

**תוכן עניינים**

שער

תוכן עניינים

תקציר הפרויקט ................................................................................................................................. 3

הגדרת עולם הבעיה ............................................................................................................................. 4

מה נעשה בתחום ................................................................................................................................ 5

הפתרון .............................................................................................................................................. 5

ארכיטקטורה.......................................................................................................................................6

הצדקה אקדמאית .............................................................................................................................. 7

כלי הפיתוח, החומרה והתוכנה..............................................................................................................7

תיאור המערכת – כולל תמונות מסך................................................................................................... 8-9

מחקר ופיתוח............................................................................................................................... 10-23

קשיים.............................................................................................................................................. 24

שיפורים............................................................................................................................................24

תוצרי הפרויקט ................................................................................................................................ 25

חלוקת עבודה.....................................................................................................................................25

ביבליוגרפיה ..................................................................................................................................... 26

תקציר פרויקט באנגלית .................................................................................................................... 27

**תקציר הפרויקט**

יותר מ-250 מיליון בני אדם הם עיוורים או בעלי לקות ראייה ברמה מסוימת בעולם. בשל הלקות, במהלך היומיום שלהם הם נתקלים בקשיים רבים בתחומים שונים. אחד התחומים הבולטים הוא התחום הקמעונאי.

כיום, אדם בעל לקות ראייה משתמש באמצעים שונים בעת קנייה בסופר, להלן –

1. שימוש בחוש המישוש ע"מ לזהות את המוצר שבחר.

2. ליווי ע"י אדם נוסף לסופר.

3. בקשת עזרה מאנשים מהסביבה.

אמצעים אלה, מלבד סעיף 1, מפחיתים את תחושת העצמאות של אותו אדם. כמו כן, ישנם פריטים שמודפסים על גבי המוצר כך שחוש המישוש לא יעזור במקרה זה, כמו תוויות הסימון של משרד הבריאות (סוכר, נתרן, שומן רווי והמדבקה הירוקה) ותאריכי תפוגה.

VisiFood מעוניינת לשפר את חוויית הקנייה בסופר עבור אנשים בעלי לקות ראייה.

ע"י אפליקציה, בעלת עיצוב מינימלי שהותאם עבור אותם אנשים, VisiFood מעניקה להם את האפשרות להיות עצמאיים יותר בכך שהיא "רואה" במקומם את הפרטים על המוצרים.

האפליקציה מאפשרת למשתמש לבחור באחת משתי האופציות בזמן נתון –

1. זיהוי תווית סימון.
2. זיהוי תאריך.

תהליך הזיהוי של כל אחת מהאופציות מתבצע ע"י שני מודלים שנבנו עבור האפליקציה.

כמו כן, השימוש באפליקציה עבור המשתמש הינה פשוטה: על המשתמש לבחור באחת מהאופציות, לכוון את המצלמה למוצר ומשם האפליקציה – בעזרת המודלים שהוזכרו לעיל – שולחת למשתמש חיווי קולי שמפרט את התוצאות הרלוונטיות שהתקבלו.

VisiFood פותחת דלתות לאנשים בעלי לקות ראייה, יוצרת להם חוויית קנייה ידידותית ומפחיתה את התלות שלהם בסביבה.

**הגדרת עולם הבעיה**

*מתוך ויקיפדיה:* "... לאנשים עם לקות ראייה יש מגבלות בשילוב כל ארבעת תחומי החיים הבאים: למידה (כולל חינוך פורמלי), עבודה, **טיפול עצמי** ופעילויות פנאי."[[1]](#footnote-1)

כיום, ברשתות הקמעונאות קיים מידע רב על הפריטים שלרוב כתוב בצורה מוקטנת אשר עלולה להקשות על לקוחות בכלל ועל אנשים בעלי לקות ראייה בפרט. טיפול עצמי כולל בין היתר קנייה בסופר.

חוויית הקנייה בסופר עבור כבדי ראייה עלולה להיות חוויה מסובכת יותר מזו של חסרי הלקות. כבדי הראייה נדרשים להיעזר באדם שמגיע עמם מראש לסופר או בכזה שנמצא בסביבת הקניות שלהם ברגע נתון. דבר זה מונע מבעלי הלקות את עצמאותם ודורש מהם להיות תלויים בסביבה.

כמו כן, כבדי הראייה נעזרים בחוש המישוש שלהם על מנת לזהות את המוצר שבחרו. אך דבר זה אינו נחשב לפתרון מלא לבעיה של בעלי הלקות מכיוון שקיימים פרטים רבים על גבי המוצרים אשר מודפסים על המוצר ולא ניתן לזהות אותם ע"י מישוש.  
לדוגמא, משרד הבריאות הוציא בשנתיים האחרונות תו תקן חדש אשר הגדיר שיש להדפיס על אריזות המוצרים את אחת מהמדבקות הבאות –

1. שומן רווי ברמה גבוהה.
2. סוכר ברמה גבוהה.
3. נתרן ברמה גבוהה.
4. המדבקה הירוקה – מסמנת מוצר בריא.

מדבקות אלו מודפסות על מוצרים שעומדים בקריטריונים של אותו תו תקן.  
אדם בעל לקות ראייה אינו יכול לזהות את המדבקה אלא אם יש לידו אדם שמציין בפניו את המידע הזה.

דוגמא נוספת היא תאריכים על גבי מוצרים. כמובן שתאריך ייצור פחות רלוונטי לכלל האנשים והדבר החשוב יותר הוא תאריך התפוגה של המוצר. גם את תאריך התפוגה, בעל לקות הראייה לא יוכל לזהות ע"י מימוש המוצר אלא רק ע"י אדם שמתלווה אליו או מסייע לו ברגע נתון.

כידוע, העולם מתפתח ואיתו מתפתח גם תחום הקמעונאות, הן מבחינת כמות המוצרים המוצעת ללקוחת והן מבחינת גודל הסופרים. דבר זה מקשה יותר על כבדי הראייה כיוון שעליהם לנווט במקום גדול בין מבחר ענק של מוצרים.

הבעיות איתן VisiFood רוצה להתמודד הן:

1. חוויית הקנייה המסובכת עבור כבדי הראייה.

מתן שירות פשוט וחינמי בעבור מטרה זהה לזו של Orcam ([מפורט בהמשך](#נעשהבתחום)) והוספת נדבך לפתרון הקיים.

**מה נעשה בתחום**

**חברת Orcam**[[2]](#footnote-2)

אורקם טכנולוגיות בע"מ היא חברה ישראלית המפתחת ומייצרת התקן מבוסס ראייה ממוחשבת, המסייע לכבדי ראייה, עיוורים ובעלי מוגבלויות נוספות.

**המייסדים** – פרופסור אמנון שעשוע, זיו אבירם.

**תאריך הקמה** – 2010.

החברה מציע מכשיר לביש לכל סוגי המשקפיים אשר מאפשר למשתמשים לקרוא טקסטים, לזהות מוצרים (שטרות כסף, כרטיסי אשראי) ולזהות פנים. כמו כן, החברה מזהה מוצרים בסופר ע"י הברקודים שלהם.

