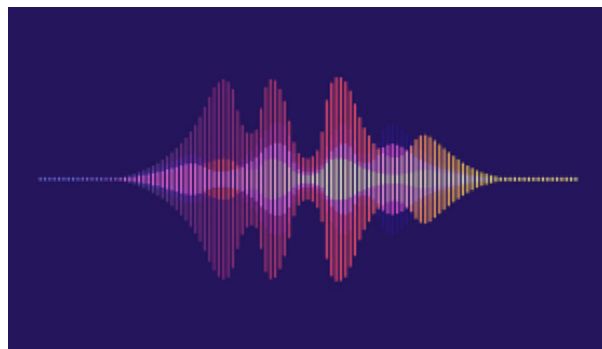


המכללה האקדמית תל חי

הפקולטה למדעי המחשב



דו"ח סיום פרויקט



דוגמאות אדוורסריות על מודל של הפרדת ערוצים: יצירה והגנה

מבצעים: נגה ענבי

איגור חמליק

מנחה: אורי בריט

תוכן העניינים

2	תוכן העניינים
3	רשימת איורים
4	תקציר
4	Abstract
5	1. מבוא
5	1.1 מוטיבציה
5	1.2 סיכום התרומה העיקרית
6	1.3 עבודות קודמות בנושא
6	1.4 מבנה העבודה
7	2. סקר ספרות
7	2.1 SPLEETER
9	2.2 DEMUCS
9	2.2.1 DEMUCS V2
11	2.2.2 DEMUCS V4
13	2.3 בחירת Hybrid Transformer Demucs (ht demucs)
13	3. מטריקות הערכה
14	3.1 SDR
15	3.2 SIR
15	3.3 SAR
17	4. סוגי התקפות
17	4.1 התקפות הרעלה
18	4.2 התקפות פרטיות
18	4.3 התקפות ניצול לרעה
18	4.4 התקפות התחמקות - Evasion Attacks
19	4.5 התקפה קופסה לבנה
19	4.5.1 התקפה מכוונת
20	4.5.2 התקפה לא מכוונת
20	4.6 התקפה קופסה שחורה
20	4.6.1 Surrogate Model - תחליפי באמצעות חיזוי תחליפי
21	4.6.2 Query-based Optimization - אופטימיזציה מבוססת חיפוש
21	1. Hill Climbing - מטפס הרים
21	2. שיטת מונטה קרלו - Monte Carlo Sampling
22	3. אלגוריתמים אבולוציוניים - Evolutionary Algorithms
22	4.6.3 שיטות ללא תוויות - Score-based or Decision-based
23	5. מימוש התקפת קופסה לבנה בפרוייקט
25	6. אלגוריתם ההגנה
26	7. מסד הנתונים MUSDB18
27	8. רשימת מקורות
27	8.1 מאמרים
28	8.2 מקורות נוספים

28.....	9. תוצאות.....
28.....	9.1 הסבר על הגרפים.....
29.....	9.2 תוצאות.....
36.....	10 פרק סיכום ומסקנות.....
36.....	10.1 ניתוח תוצאות.....
37.....	10.2 מסקנות.....
37.....	תודות.....

רשימת איורים

[איור 1 - טבלה של ביצועי ספליטיר](#)

[איור 2 - דוגמה לספקטרוגרמה](#)

[איור 3 - תרשים ארכיטקטורת דימוקס גרסה 2](#)

[איור 4 - טבלת השוואת ביצועי SDR של מודלים להפרדת ערוצים](#)

[איור 5 - תרשים ארכיטקטורת דימוקס גרסה 4](#)

[איור 6 - טבלת השוואת ביצועי SDR של מודלים להפרדת ערוצים מול הגרסה ההיברידית החדשה של דימוקס](#)

[איור 7 - תרשים של דרך הפעלת הדוגמה האדוורסרילית על המודל](#)

[איור 8 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 1](#)

[איור 9 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 2](#)

[איור 10 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 3](#)

[איור 11 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 4](#)

[איור 12 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 5](#)

[איור 13 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 6](#)

[איור 14 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 7](#)

תקציר

דוגמאות אדוורסריות הן סוג של התקפות זדוניות על מודלים של למידה עמוקה. התקפות אלו מבוצעות על-ידי שינוי מינימלי של הקלט – שינוי שאינו מורגש על-ידי בני אדם – אך גורם למודל להפיק פלט שגוי או שונה לחלוטין מהפלט הרצוי. תוקף יכול לנצל מנגנון זה כדי לגרום למערכת לבצע פעולות לא נכונות או לקבל החלטות שגויות.

במסגרת פרויקט זה, נבחן האם ניתן לבצע התקפת התחמקות (Evasion Attack) על מודל חדיש מסוג Hybrid Transformer Demucs (גרסה 4), אשר נועד להפרדת ערוצים מוזיקליים (כגון שירה, תופים, באס ולייווי) מתוך קובץ שמע מעורב.

התקפה זו בוצעה בסגנון של קופסה לבנה, כלומר, הייתה גישה מלאה לארכיטקטורת המודל, למשקליו ולקודו.

בנוסף, פיתחנו מנגנון הגנה אפקטיבי המוסיף רעש רנדומלי בלתי נשמע לקלט המותקף.

ניסויים על מסד הנתונים MUSDB18 הראו כי מנגנון ההגנה מצליח לנטרל את ההשפעה של ההתקפה ולשפר את איכות ההפרדה של המודל לאחר ההתקפה.

הערכת איכות התוצאות נעשתה באמצעות שלוש מטריקות מקובלות בתחום: SDR, SIR, ו-SAR, אשר אפשרו לבחון את מידת השיבוש שנגרמה למודל ואת הצלחת ההגנה בהתמודדות עמה.

Abstract

Adversarial examples are a type of malicious attack targeting machine learning models, particularly deep learning systems. These attacks involve applying small, imperceptible perturbations to the input data, which can cause the model to produce incorrect or unintended outputs. An attacker can exploit this vulnerability to manipulate the system's behavior in subtle yet potentially harmful ways.

In this project, we investigate whether it is possible to carry out an evasion attack on a state-of-the-art music source separation model, specifically Hybrid Transformer Demucs (version 4). This model is designed to isolate individual musical components (e.g., vocals, drums, bass, and accompaniment) from a mixed audio track. The attack was performed in a white-box setting, meaning the attacker had full access to the model's architecture and weights, and used gradient-based optimization to craft adversarial perturbations that degrade the model's separation quality.

To counter the attack, we developed an effective defense mechanism that adds random, imperceptible noise to the adversarial input. This method successfully disrupts the carefully crafted perturbations, mitigating the impact of the attack and restoring the model's separation performance.

We evaluated the effectiveness of both the attack and the defense using three standard metrics for source separation: SDR (Source-to-Distortion Ratio), SIR (Signal-to-Interference Ratio), and SAR (Signal-to-Artifacts Ratio). Experiments conducted on the MUSDB18 dataset demonstrate

the vulnerability of the model to adversarial perturbations, as well as the potential of lightweight defenses to significantly reduce their effect.

1. מבוא

1.1 מוטיבציה

בשנים האחרונות חלה התפתחות מואצת בתחום הבינה המלאכותית, במיוחד במשימות כמו זיהוי תמונה, זיהוי קול, והבנה של שפה טבעית. לדוגמה, מודלים מתקדמים כמו ChatGPT הפכו לחלק בלתי נפרד מחיינו, ונעשה בהם שימוש נרחב בתעשיות מגוונות, כולל תחומים קריטיים כגון ביטחון, רפואה ותחבורה.

למרות הרושם שמותירים המודלים הללו ולעיתים אף יכולתם לעלות על ביצועים אנושיים הם עדיין סובלים מפגיעות חמורה לתקיפות אדוורסריות. תקיפות אלו הולכות ונעשות מתוחכמות יותר, ומהוות איום ממשי על אמינות ובטיחות מערכות הבינה המלאכותית [1].

השלכות של תקיפות כאלו עלולות להיות חמורות ואף קטלניות, במיוחד כאשר מדובר במערכות הפועלות בסביבות רגישות כמו כלי רכב אוטונומיים, מערכות רפואיות מאובחנות, או מערכות הגנה ובקרה צבאיות. מסיבה זו, חשוב לחקור ולהבין את אופי התקיפות הללו, לפתח דרכים לזיהוי והתגוננות ולפתח התקפות חדשות לטובת הגנה בלבד.

מחקרים נעשו בתחומי התקפות אדוורסריות על מודלים של ראייה ממוחשבת אך מעט יחסית נעשו על מודלים של עיבוד אותות. לכן, יש עניין גדול במחקר מעמיק בנושא. יש לבדוק האם אותם אלגוריתמים לפתרונות הגנה נגד מתקפות למודלים לסיווג תמונה, יעילים גם למודלים להפרדת ערוצים של ערוצי שמע.

1.2 סיכום התרומה העיקרית

בפרויקט זה, בוצעה הדגמה של תקיפה אדוורסרית מסוג התחמקות (Evasion Attack) על מודל מתקדם להפרדת מקורות מוזיקליים – Hybrid Transformer Demucs v4. התקיפה יושמה במסגרת תרחיש קופסה לבנה, תוך שימוש בגרדיאנט של פונקציית ההפסד ביחס לקלט לצורך יצירת שיבוש אודיו מזערי אך אפקטיבי, שלא ניתן לשמוע על ידי אוזן אנושית, אשר גרם לירידה משמעותית באיכות ההפרדה של המודל.

בנוסף, הוצג מנגנון הגנה המבוסס על הוספת רעש אקראי מוגבל, אשר הוערך ונמצא שמשפר משמעותית את ביצועי המודל לאחר ההתקפה.

העבודה מהווה תרומה מחקרית חדשנית, שכן מרבית המחקרים בתחום התקיפות האדוורסריות התמקדו עד כה במודלים לסיווג תמונה, בעוד שבפרויקט זה הודגמה תקיפה מסוג זה על מודל הפועל בתחום האודיו ובמשימת

רגרסיה רבת-ערוצים. בכך מתחפתח כיוון חדש לחקר חולשות של מודלים לעיבוד אותות שמע, כמו גם לגיבוש פתרונות מיגון מתאימים.

1.3 עבודות קודמות בנושא

1. Carlini, N., & Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*.

מאמר זה היווה את הבסיס לעבודה שלנו, והיה בין המאמרים הראשונים שעבדו על התקפות אדוורסריות של מערכות שמע.

2. Qin, Y., Carlini, N., Goodfellow, I., Cottrell, G., & Raffel, C. (2019). Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In International Conference on Machine Learning (pp. 5231-5240). PMLR.

מחקר המשך שמשפר את התוצאות של המחקר הקודם ומשלב טכניקות התקפה מורכבות יותר.

1.4 מבנה העבודה

מהלך העבודה חולק למספר שלבים עוקבים. בשלב הראשון, אשר נמשך במהלך החודשים הראשונים של הפרויקט, בוצע תהליך לימוד מעמיק של נושאים תיאורטיים הנוגעים ללמידה עמוקה, רשתות טרנספורמר, עיבוד אותות דיגיטלי (בהתבסס על חומרי הקורס שנלמדו בעבר), וכן סוגים שונים של תקיפות אדוורסריות.

