

תיעוד פרויקט באינטרנט של הדברים

אביב תשפ"ב

מספר קורס :

263333

יואב יעבץ

ת.ז. : 212617864

מור לוי

ת.ז. : 212617864

גלעד שמרלר

ת.ז. : 212139240

מנחי הפרויקט :

איתי דברן, תום סופר וירון פורת

תוכן עניינים

תיעוד התקדמות של הפרויקט.....3-5

טבלת יעדים.....6-7

סיכומי פגישות.....8-9

אופטימיזציות.....11-12

גרף התקדמות.....13-14

מקורות מידע.....15

פוסטר.....16

תיעוד התקדמות של הפרויקט

| סוף שבוע | התקדמות בתחום החומרה | התקדמות בתחום התוכנה |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 | התחלת עבודה על הפרויקט וניסיון לחבר את המיקרופון ולבדוק את הפלט שהוקלט על ידי המיקרופון. | |
| 3 | הצלחנו לחבר את המיקרופון ונראה שהוא מצליח לקלוט צלילים אבל לא ברמה שמספיקה בשביל לשמוע אחר כך בפלט. | תחילת מחקר של איך עובדת רשת הניורונים של זיהוי wake word. איך כדאי ליצור את הרשת בצורה יעילה, והיכן כדאי לבצע את הזיהוי (לוקאלית או בענן). |
| 4 | הצלחנו לחבר את המיקרופון ולייצא קובץ WAV לשרת ולהוריד אותו אחר כך למחשב, אבל איכות הקול מאוד גרועה וצריך לשפר אותה. ניסינו להמשיך ולחקור כיצד לשפר את איכות הקול ומצאנו דרך מעט שונה לקלוט קול על ידי המיקרופון, ועדיין לא ניסינו אותה. | המשכנו את המחקר כיצד לעשות זיהוי של wake word, והתקדמנו עם הקוד שנדרש על מנת לטפל בקבצי השמע ולערוך אותם בשביל הרשת הלומדת. כמו כן התקדמנו עם המחקר על איפה כדאי לעשות את הזיהוי ומצאנו מספר כלים שעושים אטומיזציה של תהליך האימון של המילה ושליחת intent מתאים חזרה לבקר עם הפקודה המתאימה. |
| 5 | עדיין לא ניסינו את הדרך הנוספת שמצאנו, אבל המשכנו לקרוא עליה ולהתחיל לתכנן אותה. מסתבר שהיא די מסובכת ודורשת כמות מאוד גדולה של קוד. נמשיך לעבוד עליה ונתחיל ליישם אותה. | סיימנו את כתיבת הקוד שנועד לערוך את הקבצים של הרשת הלומדת. הרצנו את הרשת ועשינו לה בדיקות על המחשב והיא נתנה אחוזי דיוק מאוד טובים! התחלנו בכתיבת הקוד בשביל לשים את הרשת של ה-wake word על הבקר עצמו וכתיבת הקוד שנועד לקחת קובץ wav ולהריץ אותו על הרשת בשביל לבדוק אותה על הבקר. |
| תחילת שבוע 6 | ניסינו את השיטה השנייה של קליטת דגימת קול והיא עבדה!!! כלומר, יש לנו כרגע מיקרופון שעובד ומצליח להקליט קולות. | הצלחנו לשים את הרשת על הבקר ולהריץ אותו, הרשת עבדה והביאה תוצאות די יפות. אבל, עדיין יש מעט בעיות של זיהוי עם הרשת השנייה (לא של ה-wake word). |
| סוף שבוע 6 | אנחנו באמצע החיפוש של מבנה של רובוט שיוכל להתאים לצרכים שלנו, לאחר שנמצא מבנים מתאימים נקבע פגישה עם תום בשביל לדון איתו במה שמצאנו ואיך כדאי ליישם את מה שנחליט. | אז לאחר מלא בעיות (ולילה לבן) הצלחנו לפתור את הבעיות באינטגרציה של שתי הרשתות ניורונים על הבקר בו זמנית, הצלחנו לשים את שתיהן בו זמנית על הבקר וליצור מכונת מצבים שתתזמן ביניהם (מסתבר שאף אחד לא ניסה לעשות את זה לפנינו...). כאשר ניסינו לשים את הרשת הגדולה בנוסף לרשת של המילת התחלה, הבקר קרס מחוסר זיכרון. בהמלצת תום, אנו קובעים פגישה עם גיל טל ממובילאיי בשביל לנסות לפתור את הבעיה הזו. |
| 7 | חקרנו עוד יותר את סגנון הרובוט שאנחנו רוצים שיהיה בשביל הקובייה ההונגרית וחיפשנו קבצים מוכנים של הדפסה בתלת מימד. כמו כן, הסתכלנו גם על איזה חיישנים ומנועים נצטרך בשביל הרובוט ואיך כדאי להרכיב אותו. על מנת לעשות זאת בצורה טובה, התייעצנו עם תום מספר פעמים וישבנו איתו על בחירת הרובוט והחלקים הנכונים. | היו לנו כמה בעיות עם הפיצול של הרשתות ניורונים – בעיקר בתחום של ה-ring buffer, כלומר הבעיה הייתה שדגמנו את הקול פעמיים, והוצאנו מהמערך שמחזיק את דגימות הקול 2 דגימות, למרות שרצינו לעשות זאת רק על דגימה אחת. הוספנו את השינויים הדרושים בקוד. לאחר מכן מצאנו באג נוסף באחת הרשתות ניורונים, ולאחר דיבוג מצאנו את הבאג ותיקנו אותו. |
| 8 | קיבלנו מתום את מרבית החומרה הדרושה עבורנו בשביל לבנות את הרובוט והתחלנו בחיבור המנועים לבקר Arduino mega. הצלחנו לסובב 2 מנועים לסירוגין. | התחלנו לעבוד על האפליקציה ולחפש API שיתאים בשביל להציג בתלת מימד את הקובייה. בנוסף, תכננו מה אנחנו רוצים שיהיה באפליקציה בצורה כללית. התחלנו לכתוב את הקוד שיזין את המנועים ויעשה את האינטגרציה בין הפקודות הקוליות לבין המנועים. |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>התקדמנו בכתיבת האפליקציה. בחרנו את העיצוב של האפליקציה שאנחנו רוצים ואת התכונות שיהיו באפליקציה. בנוסף, התלבטנו בין 3 דרכים להעברת המידע בין האפליקציה לבין הרובוט: Bluetooth, WiFi, ובאמצעות שימוש ב-firebase real time. לבסוף החלטנו על הדרך השלישית מכיוון שהיא הכי נוחה עבורנו מבחינת הממשק לכותב הקוד.</p> | <p>9</p> <p>המשכנו עם חיבור המנועים לרובוט והדפסת החלקים בתלת מימד בשביל הרובוט. ברגע זה מרבית החלקים עדיין בתהליכי הדפסה, אבל יש לנו את התכנון הכללי של איך אנחנו רוצים שהרובוט יראה וברגע שהחלקים יהיו מוכנים נחבר אותם על מנת להרכיב את הרובוט.</p> |
| <p>סיימנו לשכתב את הקוד שמזיז את הקובייה, כלומר את הקוד של הרובוט עצמו – מה שמתבצע לאחר זיהוי הפקודה הקולית. הקוד נראה יחסית טוב, נצטרך רק לשכתב עוד קצת את הקוד שקולט את המידע מכיוון שהשתמשנו שם ב-UART וצריך להתאים בדיוק את פורמט השליחה והקבלה על מנת שזה יתבוד.</p> <p>כמו כן, סיימנו לכתוב את הקוד של האפליקציה. עדיין לא ניסינו אותו, אבל נראה שאנחנו מצליחים לתקשר בין האפליקציה לבין ה-real time firebase.</p> | <p>10</p> <p>סיימנו להדפיס את כל החלקים הדרושים עבור הרובוט. התחלנו להרכיב את החלקים אחד לשני. מסתבר שהחוסר בהוראות הרכבה הוא אכן קושי שיש להתמודד איתו ולהבין כמו החלקים יש להדפיס מכל דבר ואיך לחבר אותם בצורה הנכונה. עשינו מדידות של הגדלים שאנחנו צריכים בשביל כל אחד מהצירים, קנינו מסוריות והתחלנו לנסר את מוטות הברזל. כמו כן, קדחנו לתוך קרש עץ שהבאנו את החלקים המודפסים שאמורים להחזיק את המנועים.</p> <p>צריך להמשיך להתקדם בהרכבת הרובוט.</p> |