לעומת זאת, VisiFood מזהה מוצרים שלא באמצעות ברקוד. יתרה מכך, VisiFood שמה דגש על תוקף המוצר ולתוויות הסימון על גבי המוצרים (סוכר ברמה גבוהה, נתרן ברמה גבוהה, שומן רווי ברמה גבוהה והסימון הירוק) – דבר שהחברה לא התייחסה אליו, נכון להיום.

**הפתרון**

VisiFood נוצרה כדי לתת מענה ללקויי ראיה המעוניינים לשפר את חווית הקניה בסופר על ידי הנגשת פרטי המוצר בצורה ברורה יותר.

VisiFood היא אפליקציה שתסייע ללקויי ראייה בקלות מרבית. תוך שימוש בעקרונות של ראיה ממוחשבת, הלקוחות יוכלו לפתוח את האפליקציה בלחיצת כפתור ובאמצעות זיהוי קולי לנווט ולבצע את בדיקתם.

לאפליקציה יהיו 2 מצבים עיקריים –

1. בדיקת תווית סימון על המוצר.
2. בדיקת תוקף מוצר.

בעת לחיצת כפתור, מצלמת הפלאפון תיפתח ותתחיל לבצע סריקה. בסריקת תוויות כל frame יכנס לרשת נוירונים שמבצעת object detection והמכונה תבדיל בין התוויות השונות. לבסוף יינתן חיווי קולי לגבי כל התוויות שהמכונה זיהתה על גבי המוצר.

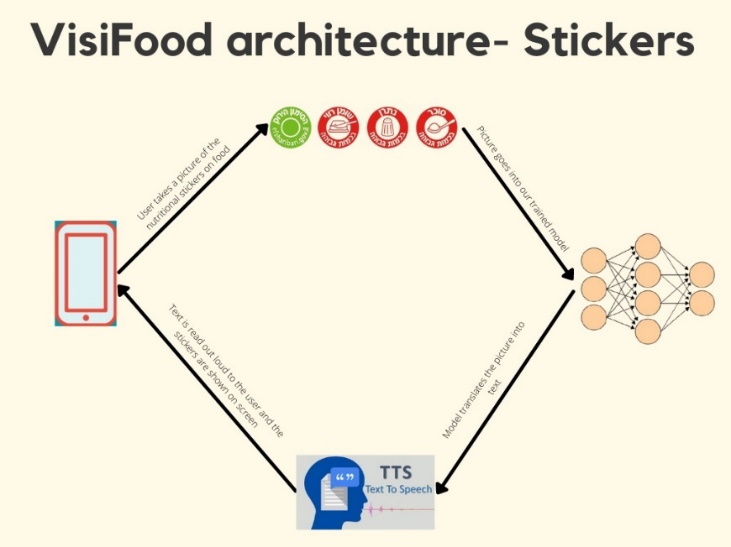
בסריקת תאריכים, מצלמת הפלאפון תיפתח ותתחיל לזהות "תבניות" של תאריכים באמצעות object detection. ברגע הזיהוי, התבנית תיחתך ותישלח למנוע OCR שיבצע "קריאה" של התאריך. לאחר מכן האפליקציה תיתן חיווי קולי למשתמש.

**ארכיטקטורה**

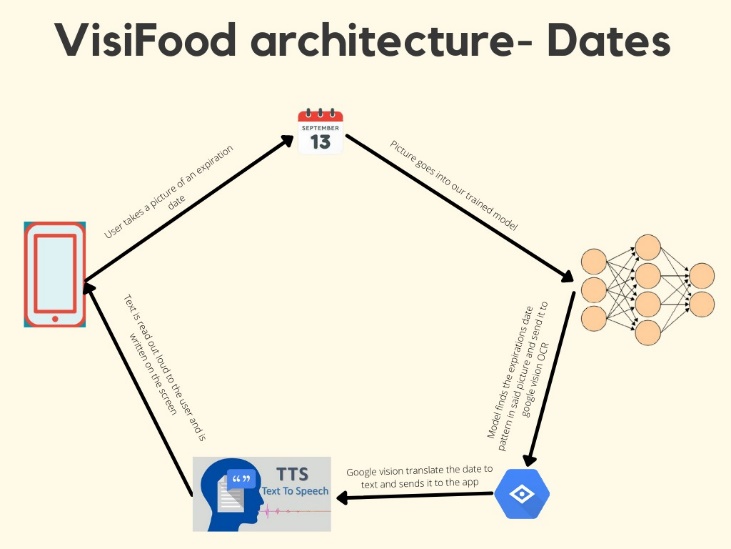
ארכיטקטורת המערכת בנויה משלוש יחידות מרכזיות –

1. **רשתות נוירונים** – רשתות אלו מבצעות object detection לצורך זיהוי תוויות ותאריכים המבוססות על data שאספנו מראש. הרשתות מוטמעות בתוך האפליקציה כך שהעיבוד מבוצע באופן לוקאלי על גבי המכשיר. הרשתות מבוססות על ארכיטקטורת YoloV4Tiny.

2. **מנוע OCR** – שימוש במנוע באמצעות ה-API של גוגל (Google Vision). המנוע מוטמע בתוך האפליקציה לצורך זיהוי וקריאה של תאריך מתוך תווית שנשלחה מרשת הנוירונים בשלב הראשון.

3. **אפליקציית המשתמש** – פיתוח על גבי מכשירי Android באמצעות Android Studio בשפה Javaתוך שימוש ב-text to speech.

**ארכיטקטורה א': תוויות סימון תזונתי**



**ארכיטקטורה ב': תאריכים**

**הצדקה אקדמית**

**החלק הראשון** הוא איסוף data באופן עצמאי (תמונות של תוויות ותאריכים – מבוצע ממצלמות של טלפונים חכמים). לאחר האיסוף, בוצע תהליך הכנה של ה Data-על ידי תוכנת LabelImg. הכנה זו כללה סימון של ה-labels המתאימים. מכיוון שה-data מחולק לשני תחומים (תוויות ותאריכים), הגדרנו את ה-labels הבאים –

**עבור תוויות סימון תזונתי**

1. סוכר ברמה גבוהה ((sugar.
2. שומן רווי ברמה גבוהה (fat).
3. נתרן ברמה גבוהה (salt).
4. המדבקה הירוקה (green).

**עבור תאריכים**

1. Date.

**החלק השני** הוא ביצוע מחקר שבו נבדוק איך אפשר לבצע עיבוד על מכשיר סלולארי ונבין אלו מודלים הם ה- state of the art של בעיית ה-object detection שיהיו הכי רלוונטיים לאפליקציה שלנו. השלב הבא הוא ביצוע אימון ומבחן על בסיס ה-data שאספנו ויצירת שני מודלים –

1. מודל שיזהה תוויות סימון תזונתי.
2. מודל שיבצע זיהוי של תבנית של תאריך.

סוף החלק השני הוא ביצוע דחיסה של המודלים בכדי שיוכלו לעבוד באופן לוקאלי על גבי טלפון חכם.