בהמשך, נלמדו לעומק שני מודלים עיקריים בתחום הפרדת מקורות מוזיקליים – Demucs ו-Spleeter על גרסאותיו השונות, תוך בחינה השוואתית של הארכיטקטורות, אופן הפעולה והביצועים בפועל.

בשלב הבא, הוכן והוטמע מאגר הנתונים MUSDB18, אשר שימש כבסיס לכל הניסויים שבוצעו בהמשך. המודלים הותקנו והורצו באופן מקומי, ונערכה סדרת ניסויים שכללה שימוש בדוגמאות מוזיקליות מגוונות במטרה לאתר מגבלות בביצועי הפרדה, זיהוי תרחישים בעייתיים, וחקר חוזקות וחולשות של כל מודל.

לאחר שלב ההיכרות וההכנה, פותח אלגוריתם תקיפה אדוורסרית מסוג קופסה לבנה, ואחריו יושמה גם אסטרטגיית הגנה ייעודית.

בשלב הסופי של העבודה, בוצעה הערכה מקיפה של תוצאות הניסויים באמצעות מדדים כמותיים מקובלים (SDR, SIR, SAR), תוך השוואה שיטתית בין מצבים של לפני ואחרי התקיפה וההגנה.

2. סקר ספרות

במסגרת חקר הספרות נחקר על מודלים שונים של פירוק ערוצים ועל דרך פעולתם. מפורט בהמשך על כל מודל שנחקר ועל המודל הנבחר.

2.1 - SPLEETER

Source code: <https://github.com/deezer/spleeter>

Paper: <https://archives.ismir.net/ismir2019/latebreaking/000036.pdf>

The paper is from 2019

How to run: `spleeter separate -o output/audio_output -p spleeter:4stems "song_path"`

Spleeter הוא מודל ל-Music Source Separation ב-Deezer פיתחו אותו כדי לאפשר יישומים שונים: קרייקי, רמיקסים, ניתוח מוזיקלי אוטומטי, ועוד.

הרעיון המרכזי של המודל:

במקום לעבוד ישירות על האודיו בזמן, המודל עובד על ייצוג של האודיו במישור התדרים: STFT (Short-Time Fourier Transform).

כלומר, הקלט למודל הוא ספקטרוגרמה.

מבנה המודל:

- רשת U-Net שפועלת על המישור הספקטרי.
- הקלט: ספקטרוגרמה מורכבת (מודולו ופאזה מופרדים - המודל מתמקד במודולו).
- הרשת מנסה ללמוד עבור כל מרכיב במיקס (שירה, תופים וכו') את ה-"mask" המתאים - כלומר, אילו תדרים שייכים לאיזה מקור.
- הפלט של הרשת הוא Masks שאותן מכפילים חזרה בספקטרוגרמת המקור כדי לקבל את הספקטרוגרמות של כל אחד מהמקורות.

ביצועים

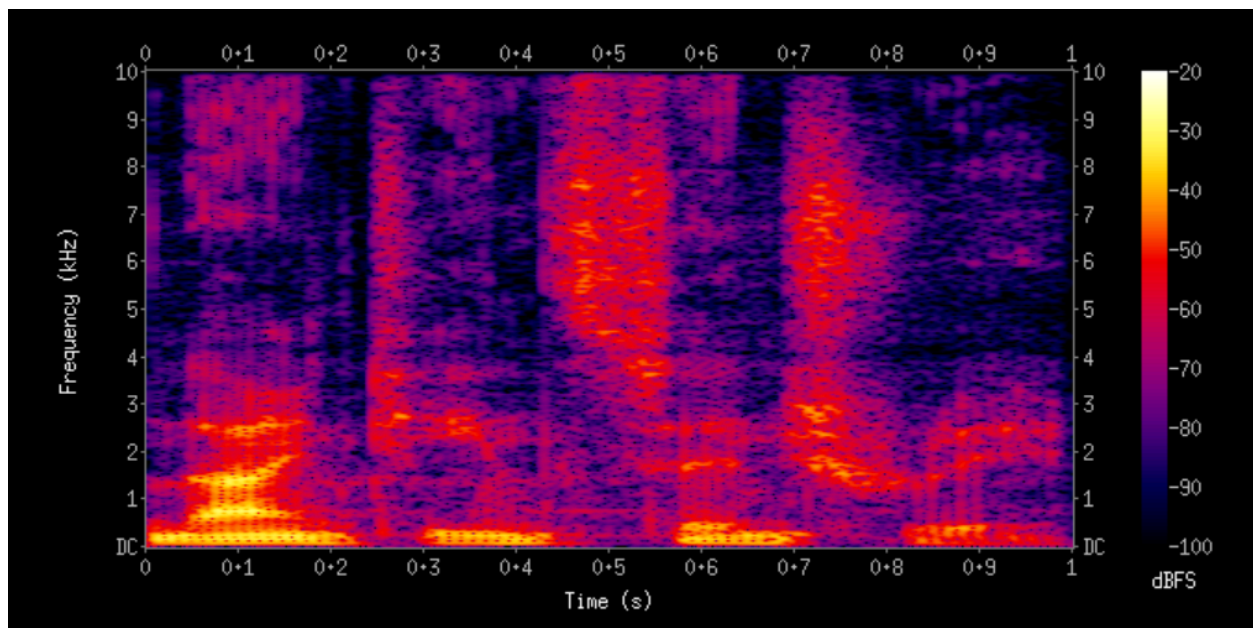
	vocals				bass				drums				other			
	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR
Spleeter Mask	6.55	15.19	6.44	12.01	5.10	10.01	5.15	9.18	5.93	12.24	5.78	10.50	4.24	7.86	4.63	9.83
Spleeter MWF	6.86	15.86	6.99	11.95	5.51	10.30	5.96	9.61	6.71	13.67	6.54	10.69	4.55	8.16	4.88	9.87
Open-Unmix	6.32	13.33	6.52	11.93	5.23	10.93	6.34	9.23	5.73	11.12	6.02	10.51	4.02	6.59	4.74	9.31

איור 1 - טבלה של ביצועי ספלייטר

מהי ספקטרוגרמה?

ספקטרוגרמה (Spectrogram) היא ייצוג חזותי של צליל בשלושה ממדים:

1. ציר X – זמן (מתי קורה הצליל)
2. ציר Y – תדר (באיזו גובה צליל)
3. צבע/עוצמה – עוצמת הצליל בכל תדר ובכל רגע בזמן



איור 2 - דוגמה לספקטרוגרמה

איך יוצרים ספקטרוגרמה?

כדי להפוך אות שמע (כמו קובץ WAV) לספקטרוגרמה, משתמשים ב-Short-Time Fourier Transform (STFT) — כלומר, מחשבים טרנספורם פורייה על מקטעים קצרים מהשמע (במקום על כולו בבת אחת). כך אפשר לראות איך התדרים משתנים לאורך זמן.

2.2 - DEMUCS

Demucs (קיצור של Deep Extractor for Music Sources) היא סדרת מודלים להפרדת מקורות מתוך שירים (Source Separation), בעיקר של הקלטה ל-4 ערוצים: Vocals, Drums, Bass, Other.

בעוד ש-Spleeter עובד על ספקטרוגרמות, Demucs פועל ישירות על האמפליטודה (waveform)

2.2.1 - DEMUCS V2

Code: <https://github.com/facebookresearch/demucs/tree/v2>

Paper: <https://hal.science/hal-02379796/document>

How to run: `demucs "song_path"`

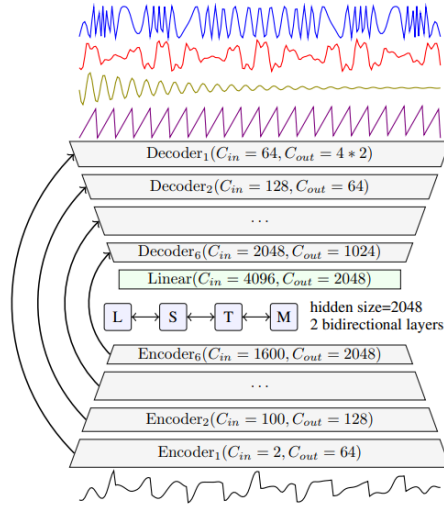
Demucs הוא מודל מבוסס encoder-decoder שמקבל כקלט מיקס סטריאו (כלומר שני ערוצים – ימין ושמאל), ומחזיר עבור כל מקור (כמו תופים, באס, שירה) הערכת סטריאו נפרדת.

הארכיטקטורה של DEMUCS V2:

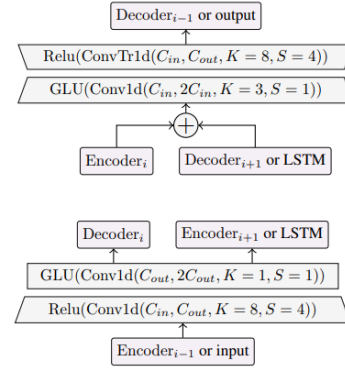
הארכיטקטורה כוללת שלושה שלבים עיקריים:

1. **Convolutional Encoder** – ממיר את גל הקול לייצוג מופשט בעומק גבוה.
2. **Bidirectional LSTM** – עיבוד של הרצף לשני הכיוונים (קדימה ואחורה בזמן), כדי להבין קשרים ארוכי טווח.
3. **Convolutional Decoder** – מפענח בחזרה את ה-waveform עבור כל מקור.

בין ה-Encoder ל-Decoder יש skip connections כמו ב-U-Net, שמאפשרים לשמר מידע מפורט משכבות מוקדמות ולחבר אותו חזרה בשכבות המאוחרות.



(a) Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections.



(b) Detailed view of the layers Decoder_i on the top and Encoder_i on the bottom. Arrows represent connections to other parts of the model. For convolutions, C_{in} (resp C_{out}) is the number of input channels (resp output), K the kernel size and S the stride.

Figure 2: Demucs complete architecture on the left, with detailed representation of the encoder and decoder layers on the right.

איור 3 - תרשים ארכיטקטורת דימוקס גרסא 2

ביצועים:

Architecture	Wav?	Extra?	Test SDR in dB				
			All	Drums	Bass	Other	Vocals
IRM oracle	✗	N/A	8.22	8.45	7.12	7.85	9.43
Wave-U-Net	✓	✗	3.23	4.22	3.21	2.25	3.25
Open-Unmix	✗	✗	5.33	5.73	5.23	4.02	6.32
Meta-Tasnet	✓	✗	5.52	5.91	5.58	4.19	6.40
Conv-Tasnet [†]	✓	✗	5.73 ±.10	6.02 ±.08	6.20 ±.15	4.27 ±.03	6.43 ±.16
DPRNN	✓	✗	5.82	6.15	5.88	4.32	6.92
D3Net	✗	✗	6.01	7.01	5.25	4.53	7.24
Demucs [†]	✓	✗	6.28 ±.03	6.86 ±.05	7.01 ±.19	4.42 ±.06	6.84 ±.10
Spleeter	✗	~ 25k*	5.91	6.71	5.51	4.55	6.86
TasNet	✓	~ 2.5k	6.01	7.01	5.25	4.53	7.24
MMDenseLSTM	✗	804	6.04	6.81	5.40	4.80	7.16
Conv-Tasnet ^{††}	✓	150	6.32 ±.04	7.11 ±.13	7.00 ±.05	4.44 ±.03	6.74 ±.06
D3Net	✗	1.5k	6.68	7.36	6.20	5.37	7.80
Demucs [†]	✓	150	6.79 ±.02	7.58 ±.02	7.60 ±.13	4.69 ±.04	7.29 ±.06

2.2.2 - DEMUCS V4

Hybrid Transformer Demucs הוא שדרוג של המודל Hybrid Demucs המקורי, בשילוב עם Transformer בין-תחומי (Cross-domain Transformer).