| <p>שכתבנו את הקוד של ה-ESP כך שאחרי כל פעולה שהוא יעשה הוא ישלח התזוזה שהוא קיבל אל האפליקציה ואל השרת ושם נשמור את התזוזה של הקובייה ונעדכן את הייצוג באפליקציה בהתאם. על פי התכנון המקורי שלנו רצינו שלאפליקציה תהיה אפשרות לשלוח גם מידע אל ה-ESP. התקשורת הדו כיוונית הזו די מסובכת מבחינת האפליקציה מכיוון שעכשיו אנחנו לא מחכים רק לפקודה קולית שיכולה לבוא רק בזמנים מוגדרים, אלא כעת אנחנו יכולים לקבל פקודה מהאפליקציה בכל רגע נתון, ואנחנו צריכים לא להתייחס אליה אם היא מגיע בחלק מהזמנים. כתבנו את הקוד שיעשה את זה אבל נתקלו בבעיה – ה-WIFI כל הזמן מתנתק ומתחבר מחדש ואנחנו לא מצליחים לקלוט הודעות דרך ה-UART.</p> | <p>11</p> <p>סיימנו להרכיב את האב-טיפוס של הרובוט. בעצם חיברנו ללוח העץ את כל החלקים על ידי הברגה, ומדדנו את המרחקים וחתכנו חוט שנחבר למנועים בציר X ו-Y כדי שהמנועים יוכלו לזוז אחורה וקדימה. ניסינו להכניס את הקובייה ולראות מה אנחנו מסוגלים להזיז, וזיהינו שיש לנו בעיות עם הזזת חלק מהמנועים – כלומר המנוע השישי (החיצוני) לא עובד לנו, ובנוסף יש לנו בעיה שהגריפרים שתופסים את הקובייה לא מספיק מדויקים והקובייה לא מחזיקה תוך בתוכם. כמו כן הבורג קרוב מדי לציר הסיבוב של הקובייה ומפריעה לה בסיבוב – צריך להדפיס גריפרים משודרגים חדשים.</p> |
| <p>הבנו מה הייתה הבעיה של החיבור ל-UART ול-WIFI. מסתבר שכאשר מחברים את ה-ESP ל-WIFI הוא מקבל הרבה מאוד חריגות על קבלת פסגות חדשות דרך ה-WIFI וה-UART לא יכול לעבוד כאשר מקבלים כל כך הרבה חריגות כל הזמן. הסתכלנו באינטרנט ולא ראינו אף אחד שהצליח לפתור את הבעיה של שילוב בין UART לבין WIFI בו זמנית. ניסינו לחשוב על פתרון לבעיה, ומצאנו פתרון ממש מגניב – ה-WIFI תמיד יהיה פעיל וכך נוכל לקבל מידע מהאפליקציה בכל רגע שנרצה, אבל לפני שנרצה לשלוח מידע למנועים שיוזזו, נכבה את ה-WIFI ונשלח להם את ההודעה. כלומר הפתרון הזה דורש מאיתנו להוסיף עוד בקר נוסף – ESP mini שרק דרכו אנחנו נתקשר עם האפליקציה. כלומר אנחנו נקלוט את הפקודה הקולית דרך ה-ESP הרגיל ואז נעלה מתח באחד הרגילים של ה-ESP mini על מנת שהוא יידע שאנחנו רוצים לשלוח לו מידע עכשיו דרך ה-UART. זה בעצם דרך לאותת ל-ESP mini שאנחנו רוצים עכשיו לשלוח לו הודעה דרך ה-UART מבלי באמת להפעיל את ה-UART כי לא ניתן לשלוח דרכו הודעה כל עוד ה-WIFI פועל. ה-ESP mini ייכבה את ה-WIFI שלו ואז הוא יוכל לקבל הודעה דרך ה-UART. במקביל נשלח את ההודעה גם ל-arduino mega שהוא ישלח את הפקודה המתאימה למנועים. לאחר שנקבל את הפקודה ב-ESP mini נדליק חזרה את ה-WIFI ונשלח את ההודעה לשרת.</p> | <p>12</p> <p>ניסינו את הגריפרים החדשים שתום הדפיס לנו והם עבדו בצורה הרבה יותר טובה אבל עדיין יש איתם מעט בעיות – הייתה גרסה של גריפרים שהיו עבים מדי, והייתה גרסה שהם היו קצרים מדי. אנחנו מנסים עכשיו גרסה רביעית שבתקווה תחזיק טוב יותר את הקובייה גם במהלך הסיבוב. חיבור המנוע השישי לא הצליח בצורה טובה וניסינו כמה פתרונות שונים בשביל לפתור את הבעיה הזו ולבסוף הצלחנו. הפתרון שמצאנו היה לחבר בקר נוסף מסוג Arduino uno שהוא ידגום כל הזמן את הערך בשני פינים מסוימים שנקבע לו מתוך ה-arduino mega והוא יעביר את מה שהוא מקבל שם אל המנוע השישי שיהיה אצלו. הסיבה בגללה לא הצלחנו לחבר מנוע שישי בדרך הרגילה לא כל כך מובנת לנו, כנראה שיש שם בעיה עם ההספק החשמלי למנוע השישי, והדרך שבה פעלנו התגברה עליו – חיברנו את המנוע השישי לספק חדש לגמרי (בקר חדש שמחובר לספק חדש) ולכן הוא עבד. קישרנו בין הרכיב המרכזי שמפעיל את חמשת המנועים האחרים לבין הרכיב החדש שמפעיל את המנוע השישי גם מבחינת תוכנה וגם מבחינת חומרה.</p> |
| <p>הטריק שהצענו קודם לא היה פשוט כל כך למימוש מכיוון שהוא דרש מאיתנו לוודא את התזמון של כל שליחת הודעה ולכתוב את הקוד בצורה שהוא יתחשב באילו רגעים יכולה לבוא הודעה מהאפליקציה ומתי לא ולהתייחס לכל מקרה בהתאם. גם התזמון של שליחת הסיגנל היה בעיה לא קטנה מכיוון שהיה מסובך שה-ESP mini יקרא את הסיגנל ויכבה את ה-WIFI ורק אז לשלוח הודעה אליו. לבסוף אחרי הרבה מאוד ניסיונות הצלחנו לסדר את זה בצורה חד כיוונית, כלומר הצלחנו לשלוח הודעה רק מכיוון ה-ESP אל האפליקציה אבל לא להפך. לאחר עוד שדרוג של הקוד</p> | <p>13</p> <p>תום הדפיס עבורנו סט גריפרים חדש ואנחנו הדבקנו בתחתית של כל גריפר בריסטול מחוספס כזה על מנת ליצור יותר חיכוך עם הקובייה ולהקטין את המרחק בין שתי הקצוות של הגריפר. השדרוג הזה עבד בצורה יחסית טובה.</p> <p>על מנת לפתור את הבעיה שהצענו קודם חיברנו את ה-ESP mini לכל שאר החומרה שלנו וחיווטנו חוט מתאים שהוא יהיה הסיגנל שלנו שאנחנו עומדים לשלוח עכשיו הודעה דרך ה-UART.</p> |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>ומציאת כמה באגים חדשים הצלחנו גם לפתור את הבעיה הזו. כך בעצם יש לנו כעת אפשרות תקשורת מהאפליקציה ושליחת פקודות כמו לערבב ולפתור את הקובייה מהאפליקציה אל הקובייה, וכמו כן הצגה של כל אחת מהפקודות שהקובייה עושה על ידי פקודות קוליות גם באפליקציה.</p> | <p>כמו כן, לאחר שסידרנו את הגריפרים החדשים, היינו צריכים להזיז את המיקום של המנועים קצת לצדדים על מנת שהם יתאימו בדיוק לקובייה ולגריפרים החדשים, אז פתחנו את כל ההברגות והזזנו והברגנו מחדש תוך כדי שמתחנו את הרצועות שמזיזות את הגריפרים בציר X ו-Y.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

