



**בי"ס להנדסת חשמל**

**פרויקט מס' 1481**

**תכנית עבודה**

**שם הפרויקט:** Audio processing platform

**מבצעים:**

ת.ז. 323829853	שם: אדר מאורי
ת.ז. 318255072	שם: בן פוזננסקי

**מקום ביצוע הפרויקט:** אוניברסיטה

**לשימוש המנהה:**

הנני מאשר את **תכנית העבודה** המצורפת

שם: ליאור ארבל

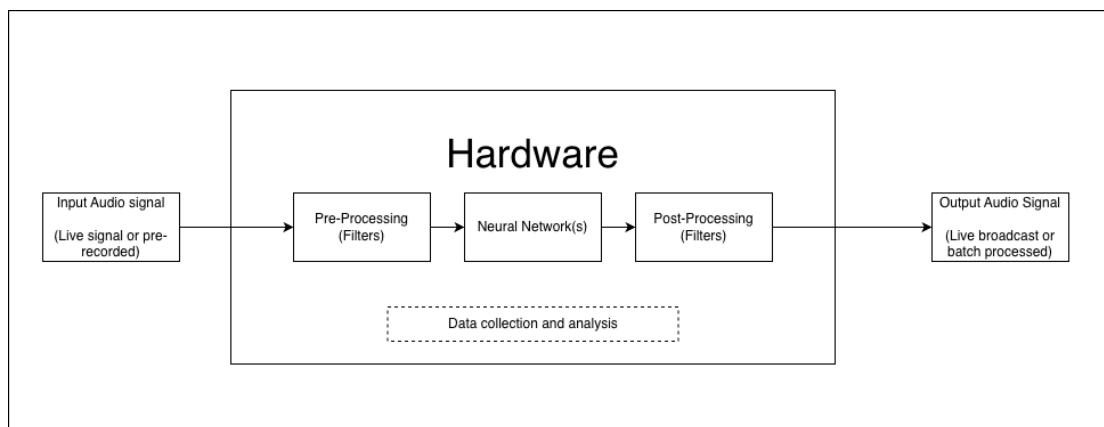
חתימה: ליאור ארבל

## 1 תקציר

פרויקט זה בוחן את האתגרים וההזדמנויות בשימוש ברשתות נירוניים (NN) עבור ישומי עיבוד אודיו הדורשים תגובה בזמן אמת (real-time). בעוד שרשתות נירוניים מוגנות יכולות מתקדמות במשימות אודיו כגון סינון רעשים, ייצור אפקטים מוזיקליים וזיהוי דיבור, הן דורשניות מאוד מבחינת כוח חישוב, דבר המהווה חסם משמעותי במערכות זמן-אמת. מטרת הפרויקט היא לבצע מחקר השוואתי של ביצוע רשתות נירוניים שאומנת מראש למטרות אודיו, על גבי פלטפורמות חומרה מגוונות. במסגרת הפרויקט, נמשך ונבחן את אותה הרשת על חומרות שונות, ביניהן מעבד מרכזי (PC), חומרה מיתכנת (FPGA) ופלטפורמת עיבוד *Bela*.

בנוסף, נחקור את ההשפעה של אלגוריתמים שונים לימוש פעולות הליבה של הרשת (כמו כפל מטריצות) על הביצועים הכלולים, בדגש על זמן ההשניה (latency) וניתוח המשאים. תוצר הפרויקט יהיה ניתוח מקיף ומתקף של החלופות השונות, ופיתוח שיטות מיושן אופטימלית/מתודות חדשות להטאמת עיבוד אודיו בזמן אמת באמצעות רשתות נירוניים, פרט פלטפורמה.

диagramma blokow sl. 1:



## **2 מוטיבציה**

המוטיבציה המדעית וההנדסית לפרוייקט זה נובעת מהפער הקיים בין יכולות התאורטיות של רשתות ניירונים (NN) בתחום האודיו, לבין הקושי המעשני בישוםן במערכות זמן אמת. כיום, רשתות ניירונים הן כלי ניידים המשמשות מרכיבות של עיבוד אותות, אך הבעיות החישובית הגבוהה שלהם מתקשה על שילובם במערכות הדורשים latency נמוך מאוד, כגון מכשירי שמיעה, מערכות תקשורת חיות, או אפקטים מוזיקליים לגינה חייה. תרומת הפרויקט היא במתן מתודולוגיה וניתוח כמוות לבחירת צורת השימוש (חומרה ואלגוריתמייה) הייעילה ביותר עבור יישום נתון, ובכך לאפשר את הרחבת השימוש בטכנולוגיות NN מתקדמות בתחוםים אלו.

