



Bar-Ilan University
אוניברסיטת בר-אילן

הפקולטה להנדסה

המסלול לעיבוד אותות

פרוייקט גמר תואר ראשון

תש"פ - 2020

2.5D Visual Sound

Based on a paper by:
Ruohan Gao | Kristen Grauman

שרה ארנסט ונעם קורנגוט

מנחה: יוחאי ימיני

מנחה אקדמי: פרופ' שרון גנות

הקדמה

4 הצגת הבעיה

5 הצגת הפתרון

6 פורמולציה של הבעיה

רקע עיוני

7 התמרת פוריה לזמן קצר – STFT

9 רשתות נוירונים ורשתות קונבולוציה

12 gradient descent | loss-12

פתרון

16 תיאור ה database

19 תיאור המודל

19 רשת unet

20 רשת resnet

23 תוצאות

28 סיכום

29 נספחים

הקדמה

אודיו בינוראלי זהו אודיו בעל שני ערוצים כך שהוא מספק למאזין תחושה כאילו הוא היה נוכח בסיטואציה ומקשיב באמצעות אוזניו. זהו בעצם של "סאונד תלת מימדי" ומאפשר חווית שמיעה עשירה של הסצנה.

עם זאת, הקלטות בינוראליות אינן זמינות והן דורשות מומחיות וציוד מקצועי ויקר. במאמר "2.5D visual sound"¹ (שנכתב במסגרת מחקר של facebook ואוניברסיטת טקסס) הוצעה שיטה להמרת אודיו חד ערוצי (מונו) לאודיו בינוראלי על ידי מינוף וידאו. הצעה זו מבוססת על העיקרון שמידע ויזואלי טומן בחובו רמזים מרחביים משמעותיים החסרים באודיו מונוראלי אך קשורים אליו מאוד.

בפרוייקט זה נממש את הרעיון שהוצע במאמר באמצעות רשת נוירונים עמוקה (CNN) שתלמד להמיר את אות השמע החד ערוצי לאות שמע בינוראלי על ידי המידע החזותי.

¹ https://www.cs.utexas.edu/~grauman/papers/CVPR19_2.5d-visual-sound.pdf

הצגת הבעיה

אדם השומע צלילים שונים ממקורות שונים מסוגל למקם (באופן משוער) את המיקום של מקור הצליל במרחב.

דבר דומה קורה גם כאשר אדם צופה בסרט בבית קולנוע: הרמקולים הפזורים באולם משמיעים את הסאונד לפי מיקום האובייקט על המסך כלומר, אובייקט הנמצא בצד ימין של המסך יושמע מרמקול ימין. כך יכול הצופה לזהות את מיקום האובייקטים על המסך משמיעה בלבד.

פעולה זו מתאפשרת מכיוון שהאודיו הוקלט במספר ערוצים הנשלחים לרמקולים נפרדים הפזורים בחלל החדר, כך שכל רמקול מוציא סאונד שונה. סאונד מסוג זה נקרא צליל היקפי (Surround Sound).

סאונד מסוג זה ממומש בסרטים, מערכות קול דיגיטליות, קולנוע ביתי, משחקי וידאו, משחקי מחשב וכו'.²

הקלטת אודיו זה דורשת ציוד יקר ומומחיות ולכן כיום משתמשים בעיקר בהקלטת צליל חד ערוצי (מונוראלי).

צליל המוגדר כ'מונו' את דיבור אשר הוקלט באמצעות מיקרופון בודד, ולכן לא מספק תחושת מרחביות.³ אף אם משתמשים במספר מיקרופונים בעת ההקלטה ובסופו של התהליך כל הערוצים מתאחדים לערוץ אחד - ההקלטה הסופית נחשבת כהקלטת 'מונו'.

טלפונים חכמים ומצלמות שנעשה בהם שימוש רב בחיי היום-יום יוצרים סאונד ללא תחושה מרחבית. אומנם הם משתמשים לרוב ביותר ממיקרופון אחד, אך המיקרופונים ממוקמים באופן שרירותי כך שהם לא יוצרים תחושה מרחבית.

לכן, כאשר אדם צופה בסרטון המלווה בשמע שהוקלט בטלפון נייד לא יוכל למקם את הדוברים רק באמצעות חוש השמיעה. אודיו שהוקלט על ידי ערוץ סאונד בודד אינו שומר על המידע המרחבי. עולה השאלה כיצד ניתן לספק לשומע תחושת שמע מרחבית מאודיו מונוראלי.

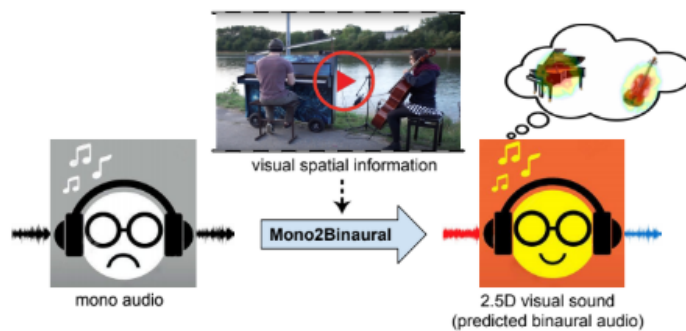
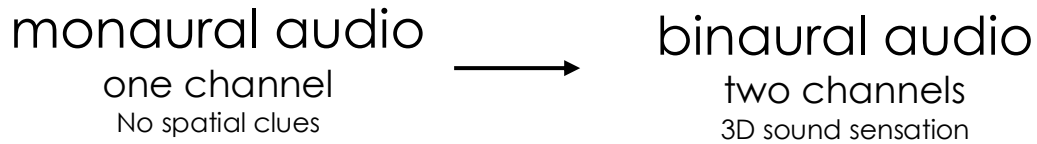
²

https://he.wikipedia.org/wiki/%D7%A6%D7%9C%D7%99%D7%9C_%D7%94%D7%99%D7%A7%D7%A4%D7%99

³ <https://en.wikipedia.org/wiki/Monaural>

הצגת הפתרון

מטרת הפרוייקט היא ליצור מודל שיהיה מסוגל לחזות אודיו דו-ערוצי באמצעות מידע ויזואלי המגיע מהסרטון הנלווה לאודיו.



איור 1: יצירת צליל ויזואלי (2.5D visual sound) ע"י הזרקת מידע מרחבי החבוי בפריימים המלווים אודיו מונוראלי

נעשה זאת לפי הגישה שהוצגה במאמר - מיפוי אודיו מונוראלי לאודיו בינוראלי ע"י וידאו. לצורך בניית המודל נשתמש ברשת קונבולוציה (deep convolutional neural network) ולאיימון המודל נשתמש בוידאו שאינו מתוייג (unlabeled). הרעיון המרכזי שעליו מתבססת בניית המודל הוא שהזרקת פריימים חזותיים עשויה לחשוף רמזים מרחביים משמעותיים, שאף שהם חסרים במפורש באודיו בעל ערוץ יחיד, הם קשורים אליו מאוד. כלומר, המידע החזותי עוזר "למנף" את ערוץ השמע השטוח לצליל מרחבי.