**החלק השלישי** הוא פיתוח אפליקציית Android בסביבת Android Studio הכתובה בשפה Java. תוך כדי הפיתוח, ביצוע מחקר שבו נבדוק באיזה מנוע OCR נשתמש בכדי "לקרוא" את התאריך מתוך התבנית שתישלח מהמודל השני שציינו לעיל. לבסוף שילוב של text to speech בתוך האפליקציה.

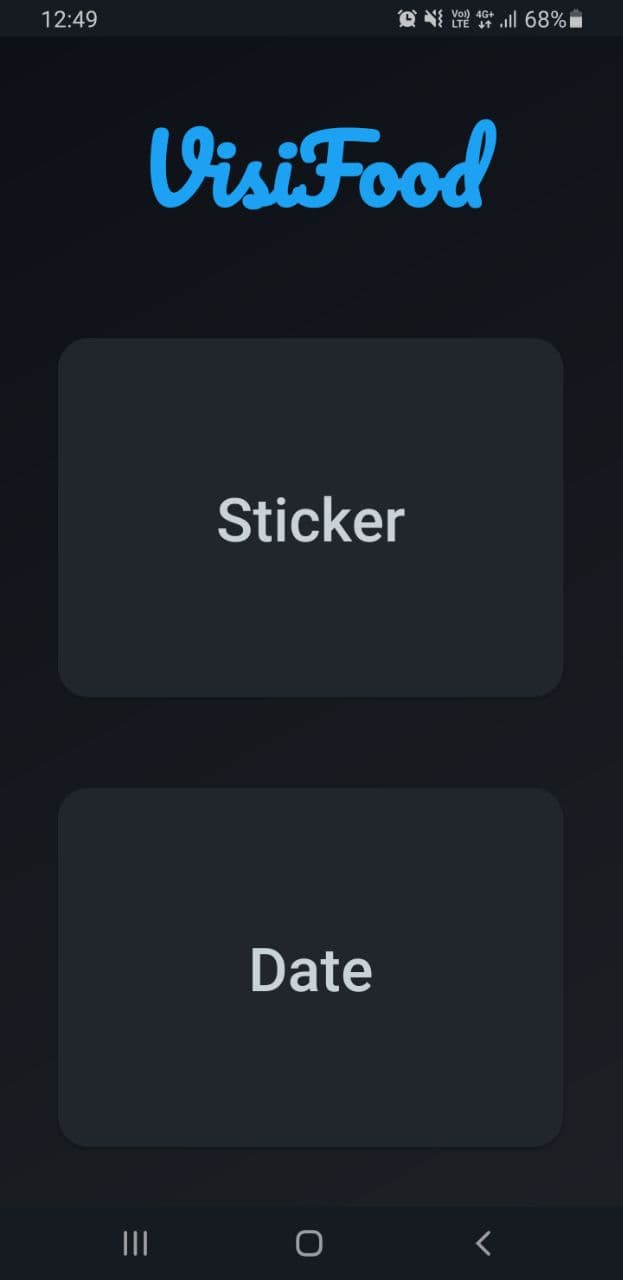
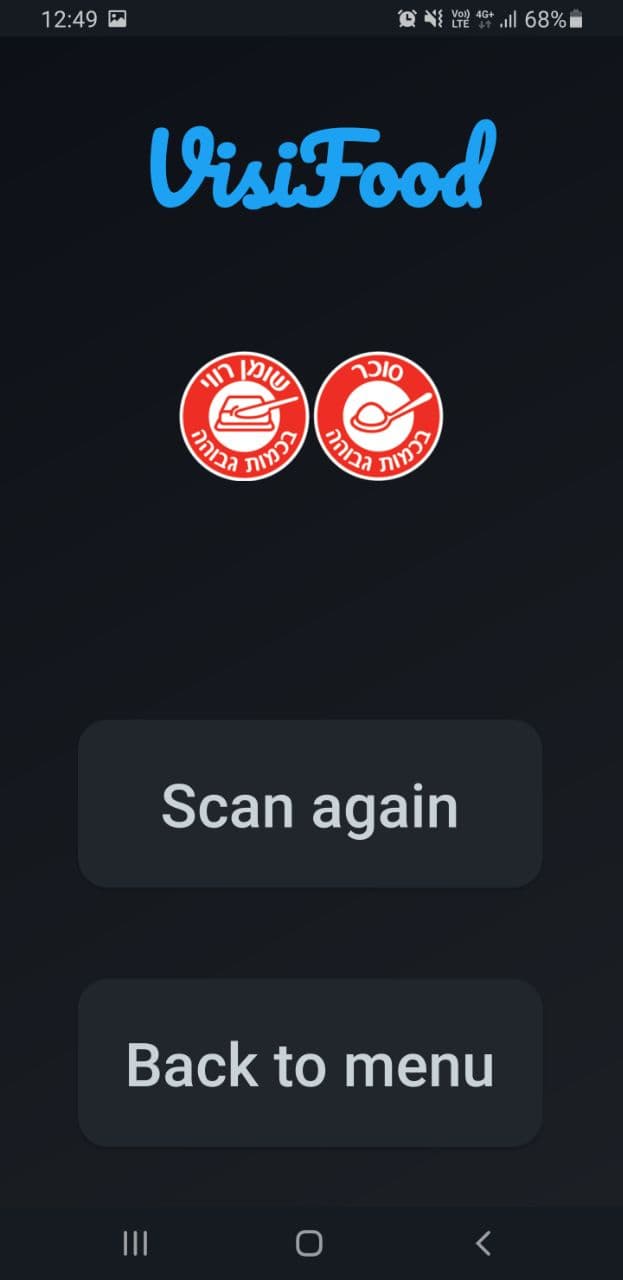
**כלי הפיתוח, החומרה והתוכנה**