השווואה לחלופות קיימות:

1. **אלגוריתמי עיבוד אותות קלאסיים (DSP):** שיטות מסורתיות כמו מסנני FIR/IIR, ואלגוריתמים אדפטיביים (LMS, RLS) הן בעליות מאד מבחינה חישובית ובעלות latency נמוך. עם זאת, הן מתקשות להתמודד עם בעיות מרכיבות ולא-ליינאריות כמו הפרדת מקורות או סינון רעשים לא-سطציוניים, משימות שבתן רשתות ניירונים מצטיניות.
2. **עיבוד מבוסס ענן (Cloud-based Processing):** חלופה זו מאפשרת שימוש במשאבי חישוב כמעט בלתי מוגבלים להרצת רשתות ניירונים מרכזיות. החיסרון המרכזי והמכריע הוא זמן ההשאה (latency) הנגרם מה צורך לשולח את האודיו לשרת מרוחק, לעבד אותו ולקבל את התוצאה בחזרה. זמן ההשאה זה אינו קביל ברוב יישומי זמן האמת.

### 3 תכולת העבודה

תכולת העבודה בפרויקט מוחולקת למספר שלבים עיקריים, החל מרכישת הרקע התאורטי ועד לניתוח התוצאות והסקת מסקנות.

#### **רקע תאורטי נדרש:**

- **רשתות נוירוניים:** הכרת ארכיטקטורות רלוונטיות לעיבוד אודיו, בעיקר רשתות קונבולוציה (CNN) ורשתות רקורסיביות (RNN/LSTM).
- **עיבוד אותות ספרתי (DSP):** עקרונות דגימה, קוונטיזציה, ניתוח ספקטרלי (STFT) וחילון.
- **ארכיטקטורת חומרה:** הבנת ההבדלים בין ארכיטקטורות SoC על PC, ומערכות משובצות (Embedded) כמו Bela, FPGA, בדגש על ביצוע פעולות אריתמטיות מוקביליות.
- **אלגוריתמים לחישובים נומריים:** לימוד אלגוריתמים לכפל מטריצות (GEMM) וקונבולוציה, והכרת ספריות ממוחשבות כמו NNBLAS, cuDNN ו-CMSIS-NN.

#### **דרישות ושלבי עבודה:**

1. **שלב א' - מחקר והקנת סביבה:**
  - סקירה ספורת ובחרית 2-3 רשתות נוירוניים מאומנות וזרימנות בקוד פתוח למשימות כמו TrainWaveNet או RNNNoise.
  - הקמת סביבות הפיתוח והבדיקה על כל אחת מהחומרות: PC (עם Python), FPGA (עם Verilog), PyTorch/TensorFlow (עם C++) ופלטפורמת Bela.
2. **שלב ב' -นำไปש ובחינת ביצועים:**
  - นำไปש גרסת בסיס של הרצת המודול (inference) על כל פלטפורמת חומרה.
  - นำไปש ואינגרץיה של לפחות שתי שיטות שונות לביצוע כפל המטריצות/קונבולוציות בראשת (למשל,นำไปש נאיבי מולนำไปש המסתמך על ספרייה מומובבת).
  - פיתוח מערכת בדיקות (benchmark) סטנדרטי למדידת ביצועים: זמן השהיה (end-to-end latency), קצב עיבוד throughput וניתוח משאבי CPU/RAM.
  - הרצת הבדיקות ואיוסוף נתונים כמותיים.
3. **שלב ג' - ניתוח וтиיעוד:**
  - ניתוח השוואתי של התוצאות והציגן בצורה גרפית.
  - זיהוי צוואר הבקבוק בביטויים עבר כל תצורה (חומרה + אלגוריתם).
  - יישוש המלצה מנומקת לשיטתนำไปש אופטימלית בהתאם למוגבלות המערכת (למשל, "מעבר latency מינימלי, מומלץ להשתמש ב-FPGA בימיosh X", או "מעבר מערכת DLL משאבי Bela עם אלגוריתם Y נתן את התוצאה הטובה ביותר ביוטר").

