

הוד השרון

הדרים

מחוז: מרכז

סמל ביה"ס: 441006



מגמת הנדסת מערכות

מכטרוניקה

עבודת גמר 5 יח"ל (805589)

הנושא: מערכת אוטונומית לניקוי במגוון שטחים יבשתיים

מגישים: נועם אלרון, ניר לונדון

מנחים:

מוטי עמר, ניר סליקטר, שמוליק שוורץ

שנה"ל: תש"פ 2021

תוכן

3	הבעת תודה
4	הצעת עבודת הגמר
12	הגדרת הבעיה
12	מבוא
13	מחקר גישוש
16	דרישות
17	תאור מלבני
18	רפלקציה
19	איסוף המידע
19	שאלות חקר לאיסוף מידע
49	פתרונות חלופיים
49	תת מערכת ראשונה- מערכת הנעה ניר לונדון
49	פתרון א: הנעת 4X4
54	פתרון ב' (מזקו"ם)
62	פתרונות חלופיים
62	תת מערכת שנייה-מערכות איסוף נועם אלרון
62	פתרון א' (גריפה מכנית)
75	פתרון ב' (מסנן רוטט)
82	בחירת פתרון מתוך חלופות
82	תיאור תהליך הבחירה (ניר)
89	בחירת פתרון מתוך חלופות
89	תיאור תהליך הבחירה(נועם)
94	מימוש הפתרון
94	תיעוד תהליך הבניה
104	תכן הנדסי - חישובים
106	שרטוט תלת מימדי
111	תוכנה
111	זיהוי אובייקטים
134	מרכז הבקרה-AKA האתר
145	קבלת מידע מהאתר והפעלת המנועים עם המידע על הרספברי פאי:
152	שרטוט חשמלי
153	רפלקציה
154	הערכה מסכמת

155
157

מילון המונחים
ביבליוגרפיה

הבעת תודה

ברצוננו להביע תודה לשמוליק שוורץ, רכז מגמת הנדסת מערכות בתיכון הדרים ומוטי
עמר על הליווי האישי והמקצועי בכל תהליך ביצוע הפרויקט. תודה שעמדתם לצידנו
ועזרתם לנו בכל שעה ובכל עניין. ברצוננו להביע לכם תודה על כל שנותינו במגמה, תודה
שסייעתם לנו להבין שתמיד אפשר ללמוד עוד ושהמגבלה היחידה בחיים שלנו הם
המגבלות אשר הצבנו לעצמנו.

הצעת עבודת הגמר

תיכון הדרים הוד השרון מגמת הנדסת מערכות התמחות מכטרוניקה

נושא הפרויקט: מערכת אוטונומית לניקוי שטחים יבשתיים

שנת הפרויקט: תשפ"ב

שמות המנחים (תואר): שמואל שוורץ (MSc), מוטי עמר (מהנדס אלקטרוניקה)

2. שמות התלמידים (ת"ז)

ניר לונדון (326630183)

נעם אלרון (20986412)

3. חלוקת משימות בצוות:

שם התלמיד האחראי	תיאור המשימה	
ניר לונדון+נעם אלרון	ארגון וסיכום מחקר שוק	1
ניר לונדון+נעם אלרון	ארגון וסיכום פתרונות אפשריים	2
ניר לונדון	ארגון וסיכום עלויות	3
ניר לונדון	שרטוט בתלת מימד	4

נעם אלרון	כתיבת הקוד	5
נעם אלרון	שרטוט סקיצה חשמלית	6
ניר לונדון+נעם אלרון	הרכבת המערכת	7
ניר לונדון+נעם אלרון	כתיבת החוברת	8

4. תיאור הבעיה:

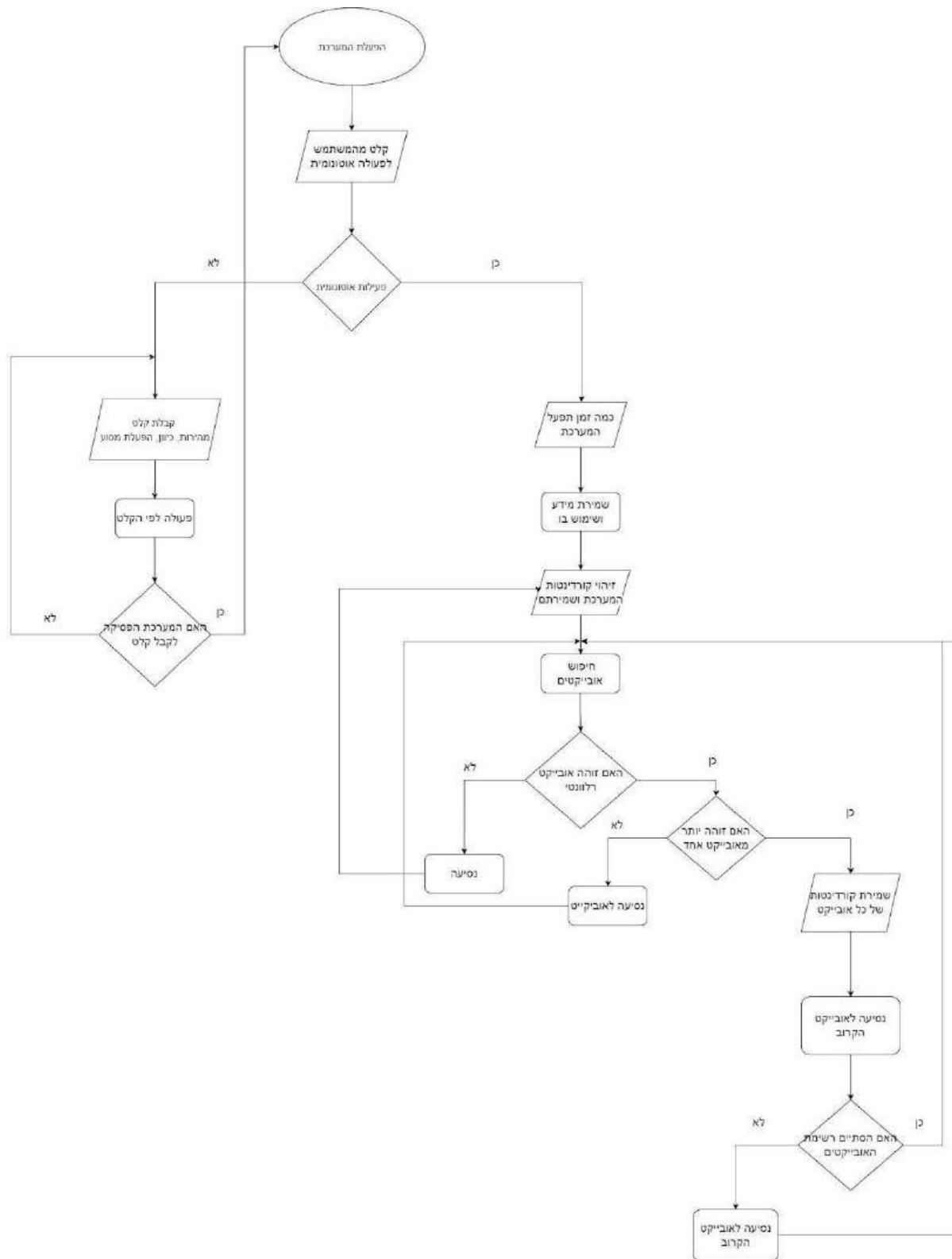
מדי שנה לשנה כמות הפסולת על כדור הארץ גדלה בקצב מחריד, החופים והאוקיינוסים שלנו מתמלאים באשפה. כיום, אלו שהכי מרגישים את ההשפעות של עודף האשפה הם בעלי החיים, אך בני אדם רבים מרגישים את ההשפעות של בעיה זאת על חייהם. ישנה מאמץ גדול של אנשים פרטיים לנסות להקטין/לפתור את המצב, אך בגלל חוסר המודעות בקרב הציבור, וחוסר הלחץ ששמים על ממשלות העולם, לא נעשה מספיק שינוי כדי לעשות הבדל משמעותי. אנחנו רוצים להצטרף למאמצים של קבוצות כמו "Team Seas" ו-"CLEAN UP THE WORLD" שעושים את מיטבם כדי להביע מודעות לבעיה, ולעזור לסביבה לפני שיהיה מאוחר מדי.

הרעיון שלנו הוא לבנות מערכת איסוף אוטונומית, כדי לנקות אזורים יבשתיים, וליצור מודעות לבעיה על ידי שימוש במערכת בציבור כדי ליצור מודעות בקרב הציבור.

5. תיאור עקרוני של המערכת:

הרעיון שלנו הוא לבנות רובוט איסוף אשפה אוטונומי, שעובד על ידי זיהוי אובייקטים עם משין לרנינג כדי לנקות אזורים יבשתיים בצורה יעילה, וליצור מודעות לבעיה על ידי שימוש ברובוט בציבור. כדי להמחיש את חשיבותנו ליצירת מודעות בקרב הציבור נוסיף על הרובוט מתקן שיאפשר לאנשים לסרוק ברקוד או דרך אחרת כדי שיוכלו להשיג עוד מידע על נושא ואופציה לחתום על עצומה לשים לחץ על ממשלות להגדיל לתקציבים ולשים עוד מאמץ על פתירת הבעיה.

6. תרשים זרימה משוער:



7. פירוט הדרישות מהמערכת:

- בדיקת זיהוי אובייקטים ספציפיים כדי לנוע בצורה יעילה.
- הבדיקה תתבצע על בסיס צילום וזיהוי אובייקטים למרחק רחוק כדי ליצור מסלול מתוחכם ויעיל.
- אפשרות עבודה אוטונומית או עבודה על ידי שליטה ידנית.
- יכולת איסוף של משקל לפריט בודד עד 500 גרם.
- יכולת איסוף של משקל כולל עד 10 קילוגרם.
- מערכת הנעה המסוגלת לנוע בשטחים פתוחים לעבור מכשולים של עד 10 ס"מ (אבן מדרכה ממוצעת).
- זיהוי מצב ותקלות במערכת (משקל מקסימלי, רכיב תקול ועוד) ושליחת הודעה מתאימה למפעיל.

8. לוח זמנים לביצוע הפרויקט:

הנושא	תאריך הסיום
סיום הגדרת בעיה	24.11.2021
הגשה של הצעת העבודה	29.11.2021
איסוף מידע ומענה על שאלות החקר	10.12.2021
פתרונות חלופיים כולל סרטוט והסבר פונקציונלי	1.1.2022
בחירת פתרון והזמנת ציוד	5.1.2022
סיום בנייה ותכנות של אב טיפוס מתפקד	10.2.2022
סיום בנייה ותכנות של דגם סופי לפני תיקונים	15.3.2022
סיום בנייה ותכנות של דגם סופי להגשה	25.3.2022
הכנה של פוסטר מדעי	30.3.2022
הצגה בערב הורים	31.3.2022
חוברת שלב א'	6.4.2022

30.4.2022	הגשה סופית (חוברת מודפסת וכרוכה, מצגת או סרטון)
-----------	---

9. משאבים:

משאבי המגמה: מדפסות תלת מימד, מכשירי המגמה, מורי המגמה, רסברי פאי, חיישנים (כגון נפח) ועוד.

רכישות נוספות: מסבים, רצועות טיימינג, מנועים, בקר מותאם למנועים, גופי אלומיניום, ברגים מותאמים, מצלמה, לוח סולרי, רכיבי תקשורת, רשת מתכתית עם "שיניים" מחוברים על ידי קפיצים, חגורת טיימינג מותאמת, צירים.

10. מקום ביצוע העבודה:

מגמת הנדסת מערכות, ובתי התלמידים.

11. רשימת מקורות:

כתב העת הבינלאומי לחקר הנדסה וטכנולוגיה (IRJET), מידע כללי על אופציות לניקוי חופים על ידי מכונות

<https://www.irjet.net/archives/V7/i2/IRJET-V7I2301.pdf>



כדי לחשב מספר רב של רכיבים במערכת ישנם מספר רב של אתרים שאנחנו נעזר בהם

<https://www.brighthubengineering.com/manufacturing-technology/83551-onsite-calculations-for-conveyor-belt-systems>



https://forbo.blob.core.windows.net/forbodocuments/7375/304_fms_transilon-calculation-methods-conveyor-belts_en.pdf



<https://www.cisco-eagle.com/category/2593/incline-conveyors>



https://www.complianceonline.com/images/supportpages/500633/sample_Belt%20Bucklet%20Elevator%20Design.pdf



<http://www.sparksbelting.com/blog/understanding-conveyor-belt-calculations>



[https://simplybearings.co.uk/shop/Info-Pages-How-to-Calculate-V-Belts-Required-Sizes/
c4746_4973/index.html](https://simplybearings.co.uk/shop/Info-Pages-How-to-Calculate-V-Belts-Required-Sizes/c4746_4973/index.html)



<https://www.renoldjeffrey.com/media/2395654/how-to-select-a-chain.pdf>



https://us.misumi-ec.com/pdf/tech/mech/US2010_fa_p3513_3534.pdf



<https://wcproducts.com/pages/calculator-belt>



<https://www.orientalmotor.com/motor-sizing/beltConveyor-sizing.html>



אתר "מק-מסתר" שבו מגוון מידע על חלקים מכנים

[/https://www.mcmaster.com](https://www.mcmaster.com)



אתר "יולו" כבסיס/השראה לזיהוי אובייקטים

[/https://pjreddie.com/darknet/yolo](https://pjreddie.com/darknet/yolo)



הגדרת הבעיה

מבוא

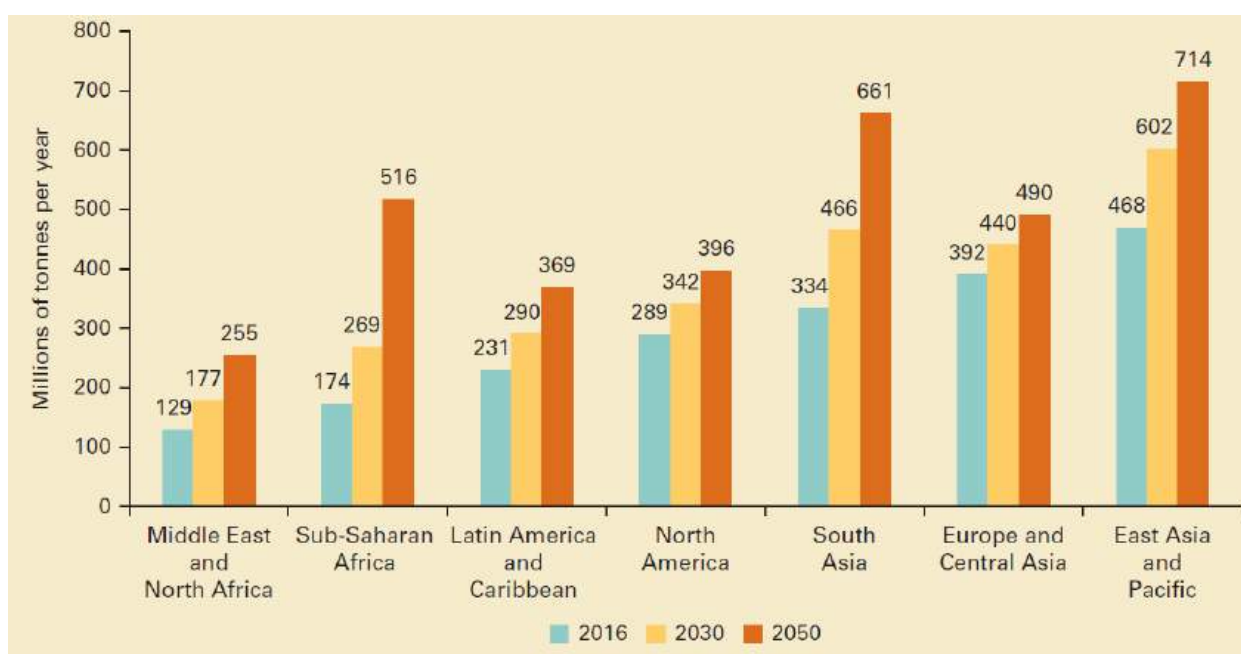
כיום ברחבי העולם מתווספים טרלוני קילוגרמים של אשפה מדי שנה לשנה. המצב עד כדי כך גרוע שבאוקיינוס השקט ישנו "אי" עשוי אשפה שגדול פי שלוש מצרפת. אנחנו החלטנו שחייבים לפעול כדי למנוע את זיהום כדור הארץ לעצמנו ולדורות הבאים. בכיתה י'א התחלנו לעסוק בתחום על ידי בניית מערכת איסוף אשפה פשוטה באזורים מימיים למען ניקוי אגמים ונהרות בסביבתנו, אך השנה רצינו להגיע לרמה גבוהה במיוחד ולקחת את פרויקט הגמר של כיתה יב לשלב הבא! מטרתנו השנה לקחת את הרעיון של איסוף אשפה משנה שעברה, אך במקום לאסוף מגופי מים, המטרה שלנו הוא לאסוף פסולת מן היבשה לפני שהוא בכלל נכנס למים. מעבר לשיפור היעילות משמעותי ממה שהיה שנה שעברה, המטרה העיקרית שלנו היא ליצור מודעות בקרב האוכלוסייה הישראלית, כי אנחנו לבד לא יכולים באמת לעשות שינוי, רק ביחד נוכל להשפיע ולשפר את העולם!

לאחרונה, כמויות עצומות של פסולת נערמו עקב מגפת COVID-19. ולמרות חלק מהמאמצים של בני האדם לצמצם פסולת ולעודד מיחזור, פשוט אין מספיק אנשים שמשתתפים באופן פעיל כדי להשפיע על המצב, ובמיוחד אין מספיק מודעות שתגרום לציבור לשים לחץ כלפי ממשלות העולם להוציא תקציבים משמעותיים להילחם בבעיה. לפי המצב הנראה, בעיית הפסולת צפויה רק להחמיר. בשל מגפת הקורונה, שיעורי המיחזור ירדו ברחבי כל העולם, ובנוסף גם יצרה כמות עודפת של פסולת חדשה בצורות של PPE ופריטים חד פעמיים. מחקר מ-*Proceedings of the National Academy of Sciences (PNAS)* הראה שהמגיפה יצרה למעלה מ-8 מיליון טון של פסולת פלסטיק ברחבי העולם ויותר מ-25,000 טון של פסולת זו נכנסה לאוקיינוס.

כמויות האשפה בעולם משפיעים על בני אדם במספר רב של דרכים, והרבה מהשפעות חדשות מתגלות מדי שנה, אך בני האדם אינם היצורים החיים היחידים שחיים על כדור הארץ, גם בעלי חיים מושפעים באותו כמות ואף יותר מבני אדם, בניית מזבלות הורסת להם את אזורי המחיה ואזורים טבעיים בהם הם אמורים לגדול, בעלי חיים נוטים לאכול אשפה מכיוון שחושבים בו כאוכל וכך מרעילים את עצמם, ומחקרים מראים לשינוי התנהגות אצל בעלי חיים, לרוב כמובן לרעה. בארגנטינה, ייתכן שפסולת שהופקה מפעילות דיג מוגברת גרמה לגידול של 37% באוכלוסיות השחפים, מה שהביא להסלמה בשיעור התקפות שחפים על לווייתנים שרבים מהם מינים בסכנה. תוך 30 שנה, עדויות להתקפות שחפים על לווייתנים עלו מ-2% בשנות ה-70 ל-99% בשנות ה-2000.

מחקר גישוש

בספטמבר 2018, האו"ם הודיע כי ייצור הפסולת העולמי שלנו צפוי לעלות ב-70% עד 2050, אלא אם ננקוט פעולות בעניין. המין האנושי מייצר כיום שני מיליארד טונות של פסולת בשנה, בין 7.6 מיליארד בני אדם. גידול האוכלוסין הוא חלק ניכר מהבעיה, אך רמות הצריכה בתוך קומץ מדינות מפותחות, והניהול השגוי שלהן בפסולת, הם שהובילו לאסון הסביבתי הזה. ארצות הברית היא הגנרטור הגדול ביותר של פסולת לנפש ברחבי העולם, כאשר כל אזרח מייצר בממוצע 808 קילוגרמים בשנה - כמעט טון - ויותר מכפול מזה של אזרחי יפן. עם זאת, כפי שמדגיש מדד הפסולת העולמי, זה לא רק יצירת הפסולת שיאיים על כדור הארץ שלנו בעשורים הקרובים - אלא הדרך בה אנו בוחרים לנהל אותה. כמות הפסולת הצפויה עד לשנת 2050 (מחולק לאזורי העולם):



כמה מהתוצאות של זריקת אשפה במקומות ציבוריים:

השפעה על בני האדם: פסולת, אם לא מופרדת ואוספה כראוי, עלולה בסופו של דבר לפגוע באנשים במגוון דרכים. לדוגמה, זריקת בדלי סיגריות ברשלנות עלולה לעורר שריפות. פסולת פריטי מזון עלולה להוביל להתפשטות של מזיקים ומחלות. פחי אשפה או אשפה לא מטופחים עלולים להכיל מלריה/יתושים מפצי דנגי. פסולת ביולוגית מסוכנת עלולה להגיע גם למקווי מים ועלולה לזהם מערכות מים שלמות.

השפעה על בעלי חיים: גם בעלי חיים יבשתיים וגם בעלי חיים ימיים טועים לעתים קרובות בפריטי פלסטיק שהושלכו כמזון. הבטן של בעלי חיים לא יכולה לעכל פלסטיק. זה יכול להוכיח להיות מזיק לבריאותם ועלול להרוג אותם. בעלי חיים רגישים גם לנזקים מחומרי פסולת מזיקים, כמו פסולת ביו-רפואית, שעלולה להכיל פתוגנים קטלניים.

השפעה על הסביבה: זריקת אשפה בצידי הכביש, ברחובות, במקווי מים, עלולה להוביל לחומרים רעילים או כימיקלים להיסחף לתוך נהרות, אוקיינוסים, אגמים, נחלים, יערות וכו'. פסולת זו תגרום בסופו של דבר. להוביל לזיהום של נתיבי המים, האדמה, אזורי היער, הקרקעות או הסביבה המימית.

עם כל המאמצים הכספיים שמומנו למטרת קמפיינים על מנת לשכנע אנשים ברחבי כל העולם לא לזרוק זבל במקומות ציבוריים עדיין מספר הזבל ברחובות, יערות וחופים עולה משנה לשנה ולכן אנו לא מסתפקים יותר באמון בני האדם אנו נזדקק את העזרה של הרובוטים. הרעיון שלנו הוא לעזור לאיכות הסביבה מבלי לסמוך על אנשים שינקו אחריהם ולכן אנו העלנו רעיון של רובוט שינקה אחרי בני האדם בצורה אוטומטית מבלי שנצטרך להתאמץ לאסוף אשפה.



כיום קיימים רובוטים אשר מטרתם לנקות בצורה אוטומטית אך הם מנקים רק בתים. כמו למשל חברת "iRobot" אשר מייצרת רובוטים לניקוי ביתי בצורה אוטומטית על ידי שיטה פשוטה, כאשר הרובוט מתנגש עם עצם מסויים הוא מסתובב 180 מעלות וממשיך לנסוע עד לסיום כל חלקי הבית. אומנם שיטה זאת מנקה את הבית ביסודיות היא לא יכולה לנקות מקומות פתוחים כי הרובוט גם לא מתאים לתנאי שטח וגם שיטת הנקיון שלו לא יעילה במקומות שהם לא סגורים.



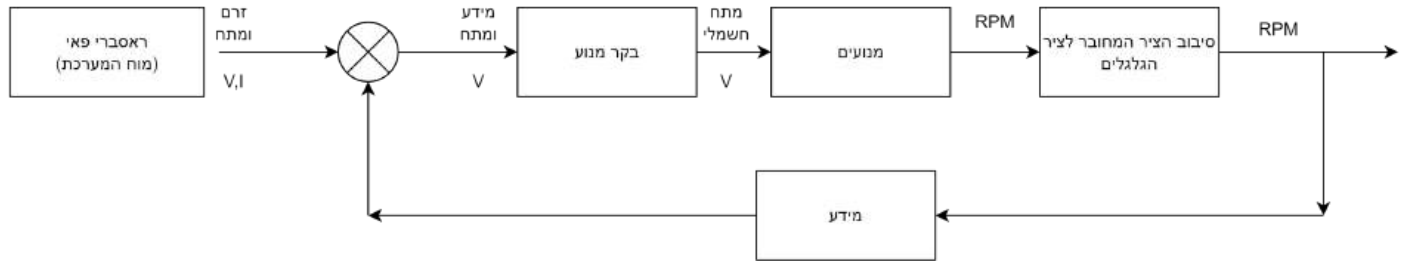
אנחנו ניקח את הרעיון של iRobot נשפר אותו ונתאים אותו לנקות אשפה בכל המקומות אשר בני אדם מלכלכים ולא מנקים אחריהם ובכך נעזור לכל העולם בבעיית איסוף האשפה.



דרישות

- המערכת יכולה לנוע לנוע במספר רב של שטחים יכולת זה נדרשת מכיוון שאשפה נמצאת בכל מקום בעולם, בין אם זה בחופים, יערות או אפילו מדברים.
- יכולת זיהוי עצמים (Object Recognition) כדאי שהמערכת תהיה אוטונומית, נשתמש בComputer vision וObject recognition
- תקשורת עם המערכת בכל מקום
- יכולת איסוף קלה ומהירה של כל סוגי האשפה הקיימים באזורים יבשתיים לדוגמא: שקיות זבל, בקבוקי פלסטיק, פחיות ומגוון דברים נוספים במידת הצורך.
- ידידותי לסביבה

תאור מלבני



רפלקציה

הבעיה שהגדרנו מאוד רלוונטית לאחד מחברי הצוות, נועם, שאבא שלו עוסק בתחום האקולוגיה ולכן מגיל צעיר חונך אותו לשמור על הסביבה בכל דרך אפשרית. מעבר לכך, לדעתנו הבעיה רלוונטית לכל בני האדם בעולם, מכיוון שכבר עכשיו ניתן להרגיש את ההשפעות והנזקים שהדורות הקודמים עשו לכדור הארץ. מספר רב של בעלי חיים כבר נכחדו, והרבה עוד נמצאים בסכנה תמידית של הכחדה. אנחנו מרגישים את ההשפעות של הזיהום על כדור הארץ בכך שגובה הים מתחיל לעלות, הכמות פחמן דו חמצני באטמוספירה עולה, עוד ועוד אזורים בעולם מוצפים באשפה, הטמפרטורה הממוצעת בעולם עולה, ויותר ויותר תופעות טבע כמו סופות, הוריקנים, התפרצויות הרי געש ועוד מתרחשים בתדירות הרבה יותר גבוהה מהרגיל. הבעיה הנפתרת ראויה להיפתר במסגרת עבודת גמר במכטרוניקה מהסיבה שהיא ראלית וקיימת צורך אמיתי בעוד ועוד פתרונות כדי לנסות להציל את העולם, דבר זה מובן לפי כך שיותר ויותר חברות אשר מפתחות מערכות דומות צצים ברחבי העולם. פתרון הבעיה יכול להועיל המון לעולם בבעיית נקיון האשפה, הרובוט יוכל להחליף עבודה של מספר אנשים מבלי לסכן חיי אדם ובתקווה בלי לפספס חתיכות אשפה קטנות שבני אדם עלולים לפספס.

איסוף המידע

שאלות חקר לאיסוף מידע

1. כיצד עובד Object detection?
2. אילו מנועים ניתן להשתמש במערכת ההנעה?
3. איזה דרכים נפוצות יש להנעה שתיצור תנועה לינארית?
4. איך ניתן לבצע תקשורת בין התת מערכות, איך ניתן לבצע תקשורת עם המערכת כלל?

זיהוי אובייקטים

זיהוי אובייקטים היא טכניקת ראייה ממוחשבת לאיתור מופעים של אובייקטים בתמונות או בסרטונים. אלגוריתמים לזיהוי אובייקטים בדרך כלל משתמשים בMachine Learning או Deep Learning כדי לייצר תוצאות משמעותיות. כאשר בני אדם מסתכלים על תמונות או וידאו, אנו יכולים לזהות ולאתר אובייקטים מעניינים תוך מספר רגעים. המטרה של זיהוי אובייקטים היא לשכפל אינטליגנציה זו באמצעות מחשב.

Deep Learning vs. Machine Learning

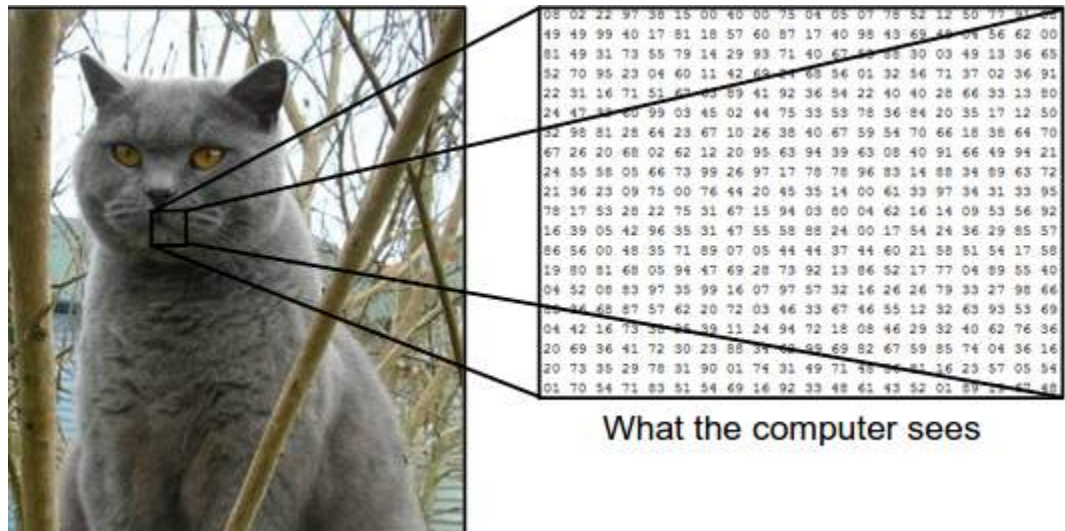
Machine Learning	Deep Learning
Machine learning uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned	Deep learning structures algorithms in layers to create an "artificial neural network" that can learn and make intelligent decisions on its own
Can train on lesser training data	Requires large data sets for training
Takes less time to train	Takes longer time to train
Trains on CPU	Trains on GPU for proper training
The output is in numerical form for classification and scoring applications	The output can be in any form including free form elements such as free text and sound
Limited tuning capability for hyperparameter tuning	Can be tuned in various ways

CNN

גישת אחת לזיהוי אובייקטים המבוססת על Deep Learning- שימוש נרחב בעולם היא Convolutional Neural Networks או CNNs.

בדומה לאופן שבו ילד לומד לזהות עצמים, עלינו להראות לאלגוריתם מיליוני תמונות לפני שהוא יוכל להכליל את הקלט ולבצע תחזיות לתמונות שלא ראה מעולם. מחשבים 'רואים' בצורה שונה מאיתנו. העולם שלהם מורכב ממספרים בלבד. כל תמונה יכולה להיות מיוצגת כמערך דו מימדי של מספרים, המכונים פיקסלים.

אבל העובדה שהם תופסים תמונות בצורה אחרת, לא אומרת שאנחנו לא יכולים לאמן אותם לזהות דפוסים, כמונו. אנחנו רק צריכים לחשוב מהי תמונה בצורה אחרת.

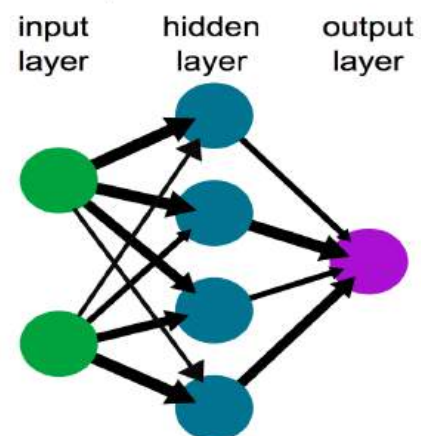


לפני שמגיעים ל-CNN's צריך להבין מה זה בעצם Neural Network:

Neural Networks

Neural Networks נועדו לעבוד בדיוק כמו המוח האנושי, הוא מורכב "מנוירונים" מלאכותיים שמאורגנים בשכבות. ישנם שלושה סוגים של שכבות: שכבת קלט, שכבת פלט ושכבה נסתרת.

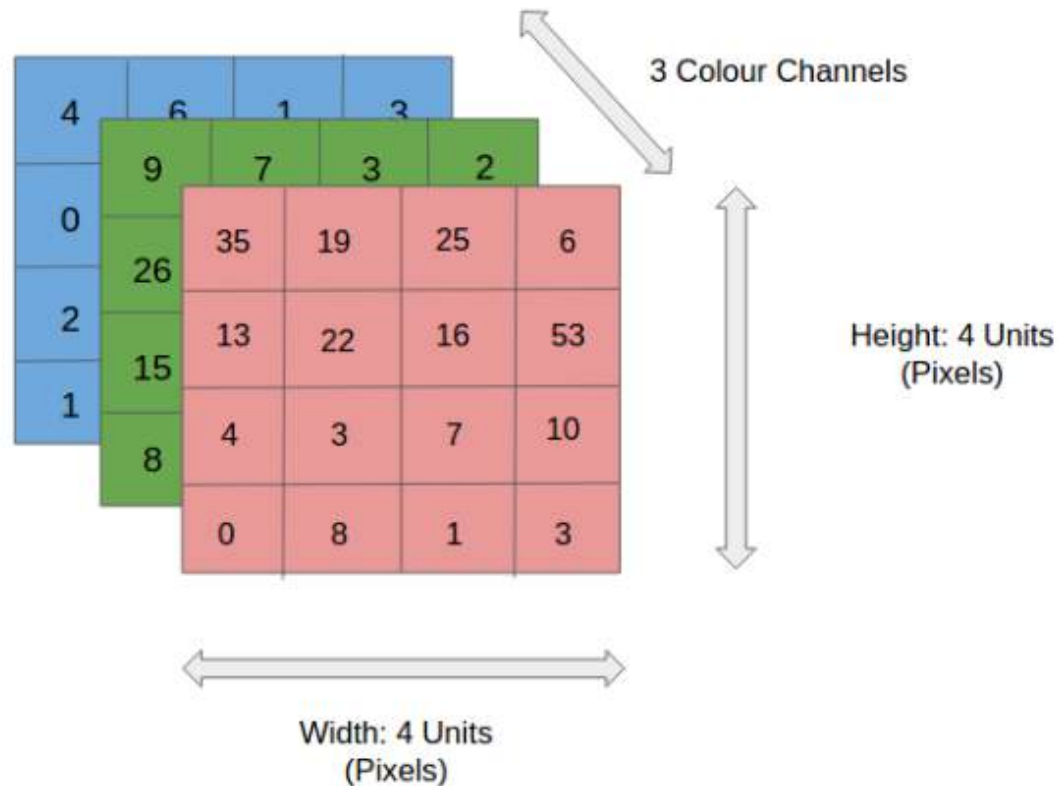
A simple neural network



ברוב המקרים, יהיו מספר שכבות נסתרות ברשת. רשתות שאין להן את השכבות הנסתרות הללו נקראות Single Layer Perceptrons.

הנוירונים בשכבת הקלט מקבלים את אובייקטי הקלט. לדוגמה, אם הקלט הוא תמונה, אובייקטי הקלט עשויים להיות פיקסלים המומרים למספר המבוסס על גווני הגריסקייל, או על גווני ה-RGB.

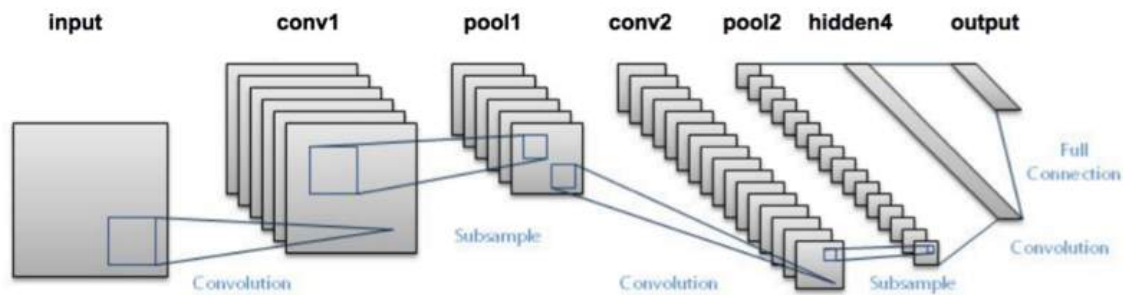
- תמונת RGB בעצם מטריצה של ערכי פיקסלים בעלת שלושה מישורים ואילו תמונה בגווני אפור זהה אך יש לה מישור יחיד. כדי להבין ניתן להסתכל על התמונה הזו:



כאשר נוירון מופעל, הוא "יורה", אשר מפעיל את הנוירונים בשכבה הבאה. כל נוירון בשכבה אחת מעביר פלט לנוירונים בשכבה הבאה. תפוקה זו מוגדרת על ידי שני גורמים: משקל והטיה/העדפה. ה"משקל" מגדיר כמה חשובה קלט מסוים לנוירון הבא, והוא גם מכון את זרימת הערכים מקלט לפלט. (בתרשים למעלה, הקווים השחורים מייצגים משקלים). מצד שני, "הטיה" היא ערך קבוע מוסף המגדיר כמה קל לנוירון להיפטר.

תהליך שליחת הנתונים משכבה אחת לאחרת נקרא התפשטות (Propagation). ישנם שני סוגי התפשטות: התפשטות קדימה (Forwards Propagation) והתפשטות אחורה (Backwards Propagation). בהתפשטות קדימה, הנתונים עוברים מקלט לשכבה נסתר לפלט. זה מסתיים בתחזית המבוססת על הקלט, שיכולה להיות מדויקת או לא מדויקת.

בהפצה לאחר, חיזוי משכבת הפלט עוברת מעקב מהפלט לשכבת הקלט, אשר מציגה את שיעור השגיאה. זה משמש לאחר מכן כדי לשנות את המשקולות וההטיות של כל נוירון, מה שמעניק לנוירונים שיעור שגיאה גבוה יותר והתאמה גדולה יותר. חשוב להתאים כל הזמן את המשקולות כדי למזער שגיאות ולהשיג דיוק גבוה יותר.

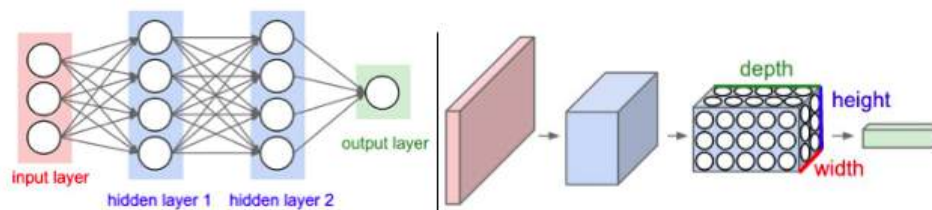


חזרה ל-CNN

CNN נבדלות מ-Neural Networks רגיל על ידי ביצועים משודרגים שלהן כאשר מכניסים אליהם אינפוט של תמונה, דיבור או שמע. ישנם מספר סיבות לכך, ל-CNN יש ארכיטקטורה שונה מזו של Neural Networks רגילות. Neural Networks רגילות משנות קלט על ידי העברתו דרך סדרה של שכבות נסתרות. כל שכבה מורכבת מקבוצה של נוירונים, כאשר כל שכבה מחוברת במלואה לכל הנוירונים בשכבה שלפני כן. לבסוף, ישנה שכבה אחרונה המחוברת במלואה - שכבת הפלט - המייצגת את התחזיות.

CNN's הן קצת שונות. קודם כל, השכבות מאורגנות ב-3 מימדים: רוחב, גובה ועומק. יתרה מכך, הנוירונים בשכבה אחת אינם מתחברים לכל הנוירונים בשכבה הבאה אלא רק לאזור קטן ממנה. לבסוף, הפלט הסופי יופחת לוקטור יחיד של ציוני הסתברות, המאורגן לאורך ממד העומק.