נשתמש בארכיטקטורה של של מקודד – מפענח (U-net). הרשת תקבל כקלט סיגנל חד ערוצי ופריים ויזואלי הנלווה אליו ותנסה לחזות סיגנל בינוראלי עפ"י הרמזים המרחביים החבויים במידע הויזואלי. הפלט המתקבל נקרא צליל ויזואלי 2.5D. כאשר נאזין למוצא נצפה לחוש את המיקום של מקורות השמע כפי שהם מוצגים בוידאו.

רקע עיוני

עיבוד במישור התדר

הצגת הפונקציה במישור התדר עשויה להקל עלינו בכיתוח האות. לדוגמא, אם ידוע לנו כי האותות המרכיבים את הגל המתקבל מורכבים מתדרים בתחומים שונים שאינם חופפים, ניתן בקלות להפריד את האות המתקבל לאותות המרכיבים אותו תוך הפעלת מסנני BP על האות.

התמרת פורייה לזמן קצר (Short Time Fourier Transform)

הגדרה: התמרת פורייה לזמן קצר מוגדרת על ידי:

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

$x[n]$ - האות בזמן בדיד אותו אנו מעוניינים להתמיר.

$w[n]$ - היא פונקציית חלון כלשהי (הנקראת חלון אנליזה) באורך L_h .

משמעות ההתמרה היא שבכל פעם לוקחים חלק מהאות (שנוצר על ידי ההכפלה בחלון) ומבצעים לו DTFT.

ככל שהחלון גדול יותר - נקבל רזולוציה טובה יותר של האות בתדר, משום שההתמרה של החלון קרובה יותר להלם (עבור חלון אינסופי אנו מקבלים למעשה את התמרת ה-DTFT הרגילה של האות). ככל שהחלון קצר יותר הרזולוציה תהיה טובה יותר בזמן.

חשיבותה של התמרת פוריה לזמן קצר מתבטאת ביישומים בהם מבצעים אנליזה ספקטרלית לאותות שההרכב התדרי שלהם משתנה כל הזמן, כמו אותות דיבור ומוסיקה. אלו אותות שאינם סטציונריים, ובהם אין משמעות לחישוב התמרת פורייה עבור זמן ארוך, אלא לחלוקה למקטעים קטנים וחישוב של התמרת פורייה בכל מקטע.

לדוגמא: עבור האות $x[n] = \cos(\omega_0 n)$ נעדיף חלון ארוך שיזהה את המחזוריות של ה-cos ויתן לנו את הפיק בתדר כמה שיותר טוב. אם ניקח חלון קטן מידי, לא נוכל לזהות את המחזוריות המאפיינת . cos

עבור אות שיש לו פיקים בזמן, אם ניקח חלון גדול מידי לא נזהה את הפיק והוא לא ישפיע על ההרכב התדרי של האות, אבל אם ניקח חלון קטן, נשים לב לשינוי בתדר.

בדרך כלל לא נעבוד עם התמרת ה-DTFT הרציפה, אלא ניקח דגימות שלה במרווחים של $\frac{2\pi}{M}k$, ונקבל התמרת DFT עבור כל חלק של האות שמובפל בחלון אנליזה $w[n]$:

$$X_{STFT}[n, k] = X_{STFT}(e^{j\omega}, n) \Big|_{\omega=\frac{2\pi}{M}k} = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\frac{2\pi}{M}km} \quad k = 0, 1, \dots, M-1$$

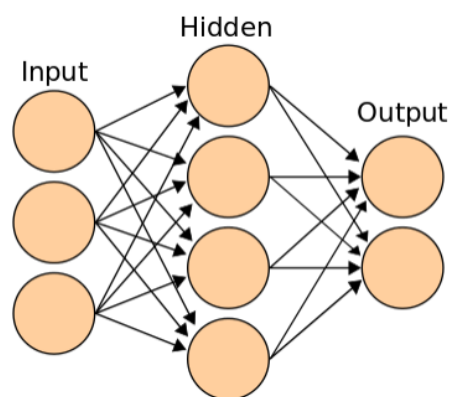
נזכור כי דגימה בתדר היא קיפול בזמן, כלומר, נתייחס לאות בזמן כאילו הוא היה מחזורי. את M בדר"כ נבחר שיהיה לפחות באורך החלון, על מנת להימנע מקיפול בזמן.

אם נניח כי האות אינו משתנה כל כך מהר, אזי לא צריך לבצע אנליזה לאות כל דגימה אלא ניתן לראות את ההרכב התדרי שלו מידי כמה דגימות, כלומר: נבצע דגימה נוספת של $X_{STFT}[n, k]$ פי R :

$$X[nR, k] = \sum_{m=-\infty}^{\infty} x[m]w[nR-m]e^{-j\frac{2\pi}{M}km} = DFT_M \{x[m]w[nR-m]\}$$

רשת נירונים (Deep Neural Network) הוא מודל מתמטי חישובי שפותח בהשראת תהליכים קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה. הרשת מקבלת קלט אשר עובר דרך שכבות עיבוד שונות הנקראות "שכבות חבויות" (Hidden Layers). כל שכבה מורכבת מתאים הנקראים נירונים. כל שכבה מבצעת חישוב פשוט יחסית. לכל תא מספר כניסות ויציאה אחת שערכה הוא פונקציה כלשהי של הכניסות. יציאה של תא מסוים יכולה להזין מספר כניסות של תאים שונים.

צורת הקישור בין היחידות, המכילה מידע על חוזק הקשר (משקלים), מדמה את אופן חיבור הנירונים במוח. בין כל שתי שכבות, יש מספר רב של דפוסי חיבור אפשריים. לבסוף השכבה האחרונה מוציאה את המידע המעובד כפלט.



איור 2: הדגמת אופן הפעולה של רשת נירונים

לרשת מספר סוגי שכבות:

1. שכבת כניסה
2. שכבות חבויות
3. שכבת יציאה

השכבה המקבלת נתונים חיצוניים היא שכבת הכניסה. השכבה המייצרת את התוצאה האולטימטיבית היא שכבת הפלט. ביניהם אפס שכבות חבויות (Hidden Layers) או יותר.

4

https://he.wikipedia.org/wiki/%D7%A8%D7%A9%D7%AA_%D7%A2%D7%A6%D7%91%D7%99%D7%AA_%D7%9E%D7%9C%D7%90%D7%9B%D7%95%D7%AA%D7%99%D7%AA

רשת נזירונים מאופיינת על ידי:

חיבורים - אופן החיבור בין הנזירונים ברשת.

משקלים - השיטה הקובעת את משקלי החיבורים בין הנזירונים.

פונקציית האקטיבציה - פונקציה הפועלת על הנזירון עצמו ועשויה להיות שונה בכל שכבה.

פונקציות האקטיבציה הן לא לינאריות, תכונה המקנה לרשת אפשרות

לייצוג תחום רחב יותר של פונקציות.

רשתות קונבולוציה (convolutional neural network - CNN)⁵

רשת קונבולוציה זו היא סוג של רשת נזירונים עמוקה המשמשת בעיקר לניתוח מידע ויזואלי כך שהקלט

של הרשת הוא לרוב תמונה.

מקור השם מהפעולה המתמטית אותה הרשת מפעילה - קונבולוציה.