Code: <https://github.com/adevossez/demucs>

Paper: <https://arxiv.org/pdf/2211.08553>

The paper is from 2022

How to run: `demucs -n htdemucs_ft "song_path"`

מבנה המודל

המבנה בנוי מ-2 מסלולים עיקריים:

1. מסלול הזמן (Waveform domain) – עם קונבולוציות זמניות.
2. מסלול הספקטרום (Spectrogram domain) – עם קונבולוציות בציר התדר.

לכל מסלול יש:

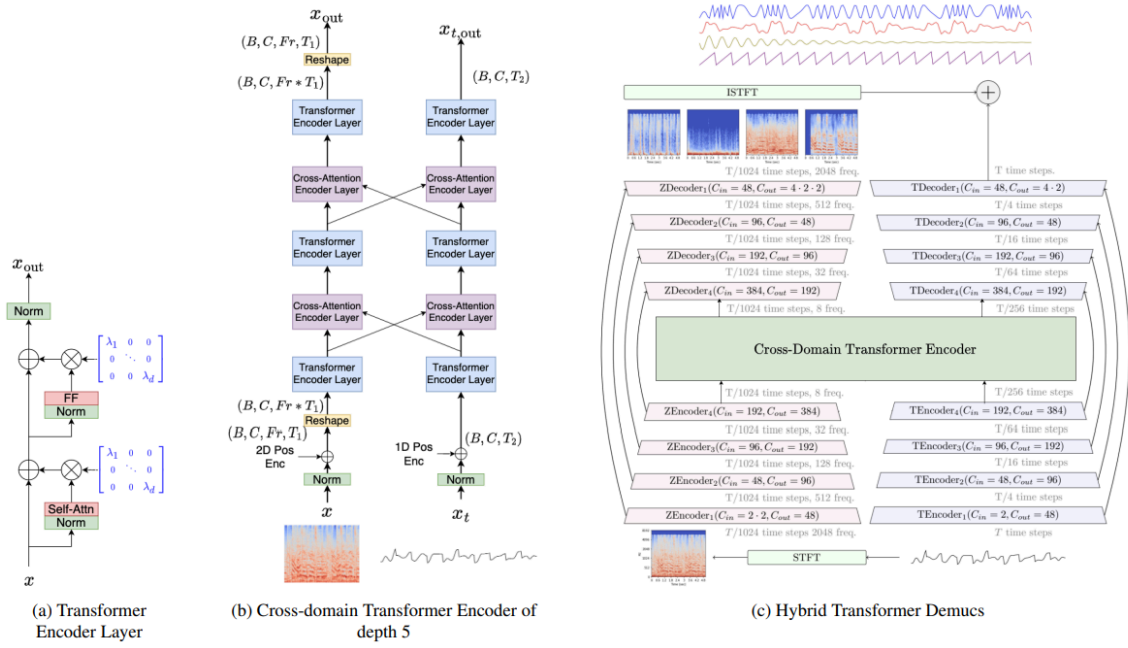
- 5 שכבות Encoder

- 5 שכבות Decoder

לאחר שכבת ה-Encoder החמישית בשני המסלולים – הייצוגים מיושרים ונחברים, ואז מוזרמים יחד ל-שכבה שישית משותפת.

חידוש המרכזי – Cross-domain Transformer:

ה-Transformer פועל גם על הספקטרום (2D) וגם על גל הזמן (1D) במקביל. כולל שילוב של Self-Attention בתוך כל תחום (spectrogram / waveform) ו-Cross-Attention בין התחומים – כלומר, תדרים וזמן "מדברים" זה עם זה.



איור 5 - תרשים ארכיטקטורת דימוקס גרסא 4

ביצועים

Architecture	Extra?	Test SDR in dB				
		All	Drums	Bass	Other	Vocals
IRM oracle	N/A	8.22	8.45	7.12	7.85	9.43
KUIELAB-MDX-Net [17]	✗	7.54	7.33	7.86	5.95	9.00
Hybrid Demucs [2]	✗	7.64	8.12	8.43	5.65	8.35
Band-Split RNN [14]	✗	8.24	9.01	7.22	6.70	10.01
HT Demucs	✗	7.52	7.94	8.48	5.72	7.93
Spleeter* [19]	25k	5.91	6.71	5.51	4.55	6.86
D3Net* [12]	1.5k	6.68	7.36	6.20	5.37	7.80
Demucs v2* [10]	150	6.79	7.58	7.60	4.69	7.29
Hybrid Demucs [2]	800	8.34	9.31	9.13	6.18	8.75
Band-Split RNN [14]	1750 [†]	8.97	10.15	8.16	7.08	10.47
HT Demucs	150	8.49	9.51	9.76	6.13	8.56
HT Demucs	800	8.80	10.05	9.78	6.42	8.93
HT Demucs (fine tuned)	800	9.00	10.08	10.39	6.32	9.20
Sparse HT Demucs (fine tuned)	800	9.20	10.83	10.47	6.41	9.37

2.3 בחירת Hybrid Transformer Demucs (ht demucs)

הוחלט לבחור במודל HT Demucs כבסיס למערכת ההגנה שלנו מפני מתקפות אדוורסריות, ממספר סיבות מרכזיות:

1. **נגישות ונוחות שימוש:**
המודל זמין כקוד פתוח, קל להרצה, וכולל אפשרות להורדה של גרסה מאומנת מראש. כך ניתן לבצע ניסויים ללא צורך באימון ממושך על מכונות יקרות.
2. **ביצועים גבוהים:**
HT Demucs השיג תוצאות מובילות במדדי הפרדת מקורות בהשוואה למודלים אחרים שנבדקו, לרבות על דאטהסטים סטנדרטיים כגון MUSDB.
3. **תיעוד ומחקר מקיף:**
המודל מלווה במאמרים עדכניים וברורים, המתארים את הארכיטקטורה והעקרונות שעומדים מאחוריה, מה שמקל על הבנה, התאמה ופיתוח נוסף לצרכים שלנו.

3. מטריקות הערכה

הסבר על מטריקות הערכה

על פי איזה קריטריונים ימדד מהו פירוק טוב?

1. שימור הערוץ שאותו רצינו למדר (כמה שומעים טוב את התופים לצורך העיניין)
2. מידור הערוצים האחרים (כמה לא שומעים את שאר הערוצים)
3. נוכחות רעשים מלאכותיים
4. איכות כוללת של הערוץ עצמו

3.1 SDR

Source-to-Distortion Ratio

מה זה SDR?

SDR measures the power ratio between the intended source and the distortion artifacts introduced during separation.

איך SDR עובד?

המדד משווה בין:

1. האות המקורי של מה שרצית להפריד (לדוגמה, רק השירה).
2. האות שהמודל שלך הפיק (השירה שהצלחת להוציא מהמיקס).

הוא בעצם מודד את היחס בין האנרגיה של האות הנקי (המטרה) לבין הטעות הכוללת (הרעש, ההפרעות והעיוותים שהמודל יצר).

SDR גבוה אומר שההפרדה טובה

SDR נמוך אומר שהתוצאה לא קרובה מאוד למקור

$$SDR = 10 * \log_{10}(E_s / E_d)$$

Where:

- E_s is the energy of the original or reference signal.
- E_d is the energy of the distortion or noise in the degraded signal.

איך להבין את המספר?

- SDR גבוה (לדוגמה, 15-20): המודל הצליח מאוד, ההפרדה מדויקת.
- SDR נמוך (לדוגמה, 5-10): יש הרבה הפרעות בתוצאה, ההפרדה לא טובה.

3.2 SIR

Signal-to-Interference Ratio

מודד את איכות הפירוק של הערוץ המופרד (למשל ערוץ של כלי נגינה מסוים) בהשוואה לרעש (interference) שמגיע ממקורות אחרים שנמצאים באותו הערוץ.

הוא מחושב על ידי השוואת הסיגנל הנקי (שאותו רוצים לשמר) לעומת הסיגנל שנמצא בשאר הערוצים (שהוא למעשה רעש או הפרעה). אם ה-SIR גבוה, המשמעות היא שהפירוק של הערוץ טוב ויש פחות רעש או הפרעה. אם ה-SIR נמוך, המשמעות היא שהפירוק פחות טוב ויש יותר רעש.

המטריקה נמדדת בדציבלים (dB), כאשר ערכים גבוהים יותר מצביעים על תוצאה טובה יותר של הפירוק.

$$\text{SIR} = 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}$$

- e_{interf} : component of the estimate that aligns with *other* true sources (interference).
- Zero interference $\Rightarrow \text{SIR} \rightarrow +\infty$.

Interpretation:

- **Higher SIR \Rightarrow better** separation (less bleed-through).
- SIR only looks at cross-talk, not at artifacts or noise.

הבדל בין SIR ל-SDR:

SDR מודד את איכות הסיגנל הסופי נגדי אם הגיטרה נשמעת מתכתית או לא אמיתית או נגיד יש מקטעים שהווליום לא אחיד לדוגמה
SIR בודק את ההפרדה מהכלים האחרים- כלומר אם שומעים קצת תופים בערך של הגיטרה אז ה-SIR היה נמוך

מה הטווחים של SIR ?

SIR גבוה = טוב מאוד (יותר מ-20 dB).
SIR בינוני = סביר (20-10 dB).
SIR נמוך = לא טוב (פחות מ-10 dB)

3.3 SAR

Signal-to-Artifacts Ratio

What it measures:

Amount of "unnatural" noise or processing artifacts introduced by the separation algorithm.

Definition:

$$\text{SAR} = 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{artif}}\|^2}$$

- e_{artif} : residual error after removing both the target component and the interference component from the estimate.

Interpretation:

- Higher SAR \Rightarrow fewer artifacts (cleaner processing).
- SAR focuses strictly on algorithmic distortions, ignoring leakage.

$$e_{\text{artif}} = \hat{s}_i - \left(\underbrace{\frac{\langle \hat{s}_i, s_i \rangle}{\langle s_i, s_i \rangle} s_i}_{s_{\text{target}}} + \underbrace{\sum_{j \neq i} \frac{\langle \hat{s}_i, s_j \rangle}{\langle s_j, s_j \rangle} s_j}_{e_{\text{interf}}} \right)$$

where:

- \hat{s}_i is the estimated i th source.
- s_i is the true i th source (the “target”).
- s_j for $j \neq i$ are the other true sources.
- $\langle x, y \rangle = \sum_n x[n] y[n]$ is the inner product.

So you subtract from your estimate both:

1. its projection onto the true target s_i , and
2. its projections onto all other sources s_j ,

and the remainder is the “artifact” component e_{artif} .

4. סוגי התקפות

ישנן כמה סוגי התקפות שניתן לבצע על מודלים. בפרק זה נסביר בקצרה על כמה מהם.