בדומה אבל ל-Neural Networks, ה-CNN's מורכבות של נוירונים מלאכותיים (Artificial Neurons). נוירונים מלאכותיים, שהם בעצם חיקוי/בנויים על בסיס של מקביליהם הביולוגיים, הם פונקציות מתמטיות המחשבות את הסכום המשוקלל של מספר כניסות ומוציאות ערך הפעלה. כאשר מזינים תמונה ב-CNN, כל שכבה יוצרת מספר פונקציות הפעלה המועברות לשכבה הבאה.

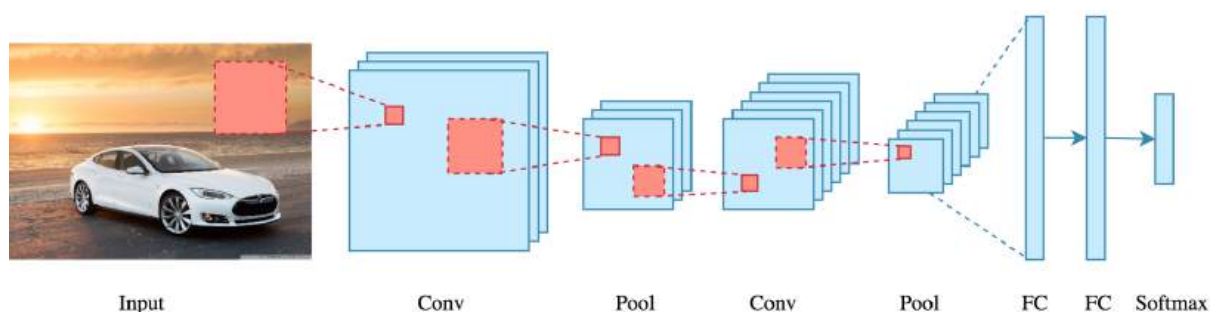


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

השכבה הראשונה מחלצת בדרך כלל תכונות בסיסיות כמו קצוות אופקיים או אלכסוניים. פלט זה מועבר לשכבה הבאה שמזהה תכונות מורכבות יותר כגון פינות או קצוות משולבים. ככל שאנו עוברים עמוק יותר לתוך הרשת, הוא יכול לזהות תכונות מורכבות עוד יותר כמו אובייקטים, פרצופים וכו'.



בנוסף Neural Networks משתמשים בשיטה מתמטית בשם Matrix Multiplication כדי לבצע חישובים/לעבד מידע, CNN's כמובן גם משתמשים בשיטה זה(הרי הם גם Neural Networks), אך בנוסף, בלפחות אחד מן השכבות שלו הוא משתמש בפעולה מתמטית בשם Convolution שהיא בעצם פעולה מתמטית על שתי פונקציות המייצרת פונקציה שלישית המבטאת כיצד צורתה של אחת משתנה על ידי השנייה.



השורה התחתונה היא שתפקיד ה-CNN הוא לצמצם את התמונות לצורה שקל יותר לעבד אותה, מבלי לאבד תכונות שהן קריטיות לקבלת חיזוי טוב.

ל-CNN יש שלושה סוגים עיקריים של שכבות, שהם:

- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

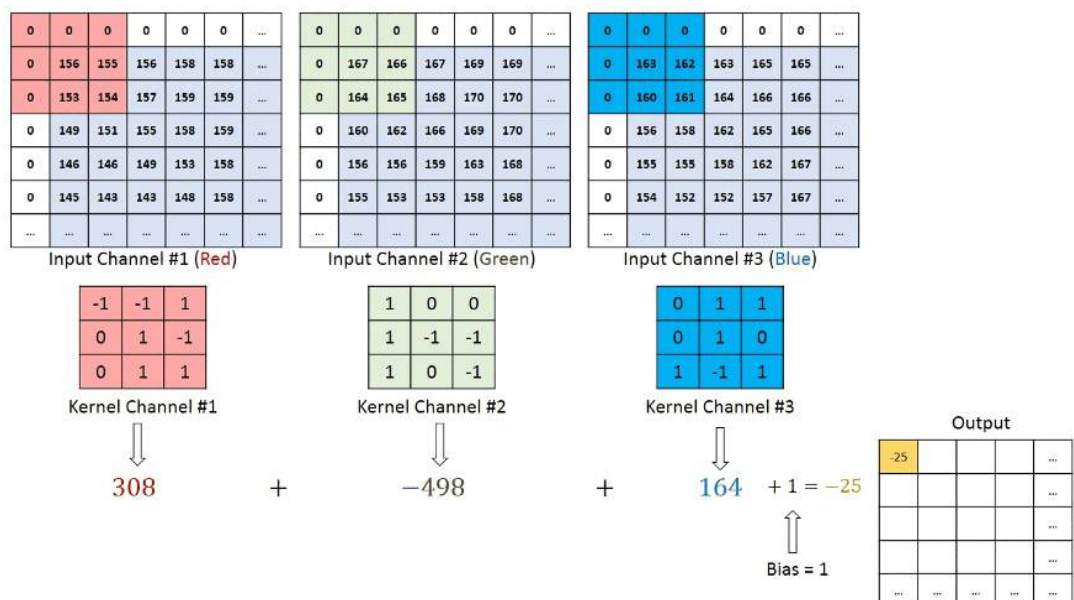
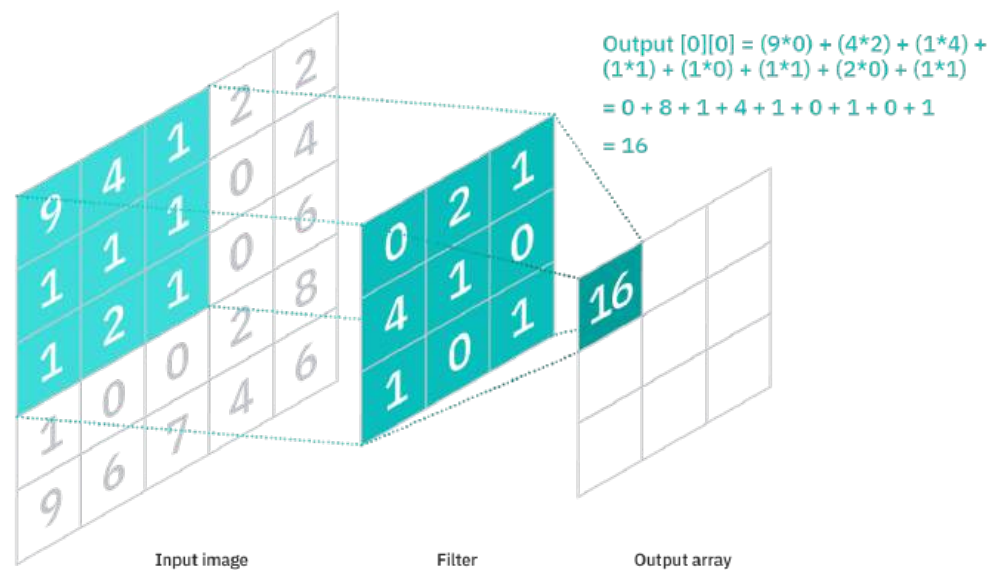
ה-CNN היא השכבה הראשונה של ה-CNN. בעוד ש-CNN layers יכולות להיות עוקבות אחר CNN layers נוספות או Pooling layer, השכבה המחוברת במלואה היא השכבה הסופית. עם כל שכבה, ה-CNN מתגבר במורכבות שלו, ומזהה חלקים גדולים יותר מהתמונה. שכבות מוקדמות יותר מתמקדות בתכונות פשוטות, כגון צבעים וקצוות. ככל שנתוני התמונה מתקדמים בשכבות ה-CNN, הוא מתחיל לזהות אלמנטים או צורות גדולות יותר של האובייקט עד שלבסוף הוא מזהה את האובייקט המיועד.

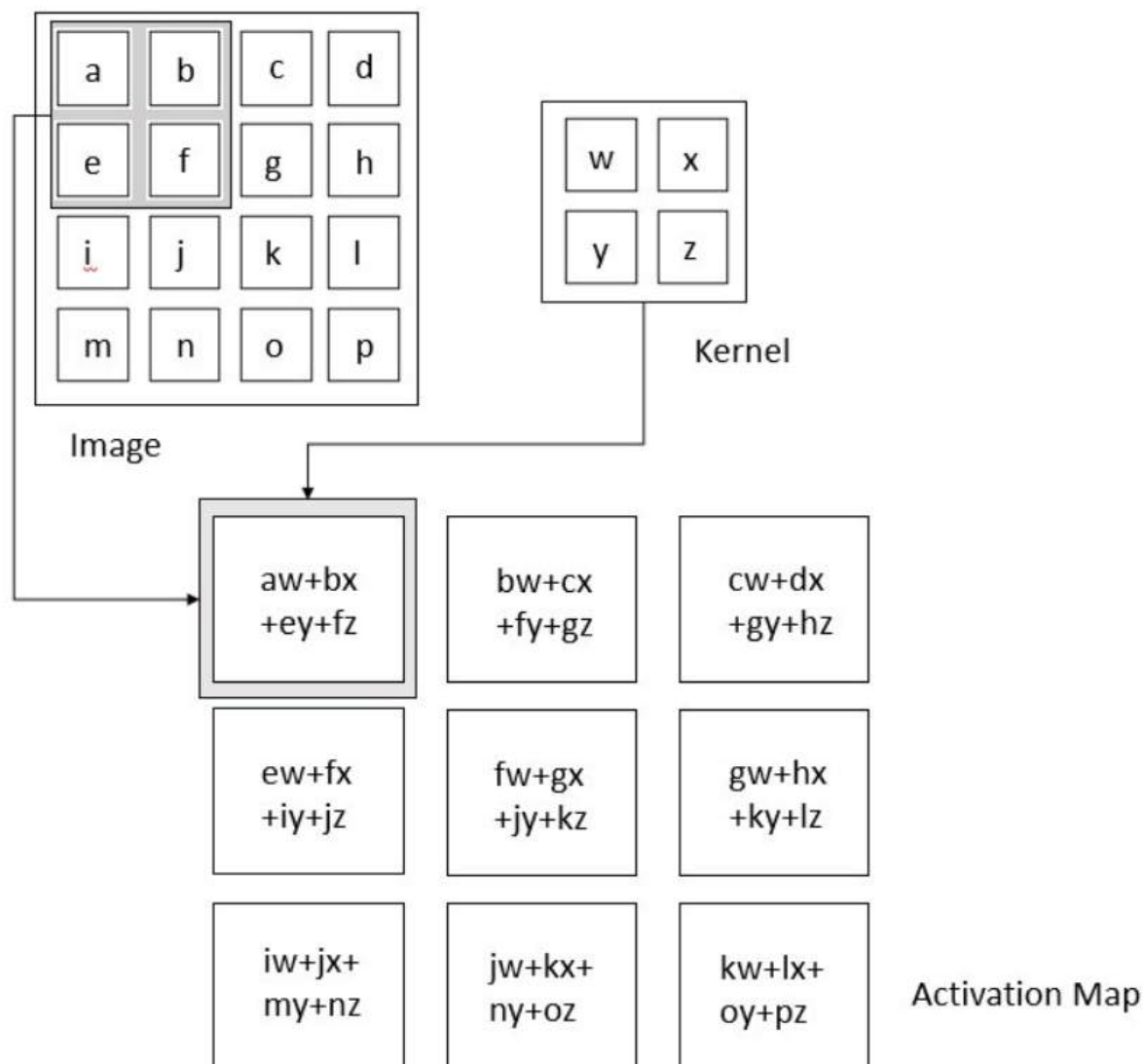
Convolutional Layer

השכבה הקונבולוציונית היא אבן הבניין המרכזית של ה-CNN, והיא המקום שבו מתרחש רוב החישוב. הוא דורש כמה רכיבים, שהם נתוני קלט, מסנן (Filter) ומפת תכונה (Feature map). נניח שהקלט יהיה תמונה צבעונית, המורכבת ממטריצה של פיקסלים בתלת מימד. המשמעות היא שלקלט יהיו שלושה

מימדים - גובה, רוחב ועומק - התואמים ל-RGB בתמונה. יישנו "פילטר" שינוע על פני השדות הקליטה של התמונה, ויבדוק אם התכונה קיימת. תהליך זה ידוע בתור Convolution.

הפילטר הוא מערך דו מימדי (2D) של משקלים, המייצג חלק מהתמונה. למרות שהם יכולים להשתנות בגודל, גודל המסנן הוא בדרך כלל מטריצה 3x3. זה גם קובע את גודל השדה הקולט. לאחר מכן הפילטר מועבר על אזור בתמונה, ומחושב תוצר נקודה בין פיקסלים הקלט לבין המסנן. לאחר מכן נקודה זו מועברת לתוך מערך פלט. לאחר מכן, המסנן "קופץ"/נע בצעדים לאורך ורוחב התמונה עד שהפילטר מועבר על פני כל התמונה. הפלט הסופי מסכום הנקודות מהקלט ומהפילטר ידוע כמפת תכונה (Feature Map), מפת הפעלה (Activation Map) או תכונה מפותלת (Convolved Feature).





כפי שניתן לראות בתמונה הראשונה למעלה, כל ערך פלט במפת התכונות לא חייב להתחבר לכל ערך פיקסל בתמונת הקלט. זה רק צריך להתחבר לשדה הקולט, שבו הפילטר מיושם/מופעל. מכיוון שמערך הפלט אינו צריך למפות ישירות לכל ערך קלט, Convolutional Layers ו-Pooling Layers מכונים בדרך כלל שכבות "מחוברות חלקית". עם זאת, ניתן לתאר מאפיין זה גם כקישוריות מקומית (Local Connectivity).

המשקלים בפילטר נשארים קבועים כשהוא נע על פני התמונה, מה שמכונה גם שיתוף פרמטרים (Parameter Sharing) כמה פרמטרים, כמו ערכי המשקל, מתכוונים במהלך האימון בתהליך של התפשטות לאחור (Backpropagation). עם זאת, ישנם שלושה היפרפרמטרים המשפיעים על גודל הנפח של הפלט שיש להגדיר לפני תחילת האימון של הרשת העצבית. אלו כוללים:

1. מספר הפילטרים משפיע על עומק הפלט. לדוגמה, שלושה פילטרים נפרדים יניבו שלוש מפות תכונות (Feature Map) שונות, היוצרות עומק של שלוש.

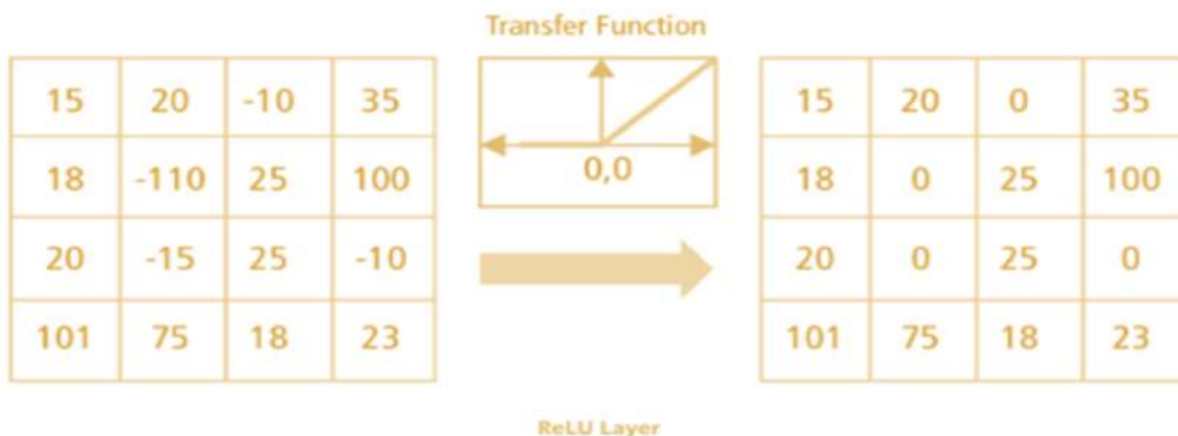
2. צעד (Stride) הוא המרחק, או מספר הפיקסלים, שהפילטר עובר על מטריצת הקלט. בעוד ערכי צעד של שניים ומעלה הם נדירים, צעד גדול יותר מניב תפוקה קטנה יותר.

3. ריפוד (Padding) (כמות הפיקסלים שנוספו לתמונה כאשר הוא מעובד על ידי הפילטר. הוספת ריפוד לתמונה המעובדת על ידי CNN מאפשרת ניתוח מדויק יותר של תמונות מכיוון שאם פילטר הוא מטריצה בגודל של 3×3 ותמונה הינה מטריצה של 10×10 , ה-Convolution רק היה קורא באזור 8×8 בתוך הגבולות. לכן כדי לשמור על יותר מדי ניתן להוסיף ריפוד) ישנם שלושה סוגי ריפוד:

- ריפוד תקף: זה ידוע גם כלא ריפוד. במקרה זה, ה-Convolution האחרון נשמטת אם הממדים אינם חופפים. (בריפוד תקף הפלט קטן מהקלט, הקטנת מידע)
- אותו ריפוד: ריפוד זה מבטיח שלשכבת הפלט יש אותו גודל לשכבת הקלט (מוסיפים אפס רק משמאל למערך ולחלק העליון של מטריצת הקלט הדו-ממדית).
- ריפוד מלא: סוג זה של ריפוד מגדיל את גודל הפלט על ידי הוספת אפסים לכל גבול הקלט. הנוסחה הכללית לחישוב התכונות ממדי התמונה בגודל $m \times m$ על בסיס גודל הפילטר $f \times f$:

$$(m \times m) * (f \times f) = (m - f + 1) * (m - f + 1)$$

לאחר כל פעולת Convolution, ה-CNN מיישם טרנספורמציה של יחידה לינארית מתקנת (ReLU) על מפת התכונות (Feature Map), כך שכל הערכים השליליים במפת התכונות הופכים ל-0 בשכבה הזאת.

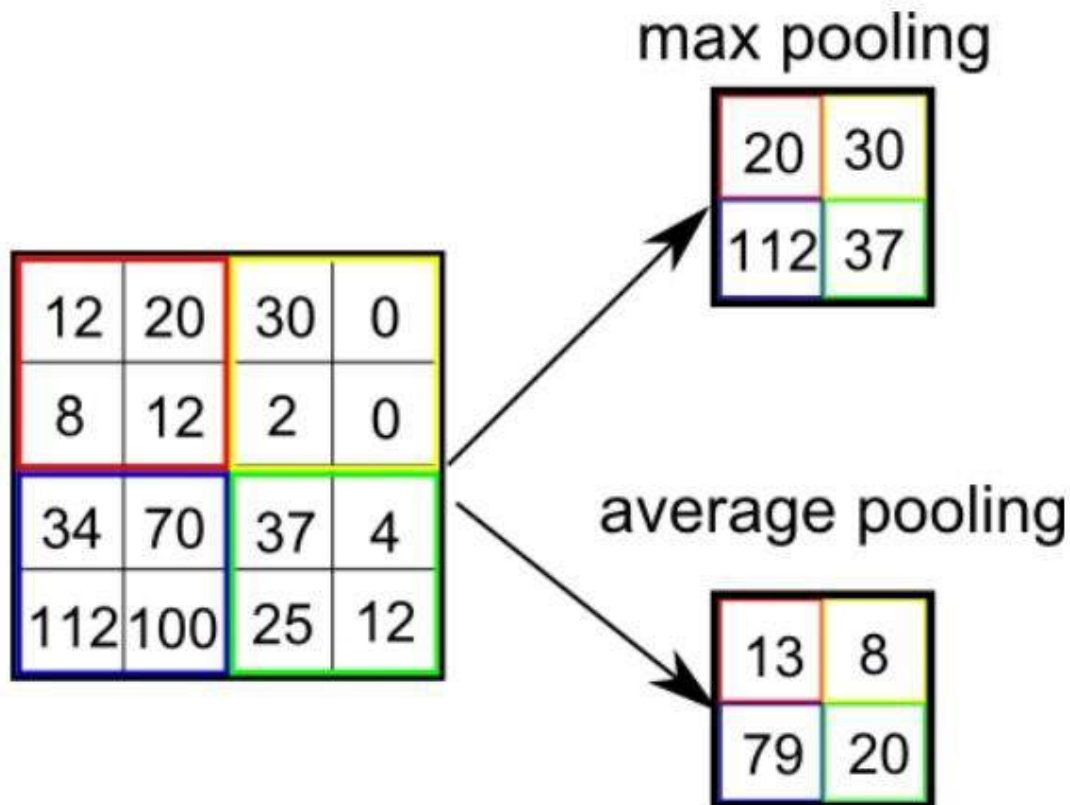


- Convolutional Layer נוספת יכולה לעקוב אחר ה-Convolutional Layer הראשון. כאשר זה קורה, המבנה של ה-CNN יכול להפוך להיררכי (Hierarchical) מכיוון שהשכבות המאוחרות יותר יכולות לראות את הפיקסלים בתוך השדות הקליטה של השכבות הקודמות.

Pooling Layer

Pooling Layers הידועים גם כ-downsampling (בדומה ל-Convolutional Layer) מעביר פילטר על פני כל הקלט, אך ההבדל הוא שלפילטר זה אין משקלים. במקום זאת, הפילטר מיישמת פונקציית צבירה על הערכים בתוך השדה הקולט, ומאכלסת את מערך הפלט. מבצע הפחתת מימדיות, ומפחית את מספר הפרמטרים בקלט סיבה אחת לשימוש בשיכבה זה היא כדי להקטין את כוח החישוב הנדרש לעיבוד הנתונים. יתר על כן, ה-Pooling Layer שימושי לחילוף תכונות דומיננטיות שהן סיבוביות ובלתי משתנות מיקום, ובכך לשמור על תהליך אימון יעיל של המודל. ישנם שני סוגים עיקריים של Pooling:

- Average Pooling: כאשר הפילטר נע על פני הקלט, הוא מחשב את הערך הממוצע בתוך השדה הקולט לשליחה למערך הפלט.
- Max Pooling: כשהפילטר נע על פני הקלט, הוא בוחר את הפיקסל עם הערך המקסימלי לשליחה למערך הפלט. מלבד זאת, גישה זו נוטה לשמש לעיתים קרובות יותר בהשוואה ל-Average Pooling.



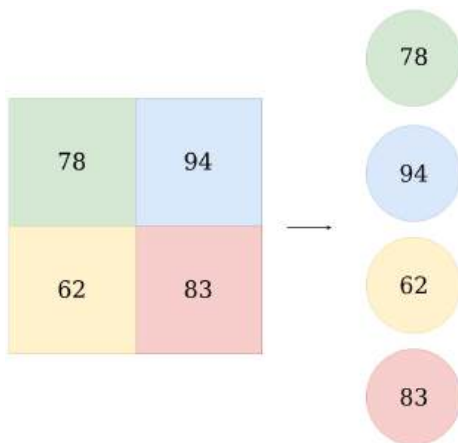
בעוד שמידע רב הולך לאיבוד בשכבת ה-Pooling, יש לו גם מספר יתרונות ל-CNN. הם עוזרים להפחית את המורכבות, לשפר את היעילות ולהגביל את הסיכון להתאמה יתר (Overfitting), התאמה יתר מתרחשת כאשר המודל שלנו הופך להיות ממש טוב ביכולת לסווג או לחזות על נתונים שנכללו בערכת האימונים, אבל לא כל כך טוב בסיווג נתונים שהוא לא הוכשר עליהם. אז בעצם, המודל התאים יתר על המידה לנתונים במערך האימונים).

ה-Convolutional Layer ו-Pooling Layer, יוצרים יחד את השכבה ה-l-th (שכבות במיקום l בסך השכבות) של ה-CNN. בהתאם למורכבות התמונות, ניתן להגדיל את מספר השכבות הללו ללכידת פרטים ברמות נמוכות עוד יותר, אך במחיר של כוח חישוב רב יותר.

לאחר התהליך לעיל, למודל אמור בהצלחה להבין את התכונות. ממשיכים הלאה, לאחר מכן צריך לשטח את הפלט הסופי ולהזין אותו לרשת עצבית רגילה למטרות סיווג.

Classification — Fully Connected Layer

Fully Connected Layer היא פשוט, Feed Forward Neural Network, כלומר היא Neural Network פשוטה שרק יכולה להעביר מידע קדימה בין ה-Nodes. ה-Fully Connected Layers הם השכבות האחרונות ב-CNN.



הקלט לשכבה המחוברת במלואה היא הפלט משכבת ה-Pooling או Convolutional Layer הסופית, אשר משטחת ואז מוזנת לשכבה המחוברת במלואה. הפלט מהשכבה הסופית Pooling Layer או Convolutional Layer הוא מטריצה תלת-ממדית, לשטח את זה הוא פשוט לגלגל את כל הערכים שלו לוקטור.

לאחר מעבר דרך ה-Fully Connected Layers במלואן, השכבה הסופית משתמשת בפונקציית ההפעלה של softmax (במקום ReLU) המשמשת כדי לקבל הסתברויות שהקלט נמצא במחלקה מסוימת (סיווג).

מושגים חשובים ב-Computer Vision

Image Classification

כנראה אחת המשימות הידועות ביותר בראייה ממוחשבת היא סיווג תמונה. זה מאפשר סיווג של תמונה נתונה כשייכת לאחת מתוך סט של קטגוריות מוגדרות מראש. דוגמא לקח היא סיווג תמונות לפי האם הן מכילות אטרקציה תיירותית או לא. נניח שמסווג (Classifier) נבנה למטרה זו ושהתמונה למטה מסופקת.



המסווג יגיב שהתמונה שייכת לקבוצת התמונות המכילה אטרקציות תיירותיות. זה לא אומר שהוא בהכרח זיהה את מגדל אייפל אלא שהוא ראה בעבר תמונות של המגדל ושנאמר לו שהתמונות האלה מכילות אטרקציה תיירותית.

Object Localization

לוקליזציה היא מציאת מיקום של אובייקט בודד בתמונה. אלגוריתם זה מאתר נוכחות של אובייקט מרכזי בתמונה ומייצג אותו באמצעות תיבה תוחמת (Bounding Box). הוא לוקח תמונה כקלט ומוציא את המיקום של התיבה התוחמת בצורה של (מיקום, גובה ורוחב).



Object Detection

אלגוריתמים לזיהוי אובייקטים פועלים כשילוב של סיווג תמונה ולוקליזציה של אובייקטים. זה לוקח תמונה כקלט ומייצר תיבה תוחמת אחת או יותר עם ה-Image Label (לאיזה מחלקה/Class הוא שייך) מחובר לכל Bounding Box. אלגוריתמים מסוג זה מסוגלים להתמודד עם סיווג ולוקליזציה מרובת מחלקות, כמו גם להתמודד עם אובייקטים עם מספר התרחשויות. המילון לראייה ממוחשבת ועיבוד תמונה נותן את ההגדרה שזיהוי עצמים הוא: "גילוי של עצמים בתוך סצנה או תמונה". המיקום של אובייקט שזוהה מתואר בדרך כלל על ידי תיבה תוחמת, שהיא קופסא המשמש לקשור את קצוות העצם, והזיהוי מוצג בדרך כלל כציון הסתברות למחלקה מסוימת.



Image Segmentation

Image Segmentation מתייחס ל"הפרדה של אובייקטים בתוך סצנה או תמונה". Segmentation שונה באופן משמעותי מזיהוי אובייקט, שבו זיהוי עצמים משתמש בתיבה תוחמת לציון מיקום אובייקט, Segmentation חוזה את הפיקסלים המדויקים המשויכים לאובייקט נתון. מיותר לציון שזה פעולה מורכבת יותר מאשר זיהוי עצמים.

Object Detection



Instance Segmentation



סוגי מנועים נפוצים לשימוש במערכת הנעה

D.C מנוע

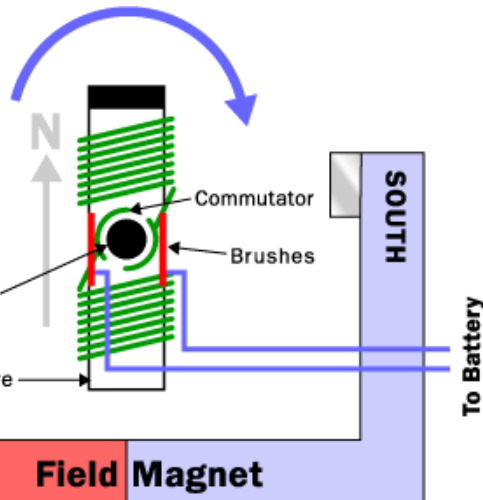
ישנם שני סוגים של מנועי D.C, בראשד (Brushed), ו-בראשלוס (Brushless).

מנוע די.סי. הוא מנוע שלוקח זרם חשמלי וממיר אותו לתנועה סיבובית מכנית. המנוע מבוסס על עיקרון של מגנטיות ומשתמש בהם כדי ליצור תנועה. כידוע למגנטים יש צפון ודרום כאשר צפון ודרום נמשכים ודרום ודרום או צפון וצפון דוחים אחד את השני. כך המנוע משתמש בתופעה זו על מנת ליצור תנועה סיבובית. המנוע די.סי. מורכב מאלקטרומוגנט גדול שלא זז שנקרא הסטטור. הוא יוצר שדה עקב

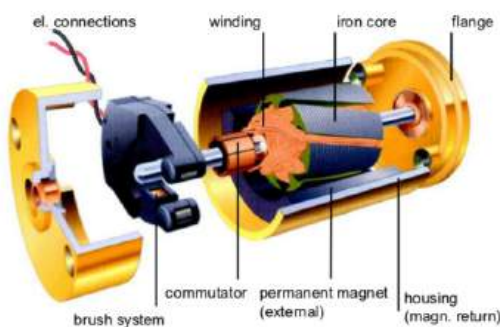
זרימה בסלילים על גביו. בין שתי קצוות הסטטור יש תיל מוליך בו זורם זרם הנקרא רוטור. כיוון שזורם שם זרם וסטטור יוצר שדה חשמלי אז פועל על התיל המוליך כוח. הרוטור שיכול לנוע ינסה להסתדר עם קצוות המגנט. כדי שהמערכת לא תעצור ברגע שהקטבים המנוגדים יתיצבו יש חלק של הרוטור בו זורם זרם אל הסלילים הנקרא מחלף. ברגע שהמנוע מגיע לנקודת האיזון מברשות מוליכות עשויות מפלמן נחושת יחליפו את כיוון הזרם על המחלף וכך יגרמו לשינוי הקיטוב של הרוטור. כיוון שהרוטור עבר את נקודת האיזון והמשיך להסתובב הוא ימשיך להסתובב כמעט עד לנקודת האיזון החדשה בגלל התנע שנוצר. ברוב המנועים יש מספר רב של סלילים על מנת ליצור תנועה חלקה.

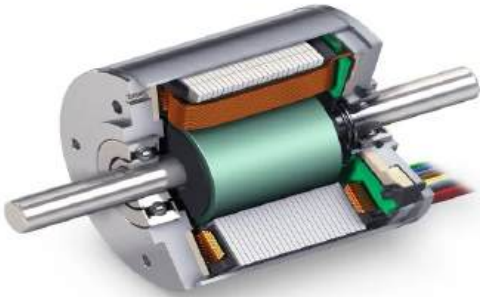
Brushed

מנועי DC משתמשים בסלילי תיל מפותלים כדי ליצור שדה מגנטי. במנוע מוברש, סלילים אלה חופשיים להסתובב כדי להניע ציר - הם החלק של המנוע שנקרא "רוטור". בדרך כלל הסלילים מפותלים סביב ליבת ברזל, אם כי ישנם גם מנועים מוברשים שהם "ללא ליבה", כאשר הפיתול נתמך בעצמו. כדי להניע מנוע מוברש, מתח DC מופעל על פני המברשות, אשר מעביר זרם דרך פיתולי הרוטור כדי לגרום למנוע להסתובב.



©2001 HowStuffWorks





Brushless

מנועי DC ללא מברשות פועלים על אותו עיקרון של משיכה ודחייה מגנטית כמו מנועי מברשת, אך הם בנויים בצורה שונה במקצת. במקום קומוטטור מכני ומברשות, השדה המגנטי של הסטטור מסובב על ידי שימוש בקומוטציה אלקטרונית. זה מצריך שימוש באלקטרוניקה בקרה אקטיבית. במנוע ללא מברשות, לרוטור מוצמדים מגנטים קבועים, ולסטטור יש פיתולים.

יתרונות:

- יש יחס ישיר בין מתח למהירות המנוע כך שהשליטה על המהירות היא פשוטה.
- היחס בין המומנט פיתול לזרם הוא ישר לכן השליטה על המומנט פיתול היא די פשוטה
- תורק התחלתי גבוה
- התקנה קלה

חסרונות:

- יקר ליצור
- שליטה גרוע במהירויות נמוכות
- דורש תחזוקה שוטפת
- רעשים אלקטרוניים, ניצוצות

רשימת מקורות:

<https://www.circuito.io/blog/arduino-motor-guide>
<http://robots.eitan.ac.il/index.php?page=MotorsDC>
<https://www.motioncontroltips.com/dc-motors>
<https://www.youtube.com/watch?v=LAtPHANefQo>
<https://www.monolithicpower.com/en/brushless-vs-brushed-dc-motors>

מנוע סרוו

מנועי סרוו הם מנועים מאוד פופולרים ויש להם שימושים רבים. מנוע סרוו הוא למעשה משלב מנוע דיסי רגיל, יחד עם חיישן כדי לקבל משוב על המיקום של המנוע. סרבו טיפוסי מורכב לרוב ממנוע דיסי קטן, מערכת גלגלי שיניים, פוטנציומטר משוב, ומספר מעגלי בקרה. בעזרת כל הדברים האלו המנוע סרוו מאפשר בקרה מדויקת למיקום.



יתרונות:

- סרווים גמישים בסביבות השימוש בהם
- לא דורש בקרה בחוג סגור
- קל לשליטה
- מדויק
- הרבה סוגים

חסרונות:

- דורש הספק מינימלי גבוה
- יקר יחסית למנועים אחרים
- שחיקה יחסית מהירה

רשימת מקורות:

<https://www.circuito.io/blog/arduino-motor-guide>

מנוע צעד

• אם במנוע זרם ישר יש בעיה, גם אם תפסיק את הזרם ברוטור ותכנן באופן מדויק מתי לעצור, עדיין לרוטור היה מומנט ולכן הוא לא יעצור מייד. גם אם נוסיף תמסורת עדיין לא נפתור את הבעיה באופן מוחלט. לעומת זאת מנועי צעד מתנהגים באופן שונה. הם לא יכולים לפעול באופן חופשי בעצמם. מנועים אלו עושים כל פעם צעד קטן. בנוסף מנועי צעד שונים ממנועי זרם ישר ביחס למומנט פיתול ומהירות. מנועי צעד טובים בליצור מומנט פיתול טוב במהירויות נמוכות

בעוד שמנוע זרם ישר עובד בעוצמה מלאה. מנוע צעד גם יוכל להחזיק את המומנט פיתול מה שמנוע זרם ישיר לא יכול לעשות.

• איך המנוע עובד: המנוע מורכב מרוטור במרכז וכמה אלקטרומגנטיים. כאשר זרם מועבר בסלילים בקומבינציה מסוימת, הסלילים כל פעם דוחפים אל הליבה המגנטית של המנוע וכך יוצרים תנועה. הסדר המסוים של הסלילים דואג שכל פעם המנועים ידחו אחד את השני וידחפו את המנוע סטפר.



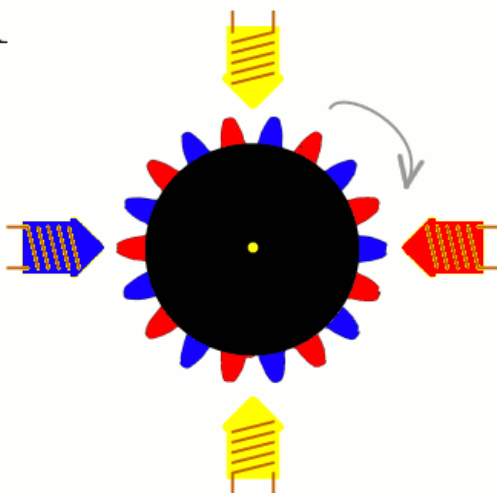
יתרונות:

- שליטה מאוד מדויקת למיקום שלו
- מאוד אמין ולא צריך הרבה תחזוקה
- תורק גבוה
- בקרה בחוג פתוח

חסרונות:

- עלול להיפגע במקרה בו מופעל עומס רב מידי
- דורש בקרה מאוד מתקדמת על מנת לשלוט בו

1



רשימת מקורות:

<http://robots.eitan.ac.il/index.php?page=MotorsStepper>
<https://www.explainthatstuff.com/how-stepper-motors-work.html>

הנעה לינארית

בוכנה פנאומטית

בבוכנה פנאומטית גז נכנס מצד אחד של מיכל לחץ ודוחף את הבוכנה עד שהלחץ משני הצדדים משתווה. אז אין כוח שדוחף את הבוכנה. כיוון שהבוכנה נמצאת בתוך מעטפת תנועה לינארית מתרחשת. כיוון שמדובר בתנועה של גז קשה ליצור תנועה מדויקת.



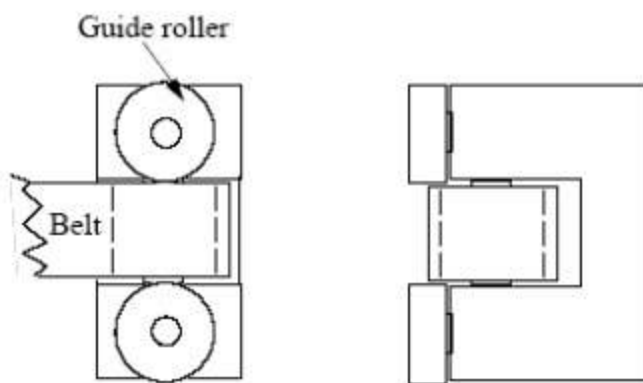
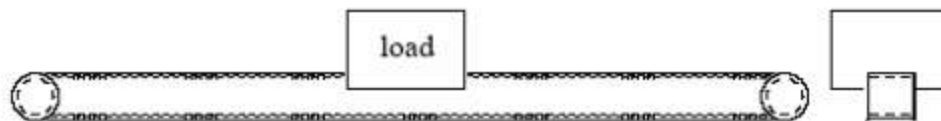
בוכנה הידראולית

בוכנה הידראולית עובדת כמו בוכנה פנאומטית רגילה רק על בסיס שמן. שמן הוא כמעט לא דחיס לכן נותן אפשרות לשליטה מדויקת. עם זאת היא עדיין לא טובה לתנועה מדויקת.



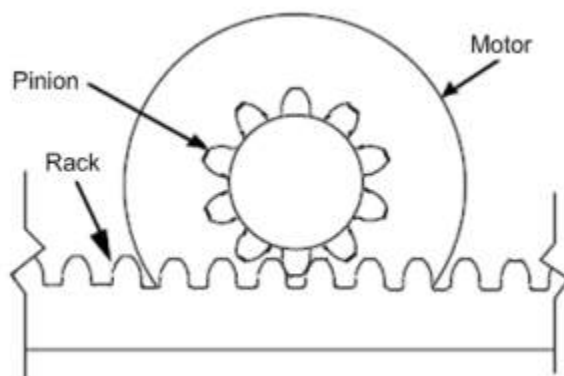
רצועות הנעה

זהו דרך נפוצה לחבר מנוע לציר שמניע גלגל שיכול להזיז רצועה. לרוב משתמשים במנועי סטפר או סרבו כדי ליצור דיוק במיקום. יש כאן הרבה גיוון בסוג הרצועה בתמסורת ועוד. רצועות עם שיניים הם לרוב הפתרון הכי טוב למחיר שגם זול וגם לא רועש יותר מידי. יש להם שימושים במגוון מקרים 12.