הערות נוספות

סוף שבוע 2: היו לנו בעיות עם הבקר שהוא לא התחבר למחשב והיה צריך להתקין דרייבר חיצוני שיצר בעיות נוספות. לבסוף לקחנו בקר חלופי, אבל גם הוא יצר בעיות, אבל אותן הצלחנו לתקן פחות או יותר.

סוף שבוע 3: הצלחנו לחבר את המיקרופון לבקר והצלחנו לראות גרף שמראה שהמיקרופון קלט צלילים מסוימים מהסביבה. מהשוואות שערכנו באינטרנט נראה שהוא קולט את הצלילים בצורה חלשה. לאחר מכן, הצלחנו לייצא את ההקלטה שעשינו לקובץ wav אבל בהקלטה שהצלחנו לייצא לא שמענו כלום. אנחנו חושבים שהבעיה היא או במיקרופון או בחיבורים הפיזיים שחיברנו, מכיוון שקטע הקוד שלקחנו עבד והצליח לייצר את הפלט הרצוי עבור פרויקטים אחרים.

סוף שבוע 5: במהלך הבדיקות של הרשת על המחשב מצאנו כמה באגים קטנים של הרשת במקרים מסוימים מאוד, ואנו חושבים שיש בעיה של overfitting עם המילה go שהיא ה-wake word שלנו. לכן, אנו מתכננים להריץ לאמן שוב פעם את הרשת עם מספר דוגמאות מתאים יותר. כמו כן, יצרנו כמה אלפי קבצי קול של מילים שדומות למילה go ועלולים ליצור בעיה ברשת, על מנת לאמן את הרשת לא להתבלבל עם המילים הללו.

בנוסף, מכיוון שעדיין יש לנו בעיות עם הקלטת קבצי הקול דרך המיקרופון, אנו מנסים להכניס ידנית קטע קול שהקלטנו על המחשב ולא דרך הבקר לתוך ה-ESP על מנת שנוכל לבדוק את הביצועים של הרשת כאשר נצליח לשים אותה על הבקר.

תחילת שבוע 6: הצלחנו להריץ את הרשת הגדולה יותר (לא של ה-wake word) והיא עבדה טוב אבל עדיין צריך לשפר אותה כי יש לה מעט בעיות. צריך לחקור האם הבעיות הן בקליטת הקול (כלומר במיקרופון) או ברשת עצמה. נבדוק את זה באמצעות הרצה של הרשת השנייה של ה-wake word שהביאה תוצאות מאוד טובות כאשר השתמשנו במיקרופון של המחשב.

אנו כרגע בודקים כיצד עדיף לעשות את האינטגרציה בין הרשת של ה-wake word לבין הרשת של זיהוי הפקודות. בנוסף, אנו מתחילים לבדוק עם אילו מנועים כדאי לנו לעבוד ואיך עושים את זה.