כלי הפיתוח בהם עשינו שימוש הם –

1. Android Studio.
2. Jupyter notebook.
3. Pycharm.
4. Google Colab.

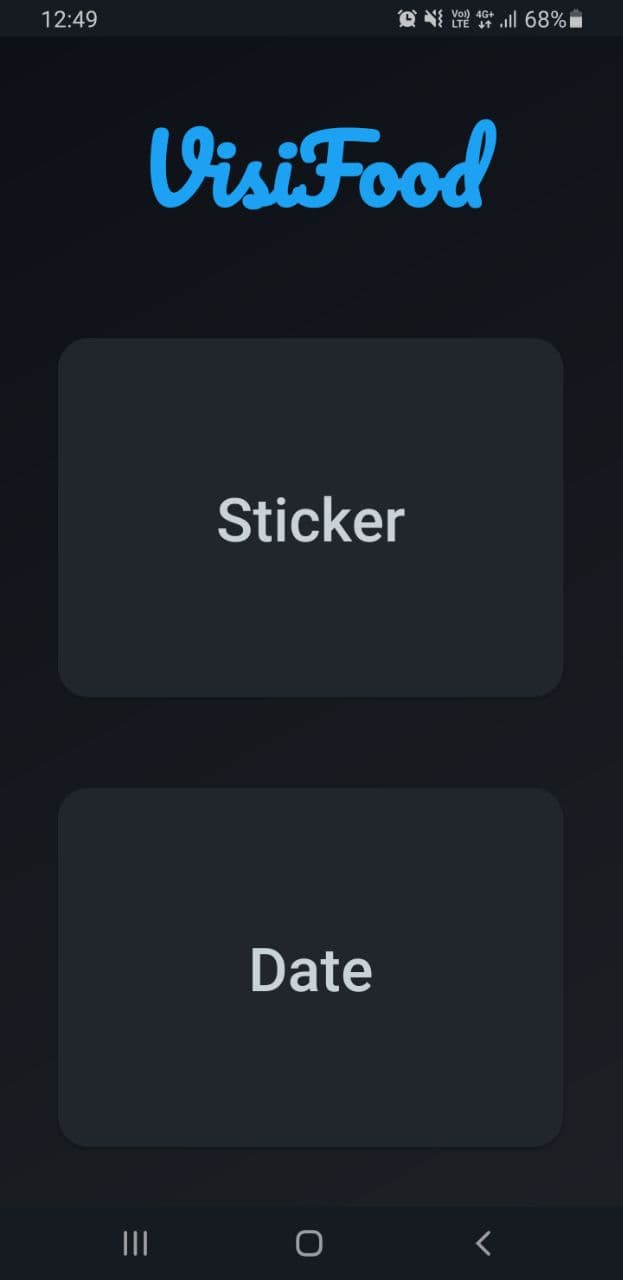
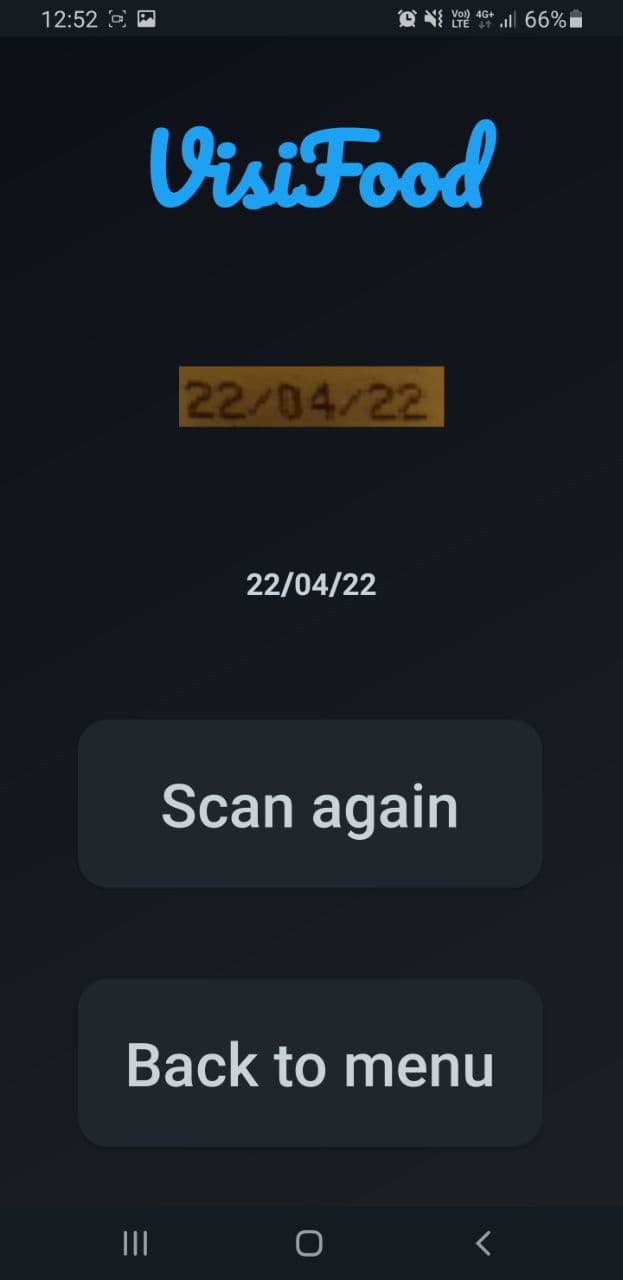
[**תיאור המערכת**](#אפליקציה) **– כולל תמונות מסך**

**אופציה א' – זיהוי סימון תזונתי**



**אופציה ב' – זיהוי תאריכים**

**אופציה ב' – זיהוי תאריכים**



**מחקר ופיתוח**

חלק זה כלל מספר שלבים –

1. [גיבוש רעיון ראשוני](#רעיוןראשוני).
2. [Data](#דאטה).
3. [אימון המודלים](#אימוןהמודלים).
4. [פיתוח אפליקציה](#פיתוחאפלי).

**גיבוש רעיון ראשוני** (חזרה ל[מחקר ופיתוח](#מופ))

תחילה, רעיון הפרויקט היה לסייע ללקויי ראיה ועיוורים לזהות באמצעות חומרה דלת הספק או אפליקציה את הפריטים הבאים –

1. פירות וירקות, הצגת צבעם ורמת בשלותם.
2. תוקף מוצרים.
3. מדבקות הסימון התזונתי על גבי מוצרים (אדום/ירוק).

לאחר פגישה עם מנחה הפרויקט, הבנו שסעיפים 2 ו-3 הרבה יותר מורכבים ומאתגרים מבחינה רעיונית ויישומית.

בנוסף, לאחר בדיקה, הבנו שקיים כבר שוק העוסק בסעיף 1 לעומת סעיפים 2 ו-3 שהמחקר בהם עדיין "צעיר". לכן, החלטנו להשקיע את מאמצינו בפיתוח זיהוי תוקף של מוצרים ומדבקות סימון תזונתי.

יתרה מזאת, הוחלט לפתח **אפליקציה** עקב הנגישות לקהל היעד והפשטה של תהליך הזיהוי (אין צורך ברכיב נוסף – הכל קורה על גבי האפליקציה). האפליקציה פותחה מעל מערכת ההפעלה Android בגלל הנוחות של סביבת העבודה.

**Data** (חזרה ל[מחקר ופיתוח](#מופ))

את ה-data אספנו באופן עצמאי באמצעות צילום של מוצרים שונים תוך התמקדות בתאריכים או בתוויות סימון תזונתי. למדנו כי ככל שיש יותר data, האימון מתבצע בצורה מדויקת יותר ולכן החלטנו בנוסף, לבצע אוגמנטציות לכל תמונה ובכך להגדיל את כמות ה-data.

בכדי לאמן את המודלים על בעיית ה-object detection, ביצענו תיוגים על ה-data שאספנו ע"י תוכנת LabelImg. כזכור, ה-data כולל שני תחומים – תוויות סימון תזונתי ותאריכים המחולקים ל-labels הבאים –

**תוויות סימון תזונתי**

1. שומן רווי ברמה גבוהה (fat).
2. סוכר ברמה גבוהה (sugar).
3. נתרן ברמה גבוהה (salt).
4. המדבקה הירוקה (green).