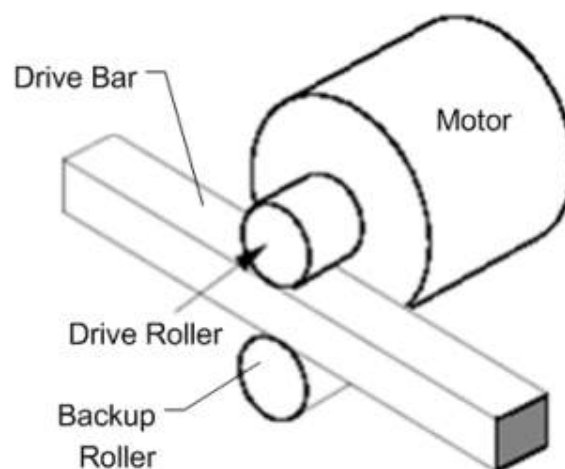
רצועת גלגל שיניים

זוהי אחד הדרכים היותר זולים ליצור תנועה לינארית מתנועה סיבובית. משתמשים בהם הרבה במכונות גדולות. קשה למצוא את היחס תמסורת המושלם. הם מסוגלים לסחוב כמה משקלים באופן עצמאי. גלגל שיניים נחשב פיתרון זול יעיל ומדויק.



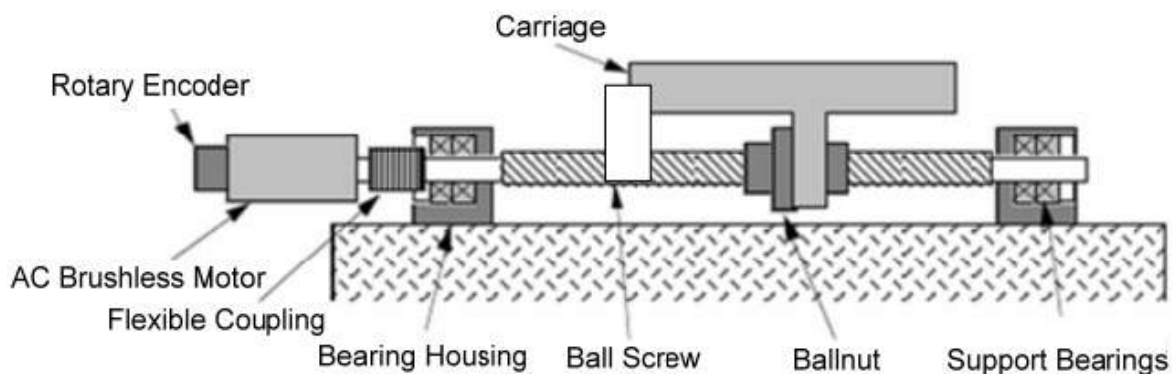
כונן חיכוך

גלגל מניע מוט שטוח הנתמך בגב מיסב רולר או הידרוסטטי שטוח. מדויק כל כך עד הננומטר. בניגוד לתיבת הילוכים רגילה עם גלגלי שיניים או גלגלי שיניים, להנעת חיכוך יש מספר אינסופי של יחסי העברה פוטנציאליים תוך שימוש במעט מאוד רכיבי הילוכים. מכיוון שההילוכים הופכים כיוון כשהגלגל המשני עובר את מרכז הגלגל הראשי, יש לו מהירויות פוטנציאליות לאחור כמו מהירויות קדימה. ניתן גם להגביר את מהירות הרכב ללא עליה מקבילה במהירות המנוע, מה שהופך אותו לאחת מתיבות ההילוכים החלקות והקלות ביותר לתפעול שהומצאו אי פעם.



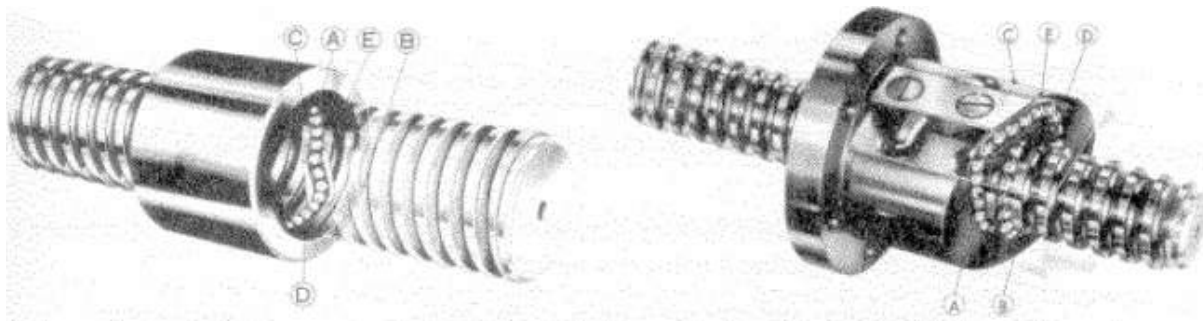
בורג הנעה

בורג הברגה מחובר למנוע. הבורג הופך למאין אום שמחובר לבורג ובכך לא יכול להסתובב. כאשר הבורג שמחובר למנוע נע סביב עצמו הבורג נע ישר. הטכנולוגיה יוצרת תנועה מהירה וחזקה עם רעש נמוך. היא גם יכולה לדאוג למיקום די מדויק. אפשר לשנות את ההברגה מה שישפיע על המהירות והכוח שהבורג יכול לתת. הבורג מתאים במקרים בהם העליות מאוד קטנות.



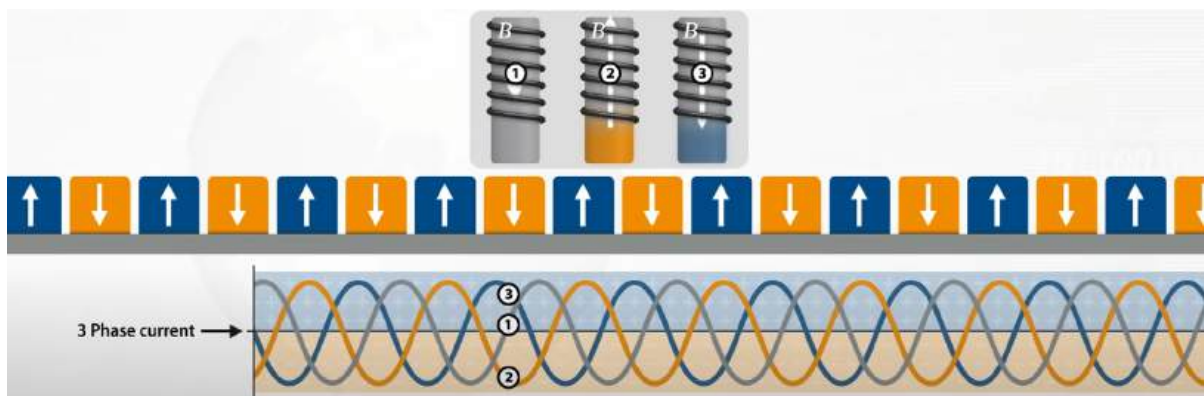
בורג הנעה כדורי

ברגי הנעה כדורי עובדים כמו בורג הנעה רגיל רק שבמקום שהאום יהיה על הבורג האום יושב על כדורים. זה מפחית באופן משמעותי את החיכוך ומגביר את היעילות. מערכות כאלו גם לעולם לא ננעלות. בורג כדורי הם מאוד מדויקים הם יכולים לספק גם מהירות וגם עוצמה בקול מאוד חלש. החיסרון שהמחיר שלהם מאוד יקר.



מנועים לינארים

מנועים לינארים הם כמו מני דיסי רק שהמנוע שוטח למסלול ארוך. המנוע הוא פס ארוך המורכב מהרבה מגנטים על פולריות מתחלפת מנוגדת. למשקל עצמו סוחב כמה אלקטרומגנטים שמשנים את השדה שלהם על פי הזרם בהם (יש בקרה עם זרם חליפין דרך האלקטרומגנטים). המנועים האלה נחשבים ברמה מאוד גבוהה, הם מאוד יקרים וצורכים המון חשמל עם נצילות לא טובה אבל הם מדויקים שקטים ומהירים.



מקורות:

<https://my.mech.utah.edu/~me7960/lectures/Topic3-LinearMotionSystems.pdf>
<https://en.compotech.se/five-technologies-for-obtaining-linear-motion-with-electric-motors>
<https://www.linearmotiontips.com/main-types-linear-actuators>
<https://en.compotech.se/five-technologies-for-obtaining-linear-motion-with-electric-motors>
https://www.youtube.com/watch?v=0_QBI6-_jJU

דרכי תקשורת אלחוטיים מרכזיות

מה זה?

תקשורת אלחוטית היא סוג של תקשורת נתונים שמתבצעת ומועברת בצורה אלחוטית. זהו מונח רחב המשלב את כל ההליכים והצורות של חיבור ותקשורת בין שני מכשירים או יותר באמצעות אות אלחוטי באמצעות טכנולוגיות והתקני תקשורת אלחוטיים.

איך זה פועל?

תקשורת אלחוטית פועלת בדרך כלל באמצעות אותות אלקטרומגנטיים המשודרים על ידי מכשיר מאפשר באוויר, בסביבה הפיזית או באווירה. המכשיר השולח יכול להיות שולח או מכשיר ביניים עם יכולת להפיץ אותות אלחוטיים. התקשורת בין שני מכשירים מתרחשת כאשר היעד או מכשיר הביניים המקבל לוכדים את האותות הללו, ויוצרים גשר תקשורת אלחוטי בין התקן השולח למקלט.

תקשורת לוויינית (Satellite Communication)

תקשורת לוויינית פרוסה בכל רחבי העולם כדי לאפשר למשתמשים להישאר מחוברים כמעט בכל מקום על פני כדור הארץ. כאשר האות (אלומה של מיקרוגל מאופנן) נשלחת ליד הלוויין אז, הלוויין מגביר את האות ושולח אותו בחזרה למקלט האנטנה שנמצא על פני כדור הארץ. תקשורת לוויינית מכילה שני מרכיבים עיקריים כמו קטע החלל וקטע הקרקע. קטע הקרקע מורכב מצידוד שידור, קליטה ועזר קבוע או נייד ומקטע החלל, שהוא בעיקרו הלוויין עצמו.



תקשורת אינפרא אדום (Infrared Communication)

תקשורת אלחוטית אינפרא אדום מעבירה מידע במכשיר או במערכת באמצעות קרינת IR. IR היא אנרגיה אלקטרומגנטית באורך גל ארוך מזה של אור אדום. הוא משמש לבקרת אבטחה, שלט רחוק בטלוויזיה ותקשורת לטווח קצר. בספקטרום האלקטרומגנטי, קרינת IR נמצאת בין גלי מיקרו ואור נראה. אז, הם יכולים לשמש כמקור תקשורת.

לתקשורת אינפרא אדום מוצלחת, נדרשים משדר LED פוטו וקולטן פוטודיודה. משדר ה-LED מעביר את אות ה-IR בצורה של אור בלתי נראה, הנלכד ונשמר על ידי קולטן הצילום. אז המידע בין המקור למטרה מועבר בדרך זו. המקור והיעד יכולים להיות טלפונים ניידים, טלוויזיות, מערכות אבטחה, מחשבים ניידים וכו' התומכים בתקשורת אלחוטית.



בלוטות' (Bluetooth)

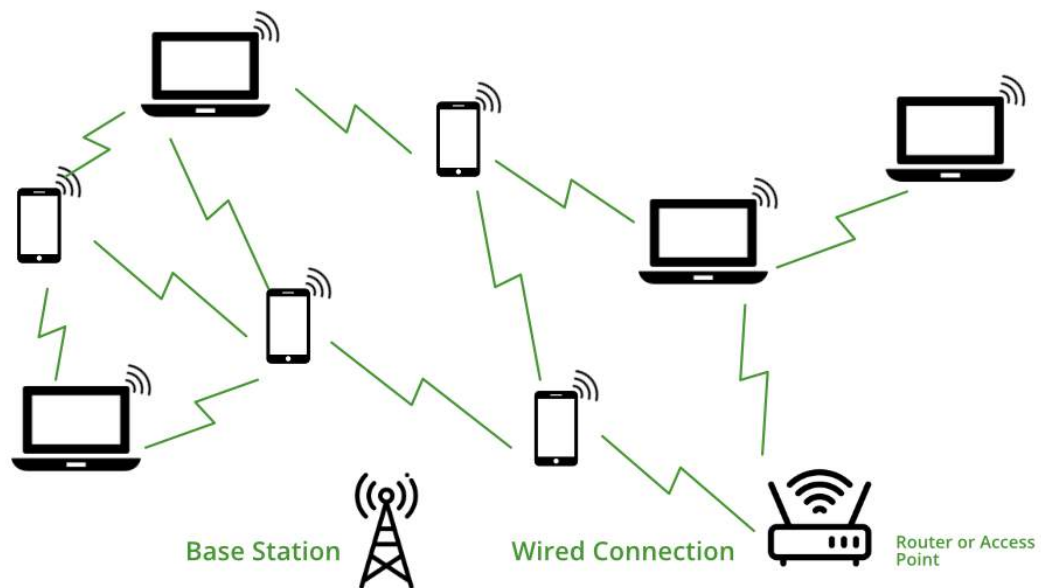
טכנולוגיה אלחוטית פתוחה להעברת נתונים אלקטרוניים קבועים וניידים על פני מרחקים קצרים. ה-Bluetooth נוצר על ידי הסרת הצורך בהתערבות המשתמש ושמירה על עוצמת השידור נמוכה ביותר כדי לחסוך בצריכת החשמל של הסוללה. כיום טווח הקליטה של הבלוטות' תלוי בסוג המכשיר שממנו הוא משדר, אך הוא יכול להגיע לטווח מקסימלי של עד כ - 400 מ'.

Wi-Fi

Wi-Fi היא טכנולוגיית רשת אלחוטית, שבאמצעותה אנו יכולים לגשת לרשתות או להתחבר למחשבים אחרים או ניידים באמצעות מדיום אלחוטי. ב-Wi-Fi, נתונים מועברים על פני תדרי רדיו בטווח מעגלי.

Wi-Fi היא טכנולוגיה אלחוטית לרשת, ולכן היא משתמשת בגלים אלקטרומגנטיים כדי לשדר רשתות. ישנן חלוקות רבות של גלים אלקטרומגנטיים בהתאם לתדירותם כגון קרני רנטגן, קרינת גמא, גלי רדיו, מיקרוגל וכו', ב-Wi-Fi נעשה שימוש בתדר הרדיו. להעברת אות Wi-Fi יש שלושה מדיומים:

1. רשת תחנת בסיס או חיבור Ethernet(802.3): זוהי הרשת ה-host הראשית ממנה מסופק חיבור הרשת לראוטר.
2. נקודת גישה או ראוטר: זהו גשר בין רשת קווית לרשת אלחוטית. הוא מקבל חיבור Ethernet קווי וממיר את החיבור הקווי לחיבור אלחוטי ומפיץ את החיבור כגל רדיו.
3. גישה למכשירים: הנייד, המחשב, מיקרוקונטרולר או כל דבר אחר שממנו משתמשים ב-Wi-Fi ובגלישה באינטרנט.



כל מכשירי האלקטרוניקה קוראים נתונים בצורה בינארית. נתבים מספקים גלי רדיו וגלים אלה נקלטים על ידי המכשירים שלנו וקוראים את הגלים בצורה בינארית.

תקשורת בגלי רדיו (Radio Frequency)

תקשורת בתדר רדיו היא כנראה צורת התקשורת הקלה ביותר בין מכשירים. גלים אלקטרומגנטיים נוצרים בעזרת תנועה מואצת של מטענים חשמליים, שהשתנותם היא בהתאם למידע שנשלח. טווח שידור גלי הרדיו תלוי בסוג המשדר וסוג הרכיבים שמשתמשים בהם.

שידור המוני - שידור גלי רדיו ממשדר לכל הכיוונים מאפשר לכל בעל מקלט רדיו מתאים לקלוט את השידורים הללו.

שידור מסוים - שידור גלים הנשלחים באמצעות אנטנות המכוונות רק לכיוון מסויים.

פרוטוקולים כמו ZigBee או ZWave משתמשים ברדיו RF בעל הספק נמוך המוטמע או מותאם לאחור במכשירים ובמערכות אלקטרוניות

פרוטוקול Z-Wave

הטווח של Z-Wave הוא כ-100 רגל (30 מ'). רצועת תדרי הרדיו המשמשת היא ספציפית למדינה שלה. לדוגמה, באירופה יש רצועת SRD 868.42 מגה-הרץ, פס 900 מגה-הרץ ISM או 908.42 מגה-הרץ (ארצות הברית), 916 מגה-הרץ בישראל, 919.82 מגה-הרץ בהונג קונג, 921.42 מגה-הרץ באזורי אוסטרליה/ניו זילנד) ו-865.2 מגה-הרץ. בהודו. משמשת להעברת מידע בצורה חשמלית באמצעות גלים אלקטרומגנטיים בתדרים גבוהים (גלי רדיו). ניתן לשדר בשני צורות עיקריות -

פרוטוקול Zigbee

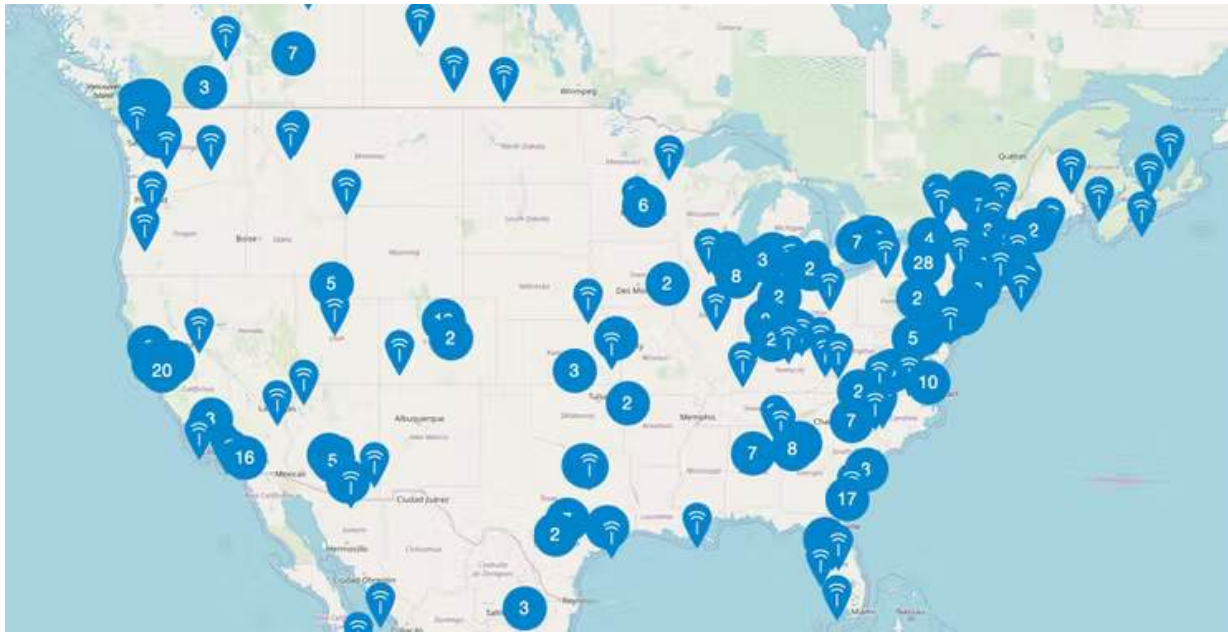
Zigbee הוא חבילה של פרוטוקולי תקשורת ברמה גבוהה המשמשים ליצירת רשתות שטח אישיות עם מכשירי רדיו דיגיטליים קטנים ובעלי הספק נמוך, כגון עבור אוטומציה ביתית, איסוף נתונים של מכשירים רפואיים ועוד דברים אשר צריכת חשמל ומידע (רוחב פס) נמוך המיועדים לפרויקטים בקנה מידה קטן הדורשים חיבור אלחוטי. מכאן ש-Zigbee הוא אופצייה שצריכת החשמל שלה נמוכה, קצב נתונים נמוך אך רק עובדת בקרבה קרובה (10m-100m). הטכנולוגיה המוגדרת על ידי Zigbee נועדה להיות פשוטה יותר וזולה יותר מרשתות תקשורת אישיות אלחוטיות אחרות (WPANs), כגון Bluetooth או רשת אלחוטית כללית יותר כגון Wi-Fi.

רשתות שטח רחבות בצריכת חשמל נמוכה (LPWAN)

מערכת שמאפשרת לשלוח מעט נתונים למרחקים ארוכים בלי הרבה כוח. האופציות העיקריות במרחב הזה הם LoRa ו-Sigfox.

LoRa: קיצור של טווח ארוך (Long Range), הוא פרוטוקול תקשורת סטנדרטי המשתמש בתדרים של 900 ו-400 מגה-הרץ כדי לשלוח נתונים בגובה של כמה קילומטרים. מודולי LoRa הם זולים וקלים לשילוב עם מיקרו-בקרים כגון התקני Arduino או מחשבים כמו Raspberry Pi.

חברות מסוימות, כמו The Things Network, מתחזקות ומפעילות סדרה של שערים מחוברים לאינטרנט המאפשרים לך לחבר את מכשירי ה-LoRa שלך לאינטרנט. שערים אלו דלילים יחסית, עם זאת, ומציעים חלק קטן מהכיסוי של רשתות סלולריות.



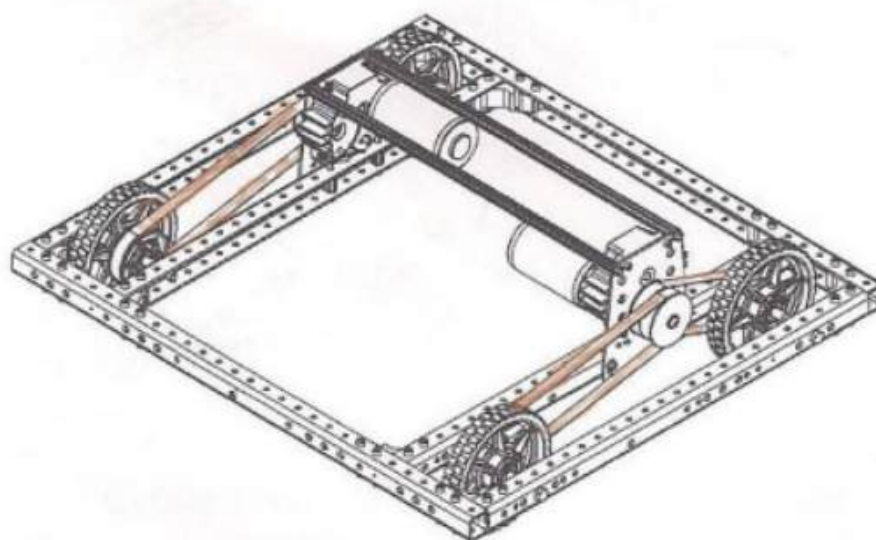
Sigfox: הוא תקן LPWAN נוסף שמשמש גם בתדר 900 MHz להעברת מידע. לא כמו LoRa, רשת Sigfox מרוכזת סביב שערים המחוברים לאינטרנט. מודולי Sigfox נוטים להיות יקרים יותר ממודולי LoRa אך מציעים יותר פיצ'רים.

פתרונות חלופיים

תת מערכת ראשונה- מערכת הנעה ניר לונדון

פתרון א: הנעת 4X4

סקיצה



מבנה

מנוע ותמסורת (אם צריך) ל 2 גלגלים (שמאל ימין) 2 מניעים ישירות מהתמסורת בעזרת שרשרת או רצועת הינע ו 2 מונעים בעזרת גלגל שיניים נוסף בשרשרת או חגורה.

אופן פעולה

אופן פעולת המערכת עובדת כך: הנעה זאת יחסית פשוטה להבנה, המנועה אשר מחובר עליו תמסורת מסתובב ומסובב את גלגל הפולי או ספרוקט היושב עליו. הפולי או הספרוקט מחוברים לגלגלי ההנעה באמצעות רצועת טיימינג או שרשרת ובכך מניעים את המערכת.

יתרונות

1. 4X4 מציעה אחיזה מעולה בתנאי כביש ושטח מאתגרים כגון בוצ', שלג, קרח, חול, סלעים, גבעות תלולות לא אחידות, משטחים מאתגרים ושטחי משיכה נמוכים רבים אחרים.
2. ריכוז משקל בחלק הקדמי או האחורי, מקנה אחיזה טובה כשהחיכוך במשטח נמוך.
3. חלוקת מומנט שווה. אשר תאפשר יציבות וחוזק למערכת מפני קריסות.
4. מערכת כבדה ביחס למערכות הנעה אחרות.

חסרונות

1. משקל נוסף משפר את האחיזה והשליטה, אך הוא גם מגדיל את מרחק הבלימה הנדרש לעצירה מוחלטת. כלי רכב קלים יותר יכולים למנוע התנגשות קל יותר מאשר כלי רכב כבדים יותר.
2. המערכת יקרה בגלל שאנו נצטרך להזמין קרוב ל 90% המרכיבים של המערכת ביניהם (גלגלי גומי, וצירים המחוברים בין הגלגלים).
3. משום שהמערכת נוסעת על גלגלים דבר זה גורם לכך שהמערכת תתקשה יותר לעבור ולנוסע במקומות בעלי מכשולים נמוכים אשר נמצאים לא מאט.

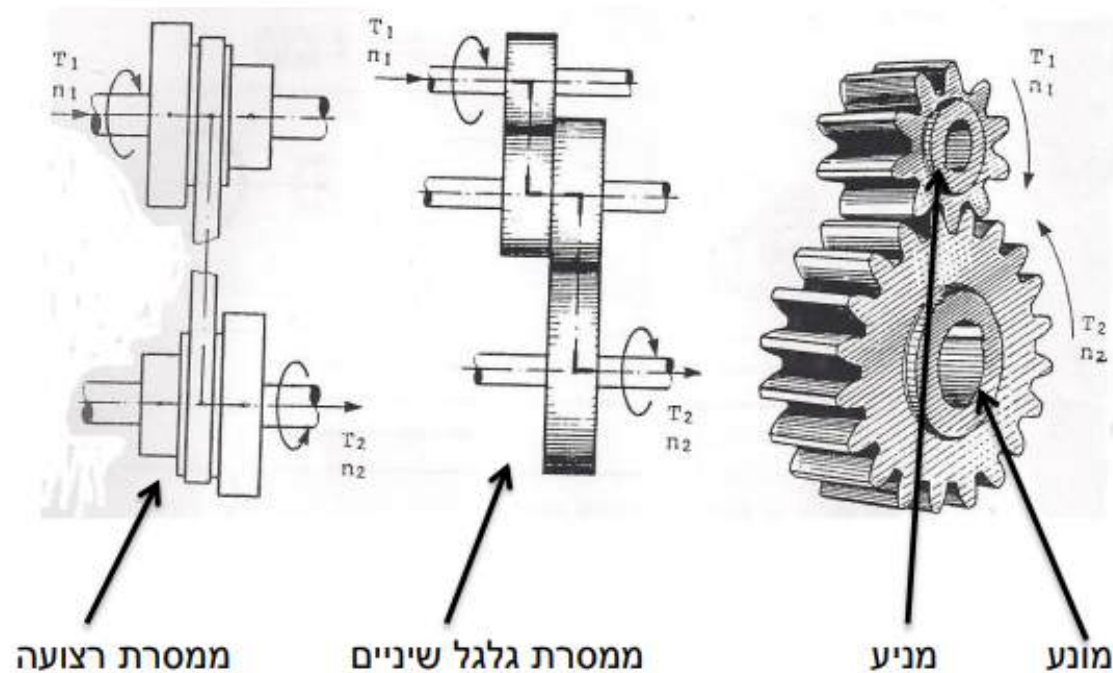
מידע נחוץ ליישום הפתרון

1. מהי תמסורת כיצד היא פועלת, ואיך נרכיב אותה במגמה שלנו?
2. באילו גלגלים נבחר ומדוע. מה עוזר לגלגל להתמודד עם מכשולים ?

1. תמסורת היא מערכת להמרת או להעברת אנרגיה מנקודה אחת לאחרת. תמסורות משמשות ברוב הפעמים להפחתת או הגברת מהירות סיבוב והגדלת הכוח סיבוב של הגלגלים. דבר זה בתבצעה בעזרת גלגלי שיניים המחוברים אחד לשני או בעזרת רצועה. בתמסורת נמצאת יחס מסויים דבר זה נקרה בשם "יחס התמסורת" והוא מבטא את יחס ההמרה של גלגלי השיניים הנמצאים בתוך התמסורת.

התפקידים העיקריים של התמסורת:

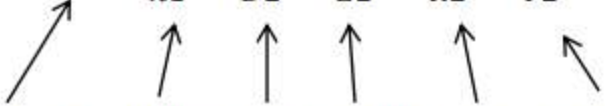
- להעביר תנועה מהמנוע לגלגלים
- לשנות את יחס המסירה.
- להמיר מהירות למומנט וההפך



סוגי יחסי המרה :

- תמסורת הפחתה מקטינות מהירות ומגבירות מומנט. (גלגל קטן מניע גדול מונע).
- תמסורת הגברה מגדילות מהירות ומקטינות מומנט. (גלגל גדול מניע קטן מונע).

$$i = \frac{n_2}{n_1} = \frac{D_1}{D_2} = \frac{Z_1}{Z_2} = \frac{R_1}{R_2} = \frac{T_1}{T_2}$$



יחס מסירה סיבוב קוטר שיניים רדיוס מומנט

- T – מומנט
- I – יחס מסירה
- n – מספר הסיבובים
- D – קוטר הגלגל
- Z – מספר השיניים
- R – רדיוס הגלגל

כיוון שיש לנו תקציב כסף מוגבל ואנו חייבים לקנות דברים שאי אפשר להכין במגמה, נעדיף למדל את גלגלי השיניים בתוכנת סוליד ונחבר אותם לגוף שנמדל בעזרת הדפסה תלת מימדית.

2. הלב של מערכת הנעה היא הגלגלים, הם אלו שיוצרים מגע עם הקרקע, עוברים מעל מכשולים ומיצבים את המערכת ולכן בחירת הגלגלים היא חשובה מאוד. לכן אנו חייבים לבחור את גלגלי המערכת בקפידה כדי למקסם את ביצועי המערכת. ראשית אני חייב להציב לעצמי קריטריונים אשר הגלגלים היו חייבים לעמוד בהם הגלגלים חייבים לעמוד כדי שנבחר אותם.

- על הגלגלים להיות מחוספסים כדי לאפשר חיכוך טוב עם הקרקע ומניעת החלקות.
- על הגלגלים להיות לא גבוהים מידי כדי לאפשר למערכת לעבור דרך מכשולים נמוכים.
- הגלגלים חייבים להיות מחומר קשיח שיוכל לסחוב את משקל המערכת + כל האשפה שאנו נאסוף.

לאחר שחקרתי ודיברתי עם שמוליק מורה המגמה מאיפה אני יכול להזמין את הגלגלים והתקציב שאנו יכולים לסת בו הגעתי למצב בו אני נאלץ לבחור בין שני גלגלים אפשריים.

גלגל סוג א : סוג גלגלים אלו מאוד טובים לנו כיוון שעומדים בכל הקריטריונים והם עשויים ממתכת אשר מספקת עמידה במשקלים כבדים מאוד ורצועה מחוספסת אשר מספקת חיכוך טוב עם הקרעה.

<https://www.swervedrivespecialties.com/products/billet-wheel-4d-x-1-5w-bearing-bore>



גלגל סוג ב : גלגל זה גם הוא טוב מאוד ועומד בכל הקריטריונים שלנו אך הוא עשוי מפלסטיק אמנם הוא עמיד ועבה אך מתכת יותר טוב. בנסוף לגלגל מחובר רצועת גומי שגם היא בעלת חיכוך טוב.
<https://www.vexrobotics.com/traction-wheels.html>



לאחר סיום החקירה על שני המנועים ושיתוף המידע עם המורה שמוליק וחבר הצוות שלי נעם הגענו להחלטה לקנות את הגלגל סוג ב כיוון שגלגל א לא היה במלאי וצפוי להגיע מאוחר מאוד מה שיתקע את כל המערכת.

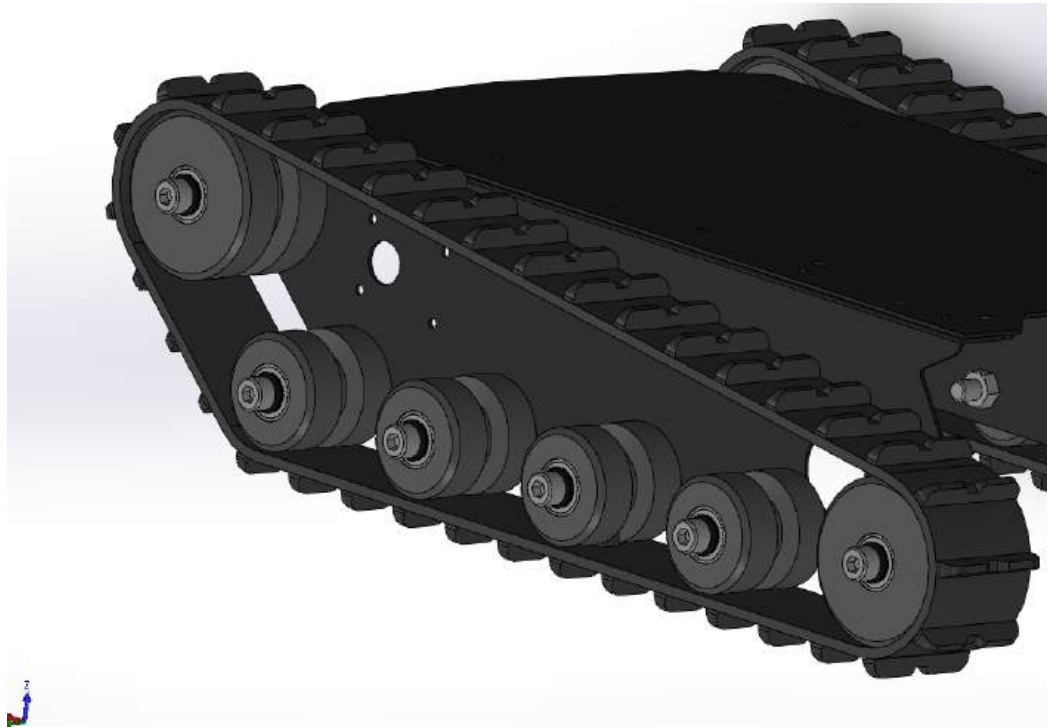
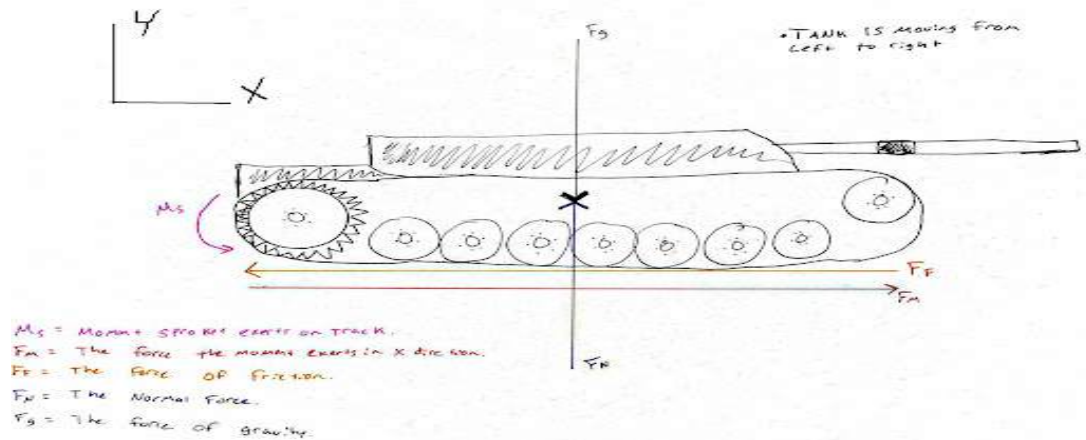
רפלקציה (תאור פתרון)

להערכת, הפתרון תואר בצורה ברורה ומפורטת כך שאדם שלא מכיר אותו יוכל להבין אותו בבירור. כמו כן, אספתי מידע לפי שאלות מנחות, והגענו למסקנה שהוא בהחלט ישים. בנוסף, המידע שאספנו סייע לנו בעיצוב הפתרון ובתכנון הטכני שלו.

לסיכום, הפתרון מציע פתרון מצויין למערכת ההנעה, ובכך יכולה להשתלב ולהיות חלק מהמערכת השלמה. אך למרות זאת אני מחפש יותר אתגר לפרויקט הסיום של כיתה יב ולדעתי פתרון זה לא מקשה עליי כמו שהפתרון השני מקשה לכן אכך זאת בחשבון בעת בחירת הפתרון למערכת.

פתרון ב' (מזקו"ם)

סקיצה



מבנה

המערכת בנוייה ממספר רכיבים. גלגלי פולי מניעים, גלגלים מונעים (מודפסים) 2 מנועים ורצועה מתוחה המחברת בין הגלגלים המונעים ומניעים. גוף המערכת בנוי ממתכת מכופפת או פוליקרבונט אשר תספק למערכת עמידות רבה ויכולת לסחוב משקל כבד. המנוע מחובר בחור מסוג "לנגלוק" המאפשר הזזה של המנוע כדי לאפשר הזזה של מיקום המנוע ולאפשר למתוח את הרצועה במיקום המתוח ביותר.

אופן פעולה

המערכת פעולת בכך שהמנועים מסובבים את גלגלי הפולי היושבים על צירי המנועים (אלו הגלגלים המניעים) אליהם יושב הזחל (רצועה), הרצועה מחברת בין כל גלגלי הפולי המודפסים (גלגלים מניעים) לגלגלי ההנעה ובכך המערכת נעה.

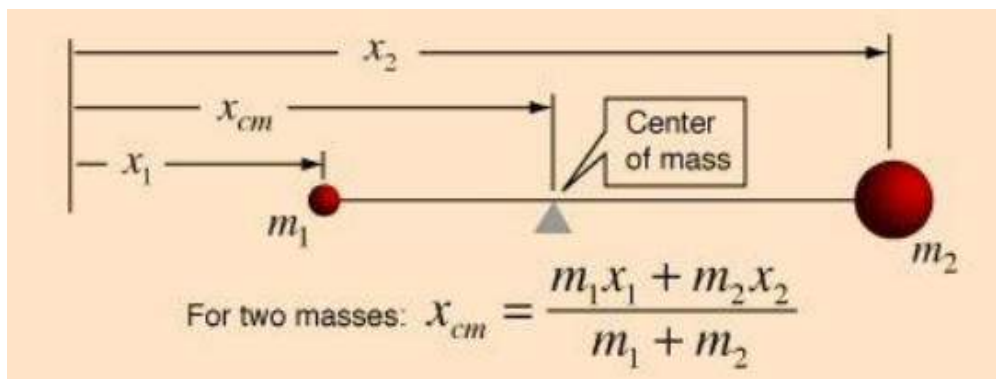
יתרונות

1. חזקה ועמידה - הזחלים מפזרים את משקל המערכת על פני שטח גדול יותר. כתוצאה מכך, הלחץ שהרכב מפעיל על הקרקע מופחת מאוד, מה שמאפשר לשאת יותר משקל על המערכת.
2. הזחל מקנה אחיזת קרקע טובה יותר ובכך הם מגדילים את כוח הגרירה של הכלי. תכונה זו מאפשרת להם לנצל את משקלם והחיכוך עם הקרקע כדי לדחוף ולגרור עצמים כבדים.
3. חומרי המערכת נגישים - החנות שממנה אנחנו קונים את הרצועה היא קרובה אלינו והמחירים לא גבוהים וגלגלי הפולי מודפסים במדפסת תלת מיימד במגמה.
4. נמוחה - המערכת לא גבוהה ומאפשרת לרובוט להיות נמוך יותר כדי שיוכל לעבור דרך מכשולים גבוהים

1. מהירות - הזחל מגביל את מהירותו של כלי הרכב בהשוואה לגלגלים.
2. קריעת הרצועה - משום שאנו משתמשים ברצועה מסוג טיימינג אשר לא מיועדת למטרה זאת הם עלולות להיקרע באט נסיעה בשטח.
3. שחיקת גלגלי הפולי - באט נסיעה ארוכה, כיוון שאנו משתמשים בגלגלי פולי מודפסים עם עלולים להתחמם על ידי הרצועה ובכך לגרום לרצועה לצאת מהמקום.