תוצאת הקונבולוציה היא מטריצה שתחושב על ידי החלקת טנזור הפילטר על פני טנזור הקלט. בכל

אזור חפיפה מחשבים את סכום המכפלות של האלמנטים החופפים בין הטנזורים וזה יהיה אלמנט

בטנזור התוצאה.

רשת קונבולוציה מורכבת משכבות כניסה ויציאה, ומשכבות נסתרות המורכבות משרשרים של

שכבות קונבולוציה, pooling, ושכבות אקטיבציה. נפרט על שכבות אלו:

שכבות ברשת CNN:

⁶Convolutional layer

"פילטר" עובר על התמונה, סורק כמה פיקסלים בכל פעם ויוצר 'מפת תכונות'.

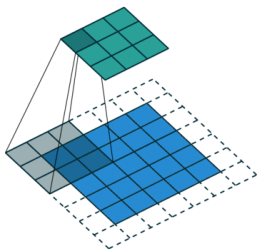
ברשת שלנו נשתמש ב Conv 2d - הקלט לשכבה זו הוא נתונים תלת מימדיים שגודלם הוא:

(עומק התמונה) x (רוחב התמונה) x (מספר הערוצים בתמונה)

והפלט הוא נתונים תלת מימדיים. ה 2d מסמן שה-kernel

עובר לאורך שני צירים. (שורות ועמודות).

פרמטרים חשובים של שכבה זו הם:



איור 3: conv2D

⁵ https://en.wikipedia.org/wiki/Convolutional_neural_network

⁶ <https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/>

- 'kernel_size' - kernel (הריבוע האפור) הוא טנזור משקולות שמוכפל בכל חלק באינפוט.
- פילטרים - עומק הפילטר המשמש לקונבולוציה הוא כמספר ערוצי הכניסה. מספר הפילטרים השונים בהם משתמשים מגדיר את מספר ערוצי המוצא⁷. ככל שמעמיקים ברשת מספר kernels אותם הרשת לומדת גדל⁸.
- Strides - על מנת לעשות downsampling נגדיר פרמטר "צעד" של הקונבולוציה לאורך ציר x ו-y של האינפוט (איור 3). ברשת זו נשתמש ב (2,2) strides. לאחר שכבה זו המימד קטן פי 2 וכמות הפיצ'רים גדלה. בתהליך למידה, המודל לומד את המשקלים האופטימלים. שכבות אלו מכונות שכבות קונבולוציה לפי מוסכמה למרות שבאופן מתמטי, זו מכפלה (cross correlation product ,dot product).

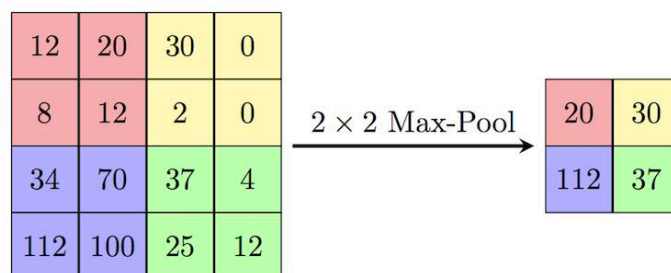
-Pooling layer

שכבת pooling מציגה גישה נוספת ל downsampling של הפיצ'רים. היא מפחיתה את כמות המידע בכל תכונה המתקבלת בשכבת הקונבולוציה תוך שמירה על המידע החשוב ביותר.

שתי מטודות מרכזיות של pooling הן:

average pooling - לוקח את הערך הממוצע של כל קבוצת ניוונים בשכבה הקודמת.

max pooling - לוקח את הערך המקסימלי של כל קבוצת ניוונים בשכבה הקודמת.



-activation function

ברשת ניוונים על המוצא של כל ניוון מופעלת פונקציה לא ליניארית.

⁷ https://keras.io/api/layers/convolution_layers/convolution2d/

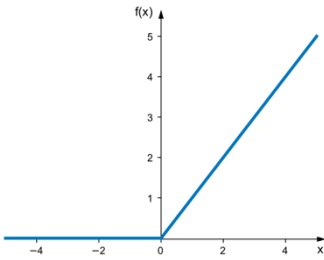
⁸ <https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>

מספר דוגמאות לפונקציות אקטיבציה:

4. **rectified linear activation function** או בקיצור **ReLU** היא

פונקציה לא לינארית פשוטה:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



5. **Leaky ReLU** -

אנו נשתמש ב **Leaky ReLU** שמאפשרת גרדיאנט קטן כאשר היחידה לא

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0.01x & \text{otherwise.} \end{cases}$$

אקטיבית:

6. **sigmoid** -

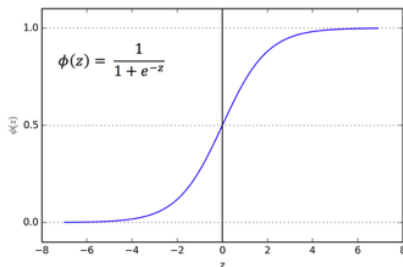
לוקח מספר בעל ערך ממשי ומעביר אותו לטווח שבין 0 ל 1. בפרט, מספרים

שליליים גדולים הופכים ל 0 ומספרים חיוביים

גדולים הופכים ל 1. אצלנו, עבור השכבה

האחרונה נשתמש כפונקציית אקטיבציה

בסיגמואיד.⁹



10 **Batch normalization** -

כדי להגביר את היציבות של הרשת ולזרז את תהליך האימון, **Batch normalization** מנרמל

את הפלט של שכבת האקטיבציה הקודמת על ידי חיסור של ממוצע batch וחלוקה בסטיית

התקן שלו.

אנחנו מנרמלים על לזרז את הלמידה של השכבות החבובות.

⁹ <https://cs231n.github.io/neural-networks-1/>

¹⁰ <https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c>

אימון הרשת

הליך הלמידה של רשת נוירונים מתבסס במהותו על אלגוריתם 'Gradient Descent'. המבנה הרב שכבתי גורם לנגזרות התאים להסתעף ולהכיל את כל הנגזרות החלקיות של התאים שמזינים אותם.

לכן, לשם האימון נשתמש באלגוריתם BACK-PROPAGATION. נתאר את שני אלגוריתמים אלו:

אלגוריתם Gradient Descent

נגדיר את השגיאה של הרשת על ידי:

$$\text{err} = \frac{1}{2} \cdot \sum_i (y_i - o_i)^2$$

כאשר y היא התוצאה הרצויה ואילו o הוא החיזוי. נרצה להביא למינימום את השגיאה על ידי מציאת המשקלים המתאימים.

נסתמך על העובדה שהליכה מנקודה מסויימת במרחב לכיוון המנוגד לגרדיאנט באותה נקודה תקרב אותנו לנקודת המינימום יותר מאשר כל כיוון אחר.

נאתחל את המשקלים באופן רנדומלי ונעדכן את המשקלים לפי הנוסחה הבאה:

$$w_{n+1} = w_n - \gamma \frac{d \text{err}(w)}{dw} (w_n)$$

כאשר γ הוא קצב הלמידה (learning rate).