#### **כלים, סביבה וחומרה:**

##### תוכנה

- C/C++

- Python (PyTorch, TensorFlow, Librosa, PANDAS)

- CUDA

##### חומרה

- (SoC) PC

- FPGA "יעודי"

- Bela Starter Kit

## 4 תוצר הפרויקט

### **תוצר לסופ סטטוס א' (הציג התקדמות):**

- דוח מסכם של סקר הספרות ובחירה המודלים והאלגוריתמיים הנבחנים לכל פלטפורמה.
- מימוש ראשוני של פחת מודל אחד על פלטפורמת Bela-PC כולל תוצאות המדגימות את פועלתו (למשל, קובץ אודיו לפני ואחרי סינון רעשים או הוספת אפקט).
- מימוש ראשוני של פחת מודל אחד על פלטפורמת-h-PC כולל תוצאות המדגימות את פועלתו (למשל, קובץ אודיו לפני ואחרי סינון רעשים או הוספת אפקט).

### **תוצרים סופיים:**

- תיאור דרישות המערכת:
  - כל החומרות:
    - זמן השהייה (Latency): שיפור בזמן הריצה תוך פגיעה של עד 5% באיכות.
    - ניצול משאבי: המימוש לא עברו 80% מניצול הזיכרון והמעבד בממוצע, כדי לאפשר פועלות מערכות אחרות.
  - PC
    - שיפור ב-20% בהשניה לעומת מודל דומה בענין, ללא פגיעה באיכות.
- קוד מקור וסימולציות:
  - מאגר קוד (Git Repository) המכיל את כל המימושים, סקרים להערכת הביקורת, וניתוח הנתונים.
  - דוח ביצועים מקייף המשווה את כל התוצאות שנבדקו.
- שיטות בדיקה וסביבת בדיקה:
  - סביבת בדיקה: המערכת מקבל קלט אודיו מקובץ WAV סטנדרטי. הפלט המעובד ישמר לקובץ WAV נפרד. מדידות latency וניצול משאבי יתבצעו באמצעות כלי פרופילינג תכונתיים (software profiling).
  - בדיקות:
    - בדיקות פונקציונליות: האזנה סובייקטיבית לתוצאות בדיקה ויזואלית של הספקטרוגרמה כדי לוודא שהעיבוד מתבצע כמצופה.
    - בדיקות ביצועים כמותיות: הרצת כל מימוש 100 פעמים על אותו קובץ קלט ומדידת ממוצע וטויות תקן של זמן העיבוד וניצול המשאביים.

## 5 לוח זמנים

הערות	תאריך יעד לביצוע	פירוט (3-2 שורות)	אבן דרך
	21.12.2025	בחינה של כל רכיב חומרה - IO, משק תובנה, קצבי עבודה. רשותות נוירונים - סקירת המודלים וההתאמתם לוחמות אלגוריתמים - IO, מנוי ריצה	השלמת סקר ספרות
	1.2.2026	הקמת סביבת עבודה בפלטפורמת PC ובחינה עם מודל/אלגוריתם פשוט	
	8.2.2026	הקמת סביבה עבודה בפלטפורמת Bela ובחינה עם מודל/אלגוריתם פשוט	הקמת סביבות עבודה
	15.2.2026	הקמת סביבת עבודה בפלטפורמת FPGA ובחינה עם מודל/אלגוריתם פשוט	
	פברואר-מרץ 2026		הגשת מצגת האמצע
	12.4.2026	כל הפלטפורמות תקין, התקדמות משמעותית בבחינת מודלים ואלגוריתמים רלוונטיים	הציג מצגת אמצע סטט
	20.4.2026	שילוב פילטרים ופקטים אחרים לפני/אחרי הריצת רשותות הבניורונים	שילוב pre ו- post processing
	29.4.2026	שילוב האלגוריתמיקה המתقدמת וביצוע אופטימיזציה למן הריצה במימוש המודלים על PC.	
	7.5.2026	שילוב האלגוריתמיקה המתقدמת וביצוע אופטימיזציה למן הריצה במימוש המודלים על Bela.	שילוב אלגוריתמים ומודלים מתקדמים
	14.5.2026	שילוב האלגוריתמיקה המתقدמת וביצוע אופטימיזציה למן הריצה במימוש המודלים על FPGA.	
	21.5.2026	ניתוח ומנוי ריצה, עלות חומרתית ואיוכית התוצאים כתלת באlgorigthmic, פילטרים, חומרה, תוכנה	ניתוח תוצאות
	24/5/2026		הגשת הפוסטר וסיום העבודה בפרויקט
	שבוע לפני הגשת ספר הפרויקט		אישור ספר הפרויקט
	יולי-אוגוסט 2026		הגשת ספר הפרויקט ומצגת הסיום