1. איך נחשב את מרכז הכובד של המערכת כדי שלא תקרוס ?
2. מה ההבדל בין סוגי הרצועות (GT2, HTD) ובאיזה אחת נבחר ?
3. כיצד נחשב את האורך של רצועת הטיימינג כדי למקסם את המתיחות ?

1. בגופים סימטריים מרכז הכובד נמצא בנקודה בה נחתכים קווי הסימטריה(קווי הציר), המרחק האופקי של נקודה זו מציר הסיבוב קובע את המומנט של משקל כל הגוף סביב ציר סיבוב נתון, למרות שהמשקל מפוזר במרחקים שונים מהציר. נקודה זו נקראת מרכז הכובד של גוף. מבחינת חישובי מומנטים ניתן להתייחס לנקודה זו כאילו המשקל של כל הגוף מרוכז בה. בגופים אסימטריים חיתוך האנכים עובר דרך מרכז הכובד. - שווי משקל יציב- מרכז הכובד במצב הנמוך ביותר. - שווי משקל רופף- מרכז הכובד במצב הגבוה ביותר - שווי משקל אדיש- מרכז הכובד באותו גובה בכל מצבי הגוף.



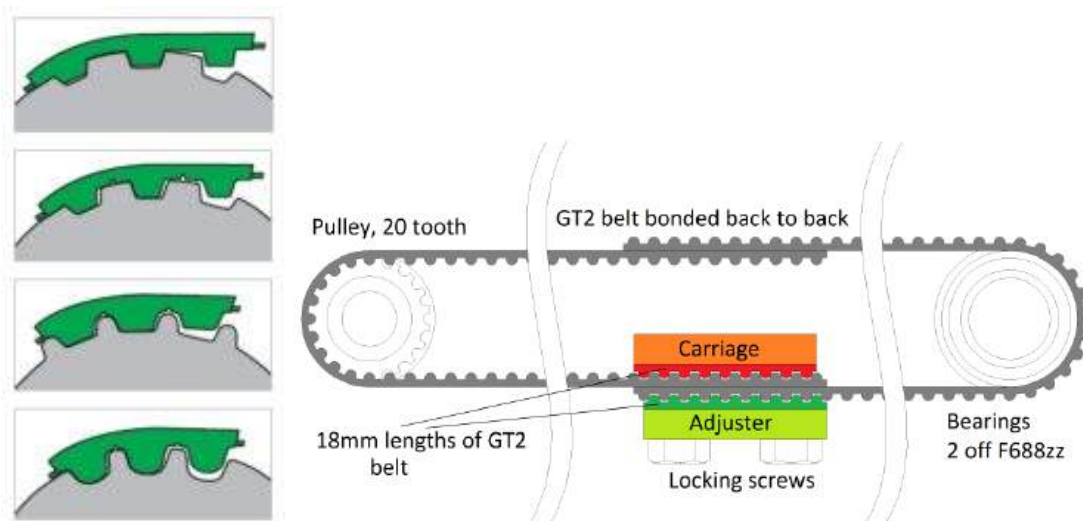
$$\bullet \text{ המרחק } X = \frac{m_2}{m_1 + m_2} x \text{ מהמסה, } m_1$$

$$\bullet \text{ ובמרחק } X = \frac{m_1}{m_1 + m_2} x \text{ מהמסה } m_2$$

2 סוגי הרצועות (GT2, HTD) והבדלם:

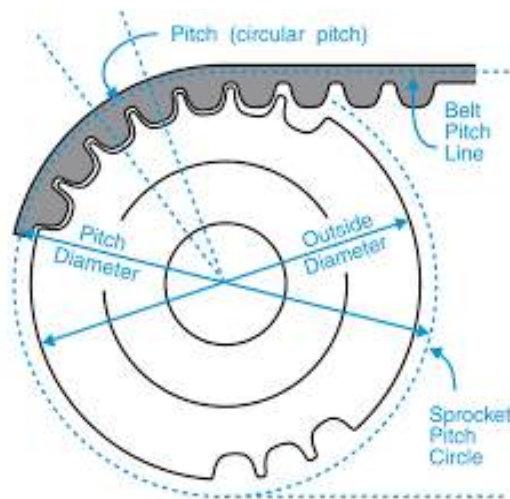
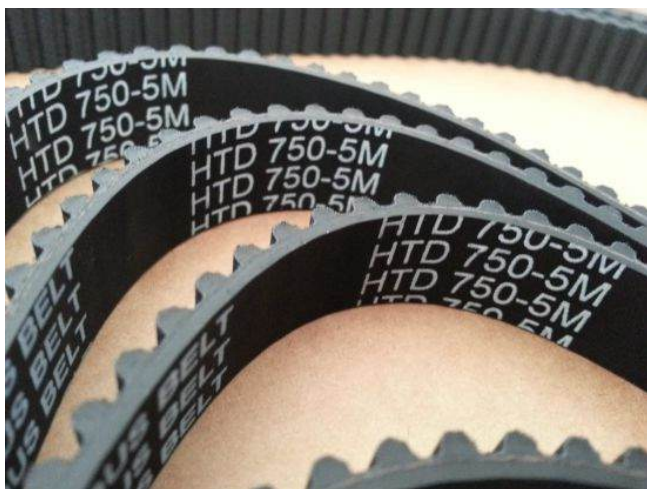
סדרת חגורות GT2 תוכננו במיוחד לתנועה ליניארית. הם משתמשים בפרופיל שן מעוגל, עם שיפוע של 2 מ"מ, המבטיח ששן הרצועה משתלבת בצורה חלקה ומדויקת בחריץ הגלגלת, כך שכאשר הופכים את כיוון הגלגלת, אין מקום לרצועה לנוע בחריץ מה שמאפשר הגנה מפני קפיצה ועצירת המערכת.

הרצועה מיועדות ליישומי עומס נמוכים יותר כגון ההפחתה הראשונה של גירבוקס או מנוע לגלגל תנופה. חגורות GT2 בעלת פיתך (Pitch) (מרחק בין מרכז שן לשן שאחריה) קטן יותר מאשר חגורות HTD מה שאומר שאפשר להשתמש בהפחתות גדולות יותר באזורים tight אלה של הרובוט שלך.



רצועות HTD (ראשי תיבות של High Torque Drive) הם רצועות בעלות שיניים עקמומיות, המשפרת את קיבולת המומנט בהשוואה לתזמון של רצועות GT2. רצועות HTD ידועות ברמה הגבוהה שלהם בהגנה מפני דילוג שיניים ודיוק בתזמון. פרופיל השן העגול-מוקשתת מספקת התנגדות מעולה לקפיצות

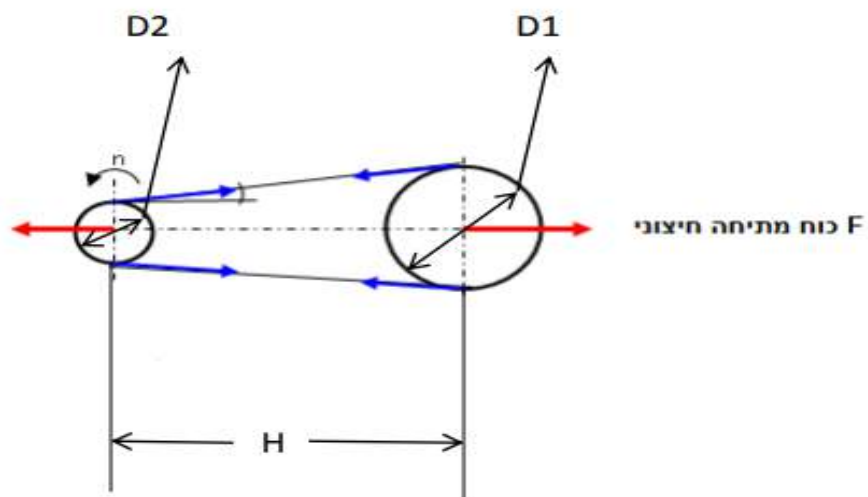
שיניים ומספקת חיבור חלק של השיניים לגלגלי השיניים/מנועה. בנוסף רמת הרעש מופחתת להפליא בהשוואה לחגורות הטרפז הקלאסיות.



יתרונות רצועת טיימינג:

1. פעולה שקטה
2. גמישות הרצועה מונעת תנודות וזעזועים
3. מחיר נמוך
4. הרכבה קלה יחסית

3. כיצד נחשב את האורך של רצועת הטיימניג כדי למקסם את המתיחות?



$$L = 2H + \frac{(D1 - D2)^2}{4C} + 1.57 (D1 - D2)$$

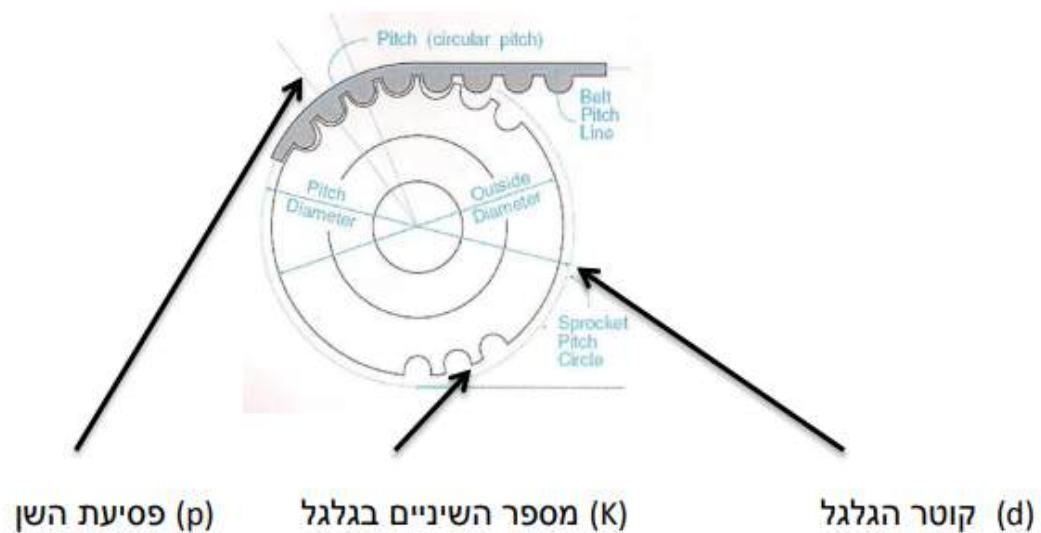
L – אורך הרצועה

H – המרחק בין שני גלגלי התזמון

D1 – קוטר הגלגל הגדול

D2 – קוטר הגלגל הקטן

נוסחה נוספת פשוטה יותר:



$$H = P \cdot \left(\frac{D}{2} \right) - k$$

P – פסיעת השן

K – מספר השיניים בגלגל

D – קוטר גלגל השיניים

H – מרחק בין הצירים

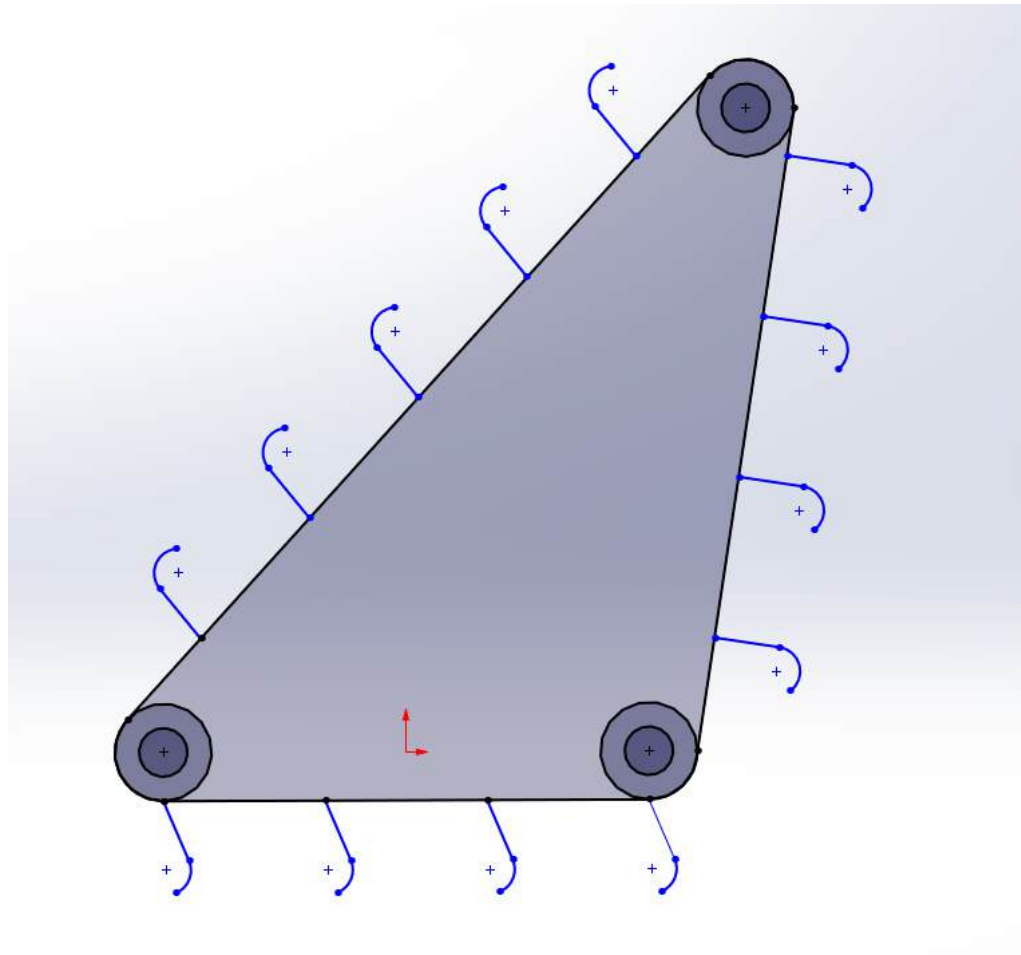
רפלקציה (תאור פתרון)

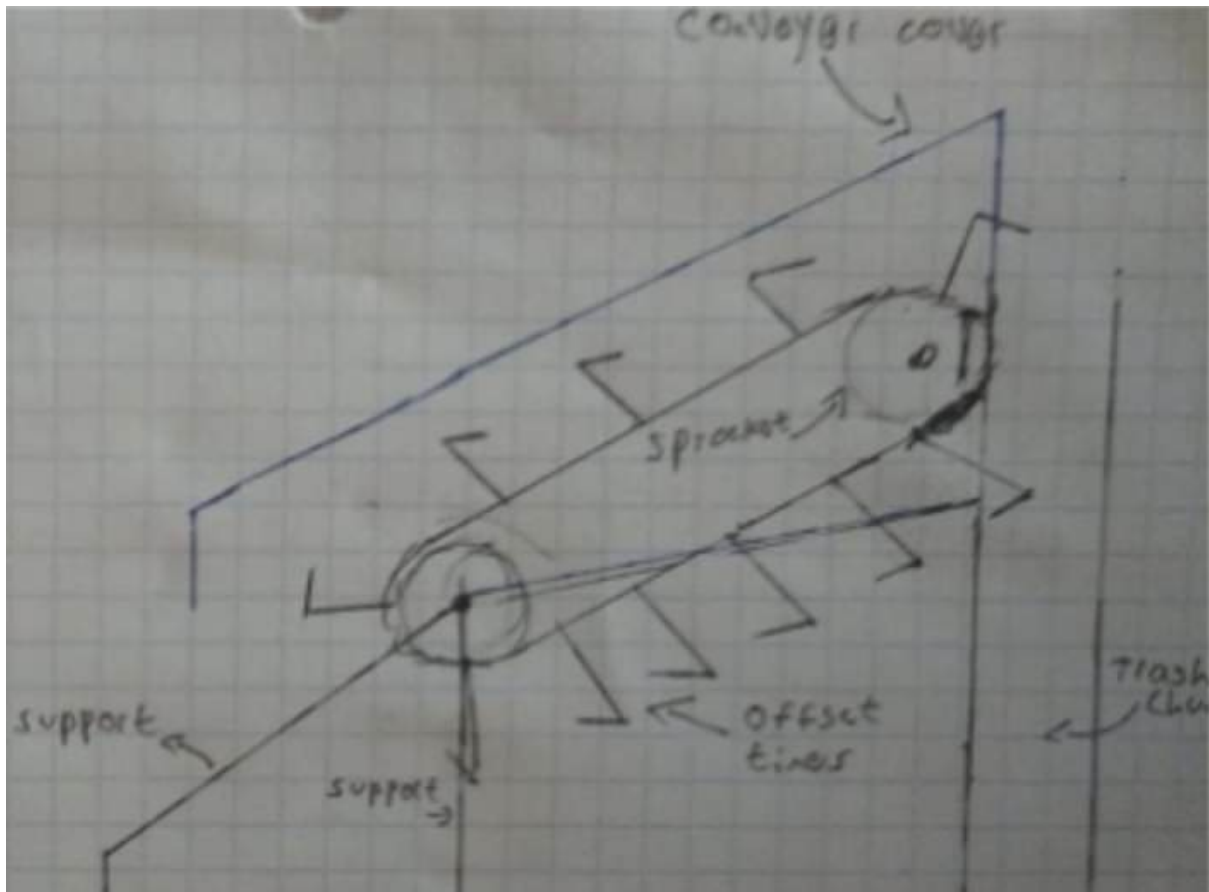
הפתרון מתואר בצורה טובה וברורה. הפתרון אכן מתאים כפתרון למערכת ההנעה מכיוון שהוא נותן מענה מלא לכל הדרישות שאנחנו דורשים שהרובוט יבצע מהמערכת השלמה. המידע שנאסף עזר לנו להבין איך לממש אותו בצורה הטובה ביותר במידה ונבחר בו. הפתרון מסקרן ומפחיד באותו הזמן מה שגורם לי לתחושת התרגשות ונכונות להוכיח לעצמי שאני מסוגל לפתח את המערכת הזאת ולהקים אותה לחיים!

פתרונות חלופיים

תת מערכת שנייה-מערכות איסוף נועם אלרון
פתרון א' (גריפה מכנית)

סקיצה





מבנה

המערכת מורכבת ממסוע, שבא גלגלת ראש (Driver pulley), גלגלת זנב (Tail/Driven pulley), מסוע כדי לתמוך בחומר הלוח, גלגלי מתח (Snub pulley) המיועד לעזור עם צמידות החגורה לפולי, חגורה, שיניים (Tines) על החגורה, גרר, דיפלקטור ופיר.

המערכת הינה בנויה משלד אלומיניום, בקדימה של השלד ישנו גרר מצד אחד ומצד השני של הגרר דיפלקטור, ישירות מאחרי הדיפלקטור ממוקם המסוע, מאחורי החלק העליון של המסוע ישנו פיר. הגלגלת ראש מחוברת ציר שמחובר לתמסורת (גירבוקס) כוח, שאליו מחובר המנוע.

אופן פעולה

הפעלה כללית של כל המערכת

כאשר אדם מפעיל את המערכת, מערכת מתחילה עם זיהוי מיקומו, את סביבתה, ומתחילה לסרוק לאובייקטים רלוונטים. עד שאובייקטים אלו לא זוהו, הוא ינוע לכיוון מסוים, ברגע שאובייקטים מתאימים זוהו המערכת תבדוק את המרחק שלו מהם ותחשב את מיקומם. אם ישנם מספר אובייקטים מפוזרים, המערכת תנוע לאובייקט הקרוב, תוך כדי שמירה על הנתונים שלהם כדי שיוכל להמשיך לעברם. כאשר זוהה אובייקט, המערכת תחל לנוע.

אופן פעולת מערכת המגרפה המכנית

כאשר המערכת נעה, הגרר (moldboard) של המערכת יפלט אזורים לא אחידים בחול כדי להבטיח משטח אחיד לניקוי וכדי לשלוט בעומק הניקוי וגם משמש למטרת דחיית אובייקטים לא רצויים. על המסוע מכוסה רצועה ישנם מספר רב מאוד של "שיניים", שיניים אלו גורפות פסולת מהחול ומפקידה אותו על גבי המסוע חשוב לציין שקיימים רווחים בין השיניים, כך שהם לא לגמרי סוחבים את האובייקטים, אלה מעיפים אותם לכיוון הדיפלקטור, "העוטף" את המסוע וכך מקיימים מעין תנועת "פינג פונג" עד שהאובייקט מגיע לחלק העליון של המסוע, ומופקד לתוך הפיר אשפה. בתוך הפיר ישנו חיישן נפח, שכאשר נפח האשפה מגיע לנפח מוגדר, תשלח אות פקודה למערכת, שיגרום לו לנסוע לאזור המקוצב להורדת אשפה.

יתרונות

1. מהירות: מנקי חופים שמשתמשים בגריפת שיניים מכנית מסוגלים לנקות בדרך כלל במהירויות הרבה יותר מהירות מאשר כל אופציה אחרת, מכיוון שהם מסירים את החומרים ישירות מהחול, במקום קודם להסיר את החול מהחוף, להעביר אותו דרך מסכים כדי לסנן את החומרים, ואז להחזיר את החול לחוף.
2. ניקוי קו הגאות: למרות שמגרפה מכנית עובדת בצורה מיטבית בחול יבש ורטוב, הוא יכול גם לנקות בחול רטוב, מבלי לאסוף חול עם הפסולת או להיסתם.
3. דורש תחזוקה לעיתים מאוד רחוקות.
4. ידידות סביבתית: לא גורם לשחיקה חופית.
5. ניקוי כל דבר בלי להיסתם או לצרוך ניקוי למשל סרגסום ואצות.

חסרונות

1. דורש שיקולים מכניים רבים.
2. כאשר נדרש תחזוקה, נדרש תחזוקה מסובכת.
3. דורש חגורה יקרה/קשה ליצור.

מידע נחוץ ליישום הפתרון

1. איזה מערכת כזו כבר קיימת?
2. איזה סוג פסולת המערכת יכולה לאסוף?
3. מה עיקרון פעולה של מסוע?
4. כיצד ניתן לשנות את גובה ה-moldboard (מגרף)?
5. איזה מערכת תקשורת נשתמש?

1. מערכות קיימות

H-Barber הינה חברה בינלאומית שקיימת משנות ה-60 העוסקת באיסוף פסולת, יש לה מגוון מערכות קיימות שביניהם מגרפה מכנית. יש להם שלושה מערכות של מגרפה מכנית, שכל אחד שיפור על השני, המודל החדשני ביותר שלהם הינו 600HD. אמנם המערכת שלהם אינה אוטונומית-היא מובלת על ידי טרקטור, היכולת איסוף שלה מדהים. ה-600HD ששוקל 1800 ק"ג, הוא מגרפת החוף הגדולה והטובה ביותר שמציע H. Barber והמנקה החופים הנפוץ ביותר על ידי עיריות לניקוי חופים רחבי ידיים. הוא מנקה עד 32000 מ"ר בשעה עם רוחב ניקוי של 2.14 מטר, הוא יכול לשאת עד 2400 ק"ג של חומר. כמערכת יכולה לאסוף מגוון רב של דברים כגון: כל סוג אשפה נפוצה, אצות, אבנים, זפת, אלמוגים ועוד.



סרטון פעולה של מגרפה מכנית- קישור לסרטון



2. סוגי הפסולת שהמערכת תצטרך לאסוף

רצוננו הוא שהמערכת שלנו תוכל לאסוף סיגריות, בקבוקים(פלסטיק וזכוכית), בד, נייר, עץ, זכוכית, קרטון ופלסטיק.

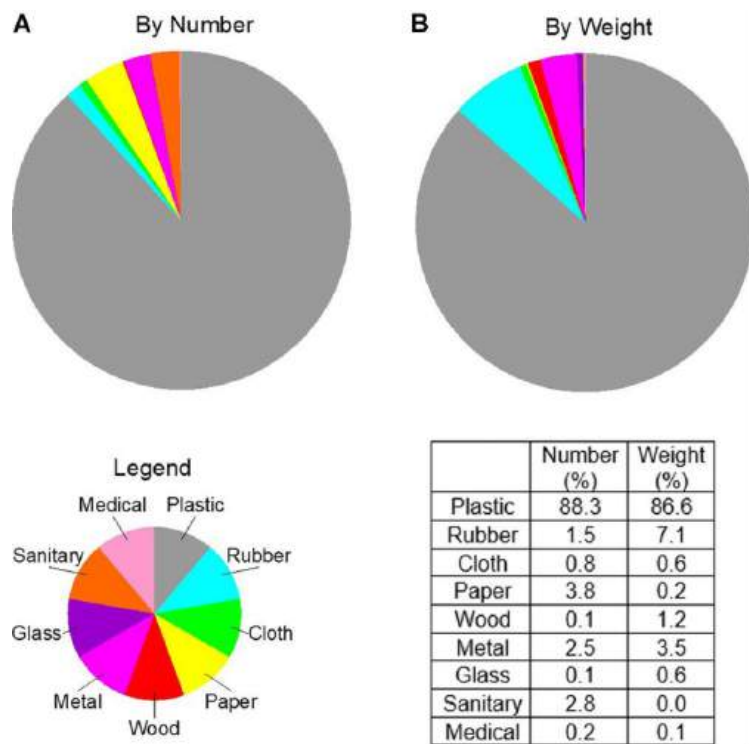
לפי מחקר סקוטי המשקל הממוצע של פריט פסולת פלסטיק היה 15.6 גרם (משקל חציוני 2.5 גרם). עבור סוגי פסולת שאינם מפלסטיק, הכבד ביותר היה עץ, עם משקל פריט אשפה ממוצע של 154.7 גרם. הזכוכית הייתה באותו סדר גודל (150 גרם).

טבלה 1

	Total Number	Total Weight (g)	Percent by Number	Percent by Weight	Min (g)	Max (g)	Median (g)	Mean (g)	Stddev (g)	np/100 m	g/100 m
All	1,618	25,797	100.0	100.0	0.01	2,600.0	2.4	15.9	81	20.2	322.5
Plastic	1,428	22,329	88.3	86.6	0.01	2,600.0	2.5	15.6	84	17.9	279.1
Non-Plastic	190	3,468	11.7	13.4	0.01	404.0	0.2	18.3	47	2.4	43.4
Rubber	24	1,843	1.5	7.1	4.90	404.0	54.8	76.8	91	0.3	23.0
Cloth	13	164	0.8	0.6	1.60	48.7	5.1	12.6	15	0.2	2.1
Paper	62	46	3.8	0.2	0.01	16.0	0.1	0.7	3	0.8	0.6
Wood	2	309	0.1	1.2	9.40	300.0	154.7	154.7	205	0.0	3.9
Metal	40	905	2.5	3.5	0.60	82.8	14.4	22.6	21	0.5	11.3
Glass	1	150	0.1	0.6	-	-	-	150.0	-	0.0	1.9
Pottery	0	-	-	-	-	-	-	-	-	-	-
Sanitary	45	12	2.8	0.0	0.0	2.8	0.2	0.3	0.4	0.6	0.2
Medical	3	38	0.2	0.1	2.0	34.0	2.2	12.7	18	0.0	0.5

Litter types are those defined by OSPAR (2010).

איור 1



במונחים של אחוז הרכב פסולת החוף (איור 1 וטבלה 1), הפלסטיק מהווה כ-87-88% מסך העומס, ללא קשר אם נמדד במספר או במשקל. עם זאת, לפי מספר, גומי, נייר, מתכת ופריטים סניטריים היוו כל אחד כ-2-3% מסך העומס, אך לפי משקל גומי ומתכת שלטו, עם 7.1% ו-3.5%, בהתאמה. תוצאות אלו פשוט מראות שהרכבים שונים הנלמדים מנתוני ניטור המלטה על החוף בהתאם לשאלה אם משתמשים במספר או במשקל כמדד, בשל ההבדל במשקל של פריטי המלטה בודדים בכל אחד מסוגי הפסולת.

3. עקרון פעולה של המסוע במערכת

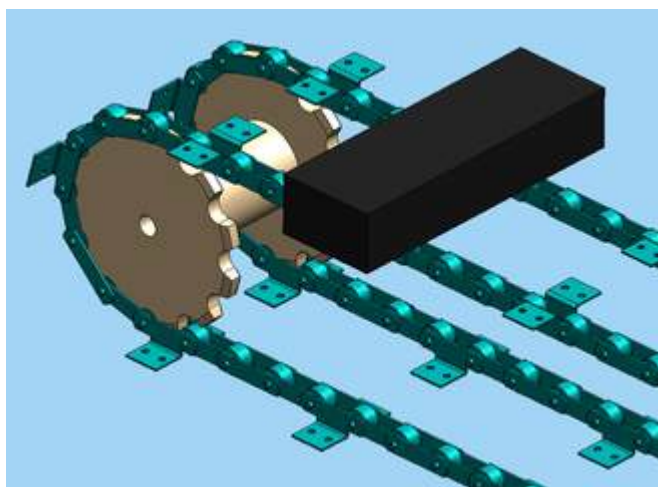
אמנם ישנם סוגים רבים של מערכות מסועים, כולם משרתים את אותה מטרה של הובלת חומרים. בדרך כלל, מערכות מסועים מורכבות מחגורה מתוחה על פני שתי גלגלות או יותר. החגורה יוצרת לולאה סגורה סביב הגלגלות כך שהיא יכולה להסתובב ללא הרף. גלגלת אחת, המכונה גלגלת ההנעה, מניעה או גוררת את הרצועה, ומעבירה פריטים ממקום אחד לאחר.

העיצובים הנפוצים ביותר של מערכת המסוע משתמשים ברוטור כדי להניע את גלגלת ההנעה והרצועה. החגורה נשארת מחוברת לרוטור דרך החיכוך בין שני המשטחים.

למערכת שלנו נשתמש בהתמזגות/שינוי של מסוע גרר, שנקרא גם מסוע תעופה. התוצאה תהיה מסוע משופע, המורכב עליה חגורה עם מספר רב של שיניים כדי "לגרור" חומרים לתוך פיר.

מסועי שרשרת משתמשים בקונספט דומה למסועי חגורה (פשוטים ובעלי יכולות הסתגלות הטובה ביותר. מסוע מתפקד על ידי תנועה של תומך מתמשך, החגורה מורכבת על פלטפורמה שעשויה להיות בה אביזרים שונים (מסילות צד, עצירות קצה, גלגלים וסוגים אחרים של אביזרים פנאומטיים ומכניים)). במקום חגורה, לעומת זאת, הם משתמשים ברשת שרשרת כדי להעביר מוצרים. שרשראות יכולות להיות עשויות מחומרים שונים ועשויות להגיע בגדלים וצורות שונות בהתאם ליישום.

מסועי גרר אינם שונים מרוב מסועים אחרים מבחינת תפעול - מנוע מפעיל שרשרת טיסות חד-גדילית אינסופית או כפולה המושכת או "גוררת" חומר דרך תחתית המסוע. השרשרת יכולה להיות סגורה חלקית או מלאה, אבל בכל המקרים, חייבת להיות צורה כלשהי של דיור כדי לאפשר גרירת חומר על ידי הטיסות לפיר/יציאה. טיסות אלו, הידועות גם כמשוטים או מגרדים, או כל כלי שנתון אופציית גרירה, נבדלות בכך שהן יכולות לנצל עד 50% מהשטח הפנוי בתוך המסוע לתנועת חומר, במקום 20-45% טיפוסיים של סוגי מסועים אחרים. לכן, הם מסוגלים להעביר יותר חומר בפחות כוח, ובדרך כלל מהווים אפשרות שינוע יעילה.



לאלמנטים בתוך המסוע ישנם מספר שיקולים:

משקל: המשקל קובע מספר אלמנטים שונים של מערכת המסוע, כגון קוטר הגלילים, דרישות מד רולר, סוג החומר למיטה רולר ועוצמת המנוע, בין היתר. אם המוצרים כבדים, יש צורך למבנה מסוע יציב עם מנוע חזק, במיוחד אם מניעים את המוצרים נגד כוח המשיכה או אופקית, כמו במקרה שלנו. גודל אובייקטים המובלים: יש צורך לשקול את האורך, הרוחב והגובה של האובייקטים כדי לקבוע אם הם יתאימו על המסוע. מוצרים גדולים יותר זקוקים למסועים עם חגורות רחבות יותר, מפרט מסילה גבוה יותר ומרכזי גלילה גדולים יותר.

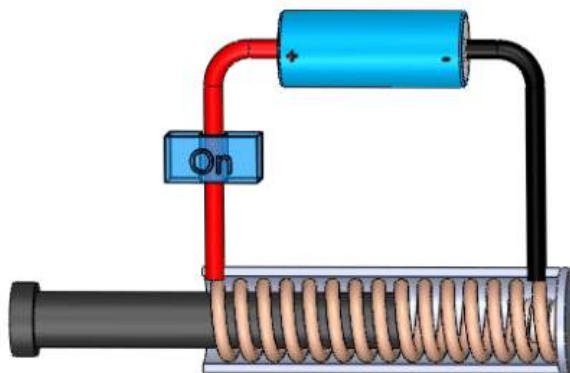
4. איך אפשר בעצם לשנות את גובה ה-moldboard (מגרף)?

כדי לשנות גובה של דברים במערכות מכניות יש צורך להשתמש במפעילים (actuators). מפעילים משרתים את המטרה הכללית של שליטה בתנועות בתוך מכונות. עם זאת, ישנם סוגים שונים של מפעילים המייצרים תנועות משתנות ומשתמשים במקורות כוח שונים. דבר ראשון, מה הוא מפעיל בעצם? מפעיל הוא חלק מכני שגורם לתנועות חלקים במערכת על ידי קבלת משוב מאות בקרה. ברגע שיש לו כוח, המפעיל יוצר תנועות ספציפיות בהתאם למטרת המכונה. מפעילים יכולים לגרום לשני סוגים של תנועה, לינארית, וסיבובית. מפעילים לפי סוג תנועה:

מפעילים לינאריים: משתמם בשמם, מפעילים ליניאריים הם מכשירים שמייצרים תנועה בנתיב ישר. הם יכולים להיות מכניים או חשמליים והם נראים בעיקר במכשירים הידראוליים או פנאומטיים. לכל מכונה, ציוד או גאדג'ט שדורשים צורה כלשהי של תנועה ישירה יש בדרך כלל מפעיל ליניארי.

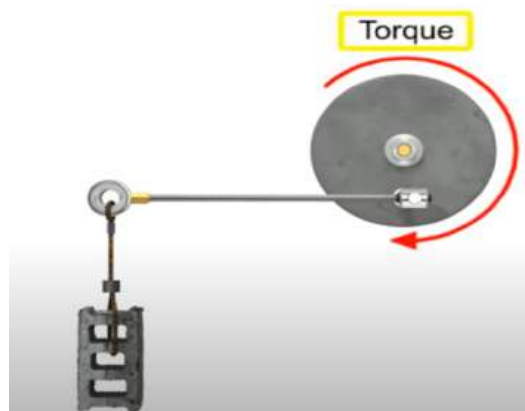
במפעיל ליניארי פשוט, יש אום, כיסוי וצינור הזזה. צינור ההזזה מספק את המרחב לתנועה, ואילו האום והכיסוי מספקים את התנועה המשתלבת השומרת על המפעיל בנתיב ישר. למפעילים ליניאריים

מורכבים אחרים יהיו חלקים נוספים, אך המערכת שהוזכרה לעיל היא הבסיס לתנועה ישרה.



מפעילים סיבוביים: בניגוד למפעילים ליניאריים, מפעילים סיבוביים יוצרים תנועה מעגלית. הם משמשים לעתים קרובות בשילוב עם מפעיל ליניארי אם מכונה דורשת לנוע קדימה, אחורה, למעלה או למטה.

מפעילים סיבוביים רבים מהם מופעלים חשמלית, אך חלקם מופעלים באמצעות מערכת הידראולית או

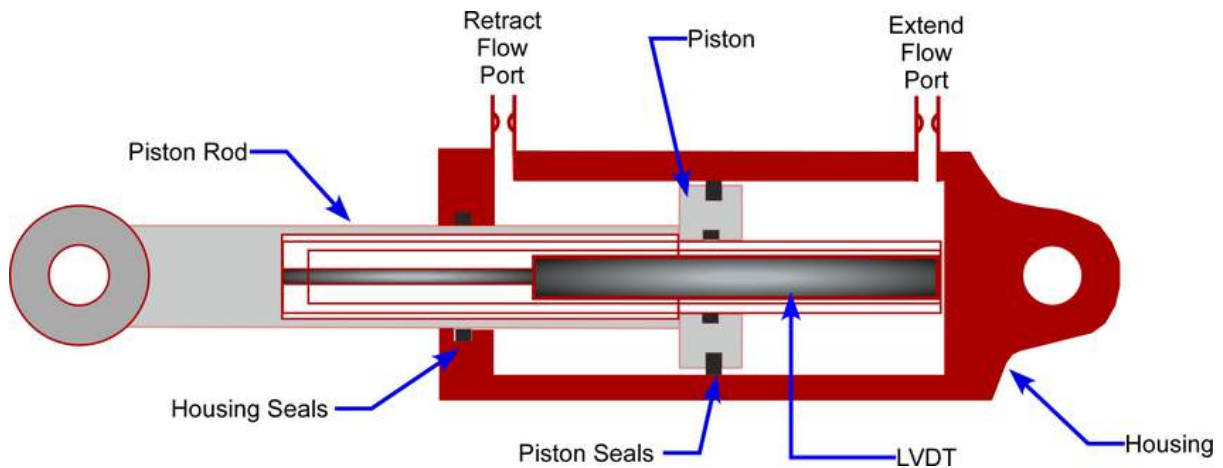


פנאומטית. אפשר יכול למצוא מפעילים סיבוביים במגבי שמשות, מאווררים חשמליים או מכונות ייצור המעבירות סחורות מאזור אחד למשנהו.

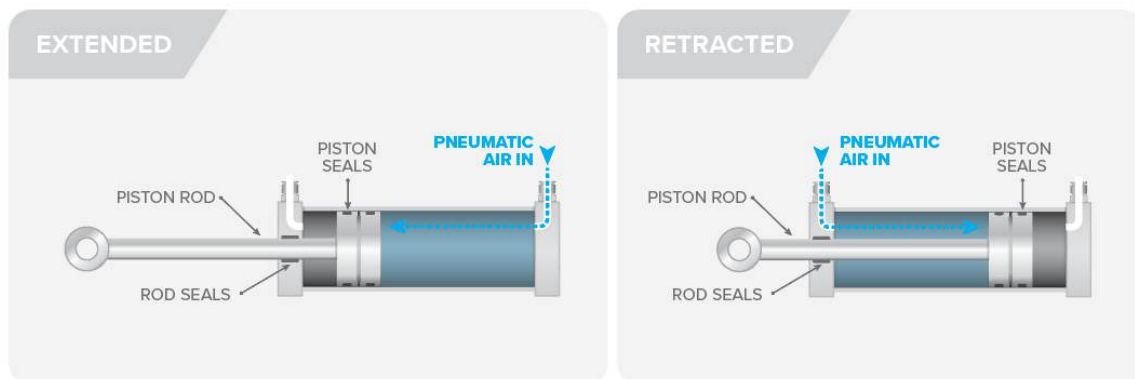
מפעילים לפי סוג כוח פועל:

מפעילים הידראוליים: מפעילים הידראוליים פועלים באמצעות צילינדר מלא בנוזל עם בוכנה תלויה במרכז. בדרך כלל, מפעילים הידראוליים מייצרים תנועות לינאריות, וקפיץ מחובר לקצה אחד כחלק

מתנועת החזרה. מפעילים אלו נראים באופן נרחב בצידוד כושר כגון צעדים או מנשאי הובלה לרכב.