| טבלת יעדים | | |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| סוף שבוע | יעדים בתחום החומרה | יעדים בתחום התוכנה |
| 2 | חיבור המיקרופון לבקר ובדיקה שהוא אכן מחובר. שיפור של הקלט הקולי שאנו קולטים מכיוון שנראה שהוא בתדרים לא נכונים | להתחיל לחקור כיצד ליישם את הרשת נוירונים ואיך לשים אותה על הבקר. בעיקר להבין אילו שיטות של AI הכי יעילות לזיהוי קבצי קול. |
| 3 | אנו רוצים להגיע למצב שאנו שומעים את מה שהקלטנו ולא רק רואים גרף שמייצג תדרים. | לכתוב את הקוד שמנרמל את קבצי הקול ויוצר מכל קובץ קול spectrogram שמתאים לאותו קובץ קול. בנוסף, לראות אילו כלים קיימים כדי ליצור את הרשת ואילו כלים קיימים עבור הבקר על מנת לחזות מילה שאנו מקליטים. |
| 4 | מכיוון שהקובץ wav שהצלחנו לייצא נשמע גרוע מאוד, אז צריך להבין איך לשפר את איכות קליטת הקול. | להחליט איפה עושים את זיהוי הקול של הרשת הגדולה. בנוסף, להמשיך לכתוב את הקבצים שעושים את יצירת הרשת ועיבוד קבצי הקול. |
| 5 | המשך של הנקודה מהשבוע הקודם. | המשך בדיקה של הרשת של ה-wake word. ייתכן שנרצה לנסות רשת חדשה שלא יהיה בה overfitting והיא תעשה למידה גם על הרבה דוגמאות נגדיות מיוחדות. לכן, יש לעדכן את הקבצים וליצור את קבצי הקול המיוחדים. להתחיל לעבוד על כתיבת הקוד שיריץ את הרשת על הבקר ויקח דוגמאות קול ויבדוק אותן על הבקר. |
| 6 | כרגע יש לבחון אילו מנועים כדאי לבחור על מנת שנצליח לסובב את הקובייה ההונגרית. בנוסף, לאחר שנבחר להבין איך משתמשים במנועים אלו. | יצירה של הרשת החדשה של ה-wake word ובדיקה שלה. כמו כן, יצירה של רשת חדשה עבור זיהוי הפקודות מכיוון שהיא מביאה תוצאות טובות אבל לא מספיק טובות עבורנו. בנוסף, להבין איך לעשות את האינטגרציה בין שתי סוגי הרשתות ואיך לשים את שתיהן על הבקר ביחד ואיך לעבוד איתן. |
| 6 (בהמשך לפגישה עם המתרגל) | לאחר שדיברנו עם המתרגל אז דיברנו גם עם תום, ובעקבות השיחה איתו, הבנו שצריך לבחור איזה מודל של רובוט של קובייה הונגרית אנחנו רוצים, ולראות האם ניתן ליצור רובוט שכזה. במידה ולא נמצא נצטרך להחליף נושא לפרויקט, כי הרובוט מעט מורכב וצריך למצוא בסיס נוח שקיים כבר כדי שיהיה לנו על מה לעבוד. | בהמשך לשיחה עם המתרגל וגם עם תום, אז הבנו שלא כדאי לשלוח את הפקודה (לאחר זיהוי מילת ההתחלה על הבקר עצמו) לשרת ושם לבצע את הפרדיקציה ויצירת ה-intent, אלא כדאי לעשות הכל על הבקר. ניסינו לכתוב סקיצה ראשונית של הקוד ומסתבר שיש מלא בעיות בזה: א. בעיות זיכרון של הבקר – לא ניתן לשים 2 רשתות בגודל בינוני – גדול על הבקר, כי הוא קורס מחוסר בזיכרון. ב. יש בעיה שהבקר מזהה רק רשת אחת מתוך השתיים ולא מתייחס כלל לרשת השנייה, זה ממש ממש מוזר כי הקוד נראה בסדר ולא ברור למה בכלל זה קורה. אנו חושבים שייתכן ובבנאי של tensorflow משתמשים במשתנים סטטיים (משיקולי יעילות מאוד הגיוניים), ולכן אנו בעצם דורסים את אחת הרשתות |

| | | |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | ששמנו. נוודא את ההשערה הזו וננסה לפתור בהתאם. |
| 8 | לקחת את החומרה מתום ולהתחיל להתעסק עם המנועים ובעיקר עם חיבור של מספר מנועים. לדבר עם תום על המשך ההדפסה של החלקים ואיך ליצור את הרובוט עצמו – כלומר אילו חלקים צריך ואיך לחבר אותם. | לחקור על ספריות מתאימות בשביל ליצור את האפליקציה – במיוחד לחפש ספרייה מתאימה בשביל להציג את הקובייה הונגרית. להבין איך מזיזים את המנועים בצורה טובה ואיך ליצור את האינטגרציה בין המנועים לבין הבקר ולבין האפליקציה. לכתוב את הקוד שיעשה את השילוב בין הפקודות הקוליות לבין המנועים ויתזמן את ההזזות שלהם. |
| 9 | לסיים את חיבור כל החומרה. להבין איך מחברים בין ה-ESP לבין הארדואינו וכיצד שולחים פקודות ומידע מאחד לשני. | לסיים את ה-GUI של האפליקציה, ולהחליט דרך איזה שירות נתקשר עם הבקר ואיך פרקטית עושים את זה. לסיים לכתוב את הקוד שעושה את האינטגרציה של החומרה והתוכנה ולהתחיל לבדוק ולדבג אותו. |
| 10 | הדפסת גריפים חדשים שיהיו מתאימים לממדי הקובייה. ניסיון להצליח לחבר בהצלחה את כל ששת המנועים – כעת עובד רק החיבור של 5 מנועים אל הבקר, כאשר המנוע השישי אינו מצליח לקבל מתח. | אינטגרציה בין הקוד שעורך זיהוי קולי על ה-ESP לבין הקוד שאחראי על הזזת הרובוט על Arduino. הפעלת תקשורת UART בין שני הבקרים, והבנת דרך התקשורת. החלטה על דרך ייצוג מצב הקובייה בכל רגע נתון בשביל האפליקציה. |
| 11 | בירור דרכים להתמודד עם חוסר זיכרון בקומפילציה ובמהלך הרצת התוכנית על גבי בקר ESP, בעת שילוב בין הקצאות רשתות נזירונים של TensorFlow ושליחת מידע בעזרת חבילת firebase. | שליחת מידע מ-ESP לfirebase ובחזרה על מנת ליצור תקשורת עם האפליקציה. כתיבת אלגוריתם פתירה של קובייה הונגרית ממצב נתון. התמודדות עם חוסר זיכרון על גבי בקר ESP32 בעת שליחת מידע לאפליקציה בעזרת firebase והפעלת זיהוי קולי של רשתות נזירונים בזמן. |
| 12 | התאמת ממדי הגריפים ועוצמת התפיסה על מנת לאפשר לעשות פעולות מסובכות כמו UP ו-DOWN היכולות להינתן כפקודה קולית על ידי המשתמש. | מציאת דרך לאפשר תקשורת WIFI ובו זמנית לאפשר שליחת מידע בין בקרים בעזרת UART. מציאת דרך לשלוח סיגנל לאפליקציה שרצה בלולאה על בקר ES32 לזיהוי קולי על מנת לטפל בפקודה שהגיעה כעת מאפליקציה. |
| 13 | עיצוב כל החומרה על מנת להציג את הפרויקט ביריד. הדבקת חוטים ביחד, הלחמת חוטים, סידור כל החומרה יפה על גבי משטח הרובוט וכדומה. | לא עשינו שיפורים מיוחדים בשבוע זה. |
| שיפורים עתידיים... | שיפור התפיסה של הגריפים את הקובייה על מנת להעלות את ההסתברות לפעולה מוצלחת של UP ו-DOWN של הקובייה (צריך לשפר אחיזה חזקה בקובייה וכיוצא בזה הפעולות הנ"ל, אשר מסתמכות על כך, יצליחו במשימתן). | שיפור רשתות נזירונים מבחינת ה-overfitting שהן מייצרות על מילת הפקודה left ביחס לשאר המילים. שיפור זמני הניתוק והחיבור מ-WIFI בכל שליחת הודעה על מנת לאפשר ערוץ תקשורת דו כיווני מהיר יותר בין אפליקציה לבין בקר. |

סיכומי פגישות

סיכום פגישה עם המתרגל – 2/5/22

הנקודה העיקרית שעלתה בפגישה היא הדיון היכן כדאי לבצע את הפרדיקציה ויצירת ה-intent בשביל ההזזה של המנועים. אנחנו חשבנו שיהיה הכי נוח עבורנו לעשות את הזיהוי של מילת ההתחלה על הבקר ואז לשלוח intent לשירות חיצוני שהוא יעשה את זיהוי הפקודה. אבל לאחר התייעצות עם המתרגל (ובהמשך גם עם תום) נאמר לנו שכדאי לעשות את כל הזיהוי רק על הבקר (על מנת לשמור על הייחודיות של הפרויקט).