עבור קצב גבוה מדי אנו עלולים שלא להתכנס, ואילו עבור קצב נמוך מדי זמן הריצה יהיה ארוך מאוד. על ידי ריצה איטרטיבית נקבל את המשקלים אשר יתנו שגיאה מינימלית.

BACK-ROPOGATION

הרעיון המרכזי של אלגוריתם זה הוא חישוב השגיאה של מוצא הרשת ביחס לערך אותו אנו רוצים שתניב ועדכון המשקלים של השכבה החבויה האחרונה. לאחר מכן נשתמש בשגיאה של המוצא כדי למצוא ביטוי לשגיאה של התאים בשכבה האחרונה, ובהתאם נעדכן את התאים של השכבה שלפניה וכן הלאה.

אלגוריתם זה מתבסס על כלל השרשרת לנגזרות המאפשר למצוא את הנגזרת של פונקציה המורכבת ממספר פונקציות אחרות כאשר הפונקציות גזירות בתחום הגדרתן.

לדוגמא: עבור $h(x) = f(g(x))$, $h'(x) = f'(g(x)) \cdot g'(x)$.

$$\frac{d h}{d x} = \frac{d f}{d g} \cdot \frac{d g}{d x} \quad \text{כלומר}$$

חלוקת סט האימון

חישוב מדויק של הגרדיאנטים לעדכון המשקולות מצריך את חישוב הגרדיאנטים עבור כל דגימת אימון ואז מיצוע של כל התוצאות. מכיוון שלרוב רשתות דורשות כמות גדולה ביותר של מידע, תהליך זה אורך זמן ריצה רב. על מנת לייעל את החישוב ולקצר את זמן הריצה מקובל לחלק את סט האימון לחלקים הנקראים batch-ים (אצווה). כל ריצה על 'batch' מתבססת על הנתונים של זו שלפניה, כך שבשלב הסופי מתקבלים משקלים משופרים.

בנוסף, מקובל הריץ את האימון יותר מפעם אחת. כל הרצה של סט אימון שלם נקראת 'epoch' כאשר כל epoch מורכב ממספר batch-ים.

החיסרון בשיטה זו הוא שהמשקלים המתקבלים מאימון על 'batch' מדויקים פחות מהמשקלים שהיו עשויים להתקבל מריצה על כל סט האימון.

כאשר מאמנים רשת נירונים קיימת סכנה שהרשת "תשנן" את הדוגמאות הספציפיות שהיא מתאמנת עליה ובעתיד לא תצליח לפענח מקרה כללי יותר, תופעה זו נקראת 'Overfitting'. לכן, נהוג להקצות קבוצה של פריטים אשר הרשת אינה מתאמנת עליהן, ולמדוד את פונקציית ההפסד על הקבוצה הזו, סט ה 'validation'. סט דוגמאות זה משמש גם לכוונון ההיפרפרמטרים (כלומר הארכיטקטורה) של מסווג.¹¹

¹¹ https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

פורמולציה של הבעיה

הרשת העמוקה MONO2BINAURAL לוקחת את האודיו המונוראלי $x^M(t)$ והפריימים היוזואלים

בקלט וחוזת את $x^D(t)$.

ראשית נגדיר :

- האות המתקבל במיקרופון אוזן ימין -

$$x^R(t)$$

- האות המתקבל במיקרופון אוזן שמאל -

$$x^L(t)$$

- האות המונו המעורב -

$$x^M(t)$$

כאשר בשלב האימון יש לנו את $x^L(t)$ ואת $x^R(t)$ ואילו בשלב הבדיקה יש לנו את מונו $x^M(t)$.

בשלב האימון את המונו המעורב $x^M(t) = x^L(t) + x^R(t)$ נלקח בקלט.

המטרה היא לפצל את אות המונו לשני ערוצים נפרדים $\tilde{x}^R(t)$ ו $\tilde{x}^L(t)$ - בהתבסס על מיקום צליל המקור.

עם זאת, במקום לחזות את שני הערוצים ישירות, אנו נחזה את ההבדל בין שני הערוצים,

$$x^D(t) = x^L(t) - x^R(t), \text{ מכיוון שהוא נותן תוצאה טובה יותר לפי המאמר}^{12}.$$

נעשה זאת על ידי רשת CNN שתחזה מסכה מרוכבת M כך שכשנבצע הכפלה מרוכבת של

ספקטרוגרמת הקלט עם המסכה המרוכבת שחזינו כלומר: $X^D = MX^M$ נקבל את ספקטרוגרמה

מרוכבת X^D עבור $x^D(t)$ - האות הדיפרנציאלי.

לבסוף, באמצעות $x^D(t)$ נשחזר את שני הערוצים - פלט השמע הבינאורלי:

$$\tilde{x}^L(t) = \frac{x^M(t) + x^D(t)}{2}, \quad \tilde{x}^R(t) = \frac{x^M(t) - x^D(t)}{2}$$

¹² מצ"ב בנספחים

פתרון הבעיה

הפרויקט ממומש בשפת Python בספריית keras.

תיאור ה database

על מנת לאמן את הרשת השתמשנו ב dataset הנקרא FAIRPlay ומכיל אודיו בינוראלי המלווה בוידאו תואם באורך של 5.2 שעות.

דאטה זה הוקלט בחדר הקלטות ע"י מיקרופונים מיוחדים המחקים את האופן שבו אדם שומע, כפי שניתן לראות באיור 2. כמו כן, המצלמה מוקמה ביחס למיקרופונים באופן המדמה אדם הנמצא בחדר וצופה בסיטואציה. (איור 2)



איור 2: חדר הקלטות שבו נאספה הדאטה

ההקלטה מורכבת משילובים שונים של כלי נגינה (כגון צ'לו, גיטרה, תוף, נבל, פסנתר, חצוצרה, בס וכו'), של מספר הנגנים ושל מיקומם בחלל.

כך הוקלט אודיו בינוראלי – בעל שני ערוצי שמע עם וידאו בקצב 30 fps.

הדאטה חולקה למקטעים קצרים באורך 10 שניות כל אחד כך שלבסוף התקבלו 1871 קליפים קצרים. במהלך הtest השתמשנו גם בקליפים בעלי ערוץ שמע יחיד (מונו) אשר נלקחו מ-YOUTUBE.

:Data Generator

על מנת לטעון את הדאטה למודל השתמשנו ב 'Data Generator' שטוען את הדאטה ב batch-ים. הדאטה שטענו לרשת היא אודיו באורך 0.63s ופריים (תמונה) תואם בגודל 224 x 448 x 3 .

על מנת לייצר פריימים דגמנו את הוידאו בקצב של 10fps באמצעות תוכנת FFMPEG.

בData Generator נבחר הפריים המתאים ע"י מיצוע בין הזמן ההתחלתי והזמן הסופי של מקטע האודיו ולקיחת פריים מהאמצע.

ההצדקה לשימוש בפריים בודד במקום בוידאו עצמו היא שהוידאו אינו משתנה בצורה משמעותית לאורך זמן קצר.

לאחר טעינת הדאטה בצענו תהליך עיבוד ('pre processing') את הפריימים חתכנו באופן רנדומלי לגודל $224 \times 448 \times 3$ וביצענו 'augmentation' שכלל שינויי בהירות שונים.