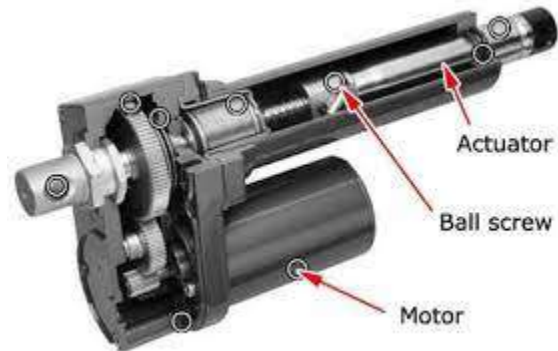


מפעילים פנאומטיים: מפעילים פנאומטיים הם אחת האפשרויות האמינות ביותר לתנועת מכונה. הם משתמשים בגזים בלחץ כדי ליצור תנועה מכנית. חברות רבות מעדיפות מפעילים מופעלים פנאומטיים מכיוון שהם יכולים לבצע תנועות מדויקות מאוד, במיוחד בעת התנעה ועצירה של מכונה. דוגמאות לצידוד המשתמש במפעילים פנאומטיים כוללים: בלמי אוטובוס, מכשירי כושר מנועי שבשבת, חיישני לחץ ועוד.

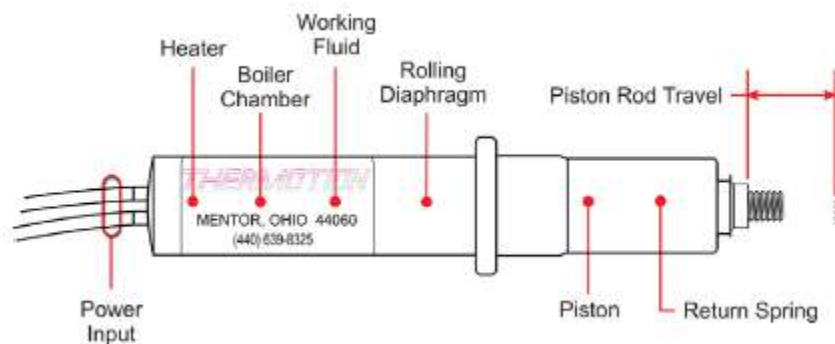


מפעילים חשמליים: מפעילים שדורשים חשמל כדי לעבוד. דוגמאות ידועות כוללות מכונות חשמליות, מכונות ייצור וציוד רובוטיקה. בדומה למפעילים פנאומטיים, הם גם יוצרים תנועה מדויקת מכיוון שזרימת הכוח החשמלי קבועה. הסוגים השונים של מפעילים חשמליים כוללים: מפעילים אלקטרו-מכניים: מפעילים אלו ממירים אותות חשמליים לתנועות סיבוביות או ליניאריות ואף עשויים להיות מסוגלים לשילוב של שניהם.

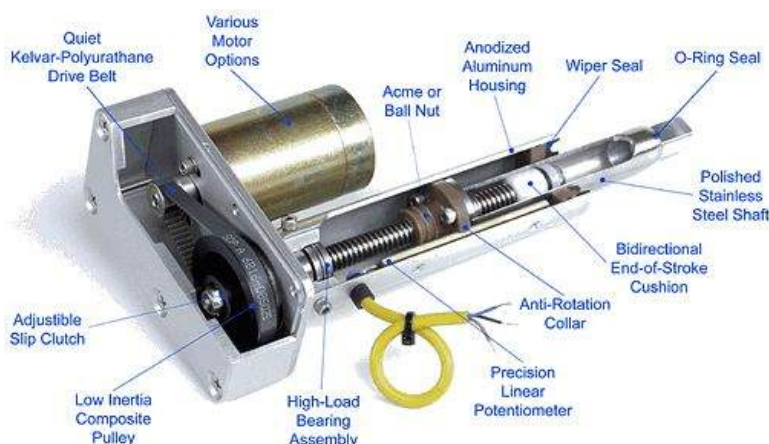
מפעילים אלקטרו הידראוליים: מפעילים מסוג זה מופעלים גם הם חשמלית אך נותנים תנועה למצבר הידראולי. לאחר מכן, המצבר מספק את הכוח לתנועה, הנראה בדרך כלל בציוד תעשייתי כבד



מפעילים תרמיים ומגנטיים: מורכבים בדרך כלל מסגסוגות של זיכרון צורות שניתן לחמם כדי לייצר תנועה. התנועה של מפעילים תרמיים או מגנטיים נובעת לרוב מאפקט ג'ול, אך היא יכולה להתרחש גם כאשר סליל ממוקם בשדה מגנטי סטטי. השדה המגנטי גורם לתנועה מתמדת הנקראת כוח לפלס-לורנץ. רוב המפעילים התרמיים והמגנטיים יכולים לייצר טווח תנועה רחב ועוצמתי תוך שמירה על קל משקל.



מפעילים מכניים: חלק מהמפעילים הם בעיקר מכניים, כגון גלגלות או מערכות מתלים (Rack and pinion). כוח מכני נוסף מופעל, כגון משיכה או דחיפה, והמפעיל ימנף את התנועה הבודדת הזו כדי להפיק את התוצאות הרצויות. לדוגמה, סיבוב הילוך בודד על סט של מתלים יכול לגייס חפץ מנקודה A



Mechanical Actuator

➤ Mechanical linear actuators typically operate by conversion of rotary motion into linear motion.

Examples

- Screw-jack
- Wheel and axle
- Cam



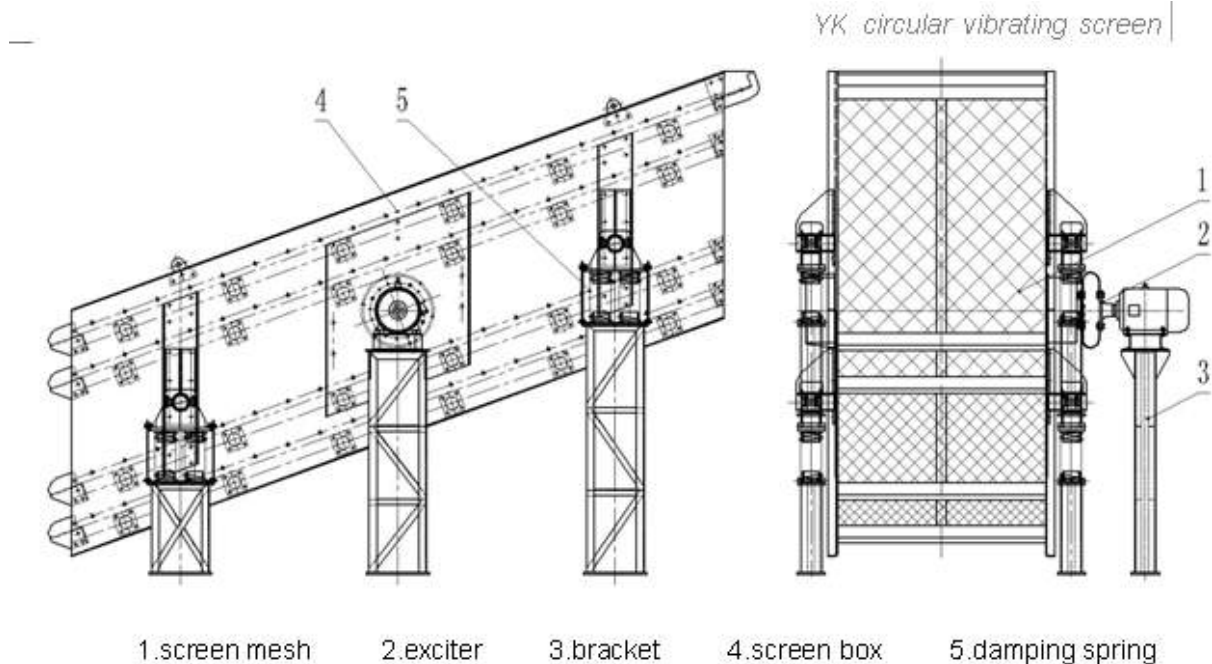
לנקודה B. תנועת המשיכה המופעלת על הגלגלת יכולה להביא את הצד השני כלפי מעלה או לעבר המיקום הרצוי.

רפלקציה (תיאור פתרון)

לדעתי הפתרון המוצע הינו תואר בצורה מובנת, הדרך שבו היא עובדת הינה מסובכת אבל לדעתי הצלחתי להעביר אותה בדרך קלה להבין לאדם שלא מכיר. למרות שהפרויקט קשה ליישום, אני חושב שהוא בהחלט אפשרי, ישנו הרבה מידע שאספתי לכמעט כל הנדרש כדי ליישם את הפרויקט, ודבר היחיד הנדרש כדי ליישם פיזית הוא להחליט על מימדים של המערכת ולחשב משם את כל הדברים הנדרשים. המערכת עצמה הינה תת מערכת למערכת השלמה שלנו.

פתרון ב' (מסנן רוטט)

סקיצה



מבנה

המסנן הרוטט מורכב מחרב איסוף, מנגנון לינארי מחובר לקפיצים, מנגנון רטט, רשת סינון מרוטט ופיר איסוף.

אופן פעולה

החול והפסולת נאספים באמצעות להב איסוף על גבי חגורת סינון, המאפשרת לחול לעבור דרכו דרך החומרים אך לא מאפשרת לחומרים האחרים לעבור דרכו (גודל החומרים שהוסרו נקבע לפי גודל החורים במסך המותקן). במערכת ישנו מנגנון רטט (שכנראה מובנה ממנגנון לינארי) שמחובר לרשת וגורמת לה לרטט. המסך הרוטט ביחד עם התנועה של המערכת, ומבנה המערכת (שיפוע קל) כולה גורם לכך שהפסולת לאט לאט זזה אחורה עד שבסופו של דבר האובייקטים נופלים לתוך פיר איסוף אשר הממוקם בחלק האחורי של הרכב. מכיוון שהחול והפסולת מועלים אל חגורת הסינון, המסננים חייבים לתת זמן לחול לנפות דרך המסך חזרה אל החוף כדי שלא יוצר עומס/פיגור.

יתרונות

1. קומפקטי: מאוד קרוב לאדמה, הגודל של כל המערכת קטנה מאוד ביחס לשיטות אחרות
2. מאוזן: אין בעיות של איזון מכיוון שלא צריך לבנות לגובה
3. איסוף אשפה קטנה אפקטיבית: אוסף כל סוג אשפה קטנה, רק תלוי בגודל החורים ברשת

חסרונות

1. משקל וגודל: לא יכול לאסוף אשפה כבדה במיוחד או גדולה במיוחד- עלול לקרוא את הרשת או לא להיכנס כלל למערכת
2. Backlog: עלול ליצור מצב שיותר מדי חול או פסולת לפני שמספיק לרדת לפיר או לעבור דרך הרשת, מצב שעלול לגרום למערכת להיתקע.
3. קושי יישום: מאוד קשה ליישם את המנגנון של האיסוף האשפה על גבי הלהט.

מידע נחוץ ליישום הפתרון

1. איך אפשר לגרום לרשת לרטט?
2. איזה סוגי פסולת המערכת תוכל לאסוף?
3. איזה מערכת תקשורת נשתמש?

1. רשת רוטטת

מנוע רטט: מנוע רטט הוא מנוע DC נטול ליבות בגודל קומפקטי שהיה רגיל אליו מודיע למשתמשים על קבלת האות על ידי רטט, ללא קול. מנועי רטט הם משומשים במגוון יישומים כולל טלפונים סלולריים, מכשירי טלפון, מכונות, מפעלים ועוד.

התכונות של מנוע רטט הוא מנוע DC שהמגנט קבוע, מה שאומר שתמיד יהיה לו את התכונות המגנטיות שלו (בניגוד לאלקטרומגנט, שמתנהג רק כמו מגנט כאשר עובר בו זרם חשמלי); תכונה עיקרית נוספת היא גודל המנוע עצמו קטן, ולכן קל משקל. יתר על כן, הרעש וצריכת החשמל שנספק על ידי המנוע בזמן השימוש נמוך.

מנועי הרטט מוגדרים בשני סוגים בסיסיים: מטבע (או שטוח) וצילינדר (או בָּר). ישנם כמה רכיבים

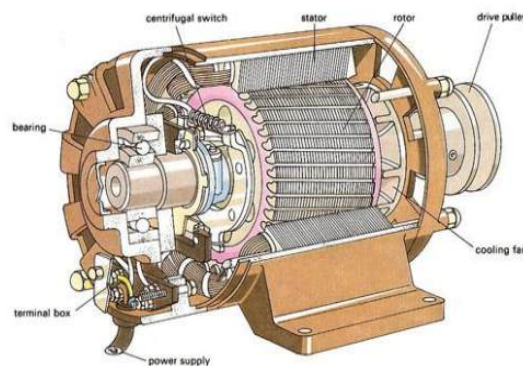


Figure 1

קונסטרוקציות הפנימיות של שניהם.

צילינדר: צורת הגליל נקראת גם מנוע רטט מסוג בר. מנוע רוטט זה הוא בעצם מנוע שאינו מאוזן. במילים אחרות, יש משקל שלא במרכז המנוע המחובר לציר הסיבובי של המנוע היוצר כוח צנטריפוגלי תוך כדי שמסתובב. כוח לא מאוזן זה מוציא מאיזון את המנוע. התזוזה המהירה שלו גורמת למנוע להתנדנד, הידוע כ"רוטט". ניתן לשנות את הנדנוד על ידי מסת המשקל שמצרפים, המרחק של המשקולת לציר, והמהירות שבה מנוע מסתובב. יתרה מכך, הכוח הצנטריפוגלי, שנוצר על ידי המשקל הלא מאוזן, גורם למנוע לרטט ב-2 צירים (ציר Z וציר X).

מטבע: בדומה למנוע רטט מסוג צילינדר, מנוע רטט מסוג מטבע מורכב ממשקולת, מגנט טבעת, רוטור עם נקודות תיווך מחוברים בחזית וסלילים מורכבים מאחור, ומברשות באספקת חשמל מחובר למגנט הטבעת. נקודות ההעברה, שהן החלק הצהוב על בתמונה התחתונה, נמצאים במגע עם קצה המברשות. זה יפעיל את הסלילים החשמליים ברוטור. אנרגיית הסלילים מייצרת שדה מגנטי והוא חזק מספיק כדי שהאינטראקציה עם מגנט הטבעת המשולב בסטטור, שגורם לסיבוב. כוח שנוצר עקב השדה המגנטי.

Frequency

$$f_{vibration} = \frac{(Motor RPM)}{60}$$

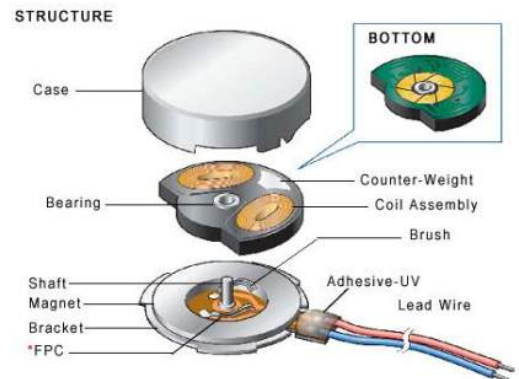
Force

$$F_{vibration} = m \times r \times \omega^2$$

m = the mass of eccentric weight
r = mass' offset distance
 ω = speed of the motor (rads⁻¹)

$$\omega = 2\pi f$$

כוח זה גורם לעקירת המשקל(תזוזה/הסטה). התזוזה החוזרת ונשנית של המשקל מייצרת כוח משתנה המורגש כרטט. נקודות ההעברה משמשות בשינוי הקטבים, כך שהרטור נע, הסלילים הופכים כל הזמן את הקטבים שלהם.



2. סוגי הפסולת שהמערכת תוכל לאסוף

רצוננו הוא שהמערכת שלנו תוכל לאסוף סיגריות, בקבוקים (פלסטיק וזכוכית), בד, נייר, עץ, זכוכית, קרטון ופלסטיק.

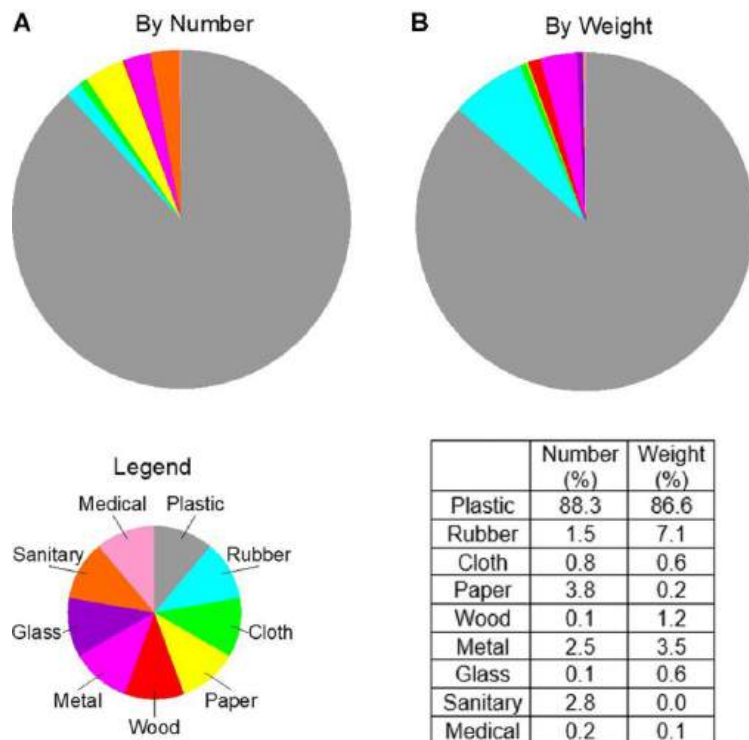
לפי מחקר סקוטי המשקל הממוצע של פריט פסולת פלסטיק היה 15.6 גרם (משקל חציוני 2.5 גרם). עבור סוגי פסולת שאינם מפלסטיק, הכבד ביותר היה עץ, עם משקל פריט אשפה ממוצע של 154.7 גרם. הזכוכית הייתה באותו סדר גודל (150 גרם).

טבלה 1

	Total Number	Total Weight (g)	Percent by Number	Percent by Weight	Min (g)	Max (g)	Median (g)	Mean (g)	Stddev (g)	np/100 m	g/100 m
All	1,618	25,797	100.0	100.0	0.01	2,600.0	2.4	15.9	81	20.2	322.5
Plastic	1,428	22,329	88.3	86.6	0.01	2,600.0	2.5	15.6	84	17.9	279.1
Non-Plastic	190	3,468	11.7	13.4	0.01	404.0	0.2	18.3	47	2.4	43.4
Rubber	24	1,843	1.5	7.1	4.90	404.0	54.8	76.8	91	0.3	23.0
Cloth	13	164	0.8	0.6	1.60	48.7	5.1	12.6	15	0.2	2.1
Paper	62	46	3.8	0.2	0.01	16.0	0.1	0.7	3	0.8	0.6
Wood	2	309	0.1	1.2	9.40	300.0	154.7	154.7	205	0.0	3.9
Metal	40	905	2.5	3.5	0.60	82.8	14.4	22.6	21	0.5	11.3
Glass	1	150	0.1	0.6	–	–	–	150.0	–	0.0	1.9
Pottery	0	–	–	–	–	–	–	–	–	–	–
Sanitary	45	12	2.8	0.0	0.0	2.8	0.2	0.3	0.4	0.6	0.2
Medical	3	38	0.2	0.1	2.0	34.0	2.2	12.7	18	0.0	0.5

Litter types are those defined by OSPAR (2010).

איור 1



במונחים של אחוז הרכב פסולת החוף (איור 1 וטבלה 1), הפלסטיק מהווה כ-87-88% מסך העומס, ללא קשר אם נמדד במספר או במשקל. עם זאת, לפי מספר, גומי, נייר, מתכת ופריטים סניטריים היוו כל אחד כ-2-3% מסך העומס, אך לפי משקל גומי ומתכת שלטו, עם 7.1 ו-3.5%, בהתאמה. תוצאות אלו פשוט מראות שהרכבים שונים הנלמדים מנתוני ניטור המלטה על החוף בהתאם לשאלה אם משתמשים במספר או במשקל כמדד, בשל ההבדל במשקל של פריטי המלטה בודדים בכל אחד מסוגי הפסולת. החלטתי שלפי המגבלות של המערכת כמו שאני מבין אותם ברגע זה מאפשרים לאיסוף של אובייקטים שלא יותר מדי גדולים: סיגריות, שקיות, גומי, זכוכית, מסכות ועוד דברים בסדר גודל זה, ולא דברים כמו קופסאות קרטון גדולים וחתיכות עץ ענקיות.

רפלקציה (תאור פתרון)

לדעתי הפתרון הוא טוב, מוצג באופן ברור על אופן הפעולה ועל היתרונות ומוגבלויות שלו, אך הוא מאוד קשה ליישם, וכדי שיעבוד אפקטיבית יש צורך בשיקולים קשים על אופן יישום. המידע שנאסף מראה באופן טוב מה ניתן לעשות כדי ליישם את הפתרון במספר אופנים. המערכת הינה תת מערכת של המערכת כולה.

בחירת פתרון מתוך חלופות

תיאור תהליך הבחירה (ניר)

מזקו"ם		4X4		סוג הנעה:			
ציון מנורמל	ציון	ציון מנורמל	ציון	משקל מנורמל	משקל	קריטריון	
13.4	85	12.6	80	15.79%	95	חיכוך עם הקרקע	1
11.7	95	8.6	85	12.28%	70	גובה (יותר נמוך יותר טוב)	2
13.4	90	13.4	100	14.91%	100	יכולת להתגבר על מכשולים בקלות יותר	3
1.6	30	4.5	95	5.26%	95	מהירות המערכת	4
15.0	95	11.1	90	15.79%	85	מסוגל לסחוב יותר משקל	5
10.4	80	11.9	70	14.91%	85	מה צורך יותר כסף	6
15.8	85	14.2	95	15.79%	90	עמידות המערכת	7
5.0	95	3.7	70	5.26%	30	רמה של קושי ביצועה	8
86		89		100.00%	570		סכ"ה :

- הערה: כתבנו את משקל המנורמל בדיוק של שתי ספרות אחרי הנקודה.

פירוט קביעת הציונים:

1. בחרתי לתת לקריטריון "חיכוך עם הקרקע" משקל 95, משום שאנו זקוקים שלרובוט יהיה רמת חיכוך כמה שיותר גבוהה עם הקרקע כדי שלא יחליק או יפול כאשר הוא עובר בשטחים מאתגרים כמו למשל (קרח, עליות תלולות, ואבנים גדולות).
2. בחרתי לתת לקריטריון "גובה" משקל של 70, מתוך שיקול שעל הרובוט להיות כמה שיותר נמוך כדי לעבור בהצלחה מכשולי גובהה כמו מחסומים, תקרה של תהלה או מערה. בנוסף, להתקרב כמה שיותר לגובהה הקרקע יעזור למערכת איסוף לאסוף אשפה בצורה קלה ונוחה יותר.
3. בחרתי לתת לקריטריון "יכולת להתגבר על מכשולים בקלות יותר" משקל 100, מהסיבה שקריטריון זה הינו הלב של כל המערכת, הוא הקריטריון המשמעותי ביותר וזאת מטרת ההענה לעבור דרך מכשולים בקלות. אני שואף לפתח רובוט שמסוגל להגיע למקומות שבני אדם לא

רוצים לסכן את חייהם כדי להגיע לשם ולכן להתגבר על מכשולים הוא כל כך חשוב שיהיה בתוך המערכת שלי.

4. בחרתי לתת לקריטריון "מהירות המערכת" משקל 40 בגלל שכדי להגיע למקומות במהירות גבוהה יותר זה חשוב, אך אין צורך גדול לזה. המהירות פחות רלוונטית כי גם כאשר אנו נאסוף זבל, מהירות ההענה תהיה נמוכה כדי לאפשר איסוף מדויק של מערכת האיסוף.

5. בחרתי לתת לקריטריון "מסוגל לסחוב יותר" משקל 85, מפני שאנו לא רוצים שהמערכת תאסוף כמות אשפה קטנה ומיד אחר כך המערכת הנעה לא תוכל לסחוב יותר ותתקע, ולכן קריטריון זה חשוב בעת בחירת המערכת הנעה.

6. בחרתי לתת לקריטריון "מה צורך יותר כסף" משקל 85, מהסיבה שבגלל שהקושי הכלכלי של המדינה שלנו בשנתיים האחרונות הנוצר עקב נגיף הקורונה פגע לנו בתקציב המגמה הכולל ובכך לכל פרויקט יש פחות כסף. ולכן בחרתי לתת משקל זה כי על המערכת אסור לחרוג את התקציב שהוקצב לנו, ולכן צריך להתחשב בכל רחיבי המערכת שהיא זקוקה ולחשב את המחיר הכולל שהמערכת תעלה.

7. בחרתי לתת לקריטריון "עמידות המערכת" משקל 90, משום שבמקרה של חבלה במערכת עקב החלקה או נפילה ממקום גבוהה, אנחנו צריכים שהמערכת תהיה חזקה ועמידה ובכך נקטין את הוצעות התחזוק שלה.

8. בחרתי לתת לקריטריון "רמה של קושי ביצוע" משקל 30, קריטריון זה הוא בחירה אישית שלי. אני מאוד אוהב להקשות ולאתגר על עצמי איפה שאני יכול, אני מאמין שככל שיותר קשה ולחוץ לנו אנחנו לומדים ומתפתחים עוד ועוד! ולכן יצרתי את הקריטריון הזה שיעזור לי להבין מה יהיה יותר מאתגר לביצוע ובכך אשתפר מפעם לפעם!

הסבר הציונים של פתרון 4X4:

1. המערכת כוללת ארבעה גלגלי גומי ושטח הפנים עם הרצפה לא כל כך גדול, כאשר לקחנו את גלגלי הגומי מהמגמה שלנו גיליתי שהחיכוך לא כל כך גדול וקיים חשש של החלקה בכל שטח שהוא לא מישורי. לכן הציון שניתן הוא 80.
2. הפתרון דורש מהמערכת להיות כמה שיותר נמוכה כדי להקל על המערכת איסוף. בגלל שאנו משתמשים בגלגלי גומי בדומא למכונית כתוצאה מכך המערכת תהיה גבוהה. הציון שניתן הוא: 85
3. הפתרון "4x4", בשל כוחה החזק של המערכת, המניעה ארבעה גלגלים במקביל קל להתגבר על מכשולים הצורכים כוח סיבוב חזק ומהירות סיבוב גבוהה. הציון הניתן הוא 100
4. המערכת מהירה מאוד בעזרת התמסורת שהיא כוללת. למנוע כוח חזק אשר מאפשר מהירות סיבוב גבוהה. הציון שבחרתי לתת הוא 95.
5. המערכת חזקה ויכולה לסת משקל לא קטן. אך עם זאת אם תעבור את המשקל הרצוי התמסורת עלולה להישחק ולפגוע בביצוע של המערכת. הציון שניתן הוא: 90
6. המערכת כוללת ארבעה גלגלי גומי, מנועה חזק שאין במגמה, סט דיפרנציאל, תמסורת ושלדה. עלות המערכת תהיה נמוכה יחסית וכל חלקי המערכת עמידים ולכן לא נצטרך לקנות אותם עוד כי הם לא ישברו או יהרסו בקלות. הציון שניתן הוא: 70
7. המערכת עמידה מאוד הסיכוי של המערכת להשבר נמוכה מאוד. עם זאת למערכת יש נקודות תורפה למשל אם המערכת תקבל מכה בדיפרנציאל היא יכול להפסיק לעבוד מיידית. ציון 95
8. לפי דעתי רמת קושי הביצוע היא לא כל כך גבוהה. הדבר העיקרי שמאתגר במערכת הוא הכנת תמסורת השינועה מהמנועים לארבעת הגלגלים. לפיכך הציון שניתן הוא 70.

הסבר הציונים של פתרון מזקו"ם :

1. הזחל מפזר את משקלו של כלי הרכב על שטח פנים גדול. בכך הלחץ שמפעיל הרכב על הקרקע קטן באופן ניכר, דבר המונע מכלי רכב כבדים לשקוע בקרקע חולית או בוצית. הציון שניתן הוא: 85.

2. גובהה המערכת הוא נמוך ביחס לפתרון א (4x4) וגלגלי הפולי מודפסים ולא גדולים. בנוסף הזחל נמצא בצידי המערכת ולא מתחתיה מה שמנמיך את המערכת מאוד. הציון שניתן הוא: 95

3. בזכות התמסורת חוזק של המערכת היא מצליחה לעבור בין כל המכשולים שעוברים בדרכה בין אם זה קרח (חיכוך גבוהה), אבנים (לעלות מעליהם), או מחסום (מערכת נמוכה) הציון שניתן הוא: 90

4. מערכת המזקו"ם ידוע בכך שהיא איטית בשל התמסורת שבתוך המערכת. יחס התמסורת שאנו נזדקק אליה הוא תמסורת כוח (מגלגל קטן לגלגל גדול). אך כפי שציינתי במשקל הפתרון, מהירות לא כל כך חשובה לנו במערכת לכן קרטיון זה לא פוגע בציון המערכת כל כך. הציון שניתן הוא: 30

5. הזחל מפזר את משקלו של המערכת על שטח פנים גדול ובעזרת זה המערכת מסוגלת לסחוב משקל כבד מבלי שיפגע בביצועי המערכת. הציון שניתן הוא: 95

6. בעזרת המדפסות התלת מימדיות שנמצאות לשירותינו במגמה אנו נוזיל את העליות של המערכת בכך שנדפיס את הפולי המונע ומניע של המערכת אך לעומת זאת רצועות המערכת עלולות להיקרע בקלות לאחר שימוש רב ולכן נצטרך להזמין כמות גדולה של רצועות. הציון שניתן הוא: 70

7. המערכת מאוד עמידה בשל המבנה המכני של המערכת. כיפופי המתכת מאפשרים חוזקה רבה למערכת, נוסף על כך הדפסות המערכת יהיו מיוצרות עם חומר חזק מאוד הנקרא "ABS". חומר זה מאפשר למערכת להיות עמידה ונוכל להשתמש בו הרבה אך לעומת זאת רצועות המערכת עלולות להיקרע באת שימוש רב בהן. הציון שניתן הוא: 85

8. לאחר שחקרתי על מערכת המזקו"ם אני הופתעתי כמה אני לא מבין איך עובדת המערכת וכיצד לבנות ולייצר אותה. דבר זה יצר בי התרגשות ורצון לעבוד עליה. הציון שניתן הוא: 95

תיאור מורחב של הפתרון הנבחר - מערכת הנעה

הפתרון הנבחר הוא הנעת 4X4 כיוון שקיבל ציון גבוהה יותר בטבלת הבחירה שערכנו ועמד ביותר קריטריונים מהמערכת השנייה. לאחר חשיבה מעמיקה ותכנון מפורט של הפתרון הנבחר, הגענו למסקנה כי יש לבצע שנוי אחד עיקרי בפתרון. בכדי להתאים את תקציב והיכולות אשר נתונות לנו במגמה ובכך נבצע את הפתרון בצורה הטובה ביותר שאנחנו יכולים עם מה שנתון לנו.

כפי שציינו בהצעת הפתרון, על המערכת להיות עמידה, בעלת רדיוס סיבוב גדול וזולה כמה שאפשר.

ראשית, בהתחלה חשבנו לעשות הנעה אשר מורכבת משני גלגלים ומנוע בכל צד. לאחר בניית אב טיפוס של מערכת ההנעה התפלאנו לגלות שרדיוס הסיבוב והמהירות של המערכת היו מאוד נמוכים ולא ענו על דרישות המערכת כראוי ולכן היינו חייבים לעשות משהו בידון. כדי לגרום לרכב אשר לא יכול לשנות את זווית הגלגלים שלו לפנות עליו לסובב צד אחד של הגלגלים במהירות קטנה יותר מהצד שאליה רוצים לפנות, אך בגלל שמהירות הגלגלים הייתה מאוד קטנה וכמעט לא היה שינוי במהירות בין שני צידי המערכת רדיוס הסיבוב שלה היה מאוד גדול. אחרי חשיבה רבה עם המורה שמוליק הגענו לפתרון אשר הכי סיפק אותנו. החלטנו להשתמש במעט הכסף האחרון שנותר לנו מהתקציב של הפרויקט כדי להזמין סט של 2 גלגלים מסוג "אומני". הדבר המיוחד בגלגלי אומני זה הגלגלות אשר מאפשרות החלקה לצדדים ובכך מאפשרת לנו להסתובב ביותר כלות ולהחליק בעט סיבוב המערכת. ובכך פתרנו את הבעיה של פניות המערכת.

תמונה הממחישה את התנועות האפשריות מהגלגל:

עכשיו הבעיה היחידה שנותרה לנו היא הבעיה של מהירות המערכת, לא היה לנו עוד כסף לקנות מנועים חזקים יותר ולכן האופציה היחידה שנשארו איתה הייתה להכין תמסורת אשר משלבת כוח של שני מנועים ומנוע אחד (ניתן לקרוא בהרחב על הכנת התמסורת ועל איך היא פועלת בעמוד 86).

שנית, השינוי האחרון שעשינו על המערכת היא החלפת רצועת הטיימינג בשרשרת הנעה. שנוי זה קרה כיוון שרצינו לשלוט באורך הרצועה במידה של שינוי המרחק בין הגלגלים באט צורך. בנוסף שרשרת הנעה מאפשר שליטה מבוקרת יותר מרצועת טיימינג ודבר זה חשוב מאוד לנו כאשר נרצה לאסוף אשפה.



רפלקציה (לתיאור המורחב של הפתרון ולתהליך בחירתו)

כאמור, להערכתתי בחרתי את הפתרון המתאים ביותר מבין הפתרונות החלופיים המוצעים לתת המערכת. תהליך בחירת הפתרון נעשה באופן ברור ומפורט, בעזרת טבלת השוואה המציבה קריטריונים, משקלים וציונים משוקללים. את הקריטריונים בחרתי בהתאם למטרות הפרויקט ובהתאם לביצוע מוצלח שלו.

הפתרון, לדעתי, הוא המתאים ביותר מאחר והוא עונה על הקריטריונים שאותם החשבנו בחשיבות עליונה מבחינת מטרות הפרויקט.

התיאור המורחב של הפתרון הנבחר מוסבר באופן ברור, ובו הצגתי את כל השיקולים והתכנונים שעשיתי לאחר בחירת הפתרון. ההסבר מלווה בסקיצות מדגומות כך שהקורא יכול להבין באופן ברור את הפתרון בצורה רחבה ומעמיקה.

בחירת פתרון מתוך חלופות

תיאור תהליך הבחירה(נועם)

מסגרת מוכנית		מסגרת רוטטת					
קריטריון	משקל	משקל מנורמל	ציון	ציון מנורמל	ציון	ציון מנורמל	ציון מנורמל
1 יכולת איסוף אובייקטים קטנים	90	16.67%	70	11.7	95	15.8	
2 יכולת איסוף אובייקטים גדולים	80	14.81%	90	13.3	70	10.4	
3 מהירות איסוף	85	15.74%	95	15.0	70	11.0	
4 עלות רכיבים	95	17.59%	60	10.6	75	13.2	
5 עניין קהל	90	16.67%	90	15.0	70	11.7	
6 חישובים מכניים	60	11.11%	70	7.8	60	6.7	
7 רצון עשייה	40	7.41%	100	7.4	85	6.3	
סכ"ה :	540	100.00%		81		75	

פירוט קביעת הציונים:

1. בחרתי לתת לקריטריון "יכולת איסוף אובייקטים קטנים" משקל 90, בגלל שרוב האשפה שמתאספת ביבשה הינה אשפה קטנה-בדליל סיגריות, שקיות, מסכות, בקבוקי פלסטיק ועוד.
2. בחרתי לתת לקריטריון "כולת איסוף אובייקטים גדולים" משקל 80, מכיוון שלמרות שרוב האשפה על היבשה הינה יחסית קטנה, עדיין יש הרבה מאוד אשפה כבדה/בעל ממדים גדולים-קופסאות, קרטונים, בקבוקי זכוכית גדולים ועוד.
3. בחרתי לתת לקריטריון "מהירות איסוף" משקל 85, מכיוון שאנחנו מאוד לא רוצים שיווצר מצב שיהיה עומס יותר מכיוון שהאשפה אינה מזרקת לפיר מהר מדי, דבר שעלול לפגוע לנו במערכת.
4. בחרתי לתת לקריטריון "עלות רכיבים" משקל 95, מכיוון שיש לנו תקציב מוגבל, ואנחנו חייבים להשתמש בכמות הזאת להשיג את כל החומרים הנדרשים, שהם לא מעט, ולא רוצים להיות במצב שנגמר לנו התקציב באמצע הפרויקט ולכן לא נוכל להשלים אותו.

5. בחרתי לתת לקריטריון "עניין קהל" משקל 90, מכיוון שלמרות שהפרויקט הינו לאסוף אשפה, המטרה האמתית הינה להביא עוד מודעות לבעיה עולמית זה. קריטריון זה הוחלט לפי סקר שערכנו בקרב חברים ומשפחה.
6. בחרתי לתת לקריטריון "חשובים מכנים" משקל 60, מכיוון שלכל פרויקט ישנם דרישות מדויקות שאליהם אנחנו צריכים לשאוף, ואם יוצר מצב שהחשובים הינם קשים מדי, עלול להיווצר מצב שבו ישנו הרבה טעויות נגררות שישפיעו באופן יעילות המערכת.
7. בחרתי לתת לקריטריון "רצון עשיה" משקל 40, מכיוון שלדעתי גם חשוב אלמנט של בחירה עצמית בבחירת פתרון כדי לעשות משהו שיותר מעניין אותך, אבל נתתי לזה משקל נמוך רק כדי שאם היה שוויון בין תוצאות הפתרונות שקריטריון זה היה שובר את זה.

פתרון א(מגרפה מכנית):

1. למערכת זו יש יכולת לאסוף אובייקטים קטנים אבל בוודאות יהיה קושי באיסוף דברים כמו בדלי סיגריות ובלתי אפשרי לנקות רכיבים אחרים שמאוד מאוד קטנים. ציון 70.
2. למערכות זו אמור להיות יכולת מאוד גדולה בלהרים דברים כבדים ורכיבים בעלי ממדים גדולים. ציון 90.
3. לפתרון זה לא אמור להיות שום בעיה עם עומס יתר, דבר היחיד שיכול לקרות זה שהמערכת עלולה לפספס ממש קצת אשפה לכן מורידים את המהירות מ100 ל95. ציון 95.
4. למערכת זו יש עלויות גדולות, כגון שלד אלומיניום ובעיקר ייצור רשת עם שיניים, דבר שעלול לעלות הרבה כסף. ציון 60.
5. פתרון מרשים, מרהיב, מעניין ובנוי לגובה אז תופס עין, כל הדברים האלו גורמים לכך שיותר בקלות יהיה ניתן למשוך את דעת הקהל לבעיה שאנחנו רוצים להציג להם. ציון 90.
6. בפתרון זה ישנם לא מעט חישובים מכניים שנדרש לעשות כדי לפעול באופן טוב, יחסי תמסורת, טורק, אינרציה, אורכים של דברים, מציאת מנועים מתאימים, חישובים קינמטיים ועוד. ציון 70.
7. פתרון זה הינו פתרון שראיתי לפני מספר שנים וישר משך אותי אליו, הוא מעניין ביותר ומהווה אתגר ובנייה מכנית מרשימה. ציון 100

פתרון ב(רשת רוטטת):

1. למערכת זו יכולת טובה מאוד באיסוף אובייקטים קטנים, יכול לאסוף כל דבר שקצת גדול יותר מגרגיר חול לפי גודל חורי הרשת, הדבר היחיד שלא ניתן להיאסף הם חומרים מיקרוסקופים שאינם דבוקים לדברים אחרים. ציון 95.
2. יכולת יחסית טובה בלהרים דברים כבדים אבל בגלל אופן שבוא בנוי לא ניתן לאסוף אובייקטים גדולים מדי. ציון 70.
3. הבעיה עם מהערכת זו שיש מגבלה במהירות איסוף, בגלל שלוקח זמן לחול לעבור דרך הרשת יכול להיווצר מצב של Backlog, עומס יתר, כאשר יותר מדי חול ואשפה על המערכת, דבר שיכול לעצור ואף לפגוע במערכת. ציון 70

4. לפתרון זה ישנם כמה מרכיבים יקרים כגון גוף מאלומיניום ומרכיבים בשביל תזוזה לינארית.
ציון 75.
5. פתרון זה הינו מרשים, אבל בגלל הגודל והמראה שלו הוא לא מושך יותר מדי את העין של הצופים בוא, אבל עדיין מעניין ויכול לגרום לשיפור דעת קהל בעניין הנושא שאנו רוצים ללמד/להבהיר. ציון 70.
6. למערכת זה כמות גדולה מאוד של חישובים מכניים קשים. ציון 60.
7. פתרון זה הינו מעניין ומרשים. ציון 85.