הערה: שאלות לשאול בפגישה עם המתרגל:

1. התייעצות לגבי הרשת – האם עדיף ניתוח על שרת חיצוני כמו WitAi או ניתוח על הבקר.
2. במידה ומחליטים ללכת על התקנת רשת על הבקר, עד איזה גודל ניתן לשים את הרשת? ואם זה חורג מהמקום הפנוי, איך אפשר להשתמש בדיסק חיצוני?
3. במידה ומחליטים ללכת על רשת על הבקר, איך לשפר את אחוזי השגיאה? אולי עדיף להשתמש בשירות אימון חיצוני ולא להשתמש בקוד שקיים לנו היום?
4. מנועים – איזה סוג של מנועים מומלץ להשתמש? מתי נוכל לקבל אותם?
5. איך נחבר את הקובייה ההונגרית למנועים – האם נדביק אותה? איך נמרכז אותה באמצע.
6. מדפסת תלת מימד – גישה למדפסת על מנת להדפיס חלקים נדרשים לחיבור בין המנוע לבין הקובייה ההונגרית.
7. התייעצות לגבי בניית אפליקציה – מה הוא חושב על זה? פרויקטים קודמים שבנו גם כן אפליקציה. באיזה שירות תכנות עדיף לעבוד על אפליקציה.

סיכום פגישה עם תום – 3/5/22 (הסיכום נכתב על ידי תום)

- פתרונות אפשריים לבעייה של חוסר יכולת להריץ 2 רשתות על אותו לוח בו זמנית -
 1. לייצר HACK שבעלייה מעלה רק רשת אחת, ואחרי זיהוי מוצלח - מעדכן פרמטר מסויים ב-FLASH ועושה RESET ללוח כך שבעלייה הבאה יעלה רשת השנייה - תתייעצו עם גיל טל מנטור של הקורס - gil.tal@intel.com, gilly.tal@gmail.com
 2. לוותר על CLASS אחד ברשת הגדולה יותר, ולאמן מחדש כאשר WAKE WORD הוא CLASS ברשת הגדולה.
 3. העלאה לשירות חיצוני בענן - לא ממליץ, בואו נשמור על הפרויקט הזה OFFLINE גם משיקולי LAG וגם ייחודיות
- מבחינת האפליקציה שנבחר ליישם סביב הזיהוי מילים - תחשבו על כמה רעיונות פשוטים יחסית (לדעתי קובייה הונגרית יש להתבסס על עיצוב זמין להורדה thingiverse.com instructables.com או בגוגל (diy rubik cube solver) דוגמא - <https://fourboards.co.uk/rubix-cube-solving-robot#cad>

נקבע פגישה נוספת בהקדם!

הערה: קבענו עוד פגישות נוספות עם תום, אך לא כתבנו עליהן סיכום פגישה. בין היתר תוכן הפגישות כלל עזרה בבניית הרובוט – הדפסת חלקים, חיבור החלקים, טיפול בחלקים שאינם טובים; עזרה בחומרה – התייעצות לגבי חיבור 6 מנועים אל בקר ארדואינו מגה אחד, חיבור UART בין בקרים, חיבור ביפר לשם נוחות המשתמש; ועזרה בתוכנה.

סיכום פגישות עם גיל טל (מובילאיי)

לאורך הסמסטר קבענו כמה פגישות עם גיל טל לצורך התייעצות והמשך התקדמות בפרויקט:

פגישה ראשונה – 10.5.2022 – שילוב 2 רשתות נוירונים על הבקר

הפרויקט שלנו מכיל בין היתר שתי רשתות נוירונים שונות, על מנת להפעיל את הזיהוי הקולי: הרשת הראשונה מיועדת לזהות רק את המילה go שהיא ה-Wake Word שלנו. רשת זו, מכיוון שמזהה רק מילה אחת, לא תופסת הרבה נפח זיכרון. רשת שנייה, מיועדת לזהות אחת מששת המילים הבאות: Left, Right, Backward, Forward, Up, Down. מכיוון שרשת זו מכילה מילים רבות הנפח שהיא תופסת בזיכרון יחסית גדול.

הצלחנו להריץ כל אחת מרשתות אילו בנפרד על בקר ה-ESP32 אבל בעת ניסיון להריץ את שתי הרשתות ביחד נתקלנו בבעיה – ESP32 אינו מכיל מספיק מקום על מנת להריץ את שתי הרשתות ביחד בו זמנית (כל רשת נוירונים מיוצגת על ידי מערך tensorflow גדול והעלאת שני מערכים אילו בו זמנית על הבקר לא הייתה אופציה אפשרית). לכן קבענו פגישה עם גיל, במהלכה הצלחנו לסדר את בעיה זו על ידי העברת המערכים להיות סטטיים בזיכרון של הבקר.

פגישה שנייה – 31.5.2022 – הגדלת מקום על בקר לשילוב רשתות נוירונים Firebase

בשלב זה הגענו למצב בו הזיהוי הקולי על הבקר עובד בצורה טובה, וכן הרובוט לפתירת הקובייה גם כן עובד בצורה סבירה. אנחנו בנינו אפליקציה לשם ויזואליזציה של מצב הקובייה עבור משתמשים, ולשם יצירת ערוץ תקשורת נוסף בין משתמש לבין הרובוט (נוסף על ערוץ התקשורת של הזיהוי הקולי, שכבר קיים).

על מנת לעשות את החיבור בין האפליקציה לבין הרובוט, נדרשנו למצוא דרך לתקשר בין ה-ESP והפעולות שהוא שולח לרובוט לעשות (הנשלטות על ידי פקודות קוליות מהמשתמש) לבין האפליקציה – עדכון מצב הקובייה באפליקציה לאחר כל פעולה שעשינו דרך הזיהוי הקולי. זאת בחרנו לעשות באמצעות firebase.