מתוך קטעי האודיו הנמצאים ב dataset לקחנו מקטעים רנדומליים באורך של 0.63 שניות.

את מקטע זה דגמנו בקצב דגימה של 16kHz מכן העברנו אותו בבלוק של STFT. התמרת ה-STFT מספקת את המידע התדרי של האות המשתנה לאורך זמן. כפי שהסברנו ברקע העיוני, הטרנספורמציה מתבצעת על ידי חלוקת האות למקטעים קצרים באורך שווה וחישוב טרנספורמציות פורייה בנפרד על כל קטע. חשבנו את ההתמרה באמצעות חלון 'Hann' באורך 25 ms עם 'hop size' של 10 ms ו FFT בגודל 512. את החלק הממשי והחלק המדומה של ה-STFT משרשרים לאורך ציר ה-channel.

המוצא של ה Data Generator נכנס לרשת ה ResNet וה- Unet עליהן נרחיב בהמשך.

חשוב לציין כי במהלך ה test בו ניסינו לשחזר מקטע של וידיאו בן 10s לקחנו מקטעים בני 0.63s עם חפיפה בגודל 0.05s ביניהם ובתהליך האיחוד מיצענו בין המקטעים החופפים.

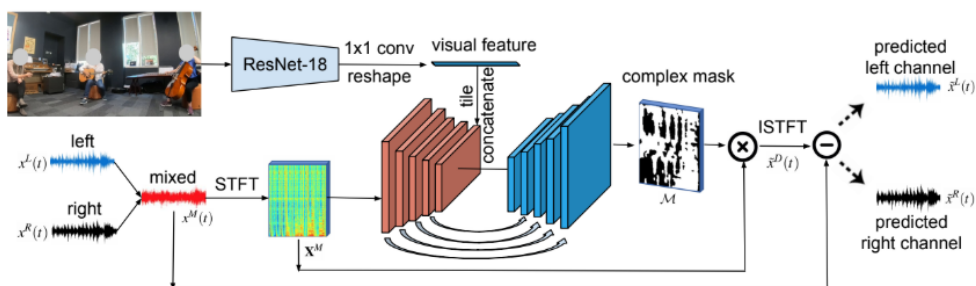
כמו כן במהלך הtest הכנסנו גם דאטה שבמקור היא מונואורלית כך שלא היה צריך לאחד בין הערוצים שלה.

הכניסה לרשת מורכבת משני קלטים:

1. ספקטוגרמה של אודיו חד ערוצי המורכב מסכימת שני הערוצים $x^L(t)$ ו $x^R(t)$ של האודיו

$$x^M(t) = x^L(t) + x^R(t) \text{ על: stft כלומר}$$

2. דאטה ויזואלית מתאימה (פריים בודד).



איור 3: המודל

המודל מורכב משני "ענפים" מקבילים:

ענף ויזואלי וענף אודיו. נרחיב על כל אחד מהם:

visual branch

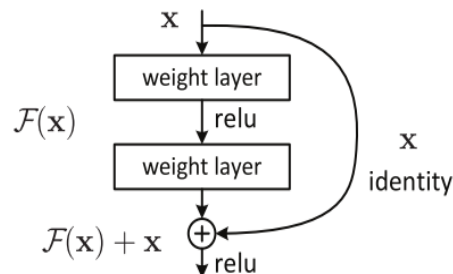
ResNet

Residual Neural Network היא רשת נירונים.

הרעיון המרכזי של ResNet הוא שמוצא של שכבה מועברת במהירות לשכבה עמוקה יותר ברשת הניורונים כדי ללמוד את משקליה.

הדבר נעשה על ידי שימוש ב"דילוגים" או "קיצורי דרך" (כפי שמתואר באיור 4) כדי לקפוץ מעל כמה שכבות המכילות אי-לינאריות בין לבין. תהליך זה מזרז את הלמידה ומקטין את שגיאת האימון.

ל ResNet נעביר את וקטור תמונות התואמת למקטעי הוידאו הנבחרים (סהכ וקטור בגודל $448 \times 224 \times 3$) לאחר המעבר ב ResNet, הפייצ'רים הוויזואלים מושטחים ומשוכפלים לווקטור תמונות חזותיות יחיד לשימוש עתידי שגודלו $8 \times 2 \times 784$.



איור 4: בלוק ברשת ResNet

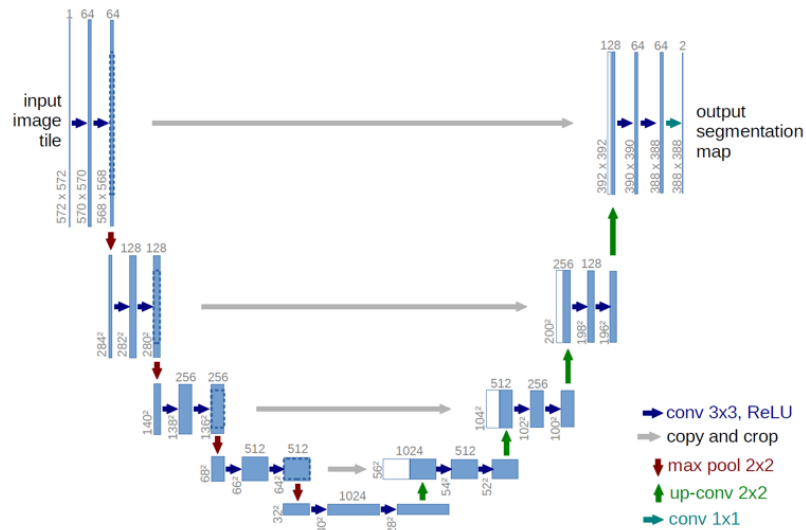
במודל שלנו, הוצאת הפייצ'רים הוויזואליים נעשית באמצעות רשת ResNet-18 שאומנה על מאגר המידע 'ImageNet'. 'ImageNet' הוא בסיס נתונים חזותיים גדול המיועד לזיהוי אובייקטים חזותיים במחקר ומורכב מיותר מ-20,000 קטגוריות כאשר כל קטגוריה מכילה מאות תמונות. אנו נקח את המוצא של השכבה לפני האחרונה (באחרונה כבר יש labels) ונשרשר עם האודיו בשכבת ה'neck' ברשת ה'U-Net'.

audio branch

U-Net

היונט היא רשת קונבולוציה. הארכיטקטורה של הרשת היא בצורת האות 'U' ומכאן מקור השם. הרשת כוללת 3 חלקים:

1. Contraction (כיווץ)
2. Bottleneck ('צוואר בקבוק')
3. Expansion section (קטע הרחבה)



איור 4: דוגמא לרשת U-Net

החלק של ה'Contraction' עשוי מבלוקים רבים, אשר כל אחד מהם לוקח קלט, מיישם שבבת קונבולוציה ואז מצמצם את מימד השבבה. מספר התכונות מכפיל את עצמו אחרי כל בלוק, כך שארכיטקטורה זו יכולה ללמוד מבנים מורכבים בצורה יעילה.