תיאור מורחב של הפתרון הנבחר

רפלקציה

להערכתי, הגדרתי את הפתרון בצורה ברורה למרות האתגר הכרוך בכך. מדובר במערכת הכוללת אתגרים מכנים גם בשלב אסיפת הזבל וגם השינוע שלו ממקום האיסוף הישר אל הפח, ולכן הבנת מבנה המערכת נחוץ לשם הבנת הפתרון, ומאחר והוא מעט מורכב נתקלתי בקושי להסביר אותו באופן ברור לקורא. אך עם מחשבה מעמיקה התגברתי על קושי זה בהצלחה.

לשם הבטחת יישום הפתרון בשלב התכנוני, היה צורך בליצור את "המגרפה" אשר תשב על המסוע ותאסוף את האשפה בבטחה ולהבטיח שהאשפה תאסף. דבר זה לא היה פשוט, ומכיוון שהתקציב שלנו היה מוגבל היינו צריכים לבצע אילתורים, ומכיוון שעוד לא הגיעה הרצועות (מגיע ב-8.6.22) אנחנו עוד לא יודעים את היכולת שלו לאיסוף אשפה בצורה יעילה.

הפתרון מציג את התת המערכת בצורה ברורה ומלאה ועל כן הוא ראוי להיות חלק מן המערכת השלמה.

מימוש הפתרון

תיעוד תהליך הבניה

מפגש 1

תחילת מחקר על הפרויקט הקטן, מידע שילך בהמשך לפרויקט הגדול.

מפגש 2

חשיבה על פתרונות אפשריים למערכות ההנעה והאיסוף.

מפגש 3

סיום הפרויקט הקטן (אב טיפוס למנגנון של הפרויקט הגדול)

מפגש 4

מתחילים לעבוד על הפרויקט הגדול, הכנו הצעת עבודת גמר.

מפגש 5

התחלת מחקר גישוש ואיסוף מידע, חלק מהמידע נאסף באת הכנת הפרויקט הקטן.

מפגש 6

סיום מחקר גישוש, איסוף מידע והכנת שאלות חקר.

מפגש 7

הזמנת ציוד למגמה.

מפגש 8

נועם: תחילת קוד סופי לפרויקט.

נר: תחילת שרטוט תלת מימד "סוליד" סופי לפרויקט

מפגש 9

תחילת בניית אבטיפוס בגודל אמיתי מקרטון עד להגעתם של החלקים שהזמנו.

מפגש 10

המשך בניית אבטיפוס בגודל אמיתי מקרטון

מפגש 11

עבודה מהבית-תרגיל ארצי, המשך עבודה על החוברת ומערכת פריצינג.

מפגש 12

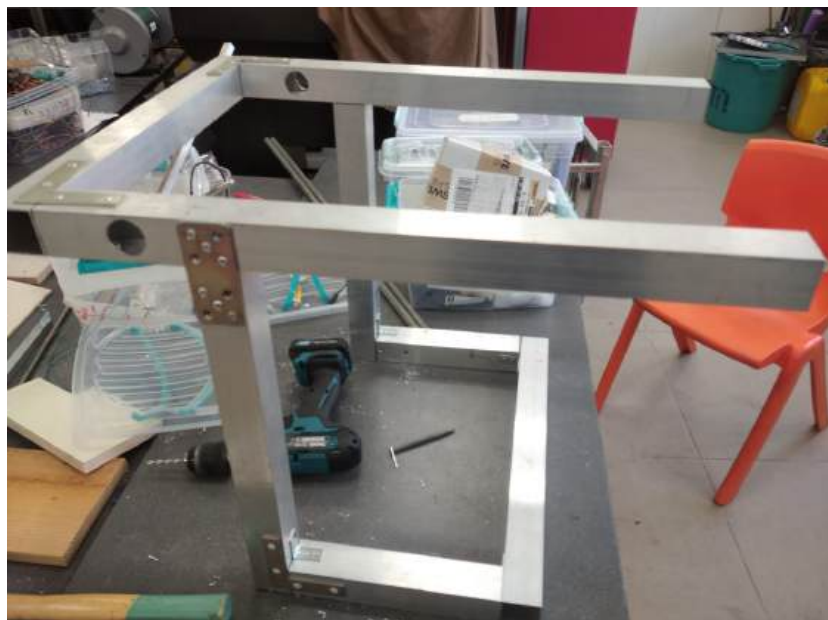
סיום אבטיפוס וסיום הפוסטר ההנדסי

מפגש 13

איסוף ציוד נוסף

מפגש 14

תחילת בניית הפרויקט האמיתי. קיבלנו כבר חלק מהציוד אז התחלנו להכין את השלדה של הפרויקט מאלומיניום.



מפגש 15

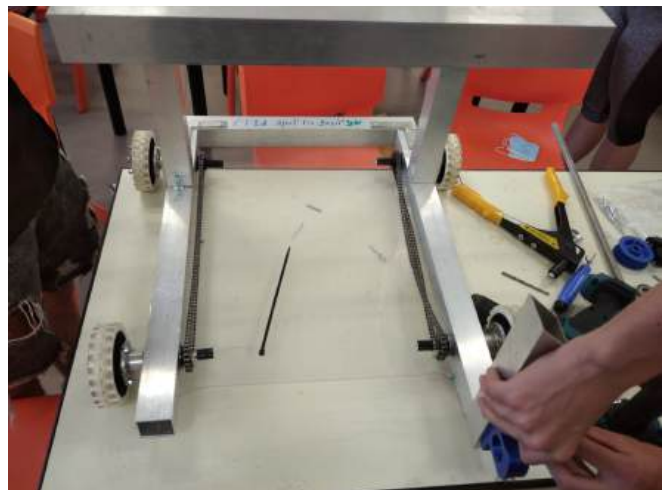
המשך עבודה על השלדה.

מפגש 16

קדיחת חורים למיסבים על ציר הגלגלים וסיום שלדה מאלומיניום(בתקווה).

מפגש 17

האקתון על הפרויקטים : בניית פיר אשפה מעץ, חיבור הגלגלים והתעסקות עם שרשראות ההנעה וחיבור הדפסות לחזית הרובוט.



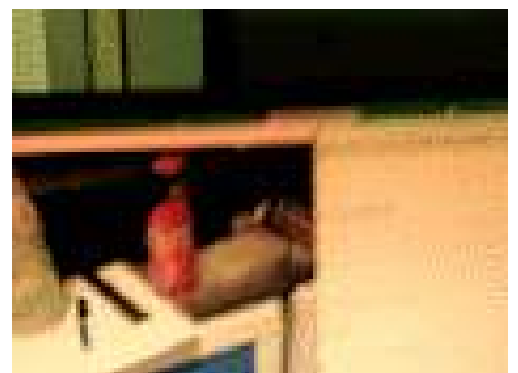
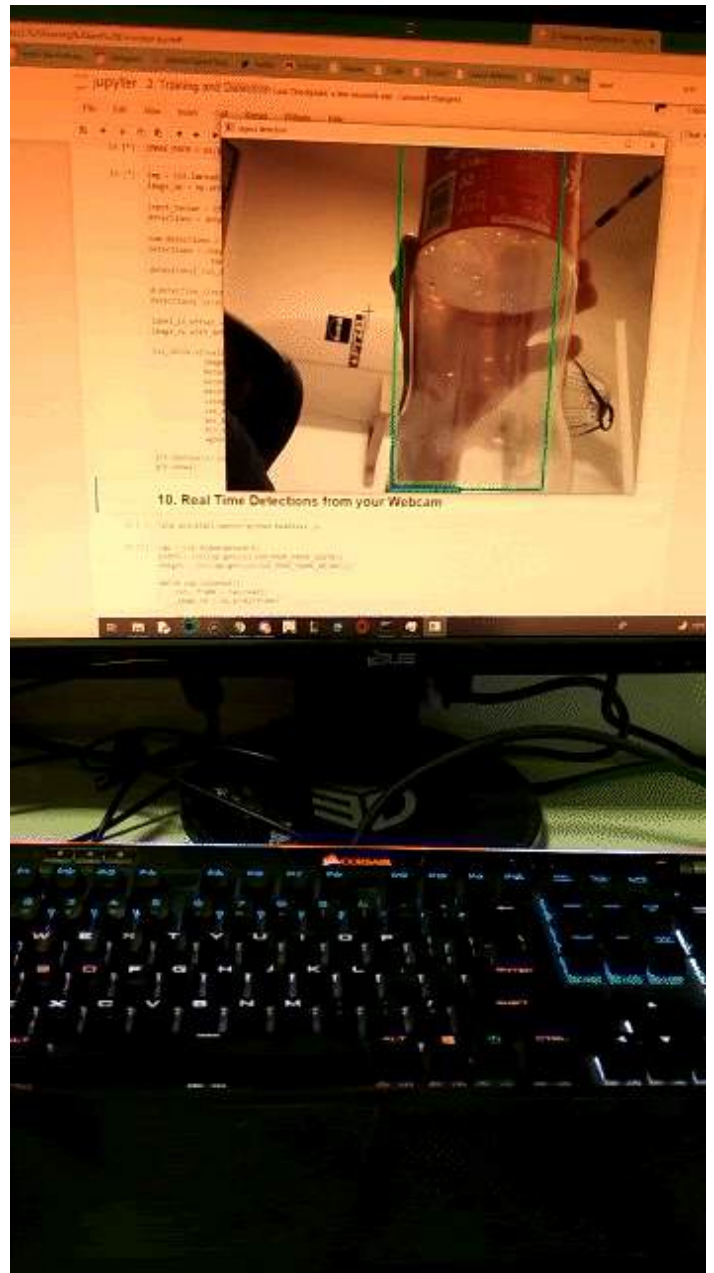
סיום זיהוי האובייקטים שעובד על המחשב

<https://youtube.com/shorts/DaJlREg1YFE>

<https://youtube.com/shorts/AOVGerlBFd8>

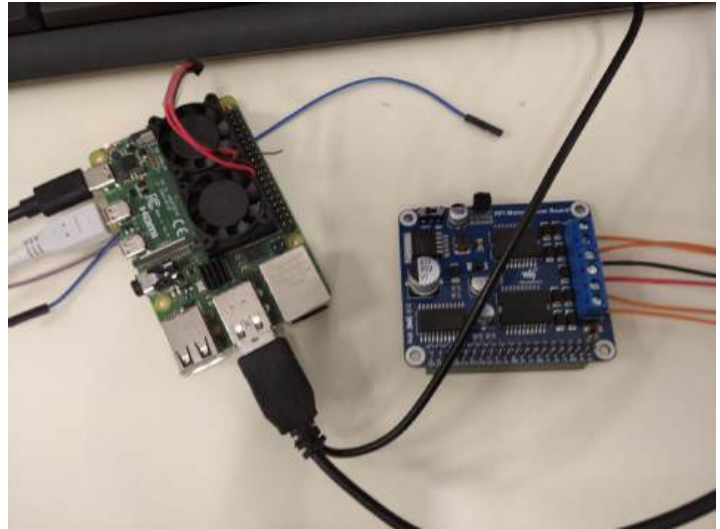
<https://youtube.com/shorts/E1WAlz7MVEo>

<https://youtube.com/shorts/F1opewmzqQc>
https://youtube.com/shorts/nWhUZJh_61c



מפגש 18

המשך עבודה על שרשראות הנעה-עיצוב מותחן בשביל השרשרת.
הכנת חיבור בקר מנוע ומנועים לרספברי פאי.



מפגש 19

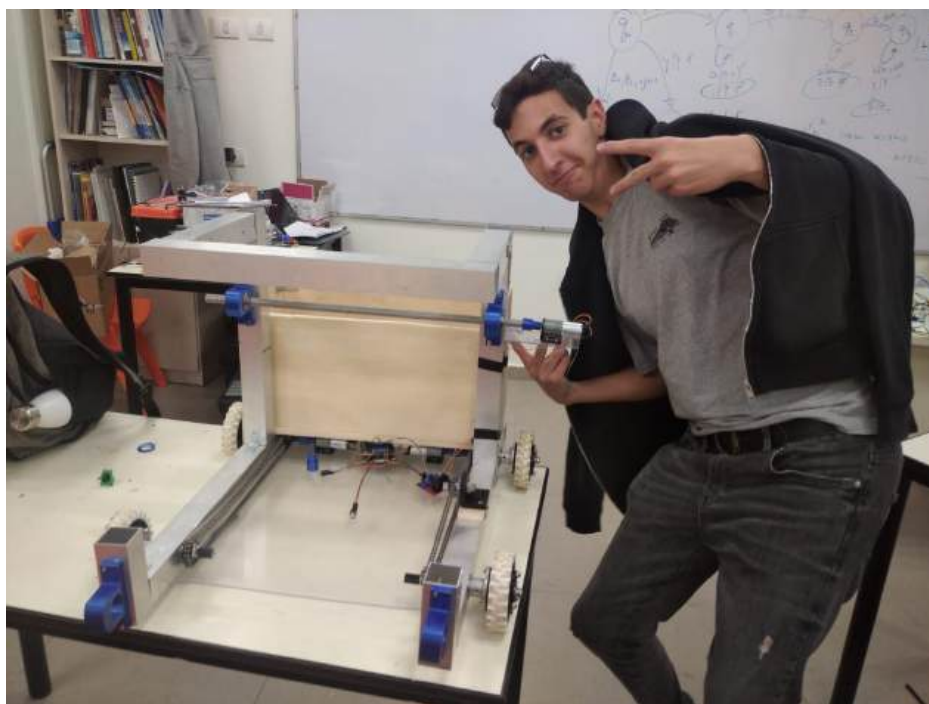
הגעת משטחי פולי-קרבונייט, חיתוך משטח בשביל בסיס השלדה
וחיבור שלו. בניית GUI בשביל התחרות כדי לשלוט על הרובוט:

<https://youtu.be/WrLBN4Wm0vk>



מפגש 20

תקלה באלקטרוניקה(שריפת בקר מנוע...) והאקטון על דברים אחרונים בפרויקט.
בסוף ויש רובוט שעובד(חסר רשת איסוף אבל הרובוט עצמו עובד כמו שצריך)



מפגש 21


חמישה באפריל(5.4.2022) : יציאה לתחרות הנדסת מערכות בחיפה!

TANJIROBOT

מערכת אוטונומית לאיסוף אשפה למען שמירה על הסביבה ויצירת מודעות בקרב קהל הציבור
מגישים: נועם אלרון, ניר לונדון
מנחה: שורן שמוליק

הפתרון שלנו

המערכת שלנו לוקחת את המשימה מתחנת נעילה ופועלת באופן אוטונומי. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה.



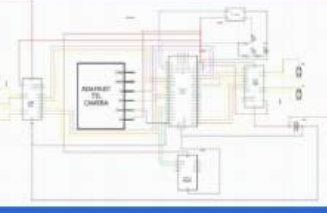
אנו מפתחים מערכת אוטונומית לאיסוף אשפה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה.

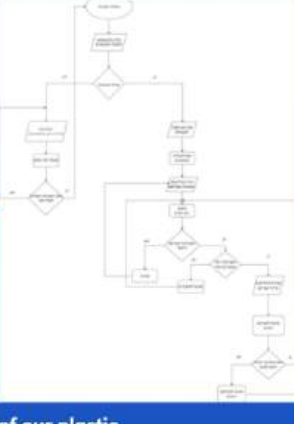
אנו מפתחים מערכת אוטונומית לאיסוף אשפה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה.


הדרכת הבעיה

הבעיה שהנחנו לעצמנו היא איסוף אשפה מהרחב הציבורי. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה.

הבעיה שהנחנו לעצמנו היא איסוף אשפה מהרחב הציבורי. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה. המערכת מזהה אשפה על הרצפה של המרחב הציבורי ומסירה אותה.







8 million metric tons of our plastic waste enter the oceans from land each year.



מפגש 22

ניסיון דיבאגינג (debugging) של בעיות ברסברי פאי



מפגש 23

המשך ניסיון של debugging, התקנה מחדש של כל הרספברי פאי מספר פעמים.



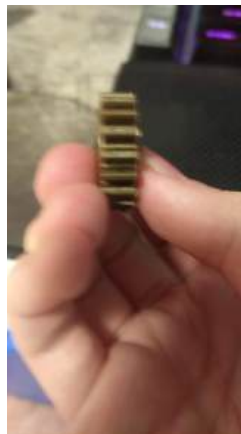
מפגש 24

החלפת הרספברי פאי משום שהיה בעיה בתצוגה שלו.

ניסיון לפתור את הבעיה שהמנועים שלנו חלשים מדי, עליה על הרעיון שליצור gearbox על שני מנועים כדי להוציא לאותו ציר פעמיים מומנט סיבוב

מפגש 25

הדפסות ראשונות של גלגלי שיניים לגיר-בוקס



מפגש 26

נסיעה לרעננה לחנות גיגי ירום כדי לנסות לקנות רצועות טיימינג מתאימות למערכת שלנו. לא מצאנו רצועות מתאימות אבל הפנו אותנו לחנות בשם יאיר פישר.

מפגש 27

ניסיון שני להדפיס את גלגלי השיניים לגיר בוקס ב-tolerance אחר.

#להוסיף תמונות של ההדפסות השניות

מפגש 28

הדפסות עובדות של גלגלי שיניים של ה-gearbox, הדפסה של מכסה הגיר-בוקס והרכבה שלו
#להוסיף צילום של המכסה המודפס ותמונה של ה-gearbox מורכב

מפגש 29

הערכה לתהליך עתידי, ההזמנה אמורה להגיע ברביעי- 8.6.2022 - הגעת רצועות טיימינג וחגורת מסוע והרכבתם על המערכת.

מפגש 30

הערכה לתהליך עתידי, סיום של הרובוט ב-100%, כולל דברים שעוד לא הספקנו, למשל להוסיף את ה-moldboard, וחיישנים שעוד לא הספקנו להוסיף.

תכן הנדסי - חישובים

חישוב מומנט סיבוב קבוע מינימלי למנועים

T_{req} - מומנט סיבוב קבוע מינימלי הנדרש בשביל להניע את המערכת.

M_{max} - משקל מקסימלי של כל המערכת, כולל משקל שאשפה מתאסף. 20kg

a - תאוצה של המערכת. 0.2m/s^2

v_{max} - מהירות מקסימלית של המערכת. 1m/s

g - תאוצת הכובד על כדור הארץ. 9.81m/s^2

θ_{max} - זווית מקסימלית שהרובוט צריך לעלות. 20° מעלות.

r - רדיוס הגלגלים. 0.096m

eff - הנחה סטנדרטית לגבי יעילות כל הרכיבים במערכת. 60%

$$T_{req} = \frac{100}{eff} \cdot M_{Max} \cdot v_{max} \cdot \left(a + g \sin(\theta_{max}) \right) \cdot r$$

$$T_{req} = \frac{100}{60} 20 \cdot (0.2 + 9.81 \cdot \sin(20)) \cdot 0.096$$

$$T_{req} = 11.377\text{Nm}$$

$$T_{req - in - kg - cm} = T_{req} \cdot \frac{100}{9.81} = 11.377 \cdot \frac{100}{9.81} = 115.97\text{kg}_{cm}$$

חישובים לחישוב אורך רצועת הטיימינג למערכת איסוף

C - מרחק מרכז למרכז של הפוליצ. $44.5 =$ סנטימטר $= 445$ מילימטר

D_1 - Pitch Diameter של הפולי הגדול. $= 2.256$ אינץ $= 57.3024$ מילימטר

D_2 - Pitch Diameter של הפולי הקטן. $= 1.128$ אינץ $= 28.6512$ מילימטר

$$Length = 2C + \frac{\pi(D_1 + D_2)}{2} + \frac{(D_1 - D_2)^2}{4C}$$

$$Length = 2 \cdot 445 + \frac{\pi(57.3024 + 28.6512)}{2} + \frac{(57.3024 - 28.6512)^2}{4 \cdot 445}$$

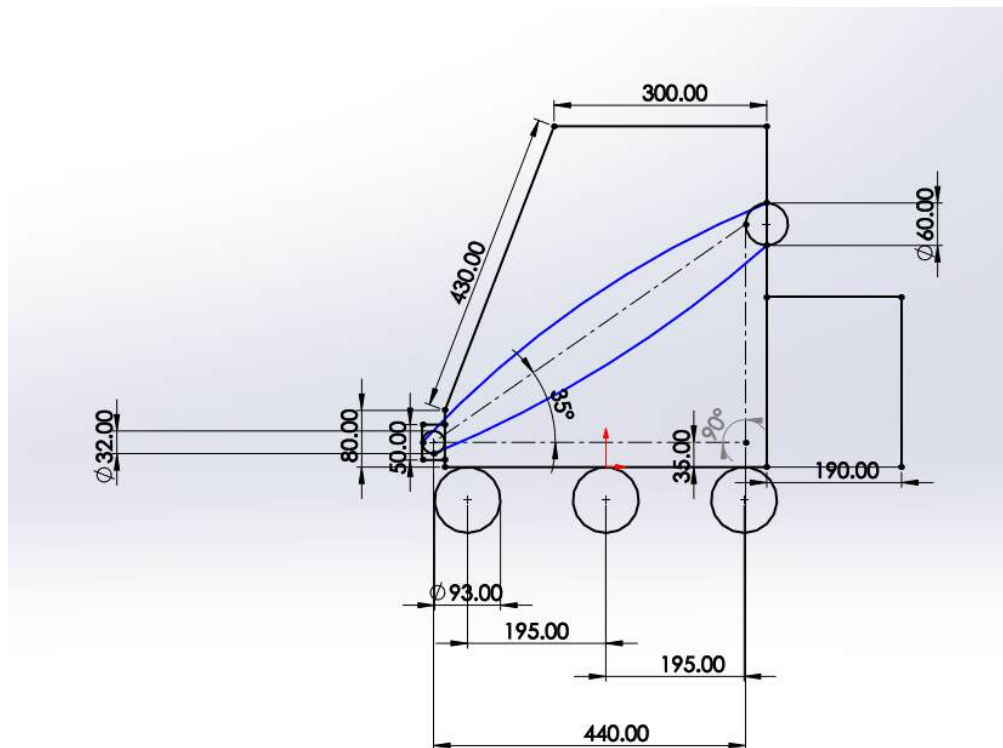
$$Length = 1025mm$$

שרטוט תלת מימדי

הסבר על גוף המערכת ובחירת הצורה שלו:

לפני כל הקמת מערכת מכנית חייב לבוא שרטוט דו מימדי אשר ידמה את גודלי המערכת וכיצד היא ממוקמת.

תחילה העלנו את הרעיון של המסוע אשר יאסוף את האשפה אך לא ידענו איפה למקם אותו ובאיזה גודל הוא יהיה, ולכן סרטטנו שרטוט דו מימדי (2D sketch) אשר יעזור לנו לראות איפה אנחנו יכולים למקם את הגלגלי פולי ובאיזה גובה הם יהיו, מה הזווית שנוצר כתוצאה משינוי הגובה של הגלגל פולי וקיצר השלדה החיצונית תיראה כדי למרכז את משקל המערכת בצורה הטובה ביותר. ראשית ידענו שאנחנו צריכים שהמסוע יהיה בשיפוע. לאחר מכן מיקמנו את הגלגלים המניעים ואת כדי לראות את פרופורציית המערכת, ומשם כל הגדלים כבר היו מוכנים פחות או יותר.

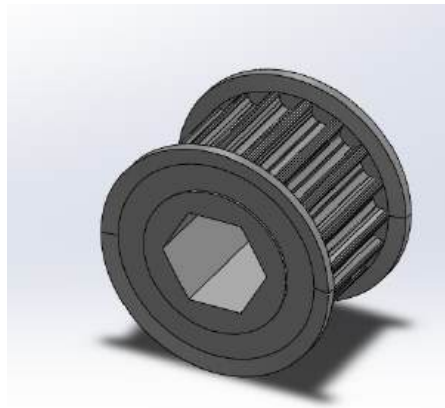


שרטוט המערכת היה שלב לא פשוט ולקח לי שעות רבות של ישיבה ממושכת של שרטוטים בתוכנת סולידוורקס (solidworks).

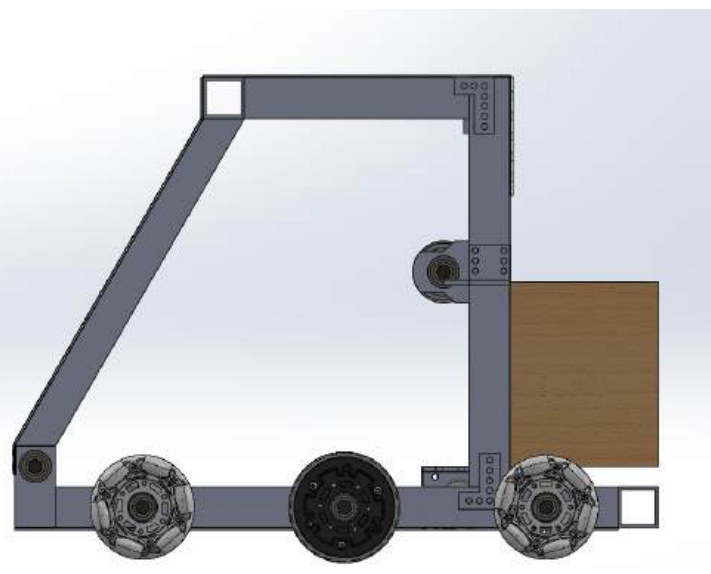
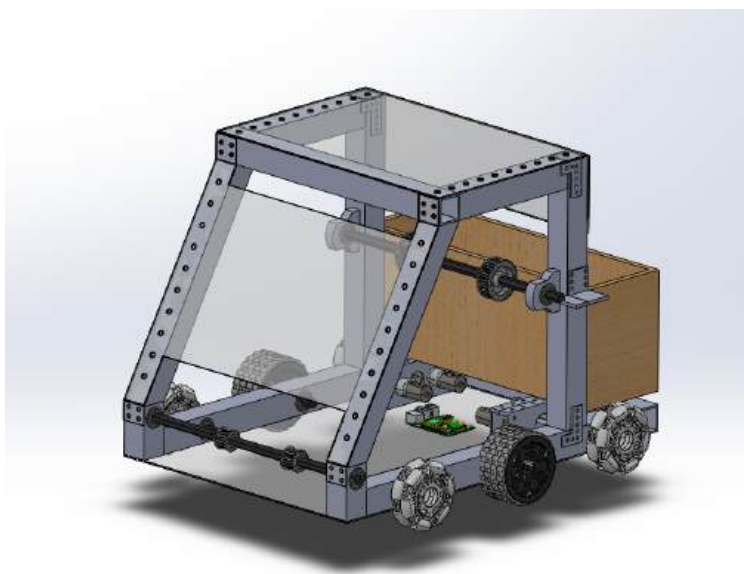
אחרי שהיה לנו שרטוט דו מימדי להתבסס על , התחלנו למדל את חלקי המערכת אחד אחד. מידלנו את פרופילי המערכת בגודל של 4X4 (גודל הפרופיל שהיה לנו במגמה). כדי לחבר את הפרופילים עשינו שימוש בניטים, ברגים. עם אומים וסוגרים בצורת L. בנוסף השתמשנו בחומר בשם

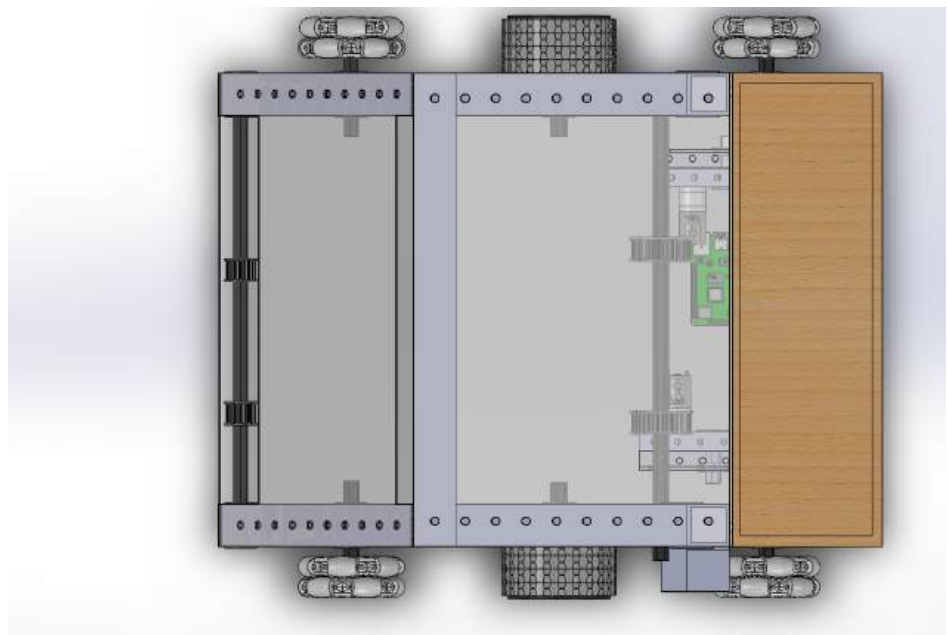
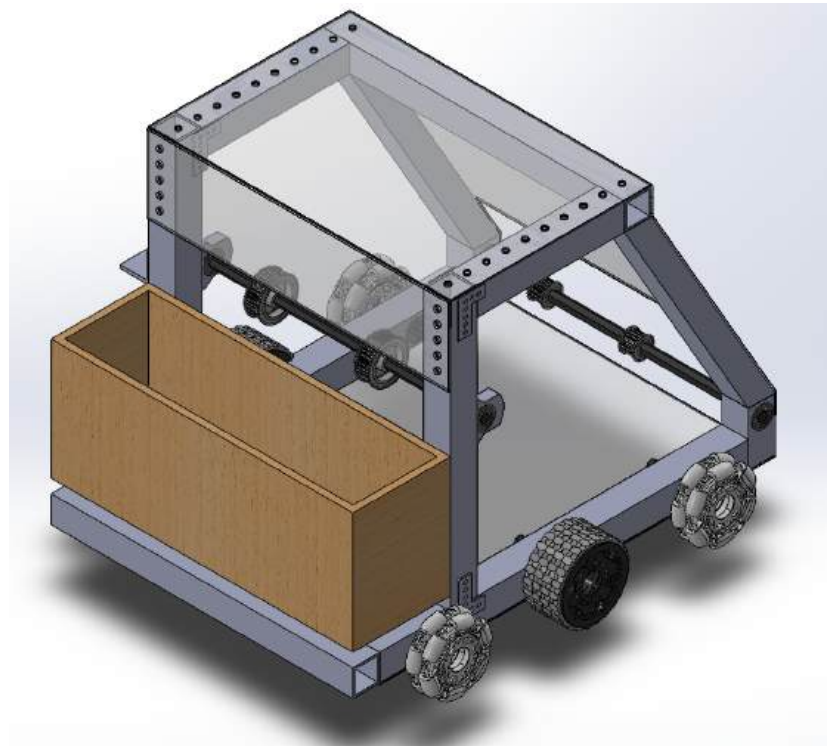
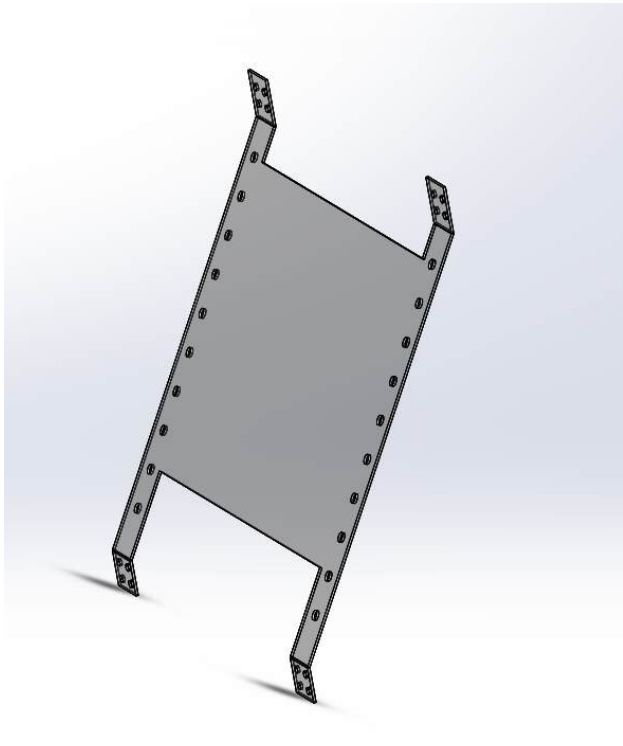
פוליקרבונט (Polycarbonate) כבסיס המערכת, גג המערכת וכהדילקטור שלנו בשל התכונות החזקות והגמישות שלו.

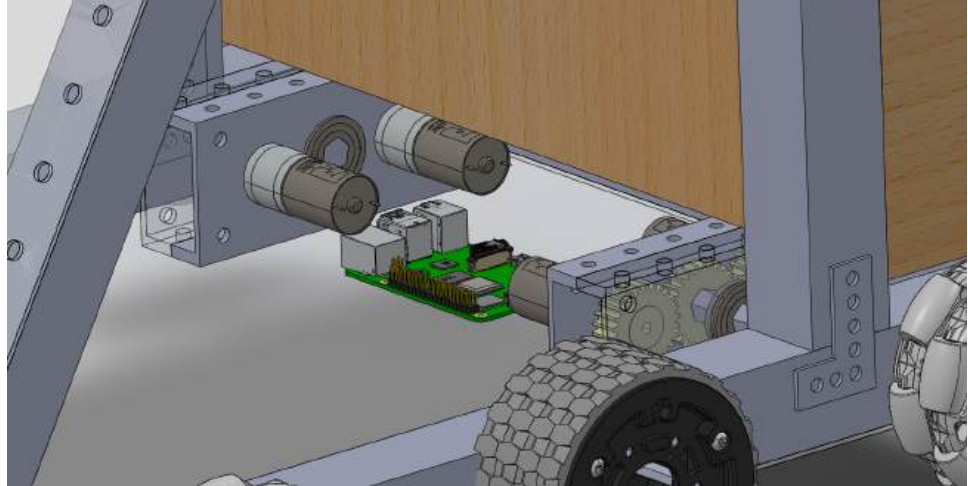
שרטטי את גלגלי הפולי של המערכת, את הקופסא ואת גלגלי ההנעה.



ולבסוף המערכת נוצרה בתלת מימד והיינו ממש שמחים ומרוצים מהתוצאה.







תמסורת מערכת ההנעה:

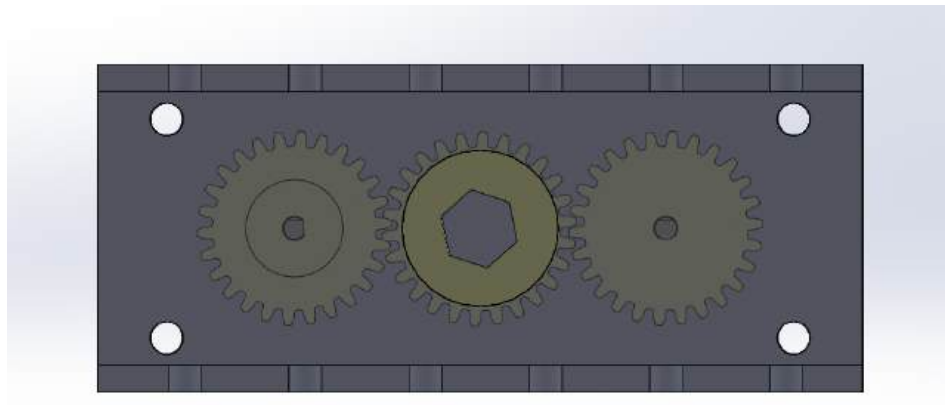
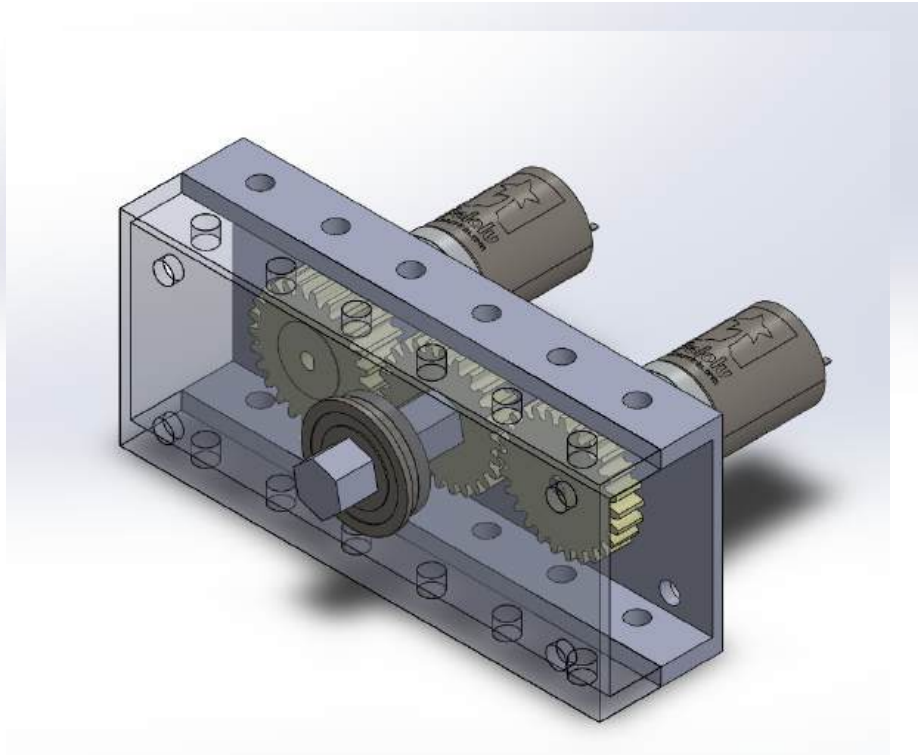
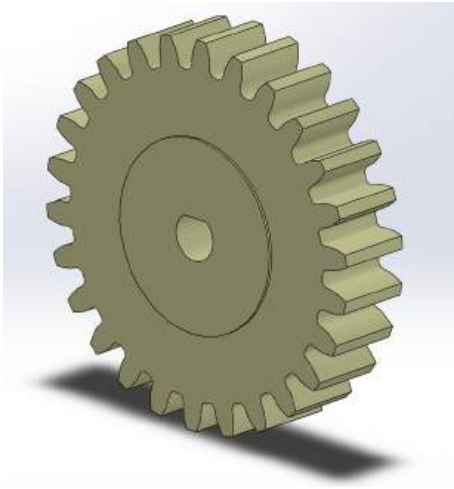
המנועים שהיו לרשותינו במגמה לא סיפקו מספיק מומנט סיבובי ביחס לזרם שהם צורכים. שמוליק הזמין לנו מספר מנועים שלצערנו לא סיפקו לצרכי המערכת. לאחר מכן חשבנו להזמין מנועים גדולים וחזקים יותר בעלי מומנט סיבוב גדול יותר אך לצערנו נגמר לנו התקציב גם כדי לקנות עוד מנועים, אבל גם כדי לקנות עוד מנועים היה צורך בלהזמין חלקים יקרים נוספים כמו בקר מנוע שיכול להעביר דרכו זרם תקיעה מספיק גבוהה כדי לפעל את המנועים, וגם ספק כוח 12V שמספק מספיק Ah (אמפר לשעה) כדי שהמערכת תוכל לפעול למשך זמן סביר.