מסתבר כי firebase היא חבילה גדולה ותופסת הרבה מנפח הזיכרון של תכנית אותה צורבים על בקר ה-ESP. מכיוון שכבר היו קיימים רשתות נוירונים על הבקר שלנו (לשם הזיהוי הקולי), הוספת חבילת firebase לקמפול הבקר גרר שגיאה – לא היה לנו מספיק זיכרון על הבקר.

קבענו פגישה עם גיל במהלכה פתרנו בעיה זו, זאת על ידי הגדלת נפח הזיכרון שבקר ה-ESP מקצה לתוכנית אשר צורבים, וכתוצאה מכך הקטנת נפח הזיכרון של מערכת ההפעלה על הבקר.

פגישה שלישית - 18.06.2022 – אין מספיק מקום על הבקר

לאחר כל פעולת זיהוי קולי של הבקר ושליחתו לאפליקציה באמצעות firebase בקר ה-ESP קרס – זאת כתוצאה ממחסור בזיכרון במהלך ההרצה (לא לפני ההרצה, שזו הייתה הבעיה הקודמת עליה דיברנו בפגישה ב-31.05). התייעצנו עם גיל איך כדאי ניתן לפתור את הבעיה. ראשית הוא הציע לנו להשתמש בבקר מסוג אחר – לדוגמה ESP Cam, בו יש יותר DRAM לכן הוא כנראה יוכל להקצות את כל הזיכרון הדרוש לנו להרצת התוכנית באופן שוטף. מכיוון שזה כבר היה שלב מאוחר בסמסטר, ורוב הקוד היה כתוב כבר עבור ESP – לא רצינו לשנות הרבה קוד נוכחי, והעדפנו למצוא פתרון אחר.

בסופו של דבר לאחר התייעצות עם גיל וגם עם תום, נקטנו דרך פעולה אחרת – פיצלנו את שליחת המידע ב-firebase לבקר אחד – ESP mini, ואת הזיהוי הקולי על בקר שני – ESP32. כך כל בקר היה יכול להריץ את כל התוכנית ללא שגיאות זיכרון. זה כמובן יצר לנו בעיות חדשות, בין היתר כיצד לנהל את התקשורת בין שני בקרים אילו, איך URAT יכול לעבוד בשילוב עם WIFI ועוד.

אופטימיזציות

קליטת קול:

כפי שכתב לעיל היו לנו הרבה בעיות עם זיהוי הקול. חיבור המיקרופון לבקר וקינפוג הפרמטרים הנכונים על מנת לקלוט קול בצורה טובה, היו בעיה רצינית שלקחה לנו מספר שבועות להתגבר עליה. לאחר הרבה ניסיונות הצלחנו למצוא את הפרמטרים האופטימליים עבור המיקרופון. את הבדיקה של איכות הקליטה ביצענו באמצעות הדפסה של גרף שמתאר את תדר הקול שהמיקרופון קלט כפונקציה של הזמן. את הגרפים שיצאו לנו השווינו עם גרפים של אותו קטע קול שהקלטנו באמצעות המיקרופון של המחשב ויצרנו את גרף התדרים שלו.

לאחר שהגענו למצב בו הגרף שיצרנו היה מתאים וטוב, ניסינו לעלות לשרת את קטע הקול שהקלטנו דרך המיקרופון של הבקר ולשמוע אותו. התוצאות שקיבלנו היו לא טובות – לא היה ניתן לשמוע דרכן כלום, אלא רק רעש סטטי. לכן, ניסינו לקנפג את הפרמטרים בצורה אחרת, ולא הצלחנו להגיע לתוצאה טובה יותר.

בסופו של דבר, מצאנו דרך נוספת לקלוט קול דרך המיקרופון של הבקר – על ידי שימוש בספריות אחרות, והשתמשנו בה. לאחר קינפוג נוסף של הפרמטרים בספרייה הזו, הצלחנו בסופו של דבר לקלוט קטע קול באיכות טובה יחסית שיהיה ניתן להביא לרשתות הניורונים בשביל להריץ עליו פרדיקציות.

זיהוי מילים:

על מנת לעשות את זיהוי המילים על הבקר היינו צריכים להחליט מה היא הדרך הטובה ביותר עבורנו לעשות את זה. האפשרות הראשונה שנראתה לנו הכי טובה היא לבצע זיהוי רק של ה-wake word על הבקר ואילו את שאר כל המילים לעשות את הזיהוי שלהן בשירות חיצוני של פייסבוק. הסיבה לכך היא שכוח החישוב והזיכרון של הבקר הוא מאוד קטן ולהריץ עליו רשתות ניורונים כבדות זה משימה כמעט בלתי אפשרית. אבל, לאחר התייעצות עם המתרגל, החלטנו לעשות את כל זיהוי המילים בצורה של **offline**. לכן, היינו צריכים ליצור רשתות ניורונים שיהיו מצד אחד מדויקות וטובות מאוד ומצד שני יהיו קלות ולא יהיו גדולות מדי.

לאחר מחקר קצר בתחום של NLP ראינו שהשיטה הנפוצה ביותר כיום היא לקחת את קטע הקול ולחלק אותו לחתיכות קטנות. על כל אחת לבצע התמרת פורייה, ובכך ליצור בעצם ספקטוגרמה של התדרים המרכיבים את קטע הקול. כך בעצם נקבל תמונה – על התמונה נוכל להשתמש בכלים של זיהוי תמונה (רשתות CNN) ולסווג בצורה מולטינומית את הפרדיקציה שלנו איזה מילה קיבלנו. הבעיה הייתה שהרשתות הללו הן לרוב בעומק של עשרות עד מאות שכבות ושוקלות הרבה, אז חיפשנו רשת שתהיה מספיק טובה בשביל המשימה שלנו אבל גם לא כבידה מדי. מצאנו בפרויקט של משהו שכבר בנה רשת כזו והשתמשנו בה והיא הביאה תוצאות טובות מאוד על זיהוי של מילה אחת.

לאחר מכן עברנו למשימה של זיהוי יותר ממילה אחת על הבקר. רצינו ליצור רשת שמזהה 6 מילים, אבל אחוזי הדיוק שלה היו לא טובים, מכיוון שהיא הייתה צריכה להיות קטנה יחסית, ולכן הדיוק שלה לא היה טוב מספיק. אז פיצלנו את המילים, ויצרנו 2 רשתות שונות שבכל אחת יש רק 3 מילים והרצנו פרדיקציה במקביל על כל רשת, ולקחנו את התוצאה הטובה ביותר מכל רשת והשוונו אותן וכל זיהוינו איזו מילה נבחרה.