כמות data בחלק זה – **1700 תמונות** לאחר אוגמנטציות כך שהיחס בין כמות המדבקות מכל סוג שווה

**תאריכים**

1. date

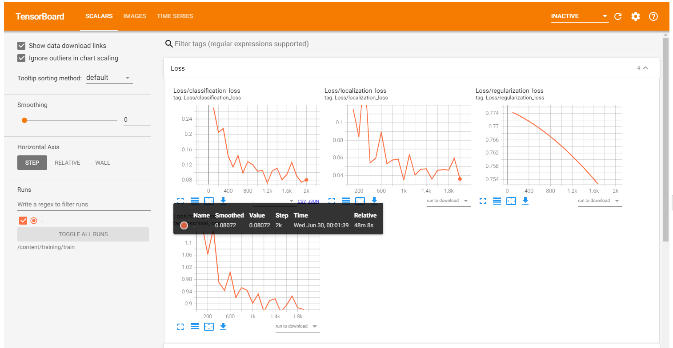
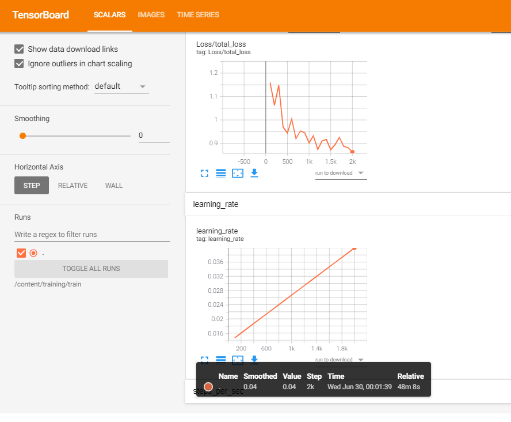
כמות data בחלק זה – **1300 תמונות** לאחר אוגמנטציות.

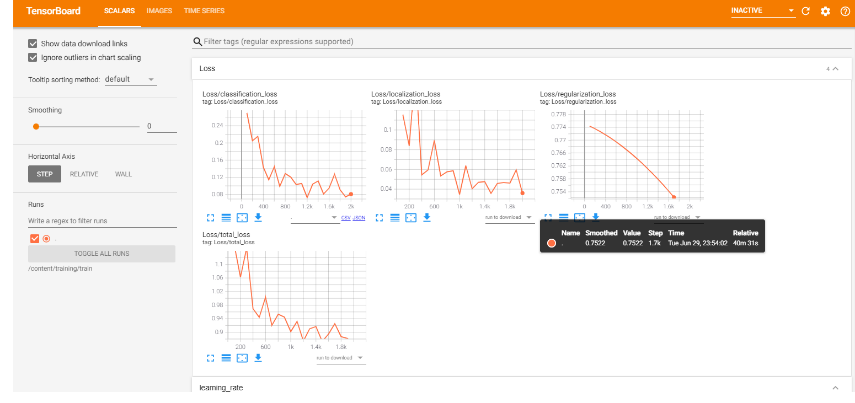
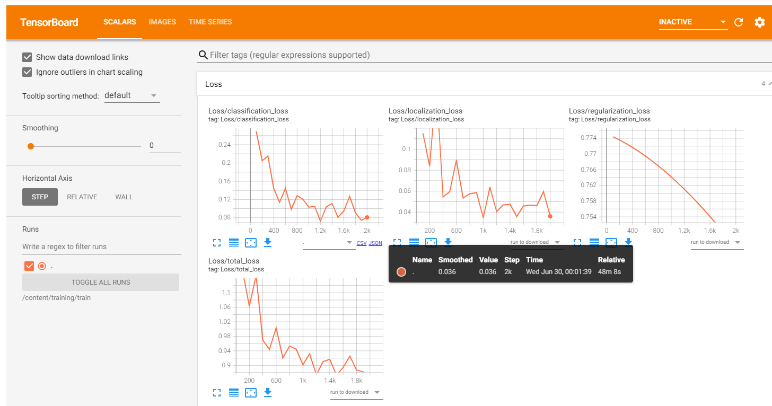
**אימון המודלים** (חזרה ל[מחקר ופיתוח](#מופ))

**מודל זיהוי תוויות סימון תזונתי**

תחילה, השתמשנו בארכיטקטורת **ssd\_mobile\_net\_v1** בכדי לייצר מודל לזיהוי סוגי תוויות. קיבלנו מודל שהניב תוצאות טובות.

מבדיקה על קבוצת האימון, אשר כללה כ-100 תמונות, המודל צדק בכ-**85%** מהפעמים.

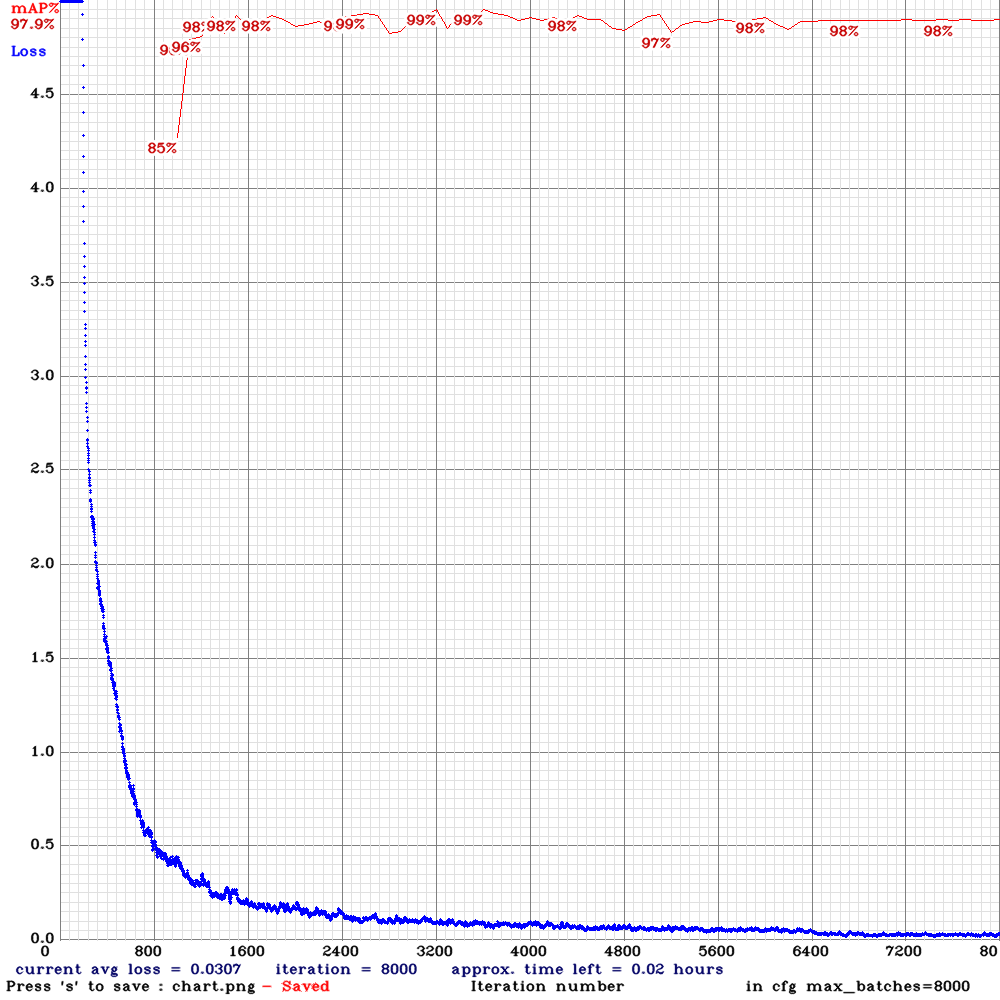
תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 300X300.



