה המון פעמים לקבלת דאטאבייס גדול ומגוון.



# תוצאות במודל החדש

את בדיקת המערכת שלנו ע"י מדד האיכות הרצנו על 200 דוגמאות. אלו מספיק דוגמאות בשביל לתת הערכה משקפת לביצועים שלנו. הערך שקיבלנו (בדציבלים) למדד הוא:

| | |
|---|---|
| 16.39723 | SIR |

מדובר בתוצאות מהקלטות שונות מאלה שהיו באימון, אבל תהליך יצירתן היה זהה לדוגמאות שבאימון, כלומר מדובר בהקלטות שעברו במערכת (סימולציה של חדר) ונכנסו למיקרופונים.

התוצאות מאוד דומות למודל הישן אך מעט יותר טובות. זה נובע כמו שהסברנו, משינוי המודל והסתמכות לא על צורת הספקטוגרמה אלא גם על מידע מרחבי שלא היה לנו קודם.

בנוסף, הרשת מתמודדת הרבה יותר טוב עם הקלטות שלא הוכנו "בתנאי מעבדה" אלא הקלטות מהיום יום שהוקלטו והן יותר רועשות, ובנוסף עם דוגמאות של שני דוברים בעלי צורת ספקטרום דומה. זה שיפור משמעותי לעומת הפעם הקודמת.

כמו כן, הרשת התמודדה גם כאשר שינינו כמה מפרמטרי הבעיה כמו גודל החדר ומיקומי מיקרופונים שונים ממה שהצגנו לה באימון.

# סיכום

**חלק א'**

בחלק זה ביצענו הפרדת דוברים באמצעות מיקרופון אחד.

המודל התבסס על מאפיינים של הלוג ספקטרום ועל כך שניתן להבחין בכך שיש שני מרכיבים בעלי מאפייני דיבור שונים.

יש למערכת כזו מספר חסרונות וקשה להגיע איתה לביצועים טובים על דוגמאות מחיי היום יום.

**חלק ב'**

בחלק זה ביצענו הפרדת דוברים באמצעות שני מיקרופונים.

המודל התבסס על מטריצת ה-RTF שמבטאת מאפיינים מרחביים של החדר בו בוצעה ההקלטה ואת מיקומי הדוברים והמיקרופונים.

המערכת יכולה להתמודד בצורה טובה יותר עם בעיות מחיי היום יום ועם דוגמאות קשות יותר.

**סיכום כללי**

ראינו איך באמצעות קונספטים של למידה עמוקה אפשר לעבד אותות דיבור, ובפרט לבצע הפרדת דוברים.

בשונה מפרויקטים דומים בתחום, הרשת בה השתמשנו היא רשת קונבלוציה, שהיא יעילה יותר, ובמקרה הזה מאפשרת גם עבודה בזמן אמת עם השהייה נמוכה.

# ביבליוגרפיה


- מאמר מנחה –

**Multi-talker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks** (Morten Kolbæk, Student Member, IEEE, Dong Yu, Senior Member, IEEE, Zheng-Hua Tan, Senior Member, IEEE, and Jesper Jensen.)

- ויקיפדיה - https://en.wikipedia.org
- python - https://www.python.org
- numpy - https://numpy.org
- pytorch- https://pytorch.org
- rir generator - https://github.com/ty274/rir-generator
- מאגר TIMIT - https://catalog.ldc.upenn.edu/LDC93S1
- matplotlib - https://matplotlib.org
- UNet - https://github.com/milesial/Pytorch-UNet