אבל פתרון זה יצר בעיה נוספת – אין מספיק זיכרון על הבקר על מנת לשים עליו 3 רשתות ניורונים (2 רשתות לזיהוי המילים ועוד רשת לזיהוי ה-wake word). לכן היינו צריכים להרחיב את הזיכרון של הבקר. עשינו זאת באמצעות התייעצות עם גיל שהסביר לנו כיצד יש לעשות את זה. בעיה נוספת שנתקלנו בה הייתה הצורך בשכפול המידע במערך הציקלי שמייצג את דגימת הקול שקיבלנו. מכיוון שאנחנו לא יודעים באיזה רגע נקבל את המילה, הגדרנו מערך ציקלי שבו בכל רגע אנחנו מנסים לעשות זיהוי באמצעות הרשתות של המידע שנמצא כעת בתוך המערך, וכל פעם דוחפים את המידע הישן ביותר ומכניסים מידע חדש. היינו צריכים לשכפל את המידע הזה פעם נוספת מבלי לפגוע במידע שנכנס באותם רגעים אל המערך וכמו כן במידע שנמצא כבר במערך.

חיבור 6 מנועים מסוג Stepper למען הפעלת הרובוט :

לאחר לבטים רבים החלטנו ללכת על תכנון רובוט – Qbot – The Open Source Rubik's Cube Solver, כאשר רובוט זה בין היתר דורש הפעלה של 6 מנועים מסוג Stepper Motor. על מנת לעשות את הפעלת 6 המנועים יעילה החלטנו (לאחר התייעצות עם תום) להשתמש ב-RTOS – רכיב חשמלי אותו ניתן להתקין על גבי Arduino Mega ודרכו לנהל עד כ-6 מנועים. הצלחנו לחבר ולהפעיל 5 מנועים, אך בעקבות בעיית מתח לא הצלחנו להפעיל את המנוע השישי.

הפתרון בו נקטנו הוא חיבור בקר נוסף (אליו גם מחובר מקור מתח נוסף), בקר מסוג Arduino Uno, אליו נחבר באופן חיצוני את המנוע השישי. כך נפתרנו מבעיית המתח, אך נתקלנו בבעיה חדשה – הקוד הצרוב על Arduino Mega אחראי על הזזת המנועים בהתאם לפקודה שאותה שלחנו אליו (הכוונה בפקודה היא לדוגמה דרישה של המשתמש להזיז את הפאה השמאלית של הקובייה, ובהתאמה נשלחות רצף פקודות הזזה לששת המנועים). הוצאת המנוע השישי וחיבור לבקר חיצוני מוציאה אותו מן המערכת של המנועים, ומונעת ממנו אפשרות לשליחת פקודות אליו. פתרנו את בעיה זו באמצעות חיבור חוטים להעברת מתח חשמלי בין הפינים שאחראים לשלוח את הפקודות למנוע השישי בבקר ה-Arduino Mega לבין הפינים השולטים על הזזת המנוע השישי כעת ב-Arduino Uno, וכך למעשה עקפנו את הבעיה.

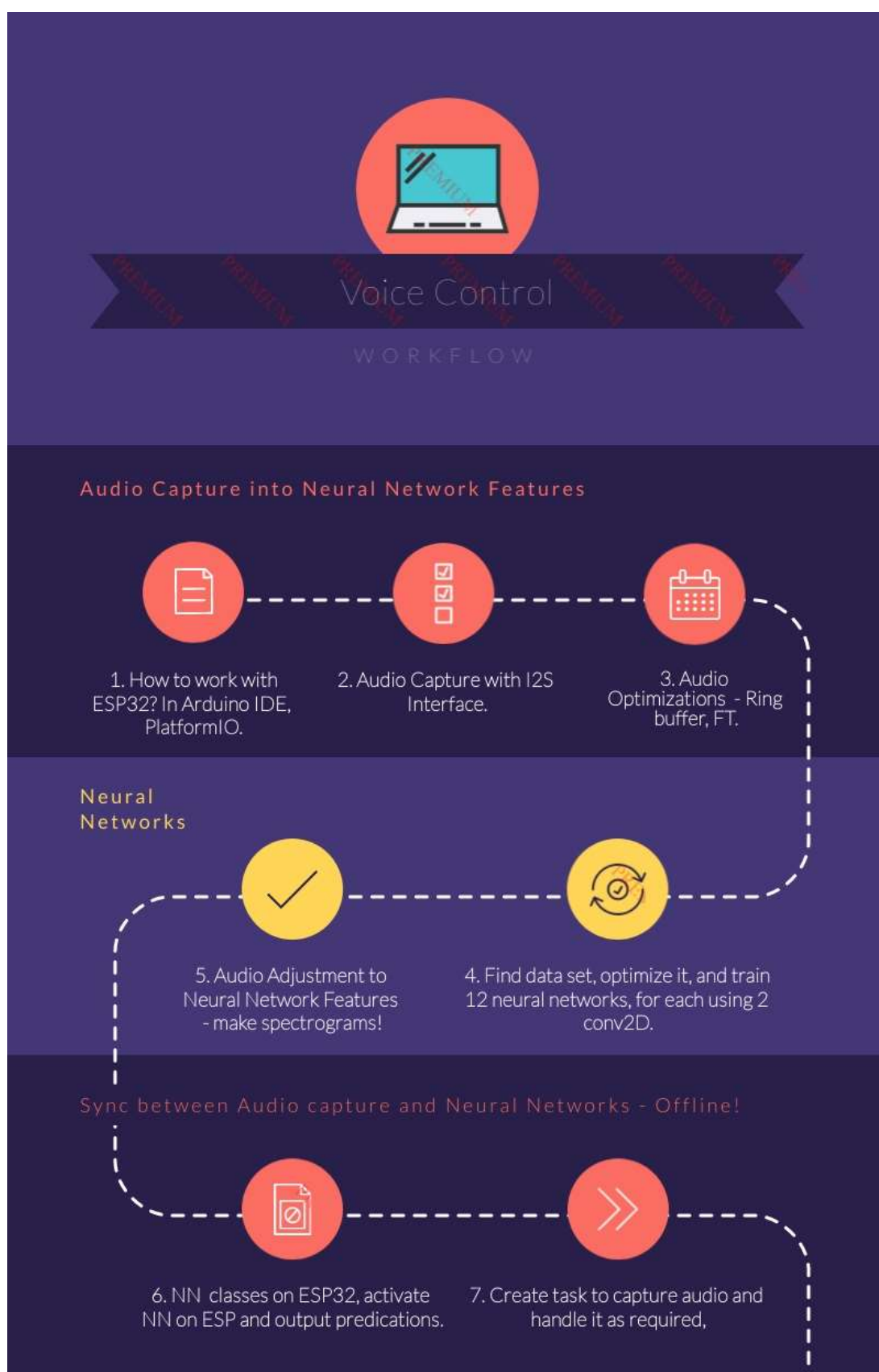
חיבור בין רובוט, זיהוי מילים ואפליקציה :

כפי שהזכרנו קודם, אחת הבעיות שהיו לנו עם החיבור בין ה-ESP שקולט את הפקודה הקולית לבין השליחה לאפליקציה זה העובדה שלא ניתן לשלוח הודעות דרך UART ודרך WIFI בו זמנית. לכן, פתרנו את הבעיה הזו על ידי שני דברים :

- הוספת ESP mini שדרכו נשלח את הפקודה שקיבלנו אל ה-firebase ודרכו נקבל מידע מהאפליקציה.
- שליחת סיגנל בדרך עקיפה שלא דורשת UART לפני כל ביצוע פקודה על מנת לנתק את ה-WIFI כדי שנוכל לשלוח הודעה דרך ה-UART.