4.1 התקפות הרעלה

התקפות הרעלה מתבצעות בשלב האימון של המודל, כאשר התוקף מחדיר אליו דוגמאות קלט מזיקות או שגויות במטרה לשבש את תהליך הלמידה. מטרת התקפה זו היא לפגוע באמינות המודל, כך שבמהלך השימוש בו בעתיד הוא יניב תוצאות שגויות או לא צפויות.

התקפה מסוג זה עלולה להתבצע, למשל, על ידי שחקן עוין כמו חברה מתחרה, אך תיתכן גם באופן בלתי מכוון – כאשר מפתחי המודל משתמשים במאגר נתונים שכולל שגיאות, הטיות או דגימות פגומות, מבלי להיות מודעים לכך.

4.2 התקפות פרטיות

בדרך כלל התקפה זו נעשית על צ'אט בוטים ומטרתה להפיק מידע מהמודל אודות המודל עצמו או כל מידע רגיש כזה או אחר או לחלופין נקודות תורפה של המודל שהמודל עלול לגלות. לדוגמה, אם המודל אומן על מידע רגיש התוקף עלול לחשוף מידע זה.

4.3 התקפות ניצול לרעה

התקפה נוספת שנפוצה במודלים של שפה טבעית. במקרה זה, התוקף משנה או עורך מקור שנחשב כאמין על ידי המודל בזמן ההפעלה של המודל (למשל ויקיפדיה). התקפה זו יכולה לשמש למטרות פרופגנדה, ממשלות או ארגונים שונים.

4.4 התקפות התחמקות - Evasion Attacks

התקפות התחמקות הן סוג של תקיפות אדוורסריות המתרחשות בשלב ההפעלה של המודל (ולא בשלב האימון), ובהן התוקף משנה את הקלט המקורי באופן מזערי ומבוקר, כך שהשינוי אינו ניתן לזיהוי בעין או באוזן האנושית, אך מספיק כדי לגרום למודל להפיק פלט שגוי או שונה מהצפוי.

במקרים מסוימים, מטרת התוקף היא לגרום למודל להפיק פלט שגוי כללי, ובמקרים אחרים – להפיק פלט מסוים ומוגדר מראש, בהתאם לאינטרס של התוקף.

סוג זה של התקפה עשוי להוביל לתוצאות חמורות, במיוחד כאשר המערכת נמצאת בשימוש רגיש.

זהו סוג ההתקפה שיושם בפרויקט הנוכחי, כחלק מבדיקת עמידות של מודל להפרדת מקורות מוזיקליים מפני שינויים אדוורסריאליים זעירים בקלט.

דוגמאות למערכות רגישות שיכולות להיפגע מהתקפות אלו:

1. **מערכות זיהוי פקודות קוליות:**
במערכות המקבלות קלט קולי לצורך הפעלת פקודות (כגון עוזרות קוליות או מכשירים חכמים), תוקף יכול לשלב פקודה מוסווית בקובץ אודיו תמים – למשל שיר המתנגן ברקע – כך שהמערכת תזהה פקודה כלשהי ותבצע פעולה לא רצויה.
2. **רכב אוטונומי:**
רכב אוטונומי מסתמך על מערכת ראייה ממוחשבת לזיהוי תמרורים. תוקף עשוי לשנות תמרור (באופן שלא ניתן להבחין בו בעין אנושית) כך שהמערכת תזהה אותו כתמרור שונה לחלוטין – למשל, להפוך תמרור עצור לתמרור נסיעה חופשית.
3. **סימנים סמויים על הכביש:**
הוספה של דפוסים גיאומטריים או צבעים שאינם מורגשים לעין האנושית עשויה לשבש את מערכת הניווט של רכב אוטונומי ולגרום לו לסטות ממסלולו.
4. **מערכות סיווג תמונה:**
תוקף יכול לשנות קובץ תמונה בשינויים מינימליים כך שמערכת סיווג תזהה את התמונה כקטגוריה שגויה, מה שעלול לשבש תפקוד של מערכות אבטחה, ניטור רפואי, בקרת איכות תעשייתית או כל מערכת שמשתמשת במודל לסיווג תמונה.

בתוך ההתקפה הזו ישנן שתי תתי התקפות עיקריות:

4.5 התקפה קופסה לבנה

מתבצעת כאשר הקוד, הארכיטקטורה והמשקלים של המודל נתונים לתוקף במלואם.

התוקף עלול להשתמש בגזירה לאחור (backpropagation) בשביל לחשב את את נגזרת פונקציית ההפסד (הגרדיאנט) ביחס לקלט (בניגוד לחישוב נגזרת פונקציית ההפסד ביחס למשקלים בעת אימון המודל).

$$\nabla_x L(x,y)$$

כאשר x הוא הקלט, y הוא הפלט ו- L היא פונקציית ההפסד

בהתבסס על גרדיאנט זה, התוקף מבצע סדרת איטרציות, כאשר בכל איטרציה הוא משנה את הקלט בשינויים קטנים מאוד (שלא ניתנים לזיהוי בעין או באוזן אנושית), כך שהפלט של המודל יוסט באופן משמעותי מהתוצאה הצפויה.

באופן זה, ניתן לגרום למודל להיכשל במשימתו, או אפילו להפיק פלט מסוים שהוגדר מראש – כל זאת מבלי לשנות את הקלט באופן מורגש.

ישנן שתי גישות להתקפה זו:

4.5.1 התקפה מכוונת

*בהסבר הבא תיהיה התייחסות למודלים של סיווג ולא מודלים גנרטיביים (Generative Models)

במקרה זה, התוקף מעוניין בפלט ספציפי שהוגדר מראש, לכן הוא יעשה את חישוב פונקציית ההפסד ביחס לפלט שהוא רוצה, ובכל איטרציה הוא ישנה מעט את הקלט בעזרת הגרדיאנט עד שפונקציית ההפסד תהיה מספיק נמוכה ביחס לפלט זה.

דוגמה:

נניח כי ברשותנו מודל למידת מכונה המסווג בין תמונות של כלבים לחתולים. בהתקפה אדוורסרילית, ניתן לעבד תמונה של כלב באמצעות שינויים מינוריים ובלתי נראים לעין האנושית, כך שלאחר ההתקפה, המודל יסווג את אותה כתמונה של חתול – למרות שבעיני המתבונן האנושי, התמונה נותרת בבירור תמונת כלב.

הסבר מתמטי:

Given model $f(x)$ where x is input, Y is the true label and T is the desired label

$$Y \neq T$$

Find $\hat{x} = x + \delta$ where's δ is the perbutation that gives $f(\hat{x}) = T$

4.5.2 התקפה לא מכוונת

כאן המטרה היא לא לקבל פלט ספציפי אלא פלט שגוי כלשהו במטרה לשבש את המודל ולקבל תוצאות לא צפויות. כאן התוקף ישתמש בגרדיאנט ביחס לפלט הצפוי, אבל הוא ישנה את הקלט כך שפונקציית השגיאה תהיה מקסימלית במטרה לא לקבל את הפלט הצפוי.

זוהי ההתקפה אשר תשומש בפרויקט.

בשתי הגישות, הקלט החדש שנוצר נקרא דוגמה אדוורסרילית (adversarial example)

כמו כן, התוקף יגביל את הדוגמה האדוורסרילית בכדי שהשוני בין הדוגמה והקלט המקורי לא יהיו ניתנים להבחנה על ידי בן אדם.

4.6 התקפה קופסה שחורה

במתקפות אדוורסריות מסוג קופסה שחורה (Black-box Attacks), התוקף אינו מחזיק בגישה ישירה למודל הלמידה — כלומר, אין לו ידע על מבנה הרשת, משקלים פנימיים, או גזירת גרדיאנטים. התוקף יכול להפעיל את המודל כקופסה שחורה בלבד: להזין קלטים ולקבל את הפלט (למשל, תווית סיווג או ערך החיזוי), בדומה לגישה חיצונית לממשק API.

במתקפות מסוג זה, התוקף לרוב משתמש באחת מהשיטות הבאות:

4.6.1 התקפה באמצעות חיזוי תחליפי - Surrogate Model

התוקף מאמן מודל חלופי על בסיס דוגמאות קלט-פלט שהתקבלו מהמטרה, ומבצע עליו התקפה לבנה (White-box). לאחר מכן, משתמש באותם רעשי התקפה כדי לתקוף את המודל המקורי. שיטה זו מנצלת את תופעת *transferability* — יכולת של דוגמאות אדוורסריות לפעול על מספר מודלים שונים.

4.6.2 אופטימיזציה מבוססת חיפוש - Query-based Optimization

בשיטות אלו, התוקף שואל את המודל שוב ושוב עם גרסאות שונות של הקלט, ובודק אילו שינויים גורמים לשינוי בפלט. לדוגמה, אלגוריתמים מבוססי אבולוציה, חיפוש אקראי, או אלגוריתם NES (Natural Evolution Strategies) משמשים למציאת שינוי קטן בקלט שיגרום לשגיאה משמעותית במודל.

ישנן כמה שיטות לאופטימיזציה מבוססת חיפוש:

1. מטפס הרים - Hill Climbing

טכניקת אופטימיזציה שבה מתחילים מנקודת התחלה כלשהי (למשל קלט אודיו מקורי), ומבצעים שינויים קטנים ("צעדים") בקלט, כך שבכל שלב בוחרים את השינוי שמשפר את התוצאה (למשל, מגדיל את הסיכוי לטעות של המודל).

בהקשר אדוורסרילי:

התוקף מוסיף רעש קטן לקלט, בודק אם המודל שינה את הפלט (למשל, פירק את הערוצים בצורה פחות מדויקת), ואם כן – ממשיך באותו כיוון. אם לא, מנסה כיוון אחר. זהו תהליך איטרטיבי שדורש פניות חוזרות

למודל. החיסרון הוא שהמערכת שלו יכולה בקלות להיתקע במקסימום מקומי – כלומר, לא תמיד ימצא את ההפרעה האופטימלית.

2. שיטת מונטה קרלו - Monte Carlo Sampling

טכניקה הסתברותית שבה מבצעים דגימה אקראית מתוך מרחב האפשרויות, כדי לאמוד או לגלות את הפתרון הטוב ביותר.

בהקשר אדוורסריאלי:

התוקף יוצר הרבה גרסאות רועשות של הקלט המקורי (noise perturbations), ושולח אותן למודל. על סמך התגובה של המודל (למשל ירידה ב-SDR), הוא מזהה אילו שינויים גורמים להפרעה מקסימלית, ובוחר להעמיק באזורים אלו של מרחב הקלט.

יתרון: לא דורש גרדיאנט, מתאים למצבים בהם יש רק גישה לפלט.

חיסרון: דורש הרבה מאוד דגימות (queries) – יקר בזמן/משאבים.

3. אלגוריתמים אבולוציוניים - Evolutionary Algorithms

שיטה בהשראת האבולוציה הביולוגית. מגדירים "אוכלוסייה" של פתרונות (גרסאות שונות של הקלט), מודדים לכל אחת את איכותה (fitness), ובחרים את הפתרונות הטובים ביותר ל"התרבות" – שילוב ומוטציה יוצרים דור חדש, וחוזר חלילה.