ברשת שלנו, האודיו (וקטור בגודל $256 \times 64 \times 2$) מועבר דרך 5 שבבות קונבולוציה כדי שהרשת תחלץ פייצ'רים של האודיו— זהו הצד של ה'encoder' של היונט (Contraction). כפי שניתן לראות בתמונה המבנה של הencoder הולך כך:

$\text{Conv2D} \rightarrow \text{leakyRelu} \rightarrow \text{BatchNorm} \rightarrow \text{Conv2D} \rightarrow \text{leakyRelu} \rightarrow \text{BatchNorm} \rightarrow \dots$

conv2d_1 (Conv2D)	(None, 128, 32, 64)
leaky_re_lu (LeakyReLU)	(None, 128, 32, 64)
batch_normalization (BatchNorma	(None, 128, 32, 64)
conv2d_2 (Conv2D)	(None, 64, 16, 128)
leaky_re_lu_1 (LeakyReLU)	(None, 64, 16, 128)
batch_normalization_1 (BatchNor	(None, 64, 16, 128)

השכבה התחתונה ביותר היא 'צוואר הבקבוק' מתווכת בין שכבת ההתכווצות לשכבת ההרחבה. גודל שכבה זו הוא $8 \times 2 \times 512$.

זהו שלב האיחוד של שני הערוצים ה- **audio branch** וה- **visual branch**. בה נשרשר את המוצא של שכבה אחת לפני האחרונה של רשת ה- **res-net**, שזו בעצם השכבה בה יש פיצ'רים של האודיו (לפני השייך שלהם לליבלים) עם הפיצ'רים של המידע הויזואלי (לאחר שעברו שכפול ו'**reshape**' בכדי להתאים למימד של האודיו). נזכיר שהתאמנו את גודלה ל- $8 \times 2 \times 784$. דבר זה נעשה על מנת לבצע ניתוח **audio-visual** משותף.

החלק של ה- '**Expansion**' מורכב מכמה בלוקי הרחבה, ובאופן סימטרי לחלק של '**Contraction**', לאחר כל בלוק המימד עולה וכמות הפיצ'רים קטנה פי 2. דבר זה נעשה על ידי '**Up - Convolution**' על הוקטור המחובר של האודיו והויזואל כדי לייצר ספקטרוגרמה של המסיכה M . בנוסף, בחלק זה בכל פעם משרשרים (**Concatenate**) לוקטור הקלט את הוקטור משכבת הכיווץ המתאימה, פעולה זו תבטיח כי מידע שנזרק ב-**encoder** ואולי הוא חשוב יהיה זמין ל-**decoder**. מספר בלוקי ההרחבה זהה למספר בלוקי ההתכווצות. חלק זה, ה- **decoder** בנוי כך:

$\rightarrow \text{Conv2D} \rightarrow \text{Relu} \rightarrow \text{Upsample} \rightarrow \text{Conv2D} \rightarrow \text{Relu} \rightarrow \text{BatchNorm} \rightarrow \text{Concatenate}$
 $\text{BatchNorm} \rightarrow \text{Concatenate} \dots$

סדרה זו של קונבולוציות **up-convolution** ממפה את ספקטרוגרמת הקלט המונוראלית למסכה מרוכבת ה"מקודדת" את השמע הבינוראלי החזוי. מסכה זו היא מה שהרשת חזתה.

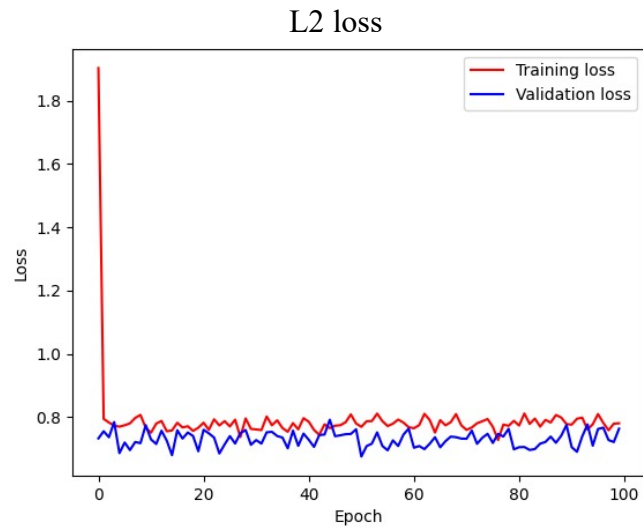
לבסוף, הספקטרוגרמה של אות הדיפרנציאלי מתקבלת על ידי הכפלה מרוכבת של ספקטרוגרמת הקלט עם המסכה המרוכבת שחזינו כלומר: $\tilde{X}^D = \mathbf{M}\mathbf{X}^M$.

לאחר מכן, באמצעות **ISTFT** נקבל את אות הפרש החזוי $\tilde{x}^D(t)$ שממנו נוכל לחלץ את שני הערוצים $\tilde{x}^L(t)$ ו- $\tilde{x}^R(t)$ - המשוחזרים ע"י:

$$\tilde{x}^L(t) = \frac{x^L(t) + \tilde{x}^D(t)}{2}, \quad \tilde{x}^R(t) = \frac{x^M(t) - \tilde{x}^D(t)}{2}$$

הloss מחושב על X^D – הספקטוגרמה של האות הדיפרנציאלי החזוי.
הרשת MONO2BINAURAL מאומנת באמצעות L2 loss כדי למזער את המרחק בין
הספקטרוגרמה המורכבת של ה "ground-truth" לזו החזויה.

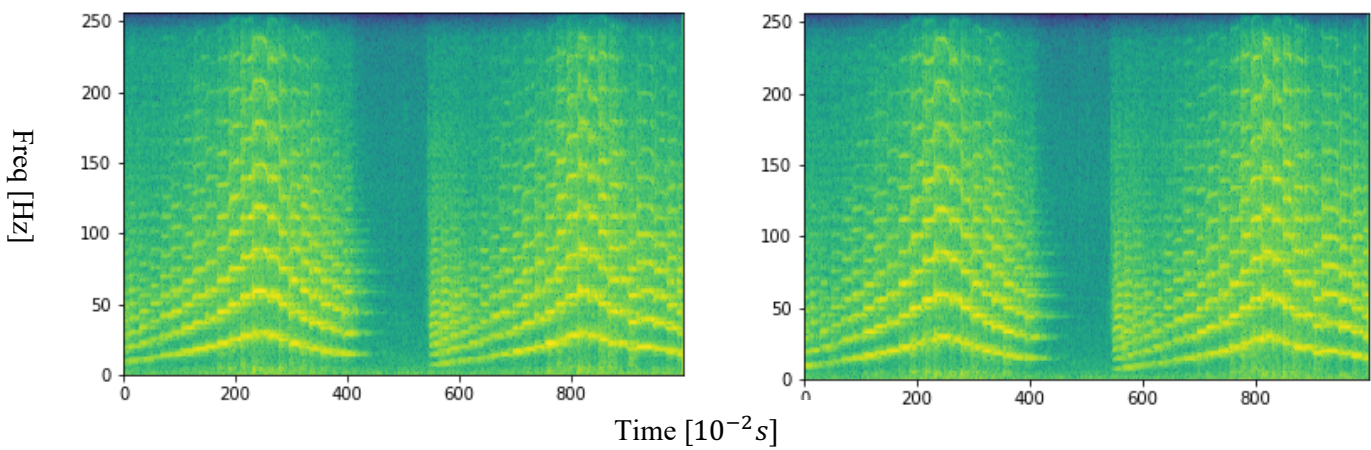
הרשת אומנה למשך כ-100 איפוקים וניתן לראות כי התבצע תהליך של למידה.