כדי לנסות בכל זאת להציל את המערכת עלינו על הרעיון שלהשמש ביחס תמסורת של 1/1 (מכיוון שסל"ד סיפק לנו) כדי לשלב אותם, ובכך להוציא משתי מנועים מומנט סיבוב לציר אחת אקוילונטי למומנט סיבוב של שניהם ביחד.

פה שרטטנו 2D sketch אשר יספק לנו את המרחק המדויק שגלגלי השיניים יצטרכו להיות בו על מנת לעבוד בצורה הטובה ביותר.

תהליך שרטוט ה-2D : ראשית בחרנו גלגלי שיניים בגודל שהיו מספיק גדולים כדי לאפשר לשני המנועים מרחק אחד מהשני כדי לאפשר לנו נוחות באת הרכבת המערכת או שנוי שלה והוספנו את זה לשרטוט. לאחר מכן הוספנו את החורים של המנועים במרכז המעגל של גלגלי השיניים. מעגלי גלגלי השיניים יהיו חייבים לחתוך אחד את השני (דבר זה מדמה את השיניים של הגלגלים אחד בתוך השני) ולבסוף יצרנו מסגרת אשר תסגור את המערכת משני צידיה.

התוצאה הסופית לאחר הרכבת כל החלקים של ה-gear box :



תוכנה

זיהוי אובייקטים

בניית התוכנה היה תהליך מאוד ארוך, עם מספר רב של שלבים. הדבר הראשון שחקרתי היה איך Computer Vision ו-Machine Learning עובדים ואיך להתחיל איתם. לאחר שעברתי קורס ב-Udacity בנוסף ללמידה עצמית נוספת מאתרי אינטרנט ויוטיוב החלטתי שהשיטה הכי נכונה היא לבנות את תוכנת זיהוי האובייקטים בשיטה של Transfer Learning, שמשמעותה לקחת מודל של Object Detection קיים, כלומר את הארכיטקטורה שלה (CNN Layers+Weights) ולאמן אותו על ה-Dataset שאנחנו יצרנו. בכדי לעשות את זה השתמשתי בסיפריה הפתוחה הגדולה ביותר של Machine Learning בשם Tensorflow. סיבת הבחירה ב-Tensorflow ולא בספריה אחרת היא ש-Tensorflow ידועה שיש לה את האינטגרציה הכי טובה עם מכשירים בעל יכולת מחשוב קטן(הם יצרו אופציה להעביר מודל שיוצרים לפורמט שלמשל רסברי פאי יכול להשתמש בו).

ראשית כל התחלתי עם דירקטורי אישי למודל ושם כל הדירקטורים המישנים שנזדקק כדי להפריד ולהשתמש הכל אפקטיבית.

קודם כל התחלתי עם נתינת גישה לסקריפט הזה למודג'ולים של OS, UUID ו-wget. OS כדי לתקשר עם מערכת ההפעלה הפועלת על המחשב ממנו מורץ הסקריפט, UUID כדי לתת לכל תמונה שם ייחודי ו-wget.

```
# Import Operating System
import os
# Import uuid
import uuid

import wget
```

פה הגדרתי איזה "classes" יהיו לי סך הכל, כלומר בסופו של דבר יהיו ארבעה סוגים של אובייקטים שימצאו: פחיות, זכוכית, נייר, ופלסטיק.

```
labels = ['cans', 'glass', 'paper', 'plastic']
```

לאחר מכן יצרתי משתנה שמצביע על דירקטורי

C:/Tensorflow/workspace/images/collectedimages, ואז ביצאתי בדיקה האם דירקטורי הזה קיים. אם הדירקטורי לא קיים מתבצעת פעולה של OS כדי לבדוק איזה מערכת מפעילה את הסקריפט, לינוקס או ווינדוס, הצורך לכך הוא שמערכת הקבצים של שני המערכות פועלת ומאורגנת בצורה מעט שונה שמכריכה שינויים כדי לפעול אחד עם השני. לאחר בדיקת סוג המערכת, בעוד שדירקטורי הזה לא קיים, מייצרים דירקטורי חדש במיקום שבו מצביע IMAGES_PATH.

לאחר מכן אותו דבר קורא אותו דבר עם המשתנה של labels. יש לולאה שעוברת על כל אחד מהlabels ויותר משתנה שעם כל לולאה משתנה, כל פעם מחבר בין IMAGES_PATH לlabel הנוכחי, ובודק עם קיים הדירקטורי הזה, אם לא, אז נוצר.

```
IMAGES_PATH = os.path.join('Tensorflow', 'workspace', 'images',
                             'collectedimages')

if not os.path.exists(IMAGES_PATH):
    if os.name == 'posix':
        !mkdir -p {IMAGES_PATH}
    if os.name == 'nt':
        !mkdir {IMAGES_PATH}
for label in labels:
    path = os.path.join(IMAGES_PATH, label)
    if not os.path.exists(path):
        !mkdir {path}
```

ראשית כל התחלנו עם לקחת תמונות מהאנטרנט של אשפה כדי לאמן את המודל העתידי שלנו. לאחר מכן אנחנו מורידים labelimg, לתוך דירקטורי עם אותו שם, ספקריפט בגישה פתוחה בפייתון שנותן ליצור bounding boxes מסביב לאובייקטים בתמונה ושומר את הנתונים בקובץ .csv.

```
LABELIMG_PATH = os.path.join('Tensorflow', 'labelimg')
if not os.path.exists(LABELIMG_PATH):
    !mkdir {LABELIMG_PATH}
    !git clone https://github.com/tzutalin/labelImg {LABELIMG_PATH}

if os.name == 'posix':
    !cd {LABELIMG_PATH} make qt5py3
if os.name == 'nt':
    !cd {LABELIMG_PATH} && pyrcc5 -o libs/resources.py resources.qrc
```



לאחר מכן יצרתי דירקטורי של Train ו-Test שבו מחלקים את התמונות וליבלים של כל המחלקות בדרך כלל 80% ל-Train ו-20% ל-Test.

```
TRAIN_PATH = os.path.join('Tensorflow', 'workspace', 'images',
                             'train')
TEST_PATH = os.path.join('Tensorflow', 'workspace', 'images', 'test')
ARCHIVE_PATH = os.path.join('Tensorflow', 'workspace', 'images',
                              'archive.tar.gz')
```

פה הגדרתי את השם של המודל שאני יוצר, את השם שם המודל שבו אני משתמש בארכיטקטורה שלו, את הקישור למודל הזה, ועוד כמה דברים.

```
CUSTOM_MODEL_NAME = 'trash detection'
PRETRAINED_MODEL_NAME = 'efficientdet_d1_coco17_tpu-32'
PRETRAINED_MODEL_URL =
'http://download.tensorflow.org/models/object_detection/tf2/20200711/
efficientdet_d1_coco17_tpu-32.tar.gz'
TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
LABEL_MAP_NAME = 'label_map.pbtxt'
```

פה אני מגדיר מספר רב של תיקיות

```
paths = {
```

```

'WORKSPACE_PATH': os.path.join('Tensorflow', 'workspace'),
#General workspace
'SCRIPTS_PATH': os.path.join('Tensorflow', 'scripts'),
'APIMODEL_PATH': os.path.join('Tensorflow', 'models'),
'ANNOTATION_PATH': os.path.join('Tensorflow',
'workspace', 'annotations'), #Folder where TF Records are stored
'IMAGE_PATH': os.path.join('Tensorflow', 'workspace', 'images'),
#Just image path
'MODEL_PATH': os.path.join('Tensorflow', 'workspace', 'models'),
'PRETRAINED_MODEL_PATH': os.path.join('Tensorflow',
'workspace', 'pre-trained-models'),
'CHECKPOINT_PATH': os.path.join('Tensorflow',
'workspace', 'models', CUSTOM_MODEL_NAME),
'OUTPUT_PATH': os.path.join('Tensorflow',
'workspace', 'models', CUSTOM_MODEL_NAME, 'export'),
'TFJS_PATH': os.path.join('Tensorflow',
'workspace', 'models', CUSTOM_MODEL_NAME, 'tfjsexport'),
'TFLITE_PATH': os.path.join('Tensorflow',
'workspace', 'models', CUSTOM_MODEL_NAME, 'tfliteexport'),
'PROTOC_PATH': os.path.join('Tensorflow', 'protoc')
}

```

ופה כמו מקודם אני עובר על כל תיקייה בודק אם היא קיימת ואיזה מערכת הפעלה יוצרים אותה, ואם לא קיימת אז יוצר אותה.

```

for path in paths.values():
    if not os.path.exists(path):
        if os.name == 'posix':
            !mkdir -p {path}
        if os.name == 'nt':
            !mkdir {path}

```

אני מבצע בדיקה האם קיים המודל של טנסורפלו, אם לא אז מורידים מ-github לספריה של .APIMODEL_PATH

```

if not os.path.exists(os.path.join(paths['APIMODEL_PATH'], 'research',
'object_detection')):
    !git clone https://github.com/tensorflow/models {paths['APIMODEL_PATH']}

```

מוריד ספריה חשובה שצריך כדי לבצע פעולות ש-Tensorflow צריך (Serialization של נתונים)

```
if os.name=='posix':
    !apt-get install protobuf-compiler
    !cd Tensorflow/models/research && protoc object_detection/protos/*.proto
--python_out=. && cp object_detection/packages/tf2/setup.py . && python -m pip
install .

if os.name=='nt':

url="https://github.com/protocolbuffers/protobuf/releases/download/v3.15.6/protoc-3.15.6-win64.zip"
    wget.download(url)
    !move protoc-3.15.6-win64.zip {paths['PROTOC_PATH']}
    !cd {paths['PROTOC_PATH']} && tar -xf protoc-3.15.6-win64.zip
    os.environ['PATH'] += os.pathsep +
os.path.abspath(os.path.join(paths['PROTOC_PATH'], 'bin'))
    !cd Tensorflow/models/research && protoc object_detection/protos/*.proto
--python_out=. && copy object_detection\packages\tf2\setup.py setup.py &&
python setup.py build && python setup.py install
    !cd Tensorflow/models/research/slim && pip install -e .
```

הורדה של ה-Base Model.

```
if os.name == 'posix':
    !wget {PRETRAINED_MODEL_URL}
    !mv {PRETRAINED_MODEL_NAME+'.tar.gz'} {paths['PRETRAINED_MODEL_PATH']}
    !cd {paths['PRETRAINED_MODEL_PATH']} && tar -zxvf
{PRETRAINED_MODEL_NAME+'.tar.gz'}
if os.name == 'nt':
    wget.download(PRETRAINED_MODEL_URL)
    !move {PRETRAINED_MODEL_NAME+'.tar.gz'} {paths['PRETRAINED_MODEL_PATH']}
    !cd {paths['PRETRAINED_MODEL_PATH']} && tar -zxvf
{PRETRAINED_MODEL_NAME+'.tar.gz'}
```

יצירת label map, מטרתו לתת לכל מחלקה אינטגר המצביע עליו כאשר התמונות עוברים עיבוד נתונים.

```
label_map = [{'name':'cans', 'id':1}, {'name':'glass', 'id':2},
{'name':'paper', 'id':3}, {'name':'plastic', 'id':4}]

with open(files['LABELMAP'], 'w') as f:
    for label in label_map:
        f.write('item { \n')
        f.write('\tname:\'{ }\'\n'.format(label['name']))
        f.write('\tid:{ }\n'.format(label['id']))
        f.write('\n')
```

יצירת TF-Records, פורמט שבוא Tensorflow שומר מידע בינארי, יש ל TF-Records יתרונות רבים, החשובים מהם הוא צמצום המידע הנצרך.

```
ARCHIVE_FILES = os.path.join(paths['IMAGE_PATH'], 'archive.tar.gz')
if os.path.exists(ARCHIVE_FILES):
    !tar -zxvf {ARCHIVE_FILES}

if not os.path.exists(files['TF_RECORD_SCRIPT']):
    !git clone https://github.com/nicknochnack/GenerateTFRecord
    {paths['SCRIPTS_PATH']}

!python {files['TF_RECORD_SCRIPT']} -x
{os.path.join(paths['IMAGE_PATH'], 'train')} -l {files['LABELMAP']}
-o {os.path.join(paths['ANNOTATION_PATH'], 'train.record')}
!python {files['TF_RECORD_SCRIPT']} -x
{os.path.join(paths['IMAGE_PATH'], 'test')} -l {files['LABELMAP']} -o
{os.path.join(paths['ANNOTATION_PATH'], 'test.record')}
```

העברת מידע של ה-Base Model לדירקטורי של Training.

```
if os.name == 'posix':
    !cp {os.path.join(paths['PRETRAINED_MODEL_PATH'],
PRETRAINED_MODEL_NAME, 'pipeline.config')}
{os.path.join(paths['CHECKPOINT_PATH'])}
if os.name == 'nt':
    !copy {os.path.join(paths['PRETRAINED_MODEL_PATH'],
PRETRAINED_MODEL_NAME, 'pipeline.config')}
{os.path.join(paths['CHECKPOINT_PATH'])}
```

אימפורט למודג'לים חשובים שנדרשים לאימון המידע שלנו.

```
import object_detection
import tensorflow as tf
from object_detection.utils import config_util
from object_detection.protos import pipeline_pb2
from google.protobuf import text_format
```

מגדיר את ההגדרות ה"בסיסיות" לגבי השכבות ה-CNN שבו יועברו המידע שלנו, כגון מספר מחלקות, מאיפה לקחת מידע (איפה קבצים כמו label map נמצאים) וכו'.

```
config =
config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])
pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "r") as f:
    proto_str = f.read()
    text_format.Merge(proto_str, pipeline_config)
pipeline_config.model.ssd.num_classes = len(labels)
pipeline_config.train_config.batch_size = 4
pipeline_config.train_config.fine_tune_checkpoint =
os.path.join(paths['PRETRAINED_MODEL_PATH'], PRETRAINED_MODEL_NAME,
'checkpoint', 'ckpt-0')
pipeline_config.train_config.fine_tune_checkpoint_type = "detection"
pipeline_config.train_input_reader.label_map_path= files['LABELMAP']
pipeline_config.train_input_reader.tf_record_input_reader.input_path[
:] = [os.path.join(paths['ANNOTATION_PATH'], 'train.record')]
pipeline_config.eval_input_reader[0].label_map_path =
files['LABELMAP']
pipeline_config.eval_input_reader[0].tf_record_input_reader.input_path[
:] = [os.path.join(paths['ANNOTATION_PATH'], 'test.record')]
config_text = text_format.MessageToString(pipeline_config)
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "wb") as f:
    f.write(config_text)
```

עכשיו מתחיל החלק האמיתי, אימון המודל שלנו. ראשית הוגדר המיקום שבו נמצא הסקריפט של האימונים, שנלקח מ-Tensorflow, ולאחר מכן מגדירים את ההגדרות שה-Training Script יקבל, למשל כמה פעמים לעבור על התמונות (ככל שיותר צעדים בדרך כלל יותר מדויק (יותר מדי מוריד דיוק), אבל מגדיל את זמן האימון משמעותית).


```

TRAINING_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research',
'object_detection', 'model_main_tf2.py')
command = "python {} --model_dir={} --pipeline_config_path={}
--num_train_steps=10000".format(TRAINING_SCRIPT,
paths['CHECKPOINT_PATH'],files['PIPELINE_CONFIG'])
!{command}

```

פה הסקריפט אימונים של Tensorflow.

```

r"""Creates and runs TF2 object detection models.
For local training/evaluation run:
PIPELINE_CONFIG_PATH=path/to/pipeline.config
MODEL_DIR=/tmp/model_outputs
NUM_TRAIN_STEPS=10000
SAMPLE_1_OF_N_EVAL_EXAMPLES=1
python model_main_tf2.py -- \
    --model_dir=$MODEL_DIR --num_train_steps=$NUM_TRAIN_STEPS \
    --sample_1_of_n_eval_examples=$SAMPLE_1_OF_N_EVAL_EXAMPLES \
    --pipeline_config_path=$PIPELINE_CONFIG_PATH \
    --alsologtostderr
"""

from absl import flags
import tensorflow.compat.v2 as tf
from object_detection import model_lib_v2

flags.DEFINE_string('pipeline_config_path', None, 'Path to pipeline
config '
                    'file.')
flags.DEFINE_integer('num_train_steps', None, 'Number of train
steps.')
flags.DEFINE_bool('eval_on_train_data', False, 'Enable evaluating on
train '
                'data (only supported in distributed training).')
flags.DEFINE_integer('sample_1_of_n_eval_examples', None, 'Will
sample one of '
                    'every n eval input examples, where n is
provided.')
flags.DEFINE_integer('sample_1_of_n_eval_on_train_examples', 5, 'Will

```

```

sample '
                                'one of every n train input examples for
evaluation, '
                                'where n is provided. This is only used if '
                                '`eval_training_data` is True.')
flags.DEFINE_string(
    'model_dir', None, 'Path to output model directory '
                        'where event and checkpoint files will be
written.')
flags.DEFINE_string(
    'checkpoint_dir', None, 'Path to directory holding a checkpoint.
If '
    '`checkpoint_dir` is provided, this binary operates in eval-only
mode, '
    'writing resulting metrics to `model_dir`.')

flags.DEFINE_integer('eval_timeout', 3600, 'Number of seconds to wait
for an'
                        'evaluation checkpoint before exiting.')

flags.DEFINE_bool('use_tpu', False, 'Whether the job is executing on
a TPU.')
flags.DEFINE_string(
    'tpu_name',
    default=None,
    help='Name of the Cloud TPU for Cluster Resolvers.')
flags.DEFINE_integer(
    'num_workers', 1, 'When num_workers > 1, training uses '
    'MultiWorkerMirroredStrategy. When num_workers = 1 it uses '
    'MirroredStrategy.')
flags.DEFINE_integer(
    'checkpoint_every_n', 1000, 'Integer defining how often we
checkpoint.')
flags.DEFINE_boolean('record_summaries', True,
                    ('Whether or not to record summaries defined by
the model'
                     ' or the training pipeline. This does not
impact the'
                     ' summaries of the loss values which are

```

```

always'

                                'recorded.'))

FLAGS = flags.FLAGS

def main(unused_argv):
    flags.mark_flag_as_required('model_dir')
    flags.mark_flag_as_required('pipeline_config_path')
    tf.config.set_soft_device_placement(True)

    if FLAGS.checkpoint_dir:
        model_lib_v2.eval_continuously(
            pipeline_config_path=FLAGS.pipeline_config_path,
            model_dir=FLAGS.model_dir,
            train_steps=FLAGS.num_train_steps,

sample_1_of_n_eval_examples=FLAGS.sample_1_of_n_eval_examples,
            sample_1_of_n_eval_on_train_examples=(
                FLAGS.sample_1_of_n_eval_on_train_examples),
            checkpoint_dir=FLAGS.checkpoint_dir,
            wait_interval=300, timeout=FLAGS.eval_timeout)
    else:
        if FLAGS.use_tpu:
            # TPU is automatically inferred if tpu_name is None and
            # we are running under cloud ai-platform.
            resolver = tf.distribute.cluster_resolver.TPUClusterResolver(
                FLAGS.tpu_name)
            tf.config.experimental_connect_to_cluster(resolver)
            tf.tpu.experimental.initialize_tpu_system(resolver)
            strategy = tf.distribute.experimental.TPUStrategy(resolver)
        elif FLAGS.num_workers > 1:
            strategy =
tf.distribute.experimental.MultiWorkerMirroredStrategy()
        else:
            strategy = tf.compat.v2.distribute.MirroredStrategy()

    with strategy.scope():
        model_lib_v2.train_loop(

```

```

        pipeline_config_path=FLAGS.pipeline_config_path,
        model_dir=FLAGS.model_dir,
        train_steps=FLAGS.num_train_steps,
        use_tpu=FLAGS.use_tpu,
        checkpoint_every_n=FLAGS.checkpoint_every_n,
        record_summaries=FLAGS.record_summaries)

if __name__ == '__main__':
    tf.compat.v1.app.run()

```

אימפורט לעוד סיפריות, ואז הגשת המידע של המודל שיצרנו מה-checkpoint האחרון שלו(ככל שמאמנים יותר את המודל יתווספו יותר checkpoints, בתוכו נשמר כל המידע על מודל וכך ניתן להתחל אותו להמשיך אימונים עליו מהנקודה הזאת ואז נוצר עוד checkpoints). (כמות ה-checkpoints שנוצר תלוי בהגדרות שנותנים כל כמה steps ליצור אחד חדש).

לאחר אתחול ה-checkpoint האחרון יצרתי פונקציה שלוקח תמונה ומחזיר את הנתונים כל הנתונים בוא. פונקציה זה משמשת גם לזיהוי תמונה יחידה, גם סרטון וגם live.

```

from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.builders import model_builder
import cv2
import numpy as np
from matplotlib import pyplot as plt
# Load pipeline config
configs =
config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])
detection_model = model_builder.build(model_config=configs['model'],
is_training=False)

# Restore checkpoint
ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'],
'ckpt-3')).expect_partial()

```

```
@tf.function
def detect_fn(image):
    image, shapes = detection_model.preprocess(image)
    prediction_dict = detection_model.predict(image, shapes)
    detections = detection_model.postprocess(prediction_dict, shapes)
    return detections
```

פה אני מבצע בדיקה על תמונה אחת יחידה כדי לראות שהכל תקין. קודם כל מבצע פעולה שהופכת את התמונה לפורמט שהמודל יכול לקבל (אובייקט של Tensor שהוא מטריצה בעל מספר ממדים). אז אני יציתי לולאה על כל מה שמוחזר מה-detect_fn על התמונה שהועברה לה, ומקבלים בעצם מתריצה שבתוכה הערכים של ה-Bounding boxes, הערכים של המחלקות (ששמורות כאינטגרים שבתחילת המספר הזה יש את ה-ID של המחלקה שבוא הוא שייך, הוגדר ב-label map), ואת הערך של ה-score, כלומר מה האחוז שהמחשב חושב שהתמונה שהועברה לו שייכת למחלקה מסוימת.

אז מוגדר כמה הגדרות לגבי מה יופיע בתמונה, במקרה זה יכול להיות עד 5 bounding boxes, ורק יופיע אם אחוז הפרדיקציה גבוה מ-80% ועוד כמה הגדרות. אז לבסוף מציירים את התמונה ובוא כל ה-bounding boxes עם המחלקה שהם שייכים אל ואחוז הפרדיקציה על המחלקה.

```
%matplotlib inline
category_index =
label_map_util.create_category_index_from_labelmap(files['LABELMAP'])
IMAGE_PATH = os.path.join(paths['IMAGE_PATH'], 'test',
'cans.8afa1c66-b46f-11ec-b268-2cfda1e04706.jpg')
img = cv2.imread(IMAGE_PATH)
image_np = np.array(img)

input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0),
dtype=tf.float32)
detections = detect_fn(input_tensor)

num_detections = int(detections.pop('num_detections'))
detections = {key: value[0, :num_detections].numpy()
               for key, value in detections.items()}
```

```

detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] =
detections['detection_classes'].astype(np.int64)

label_id_offset = 1
image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections,
    detections['detection_boxes'],
    detections['detection_classes']+label_id_offset,
    detections['detection_scores'],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=5,
    min_score_thresh=.8,
    agnostic_mode=False)

plt.imshow(cv2.cvtColor(image_np_with_detections, cv2.COLOR_BGR2RGB))
plt.show()

```

פה בעצם קורא בדיוק אותו דבר פשוט על webcam של המחשב, כלומר live. אם מחליפים את ה-0 ב-PATH לסרטון זה יעבוד בדיוק אותו דבר.

```

cap = cv2.VideoCapture(0)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

while cap.isOpened():
    ret, frame = cap.read()
    image_np = np.array(frame)

    input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0),
    dtype=tf.float32)
    detections = detect_fn(input_tensor)

```

```

num_detections = int(detections.pop('num_detections'))
detections = {key: value[0, :num_detections].numpy()
               for key, value in detections.items()}
detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] =
detections['detection_classes'].astype(np.int64)

label_id_offset = 1
image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections,
    detections['detection_boxes'],
    detections['detection_classes']+label_id_offset,
    detections['detection_scores'],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=5,
    min_score_thresh=.8,
    agnostic_mode=False)

cv2.imshow('object detection',
cv2.resize(image_np_with_detections, (800, 600)))

if cv2.waitKey(10) & 0xFF == ord('q'):
    cap.release()
    cv2.destroyAllWindows()
    break

```

לאחר שמרוצים מהמודל, כדי להעביר אותו לפורמט ש-רסברי פאי יכול להשתמש בו, יש מספר שלבים שצריך לבצע. רשית כל הוא הקפעת המודל. הקפאת המודל פירושה הפקת קובץ יחיד המכיל מידע על הגרף ומשתני ה-checkpoint, אך שמירת היפר פרמטרים אלו כקבועים בתוך מבנה הגרף. זה מבטל מידע נוסף שנשמר בקבצי ה-checkpoint כגון השיפועים בכל נקודה, הכלולים כך שניתן לטעון את המודל מחדש ולהמשיך את האימונים מהמקום בו הופסק. כלומר בעצם גורם לכך שהמידע הלא "רלוונטי" לגבי המודל כאשר לא צריכים לאמן אותו יותר אינו נשמר, וכך מקבצים הרבה מידע.

```

FREEZE_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research',
'object_detection', 'exporter_main_v2.py ')
command = "python {} --input_type=image_tensor
--pipeline_config_path={} --trained_checkpoint_dir={}
--output_directory={}".format(FREEZE_SCRIPT
,files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'],
paths['OUTPUT_PATH'])
!{command}

```

כדי לעשות הקפאה למודל השתמשתי עוד פעם בסקריפט של Tensorflow.

```

r"""Tool to export an object detection model for inference.
Prepares an object detection tensorflow graph for inference using
model
configuration and a trained checkpoint. Outputs associated checkpoint
files,
a SavedModel, and a copy of the model config.
The inference graph contains one of three input nodes depending on
the user
specified option.
    * `image_tensor`: Accepts a uint8 4-D tensor of shape [1, None,
None, 3]
    * `float_image_tensor`: Accepts a float32 4-D tensor of shape
[1, None, None, 3]
    * `encoded_image_string_tensor`: Accepts a 1-D string tensor of
shape [None]
        containing encoded PNG or JPEG images. Image resolutions are
expected to be
        the same if more than 1 image is provided.
    * `tf_example`: Accepts a 1-D string tensor of shape [None]
containing
        serialized TFExample protos. Image resolutions are expected to be
the same
        if more than 1 image is provided.
    * `image_and_boxes_tensor`: Accepts a 4-D image tensor of size
[1, None, None, 3] and a boxes tensor of size [1, None, 4] of
normalized
        bounding boxes. To be able to support this option, the model
needs

```


to implement a `predict_masks_from_boxes` method. See the documentation for `DetectionFromImageAndBoxModule` for details.

and the following output nodes returned by the `model.postprocess(..)`:

- * ``num_detections``: Outputs float32 tensors of the form [batch] that specifies the number of valid boxes per image in the batch.
- * ``detection_boxes``: Outputs float32 tensors of the form [batch, num_boxes, 4] containing detected boxes.
- * ``detection_scores``: Outputs float32 tensors of the form [batch, num_boxes] containing class scores for the detections.
- * ``detection_classes``: Outputs float32 tensors of the form [batch, num_boxes] containing classes for the detections.

Example Usage:

```

-----
python exporter_main_v2.py \
    --input_type image_tensor \
    --pipeline_config_path path/to/ssd_inception_v2.config \
    --trained_checkpoint_dir path/to/checkpoint \
    --output_directory path/to/exported_model_directory \
    --use_side_inputs True/False \
    --side_input_shapes
dim_0,dim_1,...dim_a/.../dim_0,dim_1,...,dim_z \
    --side_input_names name_a,name_b,...,name_c \
    --side_input_types type_1,type_2

```

The expected output would be in the directory `path/to/exported_model_directory` (which is created if it does not exist) holding two subdirectories (corresponding to checkpoint and SavedModel, respectively) and a copy of the pipeline config. Config overrides (see the ``config_override`` flag) are text protobufs (also of type `pipeline_pb2.TrainEvalPipelineConfig`) which are used to override certain fields in the provided `pipeline_config_path`. These are useful for making small changes to the inference graph that differ from the training or eval config.

Example Usage (in which we change the second stage post-processing score

threshold to be 0.5):

```
python exporter_main_v2.py \
    --input_type image_tensor \
    --pipeline_config_path path/to/ssd_inception_v2.config \
    --trained_checkpoint_dir path/to/checkpoint \
    --output_directory path/to/exported_model_directory \
    --config_override " \
        model{ \
            faster_rcnn { \
                second_stage_post_processing { \
                    batch_non_max_suppression { \
                        score_threshold: 0.5 \
                    } \
                } \
            } \
        }"

```

If side inputs are desired, the following arguments could be appended (the example below is for Context R-CNN).

```
--use_side_inputs True \
--side_input_shapes 1,2000,2057/1 \
--side_input_names context_features,valid_context_size \
--side_input_types tf.float32,tf.int32
"""

```

```
from absl import app
from absl import flags

```

```
import tensorflow.compat.v2 as tf
from google.protobuf import text_format
from object_detection import exporter_lib_v2
from object_detection.protos import pipeline_pb2

```

```
tf.enable_v2_behavior()

```

```
FLAGS = flags.FLAGS

```

```
flags.DEFINE_string('input_type', 'image_tensor', 'Type of input

```

```

node. Can be '
        'one of [`image_tensor`,
`encoded_image_string_tensor`, '
        '`tf_example`, `float_image_tensor`, '
        '`image_and_boxes_tensor`]')
flags.DEFINE_string('pipeline_config_path', None,
        'Path to a pipeline_pb2.TrainEvalPipelineConfig
config '
        'file.')
flags.DEFINE_string('trained_checkpoint_dir', None,
        'Path to trained checkpoint directory')
flags.DEFINE_string('output_directory', None, 'Path to write
outputs.')
flags.DEFINE_string('config_override', '',
        'pipeline_pb2.TrainEvalPipelineConfig '
        'text proto to override pipeline_config_path.')
flags.DEFINE_boolean('use_side_inputs', False,
        'If True, uses side inputs as well as image
inputs.')
flags.DEFINE_string('side_input_shapes', '',
        'If use_side_inputs is True, this explicitly sets
'
        'the shape of the side input tensors to a fixed
size. The '
        'dimensions are to be provided as a
comma-separated list '
        'of integers. A value of -1 can be used for
unknown '
        'dimensions. A `/` denotes a break, starting the
shape of '
        'the next side input tensor. This flag is
required if '
        'using side inputs.')
flags.DEFINE_string('side_input_types', '',
        'If use_side_inputs is True, this explicitly sets
'
        'the type of the side input tensors. The '
        'dimensions are to be provided as a
comma-separated list '

```

```

        'of types, each of `string`, `integer`, or
`float`. '
        'This flag is required if using side inputs.')
flags.DEFINE_string('side_input_names', '',
                    'If use_side_inputs is True, this explicitly sets
the names of the side input tensors required by
the model '
                    'assuming the names will be a comma-separated
list of '
                    'strings. This flag is required if using side
inputs.')
```

```

flags.mark_flag_as_required('pipeline_config_path')
flags.mark_flag_as_required('trained_checkpoint_dir')
flags.mark_flag_as_required('output_directory')
```

```

def main(_):
    pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
    with tf.io.gfile.GFile(FLAGS.pipeline_config_path, 'r') as f:
        text_format.Merge(f.read(), pipeline_config)
    text_format.Merge(FLAGS.config_override, pipeline_config)
    exporter_lib_v2.export_inference_graph(
        FLAGS.input_type, pipeline_config,
        FLAGS.trained_checkpoint_dir,
        FLAGS.output_directory, FLAGS.use_side_inputs,
        FLAGS.side_input_shapes,
        FLAGS.side_input_types, FLAGS.side_input_names)
```

```

if __name__ == '__main__':
    app.run(main)
```

ואז לשלב אחרון, כדי שיעבוד על הרסברי פאי צריך להפוך את המודל שלנו לפורמט בשם TFLite, פורמט שמיועד למכשירי סלולר, מיקרוקונטרולרים, ומכשירים אחרים בעלי זכרות, GPU ו-CPU חלשות.

```

TFLITE_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research',
'object_detection', 'export_tflite_graph_tf2.py ')
command = "python {} --pipeline_config_path={}
--trained_checkpoint_dir={}
--output_directory={}".format(TFLITE_SCRIPT
,files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'],
paths['TFLITE_PATH'])

!{command}

FROZEN_TFLITE_PATH = os.path.join(paths['TFLITE_PATH'],
'saved_model')
TFLITE_MODEL = os.path.join(paths['TFLITE_PATH'], 'saved_model',
'detect.tflite')
command = "tflite_convert \
--saved_model_dir={} \
--output_file={} \
--input_shapes=1,300,300,3 \
--input_arrays=normalized_input_image_tensor \
--output_arrays='TFLite_Detection_PostProcess','TFLite_Detection_Post
Process:1','TFLite_Detection_PostProcess:2','TFLite_Detection_PostPro
cess:3' \
--inference_type=FLOAT \
--allow_custom_ops".format(FROZEN_TFLITE_PATH, TFLITE_MODEL, )

!{command}

```

עכשיו שיש פורמט שהרסברי פאי יכול לעבוד איתו, פשוט צריך להעביר את הקובץ TFLite לרסבראי פאי, בנוסף פקובץ שכתוב בו את כל הלייבלים שמופיעים, ולהריץ את הקוד בדומה למה שהורץ על webcam מהמחשב.

```

import re
import cv2
from tflite_runtime.interpreter import Interpreter
import numpy as np

```

```

class ObjectDetection:

    def __init__(self, width, height):
        self.width = width
        self.height = height

    @staticmethod
    def load_labels(path='labels.txt'):
        #Loads the labels file. Supports files with or without index numbers.
        with open(path, 'r', encoding='utf-8') as f:
            lines = f.readlines()
            labels = {}
            for row_number, content in enumerate(lines):
                pair = re.split(r'[:\s]+', content.strip(),
maxsplit=1)
                if len(pair) == 2 and pair[0].strip().isdigit():
                    labels[int(pair[0])] = pair[1].strip()
                else:
                    labels[row_number] = pair[0].strip()
            return labels

    @staticmethod
    def set_input_tensor(interpreter, image):
        #Sets the input tensor.
        tensor_index = interpreter.get_input_details()[0]['index']
        input_tensor = interpreter.tensor(tensor_index)()[0]
        input_tensor[:, :] = np.expand_dims((image-255)/255, axis=0)

    @staticmethod
    def get_output_tensor(interpreter, index):
        #Returns the output tensor at the given index.
        output_details = interpreter.get_output_details()[index]
        tensor =
np.squeeze(interpreter.get_tensor(output_details['index']))
        return tensor

    def detect_objects(self, interpreter, image, threshold):
        #Returns a list of detection results, each a dictionary of

```

object info.

```
self.set_input_tensor(interpreter, image)
interpreter.invoke()
# Get all output details
boxes = self.get_output_tensor(interpreter, 0)
classes = self.get_output_tensor(interpreter, 1)
scores = self.get_output_tensor(interpreter, 2)
count = int(self.get_output_tensor(interpreter, 3))

results = []
for i in range(count):
    if scores[i] >= threshold:
        result = {
            'bounding_box': boxes[i],
            'class_id': classes[i],
            'score': scores[i]
        }
        results.append(result)
return results

def main_func(self):
    labels = self.load_labels()
    interpreter = Interpreter('detect.tflite')
    interpreter.allocate_tensors()
    _, input_height, input_width, _ =
interpreter.get_input_details()[0]['shape']

    cap = cv2.VideoCapture(0)
    while cap.isOpened():
        ret, frame = cap.read()
        img = cv2.resize(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB),
(320,320))
        res = self.detect_objects(interpreter, img, 0.8)
        print(res)

        for result in res:
            ymin, xmin, ymax, xmax = result['bounding_box']
            xmin = int(max(1,xmin * self.CAMERA_WIDTH))
            xmax = int(min(self.CAMERA_WIDTH, xmax *

```

```

self.CAMERA_WIDTH))
        ymin = int(max(1, ymin * self.CAMERA_HEIGHT))
        ymax = int(min(self.CAMERA_HEIGHT, ymax *
self.CAMERA_HEIGHT))

        cv2.rectangle(frame,(xmin, ymin),(xmax,
ymax),(0,255,0),3)

cv2.putText(frame,labels[int(result['class_id'])],(xmin, min(ymax,
self.CAMERA_HEIGHT-20)), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,255,255),2,cv2.LINE_AA)

        cv2.imshow('Pi Feed', frame)

        if cv2.waitKey(10) & 0xFF ==ord('q'):
            cap.release()
            cv2.destroyAllWindows()

def main():
    default_cam = ObjectDetection(640, 480)
    default_cam.main_func()

if __name__ == "__main__":
    main()

```


מרכז הבקרה-AKA האתר

מכיוון שרצינו שכל אחד יוכל לשלוט על הרובוט שלנו, החלטנו ששימוש באינטרנט הוא הדרך הנכונה ביותר. לשם כך יצרתי אתר, הנראה כך:



יש נביגיישן בר שברגע זה לא מצביע על שום מקום כי החלטנו להשאיר את הדברים המינורים לרגע האחרון, אז קצת על עצמנו ועל הפרויקט לא הוספנו עוד, וגם לא הוספנו קישורים לאתרים חיצוניים אבל לעשות זאת מאוד פשוט.

מתחת לנביגיישן בר יש סרטון, סרטון זה כרגע זמני, אנחנו נחליף אותו עם קישור ליוטיוב לייב של המצלמה מהפרויקט(הקוד עובד, רק יש באג עם המצלמה שצריך לנסות לתקן). מתחת לסרטון לייב, יש קונטרולים, חצים לכל כיוון. כאשר בן אדם לוחץ על אחד הכפתורים, בטקסט האפור ישתנה בו זמנית לכל בן אדם שמסתכל באתר, ובטקסט הסגול מופיע רק מה שהמשתמש הצופה לחץ על. בנוסף יש עוד טקסט נסתר שאליו נדבר בהמשך.

מתחת לקונטרולר יש קאונטרים של מספר דברים כגון כמות אשפה שהושלכה בעולם, כמה זמן עד שלא ישארו דגים בים, כמות הפחמן דו חמצני שמשוחרר לאטמוספירה וכו', כל אלו מעודכנים בזמן אמת.