בהקשר אדוורסריאלי:

ה"אוכלוסייה" היא קבוצה של קלטים עם רעש שונה. כל אחד נשלח למודל, נמדדת "כמות ההטעיה" שלו (כמה המודל טועה). בחרים את הדוגמאות הטובות ביותר, יוצרים מהן גרסאות חדשות עם מוטציה, וחוזרים על התהליך.

יתרון: יכול למצוא פתרונות טובים גם במרחב חיפוש מורכב.

חיסרון: איטי, דורש הרבה פניות למודל ודירוג של כל דוגמה.

4.6.3 שיטות ללא תוויות - Score-based or Decision-based

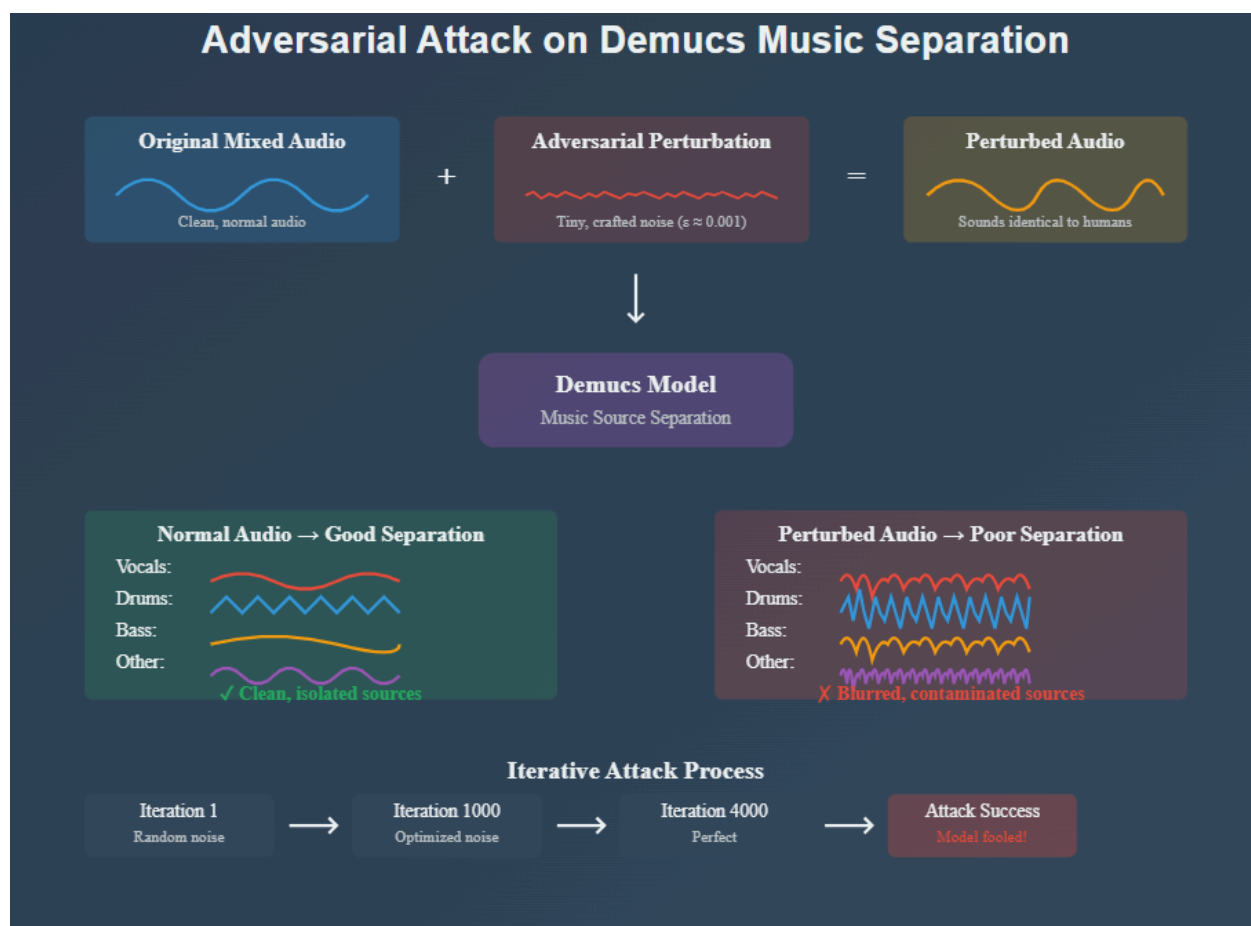
כאשר המודל מחזיר רק את התווית הסופית (ולא את הסבירות או פונקציית ההפסד), משתמשים בשיטות כמו Boundary Attack או Sign-OPT, שמחפשות את הגבול בין הקלאסים על ידי דגימה אקראית והתקדמות הדרגתית.

שלא כמו בהתקפות קופסה לבנה, בהן ניתן להשתמש בגזירת גרדיאנטים ישירה, התקפות קופסה שחורה דורשות בדרך כלל יותר פניות (queries) למודל, והן איטיות יותר, אך מתאימות למצבים מציאותיים יותר – כמו למשל במערכות בענן או שירותים סגורים, שבהם המשתמש לא נחשף למודל הפנימי.

5. מימוש התקפת קופסה לבנה בפרוייקט

כמו שצוין, בפרוייקט מומשה התקפה אדוורסרילית בסגנון קופסה לבנה לא מכוונת על מודל של הפרדת ערוצים של שיר.

עיקר ההתקפה מתבססת על מאמר של [3] wagner and carlini שהתמקד בהתקפה מכוונת על מודל של דיבור לטקסט, כלומר בהינתן דיבור כקלט המודל מפרש את זה לטקסט כפלט.



איור 7 - תרשים של דרך הפעלת הדוגמה האדוורסרית על המודל

ההתקפה נעשית באמצעות חישוב הגרדיאנט ביחס לקלט באופן איטרטיבי, כלומר בכל צעד אנו משנים מעט את השיבוש שנוסף לקלט המקורי בכיוון של הגרדיאנט ובכך מייעלים את ההתקפה.

דבר זה נעשה על ידי בניית אלגוריתם שמבצע:

א. מוסיף רעש קטן לקלט

ב. מעביר את הקלט דרך המודל של דימוקס

ג. מחשב שגיאה הפוכה, כלומר פונקציית השגיאה דומה לפונקציית השגיאה של דימוקס אבל מוכפלת במינוס אחד. (המטרה זה למקסם את השגיאה המקורית = למזער את השגיאה ההפוכה).

ד. מחשב את הגרדיאנט ביחס לשגיאה ועושה צעד בכיוון, כלומר משנה את הרעש שנוסף לקלט.

ה. אם יש צורך, מגביל את הרעש שלא יהיה חזק מדיי באמצעות אפסילון כלשהו.

ו. חזור לא'.

הגבלת הפרמוטציה

ישנן שיטות שונות ומתוחכמות להגבלת הפרמוטציה בין אם בתמונה או בשמע.

בפרוייקט יש שימוש בשיטה פשוטה שנקראת $|perturbation_i| \leq \epsilon$ The L-infinity norm of a vector:

כלומר, בהינתן אפסילון נעשה "חיתוך" ערכים בערך מוחלט שקטנים מאפסילון שיהיו לכל היותר אפסילון או מינוס אפסילון.

פרמטרים

איטרציות: 5000, אבל גם 4000 יכול לעבוד.

אפסילון: 0.0005

קצב למידה (learning rate alpha) 0.00005

6. אלגוריתם ההגנה

מנגנון ההגנה נועד לנטרל את ההשפעה של הדוגמה האדוורסריראלית על המודל. הרעיון המרכזי במנגנון זה הוא להוסיף רעש רנדומלי קטן לקלט שהותקף, כך שהרעש יהיה בלתי נשמע לאוזן האנושית ועדיין ישבש את הדוגמה האדוורסריראלית שהוספה לאות על ידי מודל ההתקפה ובכך להחזיר את הפלט של המודל לתוצאה תקינה יותר.

איך האלגוריתם עובד?

1. הוספת רעש מוגבל לפי נורמת אינסוף (L_∞):

עבור כל קובץ מותקף, נוצרת מטריצת רעש אקראית בגודל זהה לקובץ השמע. הרעש מנורמל לטווח $[-1, 1]$, ואז מותאם לגבול קבוע מראש ϵ (למשל 0.003), כך מובטח שהרעש לא יחרוג מהטווח הרצוי בכל נקודה.

2. חיבור הרעש לקובץ המותקף:

קובץ האודיו מועתק עם תוספת הרעש והתוצאה נשמרת תחת שם חדש בתיקיית הפלט.

ההיגיון מאחורי זה:

ההתקפה האדוורסרית מחושבת בצורה מאוד מדויקת. כל שינוי קטן בקלט נבחר בכוונה לפי כיוון הגרדיאנט – כדי למקסם את השגיאה של המודל. כלומר, מדובר בשיבוש מאוד ספציפי, מאוד רגיש, ולא אקראי. כשמוסיפים רעש אקראי – אפילו קטן, הוא "שובר" את האיזון הזה בכך שהוא מערבב את הקלט מחדש, והופך את השיבוש הקודם ללא אפקטיבי, בלי לפגוע באופן מורגש בשמע המקורי (כי הרעש מאוד קטן).

7. מסד הנתונים MUSDB18

שומש במסד הנתונים MUSDB18:

<https://paperswithcode.com/dataset/musdb18>

מסד זה כולל 150 שירים המחולקים ל-100 שירי אימון, ו-50 שירי בדיקה המודל של Demucs אותו נבחר לשפר, אומן על דאטהסט זה מסד נתונים זה נבחר משתי סיבות עיקריות:

1. לצורך בחינה מדויקת של השפעת התקיפה וההגנה, יש לבצע את הניסויים על שירים הדומים באופיים לאלו ששימשו לאימון מודל Demucs. שימוש בדוגמאות שאינן מייצגות את תחום האימון של המודל עלול להוביל להפרעות או שיבושים בתוצאות, אשר מקורם לא ניתן לזיהוי חד-משמעי – האם נובעים מהתקיפה עצמה או מהתאמה לקויה בין הדאטה למודל.

לדוגמה, במידה שתיבחר דוגמת שמע מז'אנר מוזיקלי שונה (כגון רוק כבד), והמפריד ייכשל בפירוק תקין של הערוצים, לא יהיה ניתן לייחס את הכישלון בהכרח להתקפה, שכן ייתכן והמודל מלכתחילה לא הותאם לז'אנר זה במהלך האימון, והביצועים הנמוכים צפויים גם ללא כל שיבוש יזום.

2. המסד הזה מספק נתוני פירוק אמיתיים של כלי הנגינה. כלומר עבור כל שיר במסד הנתונים הזה יש לנו:

- a. Mixture - השיר הרגיל
- b. Drums - פירוק ערוץ התופים, כולל כלי הקשה ואפקטי מקצבים אלקטרוניים
- c. Bass - קונטרה בס או כלי בס שונים\ ערוץ הבס המופרד, יכול להיות גיטרה בס\ בס אלקטרוני- Bass
- d. Vocals - ערוץ הקולות המופרד, כל מה שנקלט כקול אנושי - Vocals
- e. Other - כל שאר כלי הנגינה והאפקטים שלא נכנסו לאחד מהערוצים המופרדים לעיל - Other

הפורמט

כל שיר מיוצג ע"י קובץ בפורמט song_name.stem.mp4

הפורמט הזה מכיל את חמשת הקבצים לעיל

נכתב קוד שיפרק את קובץ ה stem לחמשת ערוצי השמע הרצויים בפורמט wav (באיכות גבוהה)

8. רשימת מקורות

8.1 מאמרים

[1]

Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2025). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations (NIST AI 100-2e2025). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.AI.100-2e2025>

[2]

Rouard, S., Massa, F., & Défossez, A. (2023). Hybrid transformers for music source separation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.