הloss הסופי שאליו הגענו במהלך האימון הוא 0.68

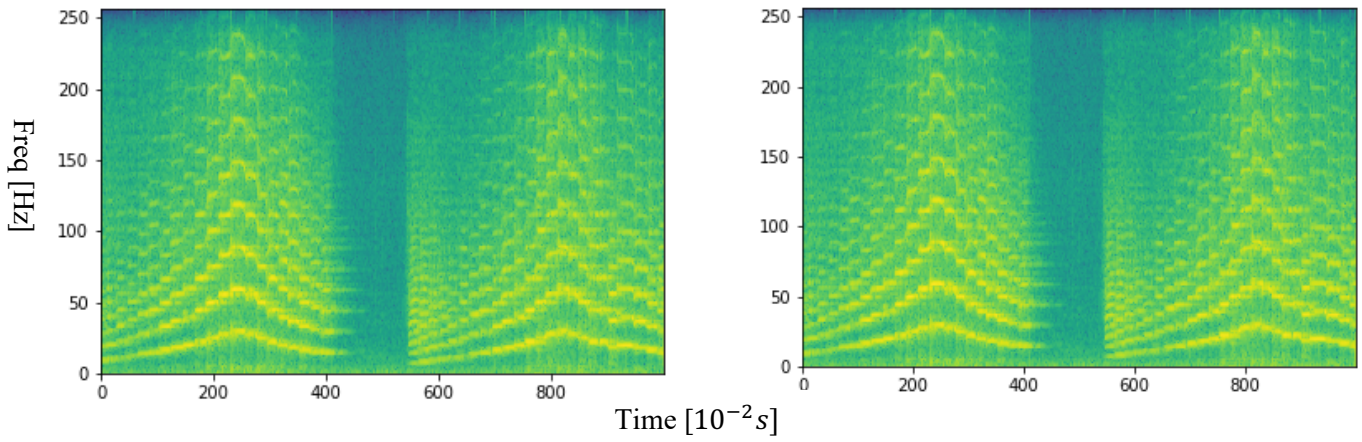
כעת נציג את תוצאות באופן ויזואלי שנלקחו על דוגמא ספציפית (רנדומלית) מסט הבדיקה.

האות ההתחלתי, אות בינאורלי לו שני ערוצים.



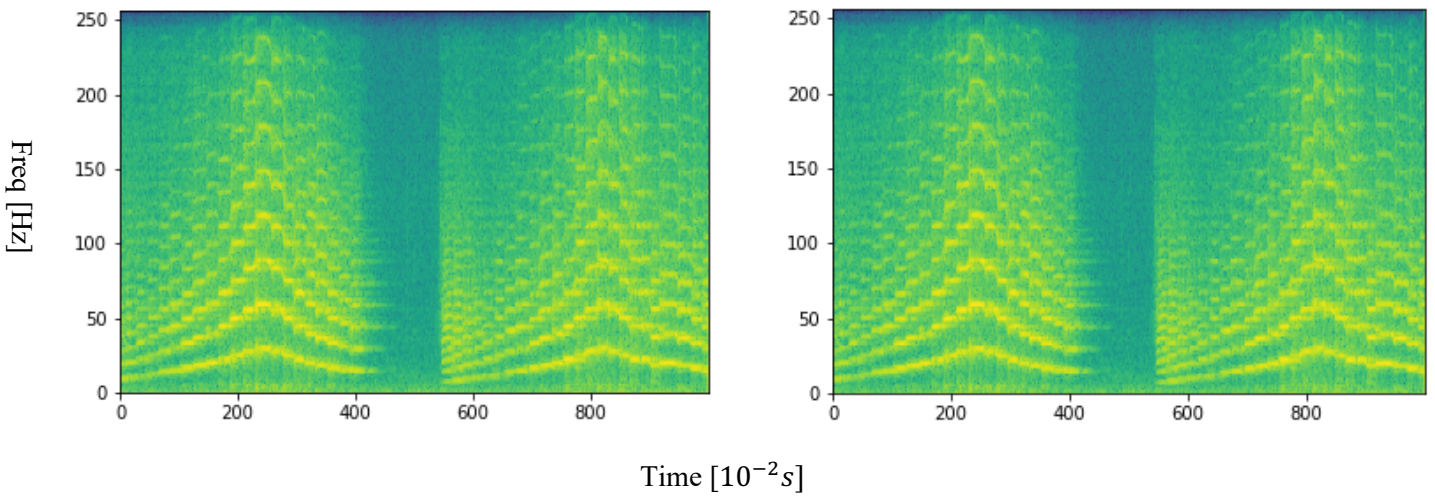
ערוץ שמאל וערוץ ימין של אות הבניסה

לאחר הפעלת הרשת מתקבלים שני ערוצים (שמאל וימין בהתאמה):



ערוץ שמאל וערוץ ימין של אות המוצא

נראה גם את אות הmono של סיגנל ה mixed ($x^M(t) = x^L(t) + x^R(t)$)



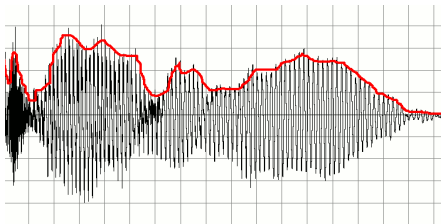
סיגנל ה mixed של אות הכניסה (מימין) והאות החזוי (משמאל)

כעת, נעריך את דיוק הרשת באמצעות 2 מטריקות:

1. STFT distance:

המרחק האוקלידי בין הספקטרוגרמה של שני ערוצי הסיגנל המקורי לבין הספקטרוגרמה של הסיגנל הבינוראלי שהמודל חזה.

$$D \{STFT\} = \|X^L - X^L\|^2 + \|X^R - X^R\|^2$$



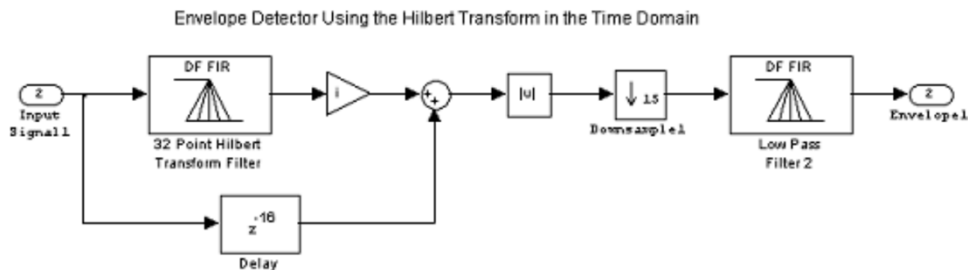
2. Envelope (ENV) Distance:

ראשית נסביר מס' מושגים:

מעטפת אות¹³ שקולה לקווי המתאר שלו¹⁴.