HTML

קודם כל ב HTML, מתחילים עם ההגדרות. השורה הראשון מגדירה את הקוד כ HTML, שהוא לא בדיוק קוד אלא שפת Markup. אחר מכן הגדרת שפה של העמוד, הכותרת של העמוד, את ה-Character set שיכול להופיע בעמוד (utf-8) הוא הסטנדרט ברוב עמודים. ההגדרה של ה-Character set עוזרת ל-Browser לקרוא את העמוד. לאחר מכן מוגדר שכל מה שבעמוד ישתנה בהתאם לרוחב של המסך הצופה(רק עובד אם בעצבים את העמוד ב-CSS כמו שצריך). אז לאחר מכן אני הגדרתי לינקים ל-CSS, JS, פונטים מאתרים באינטרנט לעזור עם עיצוב, וגם הוספתי את ה-JS ו-CSS של עצמי.

```
#HTML
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Tanjirobot</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <!-- bootstrap CDN -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap
.min.css" rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.b
undle.min.js"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js
"></script>
  <link rel="stylesheet" href="static/css/style.css">
  <script src="static/script/script.js"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display
=swap" rel="stylesheet">
</head>
```

עכשיו יש את הגוף של בעמוד, מה שבאמת מופיע ומה שמשתמשים באמת מתעסקים עם ה-Frontend. דבר ראשון הגדרתי נביגישן באר, ברגע זה הלינקים לא מובילים לשום מקום, אבל אנחנו משאירים את ייצור ה-About us, Sources, לרגע אחרון מכיוון שאינו חשובים מכיוון שיש לנו את החוברת.

```

#HTML
<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark p-3">
        <div class="container-fluid">
            <!-- navbar-brand in the <a> tag is used to highlight the
brand/Logo/project name of the page-->
            <a class="navbar-brand" href="#">Tanjirobot</a>
            <!--
                this button is used, when the device is too small or
too zoomed in, will replace the navbar
                with a button in the top right that opens a dropdown
menu for the links
            -->
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class=" collapse navbar-collapse"
id="collapsibleNavbar">
                <ul class="navbar-nav ms-auto ">
                    <li class="nav-item">
                        <a class="nav-link mx-2" href="#">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link mx-2" href="#">Control</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link mx-2" href="#">About Us</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link mx-2 dropdown-toggle" href="#"
id="navbarDropdownMenuLink" role="button"
data-bs-toggle="dropdown">Sources</a>
                        <ul class="dropdown-menu" >
                            <li><a class="dropdown-item"
href="#">HBarber</a></li>
                            <li><a class="dropdown-item"

```

```

href="#">Bebot</a></li>
        <li><a class="dropdown-item" href="#">...</a></li>
    </ul>
</li>
</ul>
</div>
</div>
</nav>

```

```

#HTML
<div class="container">
    <iframe width="560" height="315"
src="https://www.youtube.com/embed/9UMxZofMNbA" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
</div>

```

```

#HTML
<div class="container">

    <div>
        <p class="text2"></p>
        <p class="text3" id="importdata"></p>
        <p class="text1"> Local Direction: </p>
    </div>

    <div class="controller">
        <button class="control up" title="Up" aria-label="Up" id="Up"
name="button" style="font-family: 'Press Start 2P', cursive;
font-size: 12px;"> UP </button>
        <button class="control right" title="Right"
aria-label="Right" id="Right" name="button" style="font-family:
'Press Start 2P', cursive; font-size: 12px;"> RIGHT </button>
        <button class="control down" title="Down" aria-label="Down"
id="Down" name="button" style="font-family: 'Press Start 2P',
cursive; font-size: 12px;"> DOWN </button>
        <button class="control left" title="Left" aria-label="Left"

```

```
id="Left" name="button" style="font-family: 'Press Start 2P',
cursive; font-size: 12px;"> LEFT </button>
</div>
</div>
```

JAVASCRIPT

מכיוון שלא רציתי שהאתר יתען מחדש כל פעם שמישהו לוחץ על כפתור, הייתי צריך לחקור איך לעדכן את האתר בצורה דינמית. אחרי הרבה מאמצים מצאתי והצלחתי לשלב AJAX, סיפריה לJS שמטרתה לאפשר לשליחת מידע אסינכרונית(מאחורי הכלים) על חלק מהאתר, כדי שלא יתבצע refresh. אז מה קורה פה? בעצם יצרתי פונקציה ב-JS שמוצאת את כל הכפתורים לפי שכל הכפתורים מצביעים על השם "button", ואז אני יוצר קריאה ל-AJAX, אני מגדיר ל-AJAX את ה-URL שבוא הוא שולח את הפקודה אל, כדי שהסרבר יוכל להכין באיזה פעולה קראה בעצם, מגדיר שאת המידע הזה הוא שולח לסרבר על ידי "POST", והמידע שמועבר הוא הכיוון, שנקבע על פי כך ש\$(this) מצביע על האובייקט שנלחץ, כלומר מתוך הארבע כפתורים לכל אחד יש אובייקט משלו, אז הפונקציה לוקחת את האובייקט של הכפתור שנלחץ, מחלצת את ה-ID של הכפתור המסוים ושולחת אותו לסרבר. אם המידע נשלח בהצלחה אז הטסקט בסגול משתנה בהתאם לכיוון שנלחץ.

```
#Javascript
$(document).ready(function() {
    $( "button[name='button']" ).on('click', function() {
        $.ajax({
            url: '/clicks',
            type: 'post',
            data: {
                direction: $(this).attr("id")
            },
            success: function(response){
                $(".text1").text('Local Direction: ' + response);
            }
        })
    })
})
```

SERVER

כדי לקבל את המידע ולשמור אותו, יצרתי שרת, ו-Database. ל-Database יש טבלה אחת, ובטבלה שתי טורים, אחד של ID, ואחד של כיוון.

```

#Server side script
from flask import Flask
from flask import render_template
from flask import request
from flask import redirect
from flask import url_for
from flask import jsonify
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config["DEBUG"] = True

SQLALCHEMY_DATABASE_URI =
"mysql+mysqlconnector://{username}:{password}@{hostname}/{databasename}"
.format(
    username="PeterPorker",
    password="Laika246E@",
    hostname="PeterPorker.mysql.pythonanywhere-services.com",
    databasename="PeterPorker$clicks",
)
app.config["SQLALCHEMY_DATABASE_URI"] = SQLALCHEMY_DATABASE_URI
app.config["SQLALCHEMY_POOL_RECYCLE"] = 299
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False

db = SQLAlchemy(app)

class ButtonClick(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    content = db.Column(db.String(5))

@app.route('/')
def index():
    return render_template('index.html')

```

פה אני מגדיר שרק-ווסט שמתקבל, שהוא רק יכול להיות מסוג POST(כלומר השרת מקבל מידע מן המשתמש) שמגיע מהURL של האתר עם התוספת של 'clicks' (שזה מה שהגדרתי ב-JS כה-URL שנשלח) יכניס את המידע, כלומר הכיוון ל-Database, ומחזיר בפורמט של jsonify, את הכיוון שנכנס

כרגע ל-Database, ריספונס זה מתקבל בפעולת ההצלחה ב-Javascript ומעדכן את הטקסט הסגול לכיוון שהמשתמש לחץ (מראה כיוון מקומי, לא גלובלי).

```
#Server side script
@app.route('/clicks', methods=['POST'])
def clicks():
    if request.method == 'POST':
        direc = ButtonClick(content=request.form.get('direction'))
        db.session.add(direc)
        db.session.commit()
        button_direction =
        ButtonClick.query.order_by(ButtonClick.id.desc()).first()
        print(button_direction.content)
        return jsonify(button_direction.content)
```

כדי לקבל את הכיוון האחרון של כלל המשתמשים, יש צורך בלבקש את המידע הזה מהסרבר באופן קבוע. מה שקורה כאן הוא ריק-ווסט מסוג "GET", כלומר מבקש מידע מן הסרבר, ה-URL שמבצע את הריק-ווסט הוא "/data". כאשר המידע מתקבל אז נשלח לקונסולה ולטקסט האפור את הכיוון האלרון שנלחץ, ולטקסט נוסף נסתר נשלח גם כיוון וגם ID. הבקשה נשלחת לשרט כל 0.05 שניות.

```
#Javascript
setInterval(function() {
    $.ajax({
        type: "GET",
        url: "/data",
    })
    .done(function(data) {
        console.log(data);
        $(".text2").html('Last Direction From All Users: ' +
        data.direction);
        $(".text3").html(data.direction + ', ' + data.id);
    })
    }, 1000 * 0.05);
```

כאשר מתבצע הריק-ווסט למידע של הכיוון ו-ID, ה-URL שנשלח הוא "/data", הריק-ווסט היחיד שיכול להתקבל הוא מסוג "GET", כלומר הסרבר נותן למשתמש מידע. מה שקורה פה הוא שהסרבר מבקש מה-Database את הכיוון ו-ID האחרונים שצורפו אליו. מידע זה נשלח בפורמט jsonify חזרה ל-Browser.

```

#Server side script
@app.route('/data', methods=['GET'])
def data():
    if request.method == 'GET':
        instance =
ButtonClick.query.order_by(ButtonClick.id.desc()).first()
        instance_data = instance.content
        instance_id_data = instance.id
        return jsonify({'direction': instance_data, 'id':
instance_id_data})

if __name__ == '__main__':
    app.run()

```

יצירת קאונטרים של כמה דברי מידע לגבי המצב העולם, נלקח מהאתר

[/https://www.theworldcounts.com](https://www.theworldcounts.com)

```

#HTML
<br>
<br>
<div class "container">
    <iframe title='Tons of solid waste generated'
src='https://www.theworldcounts.com/embed/challenges/116?background_c
olor=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+s
ans-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='Tons of waste dumped'
src='https://www.theworldcounts.com/embed/challenges/63?background_co
lor=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sa
ns-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='Tons of CO2 emitted into the atmosphere'
src='https://www.theworldcounts.com/embed/challenges/23?background_co
lor=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sa
ns-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='World average temperature (°C)'
src='https://www.theworldcounts.com/embed/challenges/21?background_co
lor=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sa

```



```

ns-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='Rise in sea levels (cm)'
src='https://www.theworldcounts.com/embed/challenges/68?background_color=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sans-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='Time left till the end of rainforests'
src='https://www.theworldcounts.com/embed/challenges/114?background_color=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sans-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    <iframe title='Time left till no more fish in the sea'
src='https://www.theworldcounts.com/embed/challenges/115?background_color=white&color=black&font_family=%22Helvetica+Neue%22%2C+Arial%2C+sans-serif&font_size=14' style='border: none' height='100'
width='300'></iframe>
    </div>
</div>

</body>
</html>

```

עיצוב האתר:

```

#CSS
.text1 {
    cursor: default;
    user-select: none;
    font-family: "Great Vibes" cursive;
    line-height: 160px;
    font-weight: normal;
    color: #864dd5;
    font-size: 24px;
    margin-top: 20px;
    text-shadow: 0 1px 1px #fff;
}

```

```

.text2 {
  cursor: default;
  user-select: none;
  font-family: "Libre Baskerville", serif;
  line-height: 45px;
  color: #7a7c7f;
  font-size: 24px;
  margin-top: 20px;
  text-shadow: 0 1px 1px #fff;
}

.text3 {
  cursor: default;
  user-select: none;
  font-style: italic;
  color: #B6B6B6;
  font-size: 24px;
  margin-top: 20px;
  text-shadow: 0 1px 1px #fff;
  display: none;
}

.arcade {
  font-family: 'ArcadeClassic', sans-serif;
  cursor: default;
  user-select: none;
  text-align: center;
  font-size: 40px;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {

```

```

    line-height: 1.5;
    font-family: system-ui;
}

.container {
    display: flex;
    align-items: center;
    flex-wrap: wrap;
    justify-content: center;
}

.container p {
    font-size: 1.5rem;
}

.controller {
    inline-size: 20rem;
    position: relative;
}

.control {
    position: absolute;
    background-color: blue;
    color: white;
    border: none;
    font: inherit;
    inline-size: 6rem;
}

.controller, .control {
    aspect-ratio: 1 / 1;
}

.control:is(.up, .down) {
    inset-inline-start: 50%;
    transform: translateX(-50%);
}

.control.up {

```

```

inset-block-start: 0;
clip-path: polygon(50% 0%, 0% 100%, 100% 100%);
}

.control.down {
  inset-block-end: 0;
  clip-path: polygon(50% 100%, 100% 0, 0 0);
}

.control:is(.left, .right) {
  inset-block-start: 50%;
  transform: translateY(-50%);
}

.control.right {
  inset-inline-end: 0;
  clip-path: polygon(100% 50%, 0 0, 0 100%);
}

.control.left {
  clip-path: polygon(0 50%, 100% 0, 100% 100%);
}

.control:is(:focus-visible, :active) {
  opacity: 0.5;
}

```

קבלת מידע מהאתר והפעלת המנועים עם המידע על הרספברי פאי:

כדי לקחת את המידע מהאתר שלי אני משתמש בסיפריה בשם selenium. יצרתי מחלקה, ובתול המחלקה יש מודג'ולים, המודג'ולה הראשונה גורמת לכך שכאשר הסקריפט פועל, לא יפתח חלון של browser, המודג'ולה השנייה בודקת את סוג מערכת ההפעלה מכיוון שזה משפיע על סוג ה-browser שמשתמשים, ועל מיקום הדרייבר של אותו browser. המודג'ולה השלישית מחפשת את האובייקט של טקסט שנסתר בתוך האתר וקורא את המידע שלו. בהגדרת המחלקה כל המודג'ולים נקראים, כלומר מופעלים באתר יצירת אובייקט של DataSalvage.

```

import os
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By

```

```

#Data Scrape
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.options import Options
import time

#is_headless and check_os and get_data cant be static methods since
they require use of instance variables from each other.
class DataSalvage:

    chrome_options = Options()

    def __init__(self, url, headless=None):
        self.url = url
        self.headless = headless
        self.is_headless()
        self.check_os()
        self.get_data()

    def is_headless(self):
        if self.headless != None:
            self.chrome_options = Options()
            self.chrome_options.add_argument("--headless")

self.chrome_options.add_argument("--window-size=1920x1080")
self.chrome_options.add_argument("--log-level=3")

    def check_os(self):
        if os.name == 'nt':
            self.driver = os.getcwd() + "/chromedriver"
        else:
            self.driver = "/usr/lib/chromium-browser/chromedriver"
        self.driver =
webdriver.Chrome(chrome_options=self.chrome_options,
executable_path=self.driver)
        self.driver.get(self.url)

    def get_data(self):
        time.sleep(0.1)

```

```

try:
    self.element = WebDriverWait(self.driver,
10).until(EC.presence_of_element_located((By.CLASS_NAME, "text3")))
except:
    self.driver.close()
    self.element = self.element.get_attribute("textContent")
return self.element

```

הגדרת המנועים על ידי הסיפריה RPi.GPIO. קביעת הפינים המשתייכים ל-ENA, ENB, INP1, INP2, INP3, INP4 למנועי הנעה, ו-ENA, INP1 ו-INP2 למנוע של המסוע. הגדרת כל הפינים האלו כ-OUTPUT במודג'ולה של gpio_setup, הגדרת PWM ב-pwm_setup בנוסף יש מודג'לים של כיוון קדימה, אחורה, ימינה, שמאלה, עצור, התחלת מסוע, עצירת מסוע, ושחרור כל ה-GPIO's (פעולה שכדאי לעשות אחרי שמסיימים להשתמש ב-GPIO's)

```

# Importing Libraries
import RPi.GPIO as GPIO

class MotorActivation:
    PWMPin = 26 # PWM Pin connected to ENA.
    Motor1 = 20 # Connected to Input 1.
    Motor2 = 21 # Connected to Input 2.
    Motor3 = 6
    Motor4 = 13
    PWMPin2 = 12
    MotorTop1 = 17
    MotorTop2 = 27
    PWMMotortop = 22
    def __init__(self):
        self.gpio_setup()
        self.pwm_setup()

    def gpio_setup(self):
        GPIO.setwarnings(False)
        GPIO.setmode(GPIO.BCM) # BCM-GPIO Locations, Board- Physical
Locations GPIO

```

```

GPIO.setup(self.PWMPin, GPIO.OUT)
GPIO.setup(self.Motor1, GPIO.OUT)
GPIO.setup(self.Motor2, GPIO.OUT)
GPIO.setup(self.PWMPin2, GPIO.OUT)
GPIO.setup(self.Motor3, GPIO.OUT)
GPIO.setup(self.Motor4, GPIO.OUT)
GPIO.setup(self.MotorTop1, GPIO.OUT)
GPIO.setup(self.MotorTop2, GPIO.OUT)
GPIO.setup(self.PWMMotortop, GPIO.OUT)

def pwm_setup(self):
    PwmValue = GPIO.PWM(self.PWMPin, 2000) # PWM frequency to
2000.
    PwmValue2 = GPIO.PWM(self.PWMPin2, 2000) # PWM frequency to
2000.
    PwmValueTop = GPIO.PWM(self.PWMMotortop, 2000)
    PwmValue.start(75)
    PwmValue2.start(75)
    PwmValueTop.start(80)

def belt_start(self):
    GPIO.output(self.MotorTop1, GPIO.HIGH)
    GPIO.output(self.MotorTop2, GPIO.LOW)

def belt_stop(self):
    GPIO.output(self.MotorTop1, GPIO.LOW)
    GPIO.output(self.MotorTop2, GPIO.LOW)

def motor_clockwise(self):
    GPIO.output(self.Motor1, GPIO.HIGH)
    GPIO.output(self.Motor2, GPIO.LOW)
    GPIO.output(self.Motor3, GPIO.HIGH)
    GPIO.output(self.Motor4, GPIO.LOW)

```

```

def motor_left(self):
    self.PwmValue.ChangeDutyCycle(80)
    self.PwmValue2.ChangeDutyCycle(100)
    GPIO.output(self.Motor1, GPIO.LOW)
    GPIO.output(self.Motor2, GPIO.HIGH)
    GPIO.output(self.Motor3, GPIO.HIGH)
    GPIO.output(self.Motor4, GPIO.LOW)

def motor_right(self):
    self.PwmValue.ChangeDutyCycle(100)
    self.PwmValue2.ChangeDutyCycle(80)
    GPIO.output(self.Motor1, GPIO.HIGH)
    GPIO.output(self.Motor2, GPIO.LOW)
    GPIO.output(self.Motor3, GPIO.LOW)
    GPIO.output(self.Motor4, GPIO.HIGH)

def motor_anti_clockwise(self):
    GPIO.output(self.Motor1, GPIO.LOW)
    GPIO.output(self.Motor2, GPIO.HIGH)
    GPIO.output(self.Motor3, GPIO.LOW)
    GPIO.output(self.Motor4, GPIO.HIGH)

def motor_stop(self):
    GPIO.output(self.Motor1, GPIO.LOW)
    GPIO.output(self.Motor2, GPIO.LOW)
    GPIO.output(self.Motor3, GPIO.LOW)
    GPIO.output(self.Motor4, GPIO.LOW)

@staticmethod
def release_gpios():
    GPIO.cleanup()

```


כדי לחבר את קריאת המידע והפעלת המנועים ביחד החנתי סקריפט main, שבו מובא כל אחד מן המחלקות מהקבצים האחרים.

בתוך הסקריפט main יש עוד מחלקה, המחלקה העיקרית שבו מוגדר משתנה מחלקה(משתנה משותף לכל אובייקט שנוצר מן המחלקה הזאת) שבו פשוט כיוונים.

בתוך מחלקת Activate מאותחל ב-__init__ את ה-URL של האתר, האם לעשות אותו בלי לפתוח browser, אובייקט של DataSalvage שמקבל את הקישור בולטן של כן או לא לגבי פתיחת ה-browser ויצירת אובייקט של MotorActivation.

בתוך המחלקה יש מודג'לה של לקיחת המידע מן האתר-live_data, שמחזיר ID וכיוון.

אז יש את הלולאה העיקרית, שמגדירה קודם כל מצב התחלתי של ID, כדי לדעת אם התקיים שינוי, ואז מבצע לולאה תמידית שבו הוא קורא את המידע מן האתר, בודק אם ה-ID השתנה כדי לדעת אם התקיים לחיצה חדשה, ואם כן, משווה את הכיוון המתקבל לכיוונים מ-directions_list, ולפי כך מפעיל את המנועים לפי הדרישה המתאימה. המנועים פועלים למספר שניות ואז נעצרים, ובמקרה שמפסיקה את הסקריפט אז כל ה-GPIO's משוחררים.

```
from webscrapse import DataSalvage
from motor import MotorActivation
import time

class Activate:
    directions_list = ['Up', 'Down', 'Left', 'Right']

    def __init__(self, website, headless):
        self.website = website
        self.headless = headless
        self.site = DataSalvage(website, headless)
        self.motors = MotorActivation()

    def live_data(self):
        self.result = self.site.get_data()
        self.split = self.result.split(',')
        self.direction = self.split[0]
        self.id = self.split[1]
        return self.id, self.direction #return self.split[0],
self.split[1]

    def main_loop(self):
        self.old_id = self.live_data()[0]
        try:
```

```

while True:
    time.sleep(0.1)
    self.new_id, self.direction = self.live_data()

    if self.old_id != self.new_id:
        self.old_id = self.new_id
        print(self.direction)
        if self.direction == self.directions_list[0]:
            self.motors.motor_clockwise()
            print("Up")
        elif self.direction == self.directions_list[1]:
            print("Down")
            self.motors.motor_anti_clockwise()
        elif self.direction == self.directions_list[2]:
            print("Left")
            self.motors.motor_left()
        elif self.direction == self.directions_list[3]:
            print("Right")
            self.motors.motor_right()
    time.sleep(5)
    self.motors.motor_stop()

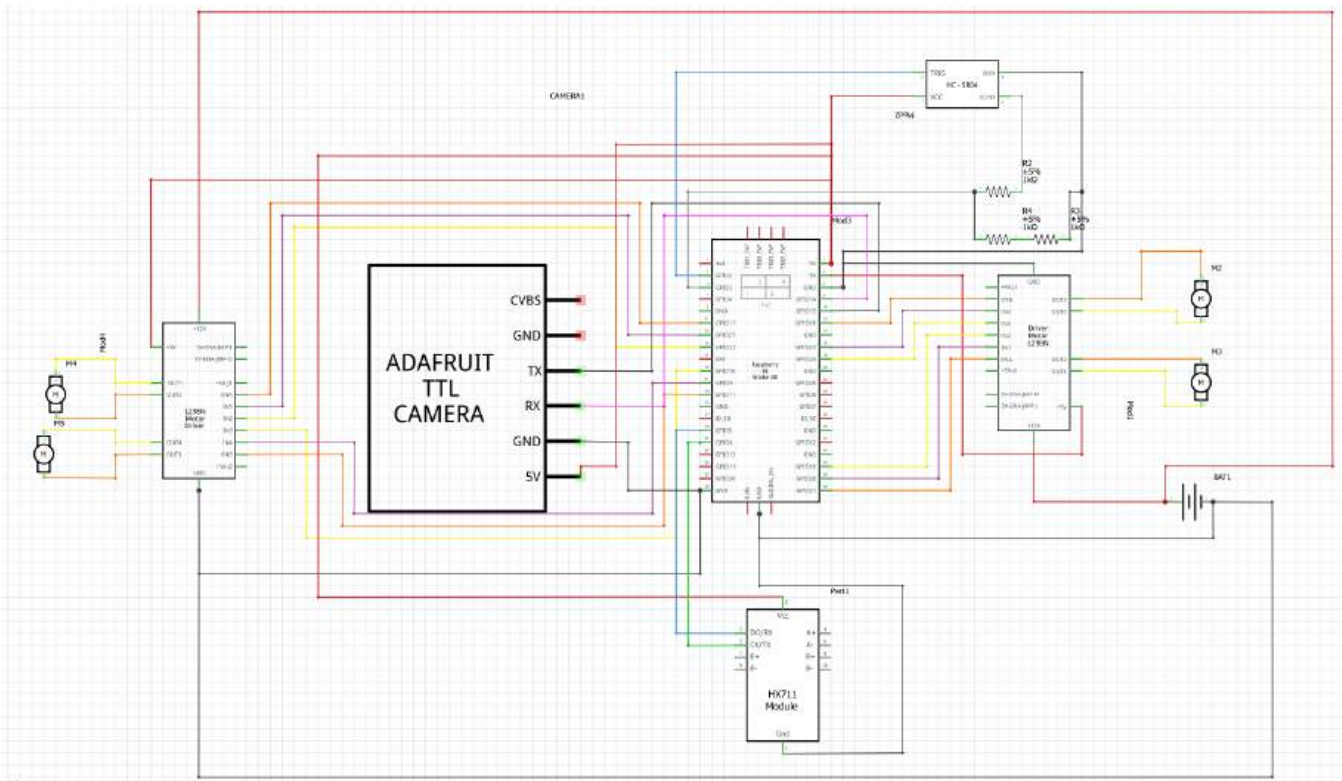
except KeyboardInterrupt:
    self.motors.release_gpios()
    print("Keyboard Interrupt!!")

def main():
    peterporker = Activate("https://peterporker.pythonanywhere.com",
True)
    peterporker.main_loop()

if __name__ == "__main__":
    main()

```

שרטוט חשמלי



רפלקציה

- לסיום הפרויקט, המערכת עונה על הדרישות והמטרות שהצבנו לעצמנו בתחילת התהליך.
- לדעתנו הצלחנו לבנות פרויקט העונה על משבר האשפה העולמי, בזכות זה שעבדנו בצורה מסודרת לפי תהליך מאורגן מראש וידענו מה נעשה בכל שלב ושלב בתהליך הפרויקט.
- העלנו רעיונות חלופיים לכל תת מערכת בפרויקט וערכנו איסוף מידע המתאים לכל רעיון. המידע שנאסף עזר לנו לבחור את התת מערכות המתאימות ביותר לצורך הרובוט שלנו. בין אם זה החלפת רצועת הטיימניג בשרשת במערכת ההנעה או שני הזווית של מערכת האסיפה.
- השינויים הוסברו באופן ברור ומדויק, והתייחסנו לכל שינוי באופן מפורט ומעמיק, והסברנו מדוע עלינו לשנות פרט זה במערכת באופן ברור ככל הניתן.
- בתחילת תהליך התכן של המערכות סירטנו סרטוט דו מיימדי אשר עזר לנו להבין את גודלי המערכת וכיצד היא תיראה ביחס למנועים, גלגלים וכו.. בנוסף, ערכנו מספר חישובים הנדסיים רלוונטיים, כמו חישוב fourqen (מומנט סיבוב) אשר המנועיים יוציאו וחישוב אורך הרצוע אשר נצטרך למערכת האיסוף.
- כל החישובים מנומקים ומוסברים כולל דרך הפתרון בפרק "תוכן הנדסי".
- החלק הטכנולוגי והטכני של הפרויקט עובד כהלכה. החלק האלקטרוני והחלק של בניית התוכנה מוכנים, ומתפקדים כשורה.
- בנינו תיאורים מלבניים של אלגוריתם התוכנה. תרשימים אלו מפשטים את אלגוריתם התוכנה ומדגימים את הדרכים בהם בחרנו לעמוד בדרישות התוכנה. האלגוריתם מתוכנן, לדעתנו, בצורה היעילה ביותר תוך התחשבות בתנאים הרצויים ובדרישות שקבענו מראש.
- משתנה המערכת מבוקר בחוג סגור כאשר הזבל נאגר בכופסא האשפה. אנו משתמשים בחיישן אולטרה סוני אשר ישב על סף התיבה וימדוד את המרחק בין האשפה שנאגרה בקופסא אל קצה הקופסא ובכך לא יאפשר להשתמש יותר האיסוף עד לריקון הקופסא.
- המערכת כוללת ממשק משתמש ידידותי, אשר יאפשר לכל אדם בכל גיל לשלוט ברובוט ולבצע איתו משימות בכל שטח.
- הדגם הסופי עונה על הגדרת הבעיה בצורה הטובה ביותר שיכולנו לבצע ואנו גאים מאוד בתוצר הסופי שלנו ובכל מערך העבודה שהייתה.
- בזכות הפרויקט שני חברי הצוות התקרבו ויצרו חוויות לכל החיים אשר נזכור ונצחק עליהם גם בעוד הרבה שנים.

הערכה מסכמת

דרישה מילולית	סוג	דרישה מנוסחת הנדסית	הערות	בוצע/לא בוצע
יכולת לנוע ולעבור מכשולים במספר רב של שטחים	מכנית	על המערכת ההנעה להיות מסוגלת להתמודד עם הכוח שפועל עליה ועל המנועים להיות בעלי מומנט סיבוב גדול על מנת התמודדות של מכשולים.	המערכת לא מסוגלת לעבור בכל השטחים אשר הצבנו לה כמטרה.	לא בוצע
זיהוי אובייקטים	תכנות	על המערכת לזהות אובייקטים ולדעת למיין אותם לפי קטגוריות.	המערכת יכולה לזהות בצורה טובה בקבוקים ופחיות, ולפעמים מצליחה לזהות זכוכית ו	בוצע
יכולת אסיפה של סוגים רבים של אפשה	מכנית	על המערכת לאסוף את סוגי האשפה בעזרת המערכת אסוף אשר תגרוף את האשפה ותעביר את הזבל לפח.	המערכת תצטרך לאסוף את כל סוגי האשפה אשר נמצאים במרחבים ציבוריים.	בוצע
תקשורת עם המערכת בכל מקום	תכנות	המערכת תוכל להשלת מכל מקום בעולם בעזרת האתר שיהיה מחובר לראזברי פאי וישלח לו פקודות מן האתר.	(נועם)	בוצע
פעילות אוטונומית	תכנות	על המערכת לנסוע לאובייקטים ולאסוף אותם. כאשר החיישן מזהה שהפח מלא ללכת לנקודת ההתחלה שממנו יצא ויחכה שיורקנו את הפח.	לצערנו לא הצלחנו לאפשר לרובוט לעשות את זה.	לא בוצע

מילון המונחים

לפניכם רשימה של מונחים חשובים מאד לתיאור המערכת שבניתם. ליד כמו מונח, ציינו האם הוא מופיע במערכת שבניתם ואם כן כיצד. (רצוי בגופן שונה) אם לא, רשמו את הגדרת המונח בלבד.

1. קלט
2. פלט
3. עיבוד
4. בקר
5. חוג סגור
6. חוג פתוח
7. ממשק
8. אלגוריתם
9. פקודות קלט
10. פקודות פלט
11. wait
12. loop
13. switch
14. מתח
15. זרם
16. התנגדות
17. הספק
18. עבודה
19. מנוע
20. מקודד
21. חיישן מגע
22. חיישן מרחק
23. חיישן אור
24. חיישן קול
25. אות אנלוגי
26. אות דיגיטלי
27. כח
28. מרחק
29. מהירות
30. עבודה

- “3D Printed Gearbox: How to Design Your Own Box.” *All3DP*, 29 Sept. 2019,
all3dp.com/2/3d-printed-gearbox-how-to-design-your-own-box/.
- “4ocean X Poralu BeBot.” *4ocean*, www.4ocean.com/pages/4ocean-x-poralu-bebot.
- “6 IoT Communication Protocols for Web Connected Devices.” *Getkisi.com*, 2018,
www.getkisi.com/blog/internet-of-things-communication-protocols.
- “A Simple Guide to Conveyor Drive Types | Mk North America.” *Www.mknorthamerica.com*,
www.mknorthamerica.com/Blog/conveyor-drive-types/.
- “A Step-By-Step Introduction to the Basic Object Detection Algorithms (Part 1).” *Analytics Vidhya*, 11 Oct. 2018,
www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/.
- “An Intuitive Guide to Convolutional Neural Networks.” *FreeCodeCamp.org*, 24 Apr. 2018,
www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/#:~:text=Convolutional%20Neural%20Networks%20have%20a.
- Arunava. “Convolutional Neural Network.” *Medium*, Towards Data Science, 25 Dec. 2018,
towardsdatascience.com/convolutional-neural-network-17fb77e76c05.
- Bai, Jinqiang, et al. “Deep Learning Based Robot for Automatically Picking up Garbage on the Grass.” *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, 1 Aug. 2018, pp. 382–389, ieeexplore.ieee.org/abstract/document/8419288, 10.1109/TCE.2018.2859629.
- Balasuthagar, C, et al. “Design and Fabrication of Beach Cleaning Machine.” *IOP Conference Series: Materials Science and Engineering*, vol. 912, 12 Sept. 2020, p. 022048, 10.1088/1757-899x/912/2/022048.

“Best Pre-Trained Models for Object Detection in Machine Learning.” *IT4nextgen*, 20 Aug. 2019,

www.it4nextgen.com/best-pre-trained-models-for-object-detection-in-machine-learning/.

“Conveyor Definitions, Terminology & Glossary | Cisco-Eagle.” *Www.cisco-Eagle.com*,
www.cisco-eagle.com/category/3385/conveyor-terminology-glossary.

Davies, Alan. “Using Python Flask and Ajax to Pass Information between the Client and Server.” *Medium*, 18 June 2021,
towardsdatascience.com/using-python-flask-and-ajax-to-pass-information-between-the-client-and-server-90670c64d688.

elbruno. “#RaspberryPi – Running a #Python Script in a Python Virtual Environment on Reboot / Startup.” *El Bruno*, 30 July 2019,
elbruno.com/2019/07/30/raspberrypi-running-a-python-script-in-a-python-virtual-environment-on-reboot-startup/.

FIRSTInternFollow. “Understanding Motor and Gearbox Design.” *Instructables*,
www.instructables.com/Understanding-Motor-and-Gearbox-Design/.

“Google Colaboratory.” *Colab.research.google.com*,
colab.research.google.com/github/dctian/DeepPiCar/blob/master/models/object_detection/code/tensorflow_traffic_sign_detection.ipynb#scrollTo=gnNXNQCjdniL.

“Guide to Different Padding Methods for CNN Models.” *Analytics India Magazine*, 4 Sept. 2021, analyticsindiamag.com/guide-to-different-padding-methods-for-cnn-models/.

Haiside Heavy Industry. *How Does the Vibrating Screen Work?* | *Haiside*. 7 June 2021,
www.hsd-industry.com/news/how-does-the-vibrating-screen-work/#:~:text=The%20vibrating%20screen%20works%20by.

- “Image Classifier Using CNN.” *GeeksforGeeks*, 23 Dec. 2017,
www.geeksforgeeks.org/image-classifier-using-cnn/?ref=lbp.
- Jason Brownlee. “A Gentle Introduction to Object Recognition with Deep Learning.” *Machine Learning Mastery*, 5 July 2019,
machinelearningmastery.com/object-recognition-with-deep-learning/.
- Majchrowska, Sylwia, et al. “Deep Learning-Based Waste Detection in Natural and Urban Environments.” *Waste Management*, vol. 138, Feb. 2022, pp. 274–284,
10.1016/j.wasman.2021.12.001.
- Mandal, Manav. “CNN for Deep Learning | Convolutional Neural Networks (CNN).” *Analytics Vidhya*, 1 May 2021,
www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.
- Mepani Vrutant N, Patel Himani N, Vataliya Mohil B, Prof. Sahu RohitKumar. 2 Feb. 2020,
www.irjet.net/archives/V7/i2/IRJET-V7I2301.pdf.
- Michaud, David. “Vibrating Screen Working Principle.” *Mineral Processing & Metallurgy*, 26 July 2015,
www.911metallurgist.com/blog/vibrating-screen-working-principle#vibrating-screen-design.
- Mikołajczyk, Agnieszka. “Waste Datasets Review.” *GitHub*, 8 May 2022,
github.com/AgaMiko/waste-datasets-review.
- Mishra, Mayank. “Convolutional Neural Networks, Explained.” *Medium*, 2 Sept. 2020,
towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.
- Mitra, Arghadeep, et al. *Detection of Waste Materials Using Deep Learning and Image Processing*. 2020.

“MobileNetV2 SSD FPN - Edge Impulse Documentation.” *Edgeimpulse.com*, 2017,
docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/mobile
netv2-ssd-fpn.

“Motor Power Calculator | Magtrol.” *Wwww.magtrol.com*,
www.magtrol.com/motor-power-calculator/.

“Neural Network | Introduction to Neural Network | Neural Network for DL.” *Analytics Vidhya*,
1 Mar. 2021, www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/.

“Object Recognition.” *Wwww.mathworks.com*,
www.mathworks.com/solutions/image-video-processing/object-recognition.html#:~:text=
Object%20recognition%20is%20a%20computer. Methods for object detection other than
CNN.

“Overfitting in a Neural Network Explained.” *Deeplizard.com*,
deeplizard.com/learn/video/DEMmkFC6IGM.

“Padding (Machine Learning).” *DeepAI*, 17 May 2019,
deepai.org/machine-learning-glossary-and-terms/padding#:~:text=Padding%20is%20a%2
0term%20relevant.

Plastic Waste Detection with Deep Learning | LearnOpenCV. 25 Jan. 2022,
learnopencv.com/plastic-waste-detection-with-deep-learning/.

“Pulley Calculator. RPM, Belt Length, Speed, Animated Diagrams.” *Wwww.blocklayer.com*,
www.blocklayer.com/pulley-belteng.aspx.

Python, Sorting Contours using, and OpenCV-PyImageSearch says. “Find Distance from Camera
to Object Using Python and OpenCV.” *PyImageSearch*, 19 Jan. 2015,
pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv

/.

Rahman, Md. Wahidur, et al. “Intelligent Waste Management System Using Deep Learning with IoT.” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, 5 Sept. 2020, www.sciencedirect.com/science/article/pii/S1319157820304547, 10.1016/j.jksuci.2020.08.016.

“Rectified Linear Units (ReLU) in Deep Learning.” *Kaggle.com*, www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook.

“RF Controlled Beach Cleaner Robotic Vehicle.” *Nevon Projects*, 7 July 2018, nevonprojects.com/rf-controlled-beach-cleaner-robotic-vehicle/.

Saha, Sumit. “A Comprehensive Guide to Convolutional Neural Networks—the ELI5 Way.” *Towards Data Science*, 15 Dec. 2018, towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

“Scorpion – Sand Cleaning Solutions.” *Sand Cleaning Solutions*, sandcleaningsolutions.com.au/scorpion/.

Seb. “Understanding Padding and Stride in Convolutional Neural Networks - Programmatically.” *Programmatically.com*, 3 Dec. 2021, programmatically.com/understanding-padding-and-stride-in-convolutional-neural-networks/.

Sloactive. “Plastic Pollution Guide - Ocean Pollution Facts & Figures.” *SLO Active*, 2016, sloactive.com/plastic-pollution/.

Smith, Lauren, and William Richard Turrell. “Monitoring Plastic Beach Litter by Number or by Weight: The Implications of Fragmentation.” *Frontiers in Marine Science*, vol. 8, 17

Sept. 2021, 10.3389/fmars.2021.702570.

Srihari, Sargur. *Deep Learning Srihari Convolutional Networks: Overview*.

“SURF RAKE by Barber.” *H Barber & Sons*,

www.hbarber.com/beach-cleaning-machines/surf-rake/.

Team, Great Learning. “Artificial Intelligence versus Machine Learning versus Deep Learning.”

GreatLearning Blog: Free Resources What Matters to Shape Your Career!, 17 Oct. 2019,

www.mygreatlearning.com/blog/artificial-intelligence-machine-learning-deep-learning/.

“Technology.” *BeachTech Beach Cleaners | Beach Cleaning Equipment*,

www.beach-tech.com/en/home/technology.html.

The World Counts. “The World Counts.” *The World Counts*, 2019, www.theworldcounts.com/.

Timing Belts and Pulleys Sizing & Measurement | MISUMI Blog, 21 Aug. 2014,

us.misumi-ec.com/blog/timing-belts-pulleys-sizing-measurement/.

“Torque and Angular Acceleration.” *Coursehero.com*, 2022,

www.coursehero.com/study-guides/boundless-physics/torque-and-angular-acceleration/.

Train, Test, & Validation Sets Explained. deeplizard.com/learn/video/Zi-0rIM4RDs.

“Udacity.” *Auth.udacity.com*,

auth.udacity.com/sign-up?next=https%3A%2F%2Flearn.udacity.com%2Fcourses%2Fud1

87.

Wate, Yash. “3 Ways to Run a Raspberry Pi Program or Script at Startup.” *MUO*, 16 Apr. 2021,

www.makeuseof.com/how-to-run-a-raspberry-pi-program-script-at-startup/.

WE CLEAN ON.

“What Is a ReLU Layer?” *Educative: Interactive Courses for Software Developers*,

www.educative.io/edpresso/what-is-a-relu-layer.

“What Is the Difference between a Convolutional Neural Network and a Regular Neural Network?” *Artificial Intelligence Stack Exchange*,

ai.stackexchange.com/questions/5546/what-is-the-difference-between-a-convolutional-neural-network-and-a-regular-neur.

“What Wireless Communication Standard to Use for Your next Project.” *Hackster.io*,

www.hackster.io/news/what-wireless-communication-standard-to-use-for-your-next-project-2db9570eabaf.