[3]

Carlini, N., & Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP) (pp. 1-15). IEEE.

[4]

Alexandre Défossez, Nicolas Usunier, Léon Bottou, Francis Bach
(2021). Music Source Separation in the Waveform Domain

[5]

Romain Hennequin, Anis Khlif, Felix Voituret, Manuel Moussallam
(2019). SPLEETER: A FAST AND STATE-OF-THE ART MUSIC SOURCE
SEPARATION TOOL WITH PRE-TRAINED MODELS

[6]

Simon Rouard, Francisco Massa, Alexandre Defossez
(2022). HYBRID TRANSFORMERS FOR MUSIC SOURCE SEPARATION

8.2 מקורות נוספים

מסד הנתונים MUSDB18:

<https://paperswithcode.com/dataset/musdb18>

Evaluation Metrics for Speech(Audio) Signal Processing:

<https://medium.com/@poudelnipriyanka/audio-metrics-their-importance-and-their-necessity-417950b0d848>

9. תוצאות

9.1 הסבר על הגרפים

Original Separation = evaluate the original channels compared to the demucs estimation separation

Attack Effect = comparing the demucs operation on the original song to the demucs operation on the attacked song

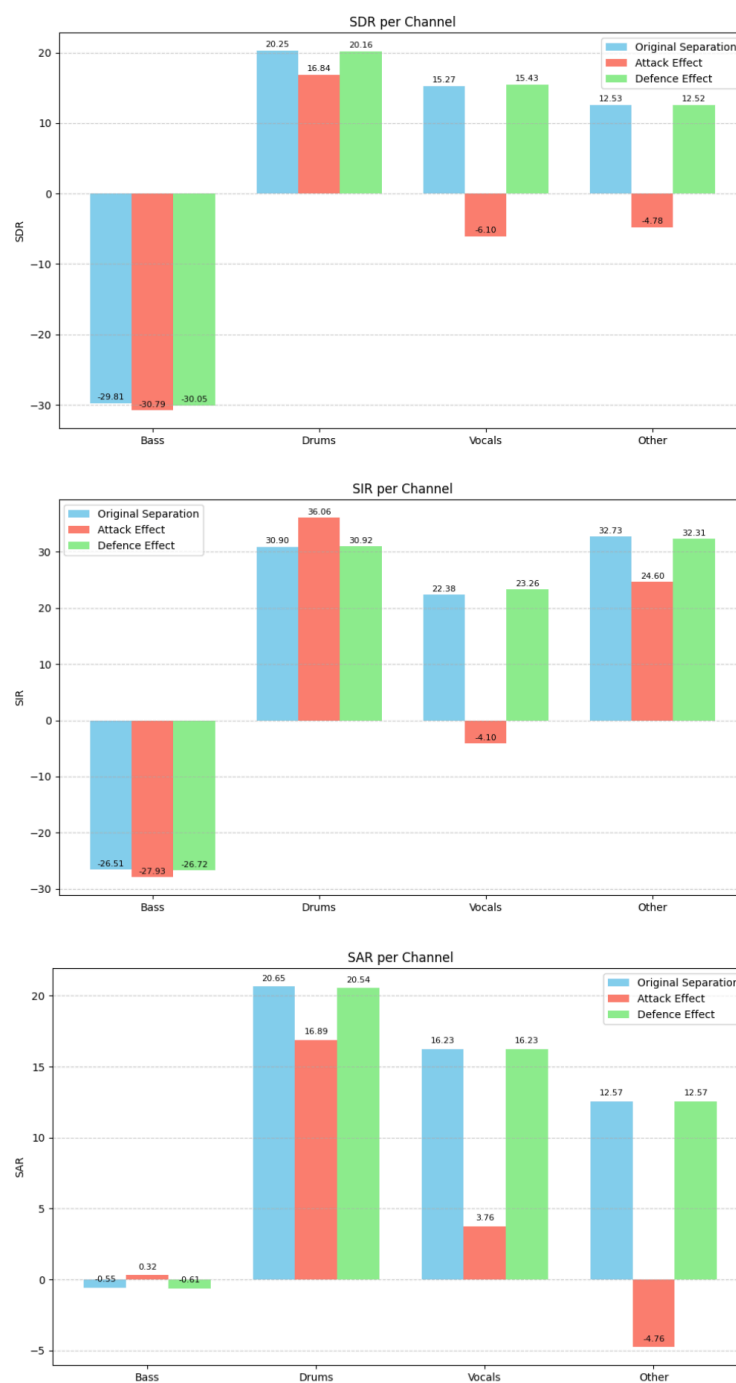
Defense effect = comparing the original channels to the demucs operation on a song that has been attacked and defended

לקחנו 7 שירים מתוך מסד הנתונים MUSDB18 מתוך תיקיית הבדיקה ועליהם ביצענו:

1. התקפה
2. הגנה
3. הערכה

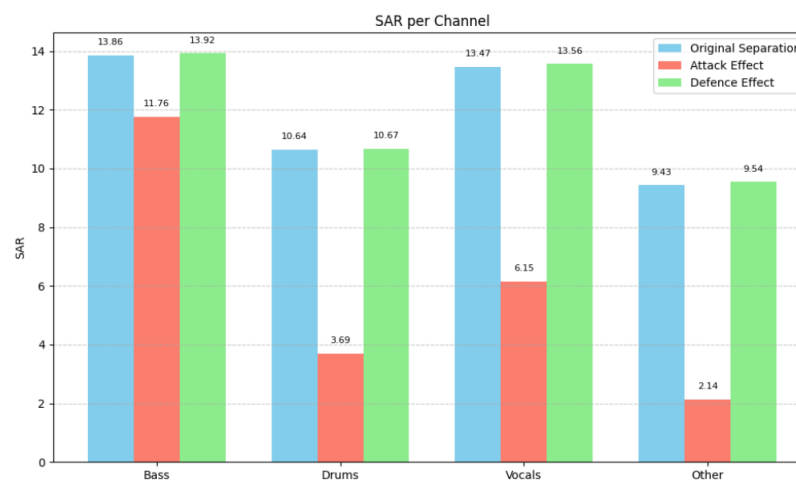
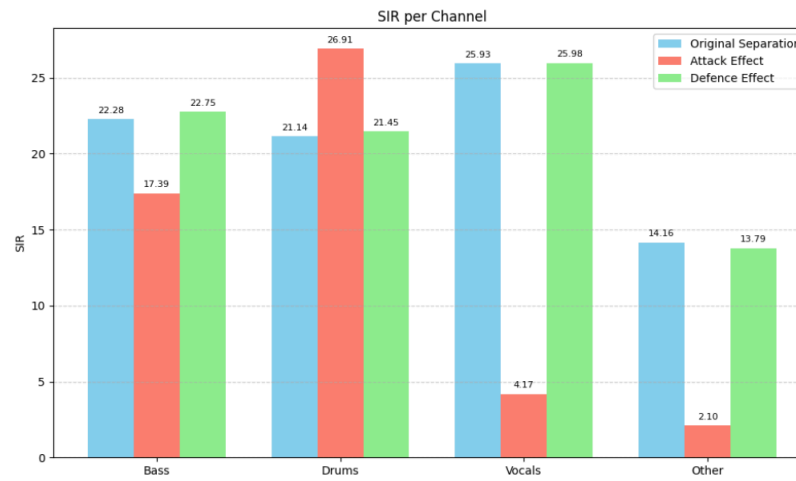
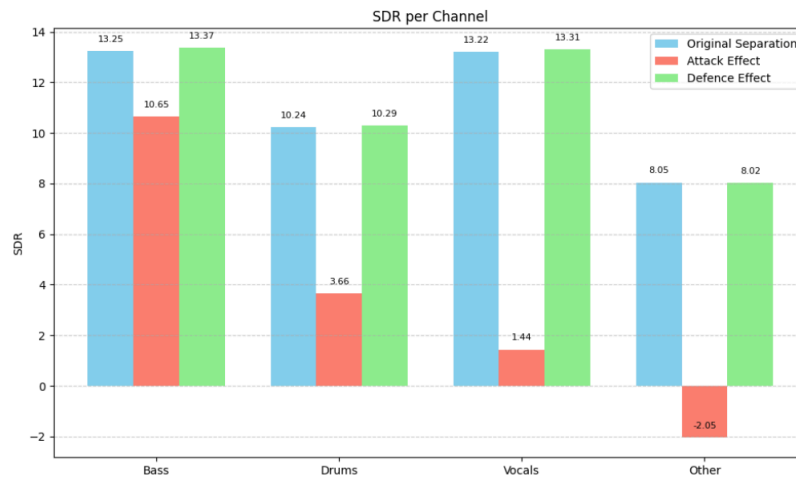
לכל שיר מודפסים שלושה גרפים, כל גרף מייצג מטריקת הערכה שונה (SAR, SIR, SDR)

Song 1: AM Contra - Heart Peripheral



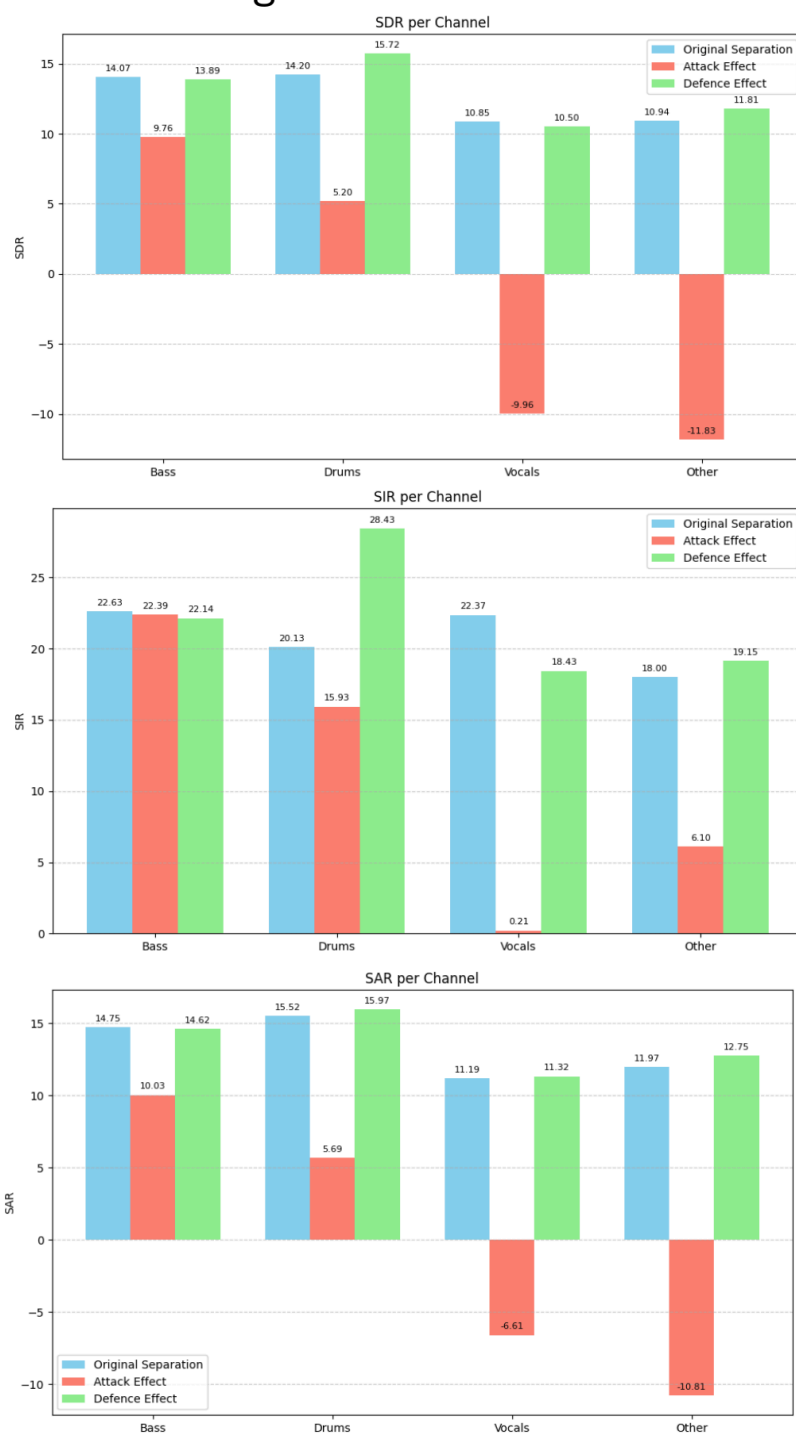
איור 8 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 1