על ידי השינויים הללו הגענו למצב בו כל התקשורת עם האפליקציה תמיד מתבצעת דרך ה-ESP mini. כלומר, ה-WIFI תמיד יהיה מחובר עליו, וכך נוכל לקבל מידע מהאפליקציה בכל רגע נתון (חוץ מזמנים מסוימים בהם הגדרנו בקוד שאי אפשר לקבל מידע מהאפליקציה), ואם נקבל מידע מהאפליקציה אז נשלח נכבה את ה-WIFI ונשלח אותו דרך ה-UART אל ה-ESP שממש נשלח את הפקודה לביצוע. אם נקלוט פקודה קולית על ה-ESP אז נשלח סיגנל דרך פין ייעודי שהקצנו עבור זה אל ה-ESP mini שהוא יקלוט את הסיגנל הוא יכבה את ה-WIFI שלו ויחכה לקבל מידע דרך ה-UART. לאחר שהוא קיבל את הפקודה ושמר אותה, אז הוא ידליק מחדש את ה-WIFI וישלח את הפקודה לשרת בשביל לשמור אותה ולעדכן אותה באפליקציה.

גרף התקדמות



Preliminary work for building the robot



9. Activate 6 stepper motors using Arduino Mega, Arduino Uno and RTOS,



8. Finding a suitable design and starting to print the parts.

Build Robot!



9. Assemble all the parts and build the robot according to QBot design.



10. Sync between robot hardware and robot software code.



11. Make final optimizations - such as improving position of robot and print new grippers.

Make an Application



12. Build application using Android Studio, which visualize the cube.



13. Save the cube state after each voice control command, add Solver algorithm and enable the user to send solve, scramble and hint commands from app to robot.

Sync between Robot and Voice commands



14. Make the connection between ESP32 and Arduino mega, in order to enable communication between voice commands and the robot.



15. Make the interface user-friendly - either by activating a beep after recognizing the wake word or by optimizing the delay times between each command and movement

Sync between the former and the application!



16. Enable two way communication - from app to ESP32 and from ESP32 to app, using firebase as a mediator.



17. Enable WIFI communication with URAT communication with hacks we talked about in documentation :)

✓ TASK DONE!

Made with VISME

מקורות מידע

פרויקטים אחרים:

1. Wifi Intercom - <https://github.com/lidor51/WifiIntercom>.
2. Voice Control Robot - <https://github.com/atomic14/voice-controlled-robot>.
3. ESP32 Audio - https://github.com/atomic14/esp32_audio.
4. TensorFlow Lite ESP32 - <https://github.com/atomic14/tensorflow-lite-esp32>.
5. Platformio Tensorflow Lite - <https://github.com/atomic14/platformio-tensorflow-lite>.
6. DIY Alexa - <https://github.com/atomic14/diy-alexa>.
7. Get I2S Sampling - https://github.com/atomic14/esp32_audio/tree/master/i2s_sampling.
8. Walkie Talkie - <https://github.com/atomic14/esp32-walkie-talkie>.
9. QBot Firmware - https://github.com/Axodarap/QBot_firmware.
10. QBot Software - https://github.com/waldhube16/Qbot_SW.

חיבור מיקרופון:

Library of many audio functions (download the git as zip and upload it in the ide):

<https://github.com/pschatzmann/arduino-audio-tools>

Getting audio input:

1. <https://diyi0t.com/i2s-sound-tutorial-for-esp32/>
2. <https://docs.espressif.com/projects/esp-idf/en/release-v4.2/esp32/api-reference/peripherals/i2s.html>
3. <https://github.com/espressif/esp-idf/tree/680710df2f248fa7b144ed58d53126158be8beab/examples/peripherals/i2s>
4. <https://youtu.be/a936wNgtcRA>
5. https://youtu.be/pPh3_ciEmzs

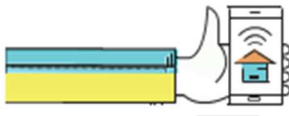
Convert i2s record to .wav file: 7. <https://youtu.be/qmruNKeIN-o>

חיבור RTOS:

<https://github.com/ShawnHymel/introduction-to-rtos>

<https://www.youtube.com/watch?v=F321087yYy4&list=PLEBQazB0HUyQ4hAPU1cJED6t3DU0h34bz>

(From the YouTube videos, if you want to understand (very basic understanding) about things we saw in Alexa and Voice Control Robot, watch videos 1-5. Especially video number 3,5.)



Voice Control

Solve your rubik's cube with voice commands!

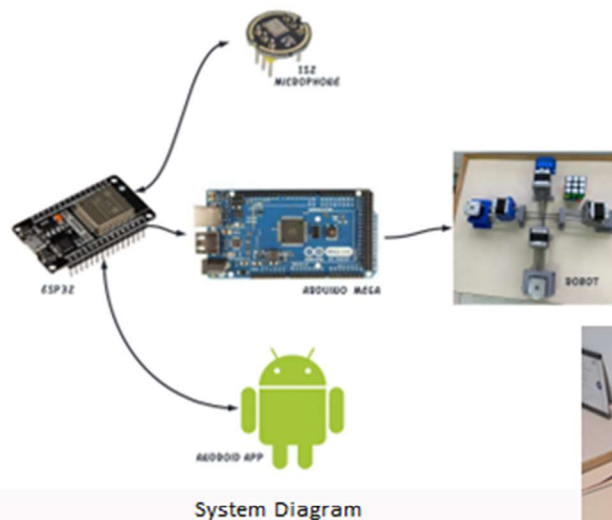
Our project is solving rubik's cube according to voice commands given by the user.

The main features of our project are:

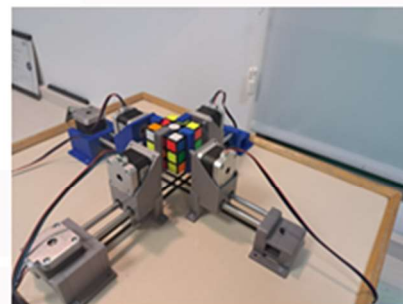
- Voice Commands Recognition (in offline mode).
- Robot that can rotate and solve the rubik's cube.
- Interactive android app with the cube's visualization.

Main advantages of our project:

- The voice recognition is done purely offline.
- The robot can solve the cube efficiently and by his own.



The App View



The Rubik's Cube Robot

**Mor Levy, Yoav Javits, Gilad Shmerler
Itay Dabran, Tom Sofer, Yaron Porat**
In collaboration with Intel

A Project in Internet of Things (IoT)