**Stickers Model ssd\_mobile\_net\_v1 image size 300x300**

לאחר מחקר נוסף, ניסינו להשתמש בארכיטקטורת **yoloV4Tiny** וראינו שאנחנו מקבלים תוצאות איכותיות יותר מהמודל הקיים (SSD). לכן, החלטנו להמשיך את הפיתוח עם המודל החדש (כלל הניסויים על המודלים הסופיים נעשו בארכיטקטורת yoloV4Tiny).

מבדיקה על קבוצת האימון, אשר כללה כ- 100 תמונות, המודל צדק בכ-**100%** מהפעמים.

תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 416X416.

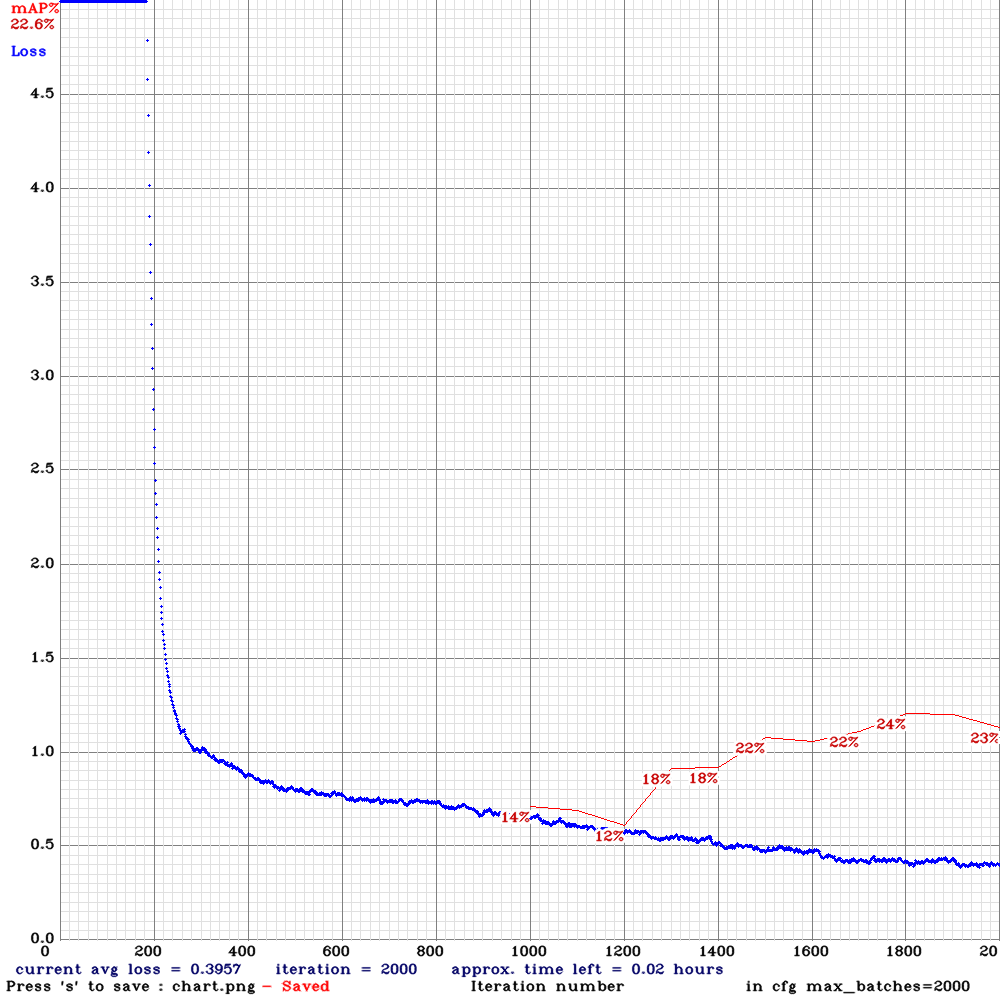
**Stickers Model yoloV4Tiny image size 416x416**

**מודל זיהוי תאריכים**

את אימון מודל התאריכים ביצענו מספר פעמים בגלל מורכבות הבעיה.  
להלן מפורטים מספר ניסויים שבוצעו –

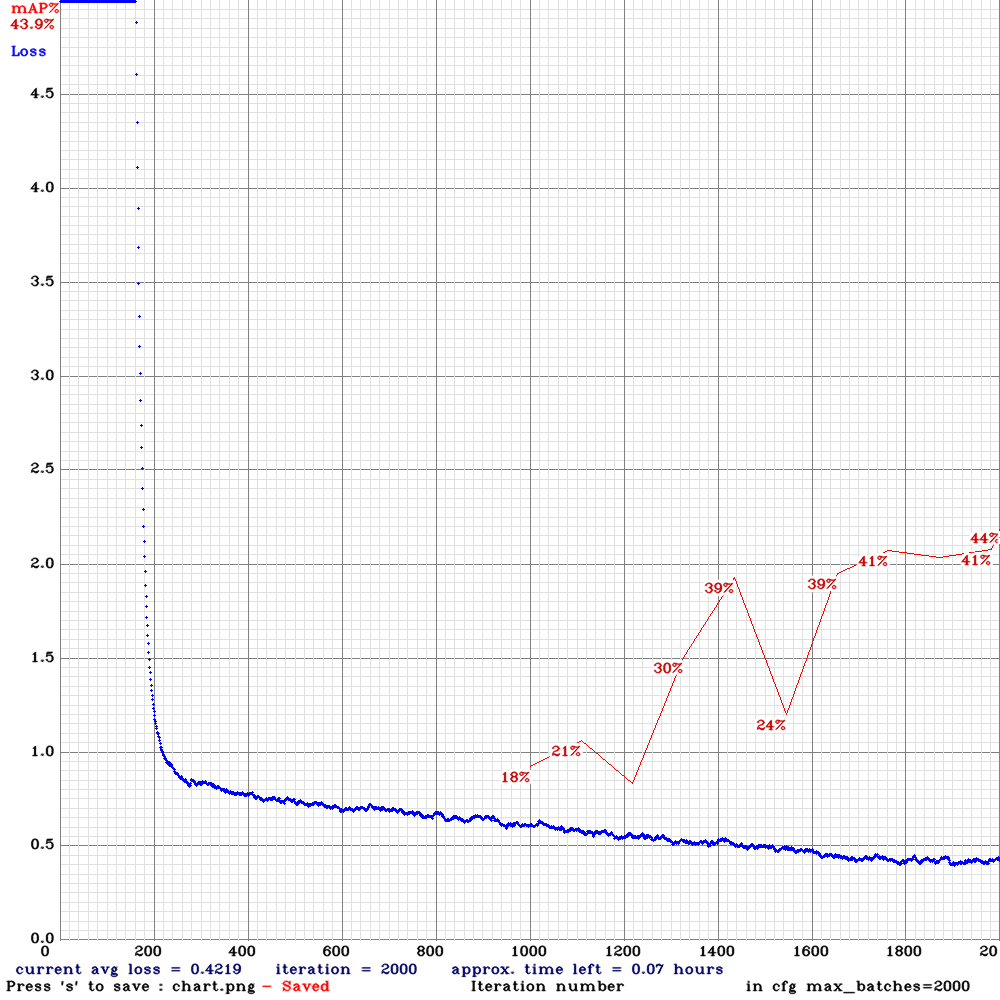
ניסוי 1 – ניסוי זה היה בשימוש הגדרות ברירת המחדל של DarkNet. המודל **לא הניב תוצאות שסיפקו אותנו**. לאחר התייעצות עם המנחה החלטנו לנסות לבצע אימון על רזולוציה גדולה יותר (ניתן לראות תוצאות ב[ניסוי 2](#ניסוי2)).

תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 416X416.

****

**Date Model yoloV4Tiny image size 416x416**

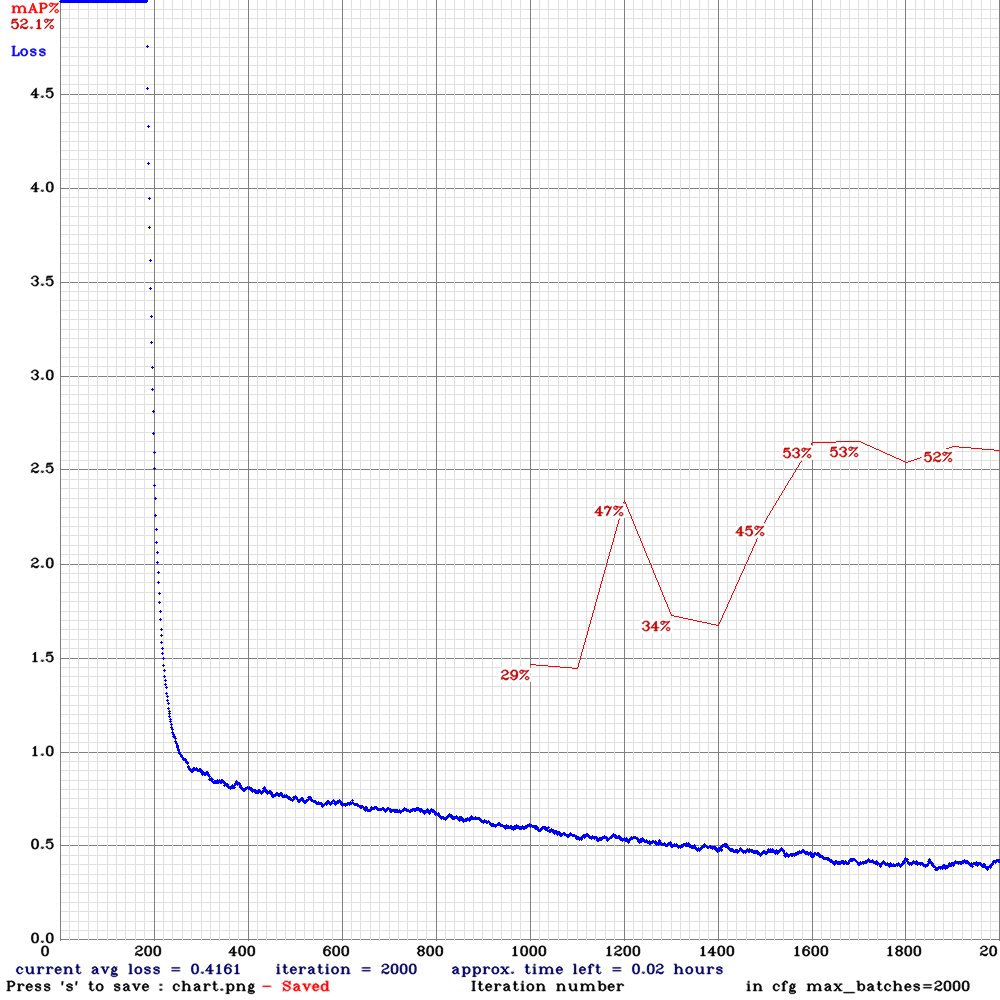
ניסוי 2 –ניתן לראות שביצוע אימון על רזולוציה גדולה יותר **אכן עזר אבל לא מספיק** בשביל לבצע object detection באפליקציה ב-.real time לאחר התייעצות נוספת עם מנחה הפרויקט, החלטנו להתאים את ה-Aspect Ratio של ה-Anchor Boxes מחדש. כמו כן, הוספנו data וסימנו את ה labels שעל גבי התמונות מחדש ובצורה מדויקת יותר (ניתן לראות תוצאות ב[ניסוי 3](#ניסוי3)).

****תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 832X832.

**Date Model yoloV4Tiny image size 832x832**

ניסוי 3 – ניתן לראות שהמודל **המשיך להשתפר**. פנינו לקהילת המפתחים של DarkNet בכדי לקבל ייעוץ נוסף ואולי להגיע לשפור נוסף. לאחר התייעצות עם גורמים בקהילת DarkNet, הובהר לנו כי עדיף להישאר עם הגדרות ברירת המחדל שלהם ולא לשנות את ה-Aspect Ratio אם אנחנו לא מומחים בתחום (ניסויים [4](#ניסוי4) ו-[5](#ניסוי5) מתארים את המודלים שהתקבלו לאחר הוספת data ושיפור הדיוק ב-labels על גבי התמונות בזמן הכנת ה-data).