Song 2: Angels In Amplifiers - I`m Alright



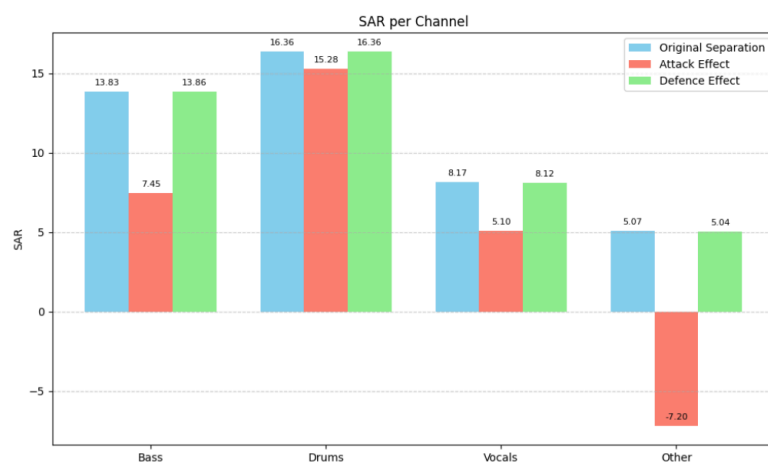
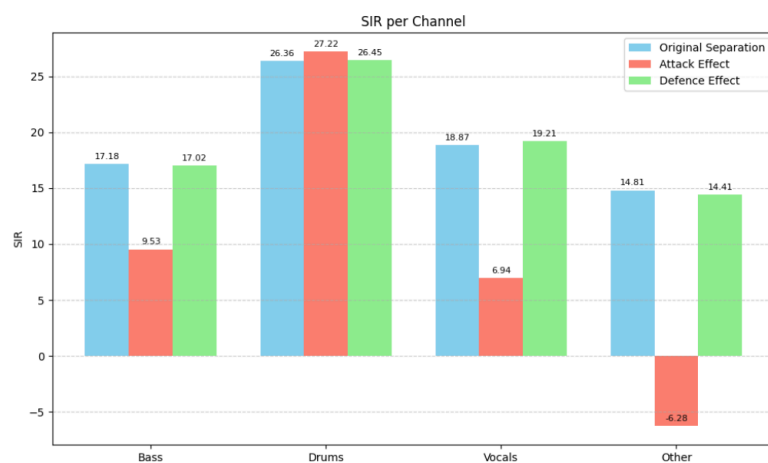
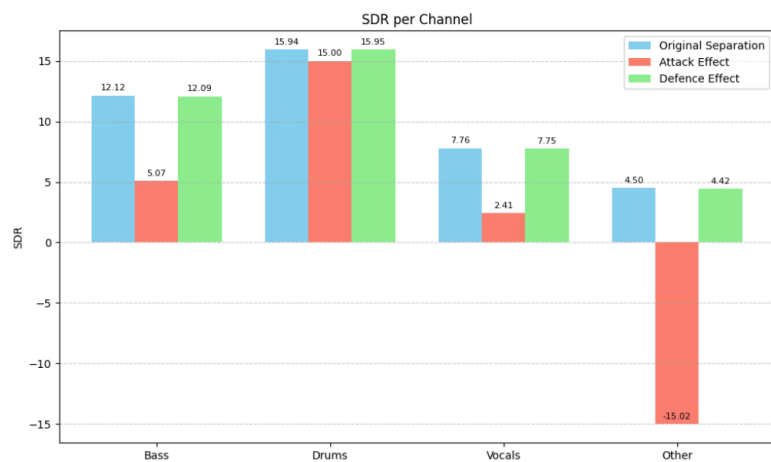
איור 9 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 2

Song 3: Ben Carrigan - Well Talk About It All Tonight



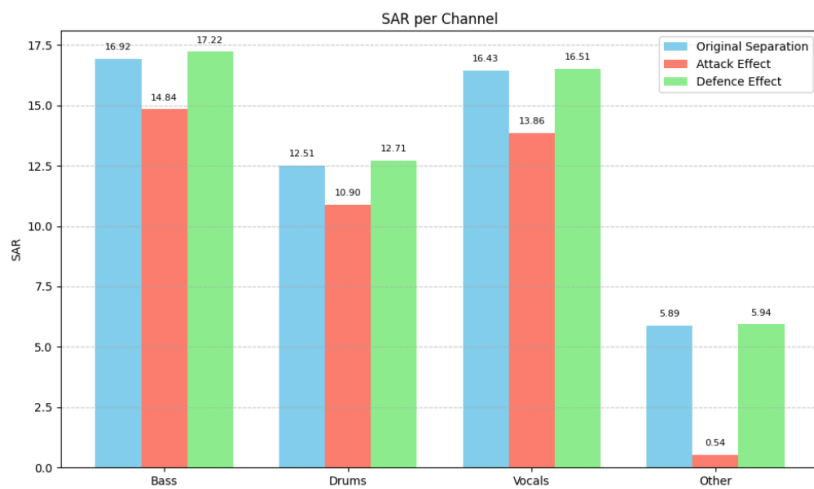
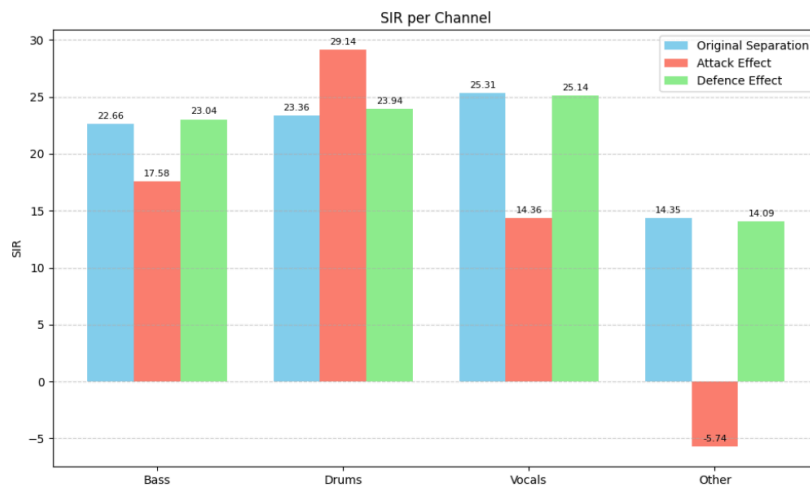
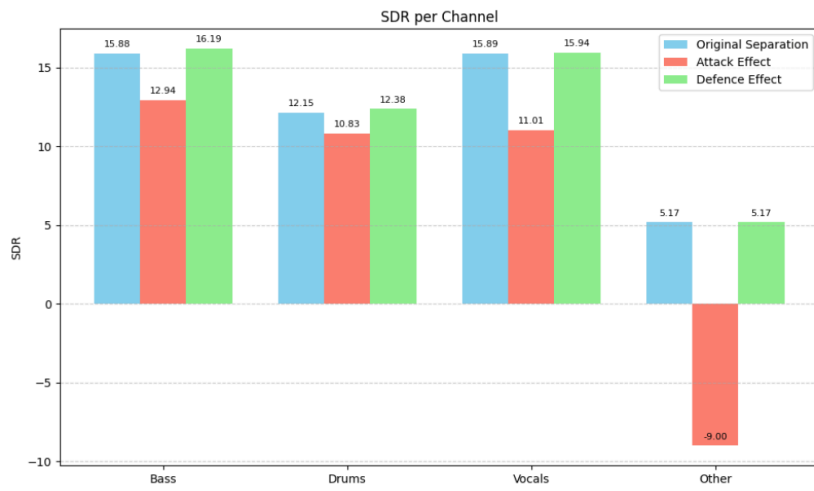
איור 10 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 3