איור 5: מעטפת אות (באדום)

גלאי מעטפת בעצם מחבר את כל הפייקים של הסיגנל ליצירת מעטפת האות. ליהוי מעטפות קיימים יישומים רבים בתחומי עיבוד אותות ותקשורת, אחד מהם הוא זיהוי משרעת



(AM). דיאגרמת הבלוקים הבאה מציגה את יישום זיהוי המעטפות בשיטת הילברט:

שיטת זיהוי מעטפות זו כוללת יצירת אות אנליטי של הקלט באמצעות טרנספורמצית הילברט. אות אנליטי זה הוא מורכב, כאשר החלק האמיתי הוא האות המקורי והחלק המדומה הוא טרנספורמצית הילברט של האות המקורי.

¹³ <https://la.mathworks.com/help/dsp/ug/envelope-detection.html>

¹⁴

https://he.wikipedia.org/wiki/%D7%92%D7%9C%D7%90%D7%99_%D7%9E%D7%A2%D7%98%D7%A4%D7%AA

המעטפת $e(t)$ של האות $x(t)$ מוגדרת כגודל האות האנליטי כפי שמוצג במשוואה הבאה:

$$e(t) = \sqrt{x(t)^2 + \hat{x}(t)^2}$$

כאשר: $\hat{x}(t)$ הוא טרנספורמציה הילברט של $x(t)$

טרנספורמציה הילברט של האות מתבצעת באמצעות פילטר **FIR** עם 32 נקודות. כדי ליצור את האות האנליטי, מכפילים את טרנספורמציה הילברט של האות ב- i ומוסיפים אותו לאות המקורי המעוכב בזמן. יש צורך להשוות את אות הקלט מכיוון שהטרנספורמציה של הילברט, המיושמת על ידי פילטר **FIR** תציג עיכוב של חצי מאורך המסנן.

המעטפת היא אות בתדר נמוך בהשוואה לאות המקורי. כדי להפחית את תדירות הדגימה שלה ולהחליק את המעטפה, דוגמים את האות בקצב נמוך (**downsampling**) ומעבירים את התוצאה דרך מסנן מעביר נמוכים (**LPF**). לבסוף המטריקה שלנו היא המרחק האוקלידי בין מעטפת הערוצים של הסיגנל המקורי והסיגנל הבינוראלי החזוי:

נסמן את מעטפת האות $E[x(t)]$.

$$D \{Env\} = ||E[x^L(t)] - E[\hat{x}^L(t)]||^2 + ||x^R(t) - \hat{x}^R(t)||^2$$

בחרנו מספר וידויים רנדומליים וחשבנו עבורם את המרחקים ב-3 מקרים:

1. **Audio only**

כדי לקבוע אם מידע חזותי חיוני לביצוע ההמרה מ'**mono**' ל'**Binaural**' נתקנו את כניסת ה'**resnet**' כך שהאודיו שימש כקלט יחיד ורק מאפייני השמע הועברו לשכבות ה'**upconvolution**' לצורך החיזוי. שאר ההגדרות זהות.

2. **Flipped Visual**

במהלך ה 'test' הפכנו את הפריימים הנלווים לאודיו כדי לבצע חיזוי באמצעות מידע ויזואלי שגוי.

3. Mono2binaural :

הרשת שיצרנו כמפורט לעיל.

התוצאות שקבלנו:

	Stft l2 distance	Envelope distance
Audio Only	1.080998	0.156836
Flipped Visual	1.08097142	0.15683906
Mono2binaural	1.02449186	0.14868609

ניתן לראות כי המרחק הקטן ביותר התקבל עבור הרשת שיצרנו.

כמו כן, מצורפות (נספח מס' 2) שתי דוגמאות להפרדה בין הערוצים המבוצעת ע"י הרשת. דוגמא ראשונה (וידיאו 1644) מציג את כניסה בינאורלי הנלקח מסט testn שהרשת הורצה על כל אורכו (במקטעים קצרים של 0.63s עם חפיפה של 0.5s שאוחדו לאחר מכן חזרה ל 10s) כך שניתן לשמוע

א. את המקור- אודיו בינאורלי, שהוא בעצם התוצאה הרצויה לנו

ב. את אודיו החזוי שאיחדנו חזרה עם הוידיאו לחוויה מיטבית.

ה mse בין האות החזוי למקורי, על ערוץ ימין וערוץ שמאל, בתחום הזמן בדוגמא זו הוא: 0.00625 (לצורך השוואה: כאשר השתמשנו בתמונות הפוכות (Flipped Visual) ה mse (בתחום הזמן) היה 0.01). (ככל שיותר קרוב ל-0 סימן שחזינו מדויק יותר)

בדוגמא מס' 2 (מתוך- הדוב בבית הכחול) הרשת הופעלה על סרטון הנלקח מ YOUTUBE כך שהאודיו המקורי הוא בעל ערוץ אחד ללא רמזים מרחבים, שוב נעשה תהליך פרדיקציה כך שלבסוף יצרנו קטע וידיאו שמורכב והפריימים ומשני הערוצים שנחזו.

ממליצות להאזין 😊

שיפורים ושינויים:

במאמר נעשה אימון גם על רשת ה-RES-NET. אנו ניסינו להריץ עם אימון על רשת הראז נט וקיבלנו תוצאות טובות יותר כאשר ה fine-tuning שלה היה מושתק. כמו כן, בניסיון לשפר את המודל ניסינו להגביל את המשקולות, אך דבר זה לא תרם באופן מובהק לשיפור ה loss. בדאטה ג'נריטור השתמשנו ב up convoloation ב upsampling ולאחר מכן ב conv2d שציפינו שיתן תוצאות טובות יותר (מאשר deconv2d) מכיוון שלא יוצר ארטיפקטים (מקור)¹⁵

¹⁵ <https://distill.pub/2016/deconv-checkerboard/>

סיכום

בפרויקט זה תיארונו מערכת הממירה שמע בעל ערוץ יחיד לאודיו בינוראלי בעל שני ערוצים באמצעות חילוץ מידע חזותי מהפריימים הויזואלים .

כמפורט לעיל הרשת בנויה בארכיטקטורת מקודד-מפענח ומנסה לחזות אודיו דו - ערוצי בינארלי המסכים לקומפוזיציה המרחבית בוידאו.

תוצאות הפרויקט בהחלט ניתן לראות כי בהסאונד החזוי יש שיפור בתחושה המרחבית.

כאשר מאזינים לאודיו הדו-אוריאלי החזוי - הצליל הוויזואלי 2.5D -המאזינים יכולים להרגיש את מיקומי מקורות הקול כפי שהם מוצגים בסרטון.

נספחים

1. קבצי קוד :

- Main.py
- Unet.py
- Resnet.py
- Data_generator.py
- Two_channel_output.py
- Mono_two_channel_output.py
- Evaluate.py

2. תקייה לתוצאות audio :

- 001644.mp4 אודיו המקור- בינאורלי
- predicted_video_1644.mp4 האודיו החזוי
- cut_10s_test_bear.mp4 אודיו מונו
- predicted_bear.mp4 האודיו החזוי

3. תקיית המאמר:

- 2.5D Visual Sound
- 2.5D Visual Sound (Supplementary Materials)

4. המשקלים