Song 4: BKS - Bulldozer



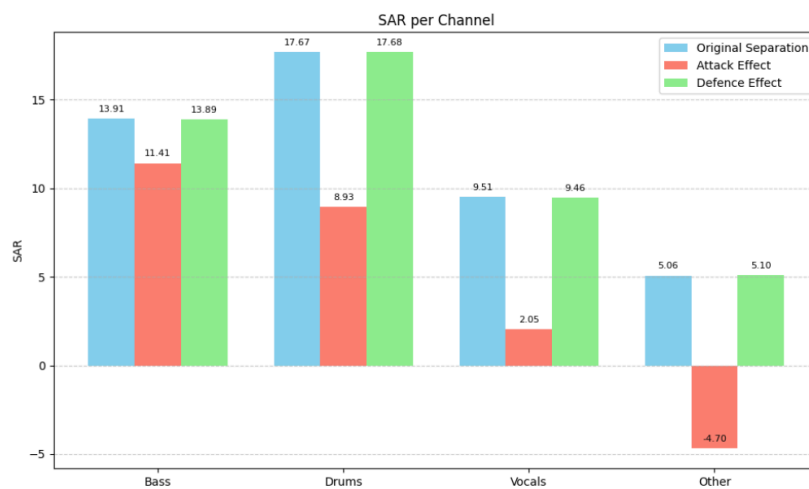
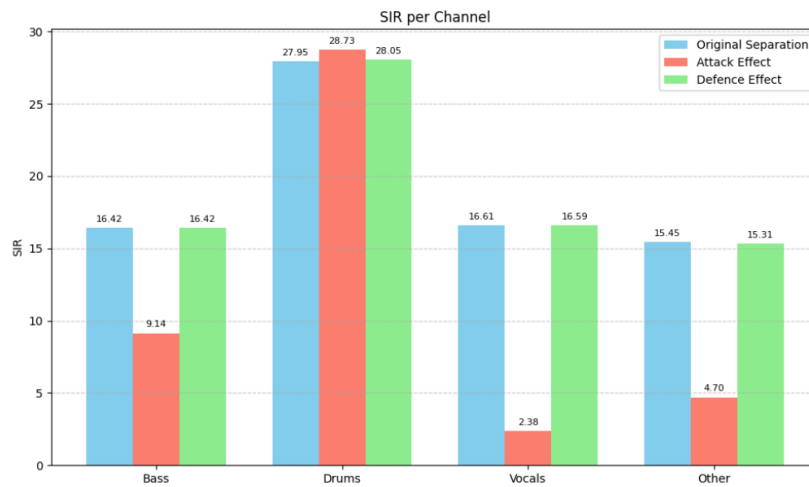
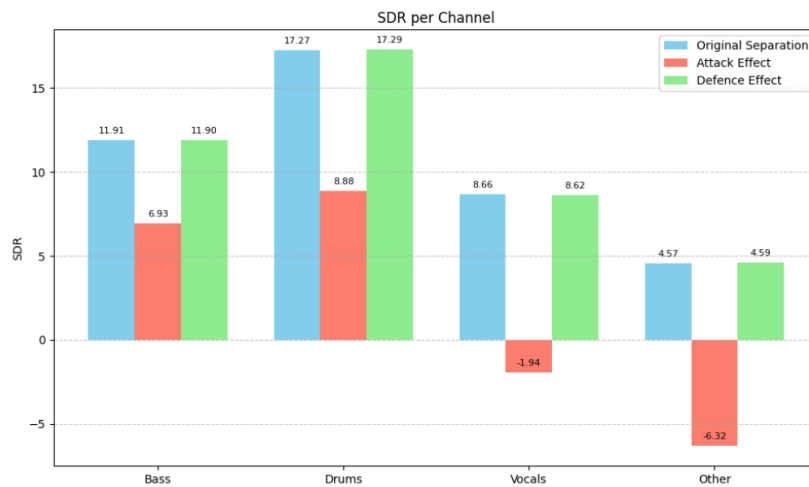
איור 11 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 4

Song 5: BKS - Too Much



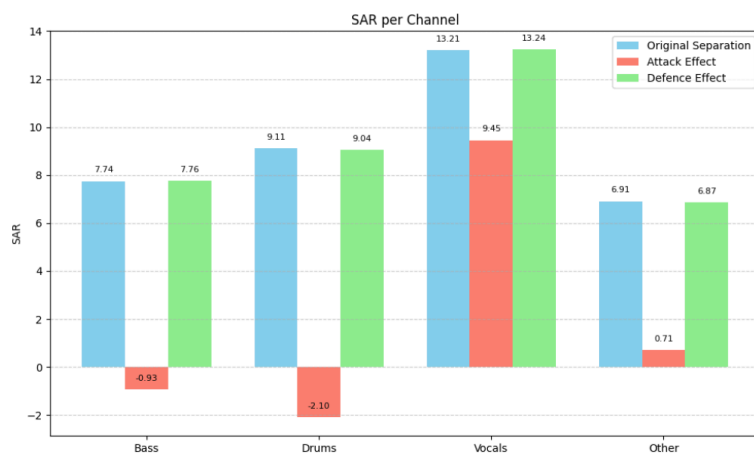
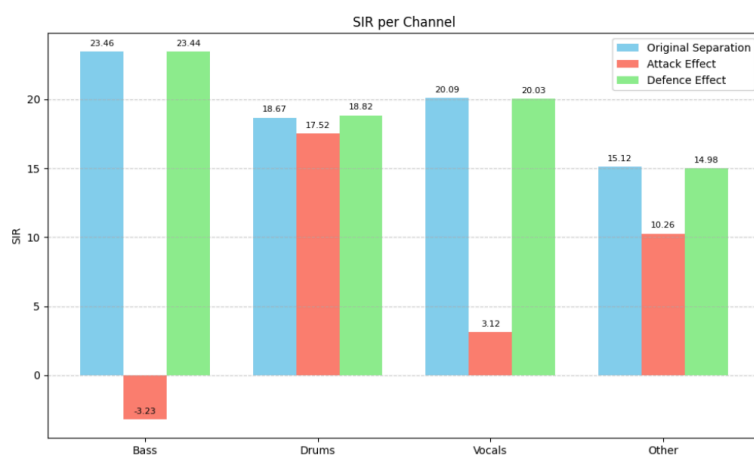
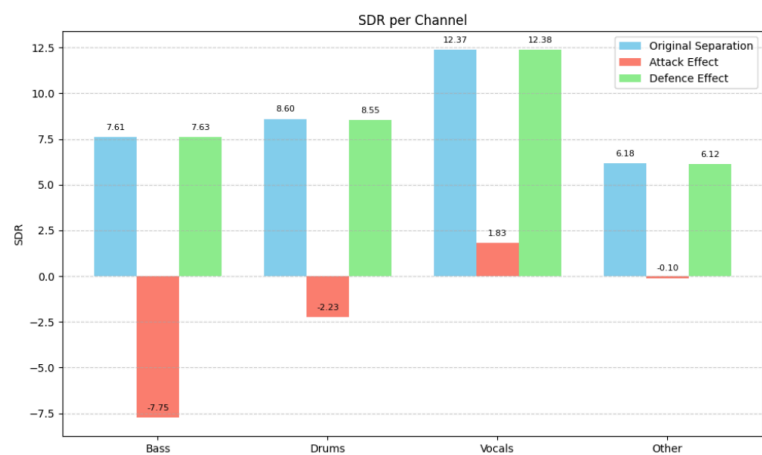
איור 12 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 5

Song 6: BKS - Buitraker - Revo X



איור 13 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 6

Song 7: Carlos Gonzalez - A Place For Us



איור 14 - הצגת ביצועי SDR, SIR, SAR של המקור, ההתקפה וההגנה של שיר 7

10 פרק סיכום ומסקנות

10.1 ניתוח תוצאות

מהשוואת תוצאות הניסויים עולה בבירור כי התקיפה האדוורסרית משפיעה באופן נרחב על איכות הפירוק שביצע המודל. ירידה משמעותית נצפתה בערכי המדדים SIR, SDR ו-SAR לאחר התקיפה, דבר המעיד על פגיעה כוללת באיכות ההפרדה, עלייה בהפרעות בין הערוצים, והופעת עיוותים ועיבוד לא טבעי באות המפוצל.

לאחר הפעלת מנגנון ההגנה, התקבלה מגמת שיפור ברורה בכל המדדים – כאשר במרבית השירים, ערכי SIR, SDR ו-SAR התקרבו מאוד לערכים שהיו לפני התקיפה, ולעיתים אף חזרו כמעט במלואם לרמה המקורית. הדבר מצביע על יעילות גבוהה של ההגנה בנטרול השפעת השיבוש.

ניתן לראות בשירים 1-6 במטריקת ה SIR בפירוק של ערוץ other ובשירים 1,3 ו-6 במטריקת ה SIR בפירוק של ערוץ ה Vocals שהערך המקורי והערך לאחר ההגנה הוא חיובי, ולאחר המתקפה הערך של ה SIR הוא שלילי. מה זה אומר?

כאשר $SIR < 0$, כלומר שלילי, זה אומר שה:

- ההפרעה חזקה יותר מהסיגנל המקורי.
- המודל הוציא ערוץ שבו רוב מה ששומעים כלל לא שייך לערוץ זה.
- **ייתכן שהערוץ כמעט ריק מהתוכן הרצוי, ומלא בשאריות של ערוצים אחרים (למשל: שומעים שירה בתופים).**

במילים אחרות:

המודל נכשל בהפרדה – הוא זיהה בצורה גרועה מה שייך לערוץ הזה, עד שה"זליגה" דומיננטית יותר מהמקור.

תוצאה מעניינת ולא צפויה עלתה במדד SIR עבור ערוץ התופים, בשירים 1 עד 6: נמצא כי לאחר התקיפה – לפני הפעלת ההגנה – נרשמה עלייה במדד SIR, כלומר לכאורה חלה הפחתה בהפרעות מערוצים אחרים דווקא בעקבות התקיפה. תופעה זו לא הופיעה בשיר 7.

תוצאה זו עשויה להצביע על כך שהשיבוש האדוורסרילי, שבמכוון נועד למקסם את השגיאה הכוללת, גרם למודל "להתמקד" בתדרים מסוימים באופן שהקטין את הדליפה מערוצים אחרים דווקא בערוץ התופים.

אחת ההשערות האפשריות להסבר התופעה היא ש-ערוץ התופים מכיל תדרים חזקים ודומיננטיים במיוחד ביחס לשאר הכלים. מאפיין זה עשוי לגרום לכך שכאשר מוזרם שיבוש אדוורסרילי לקלט, הפגיעה היחסית בתופים נמוכה יותר, או שהמודל "מתבלבל" לטובת הדגשת רכיבי הקצב, ובכך מופחתת זליגה מערוצים אחרים. לחלופין, ייתכן שהשיבוש פוגע בכלי אחרים ומחליש את ההשפעות שלהם על התופים, דבר שמוביל ל-SIR גבוה – גם אם איכות הצליל הכללית נפגעה.

יש לזכור שמדד SIR בלבד אינו משקף את כל האספקטים של איכות ההפרדה, ויש לבחון אותו בשילוב עם SDR ו-SAR על מנת להבין את מלוא המשמעות של התוצאה.

תופעה זו מצביעה על חשיבות בחינה רב-ממדית של איכות ההפרדה, ועל הצורך בזיהור בפרשנות של מדד אחד בודד ללא הקשר רחב יותר.

10.2 מסקנות

מהממצאים שהתקבלו עולה המודל **Hybrid Transformer Demucs**, רגיש באופן מובהק לתקיפות אדוורסריות מסוג **התחמקות (Evasion Attack)**.

התקפה זו, שפותחה במקור עבור מודלים בתחום ראיית מכונה (כגון סיווג תמונות), הותאמה במסגרת פרויקט זה לפעול גם על אותות שמע רציפים, והוכחה כי **עילה במיוחד בהשפעתה – הן על איכות ההפרדה והן על מידת השיבוש בתוצרי הפלט**.

המסקנה המרכזית היא כי **הפגיעות למתקפות אדוורסריות איננה ייחודית למודלים ויזואליים**, אלא קיימת גם בתחומים מבוססי שמע. לפיכך, קיימת חשיבות מהותית בהמשך מחקר בנושא זה, ובפיתוח מנגנוני הגנה ייעודיים גם למודלים שפועלים מחוץ לעולמות התמונה והטקסט.

תודות

ברצוננו להודות למר **אורי בריט**, שהנחה אותנו לאורך הפרויקט במסירות, סבלנות וראייה רחבה. תחת הדרכתו קיבלנו לא רק הכוונה מדעית מדויקת, אלא גם חופש לחקור, להתנסות ולפתח רעיונות באופן עצמאי וביקורתי. התמיכה והזמינות שלו היוו עבורנו עוגן משמעותי בתהליך הלמידה, ואפשרו לנו להתמודד עם אתגרים מורכבים הן ברמה הטכנית והן ברמה המחקרית.

הפרויקט לא היה נראה כך ללא הליווי המקצועי והאנושי שלו, ועל כך תודתנו העמוקה.