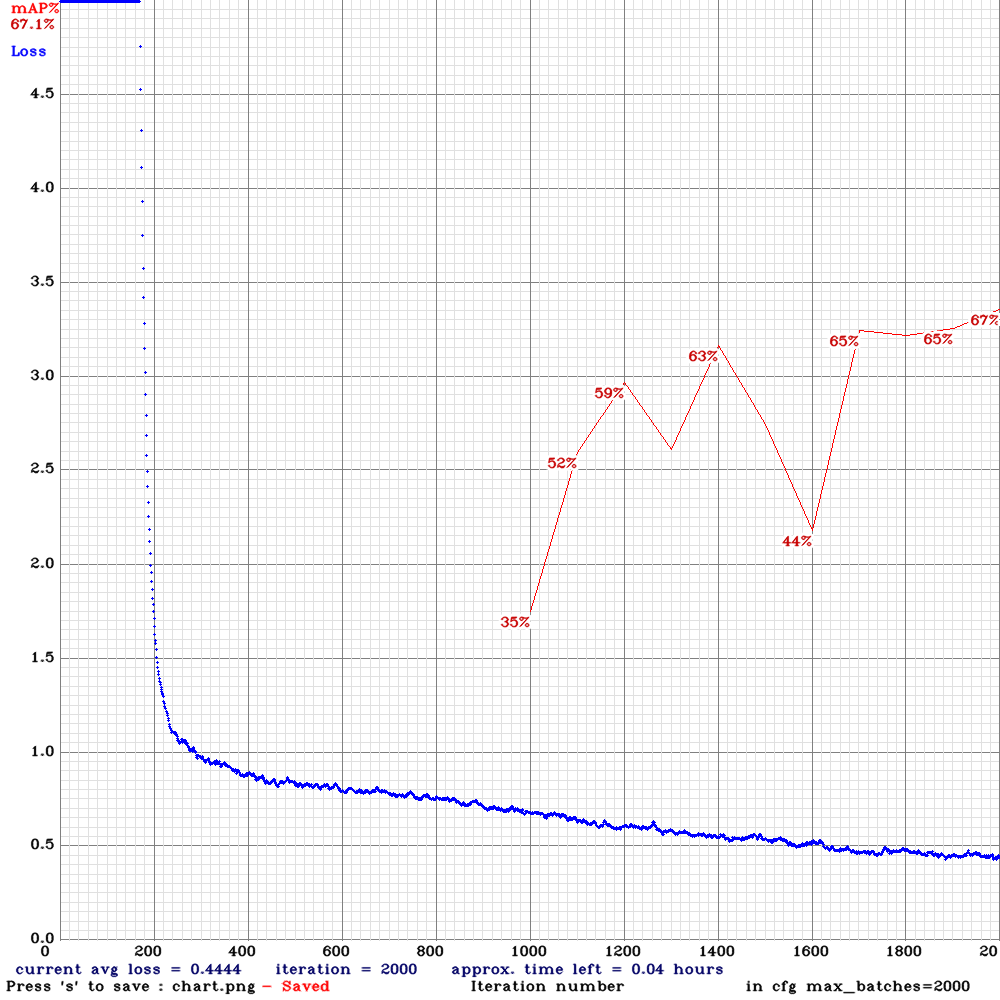
תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 416X416.

****

**Date Model yoloV4Tiny image size 416x416 b**

ניסוי 4 – תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 640X640.

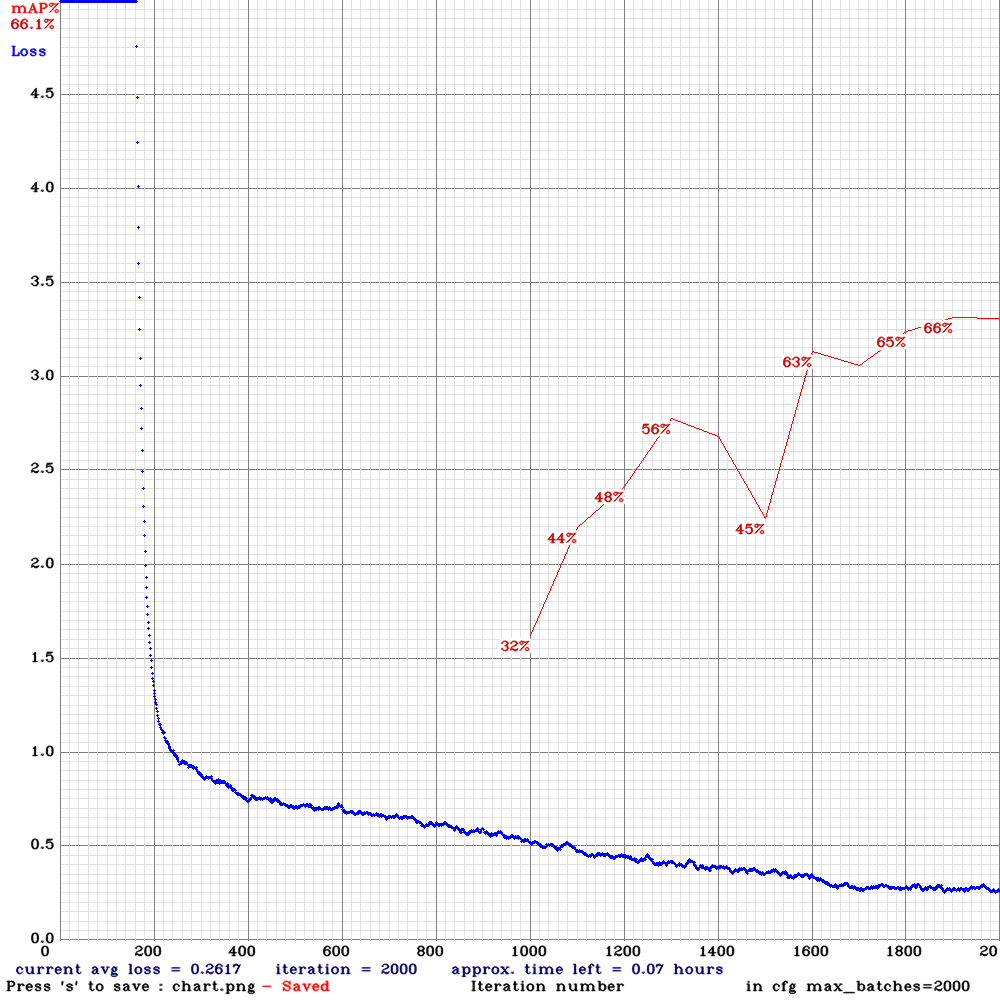
מבדיקה על קבוצת האימון, אשר כללה כ- 100 תמונות, המודל צדק בכ-**74%** מהפעמים.

****

**Date Model yoloV4Tiny image size 640x640 b**

ניסוי 5 – תמונות האימון התחלקו כך ש-80% מהן הוגדרו כקבוצת אימון וה-20% הנותרים הוגדרו כקבוצת ה-validation. כמו כן, הן נכנסו למודל ברזולוציה של 832X832.

מבדיקה על קבוצת האימון, אשר כללה כ- 100 תמונות, המודל צדק בכ-**69%** מהפעמים.

****

**Date Model yoloV4Tiny image size 832x832 b**

**סיכום והערות נוספות**

מודל המדבקות הניב תוצאות טובות כבר מהרגע הראשון ולכן המורכבות הייתה לבחון מי מבין הארכיטקטורות תשמש אותנו בצורה הטובה ביותר (כפי שצוין לעיל הארכיטקטורה yoloV4Tiny נבחרה).

בנוגע למודל התאריכים, בוצעו ניסויים רבים בנוסף לניסויים שהוצגו לעיל, כגון –

1. שינוי גודל קבוצת הוואלידציה.
2. רזולוציות נוספות שהניבו תוצאות פחות טובות או ללא שינוי.

לסיכום, המודל שאותו בחרנו הוא המודל שהוצג ב[ניסוי 4](#ניסוי4), לא רק בגלל שהוא זכה לתוצאות טובות יותר, אלא בגלל שהוא עובד טוב יותר באפליקציה בהשוואה למודלים אחרים שיצרנו.

**דוגמאות ממודל תוויות הסימון התזונתי**

**ssd\_mobilenet\_v1**

****

****

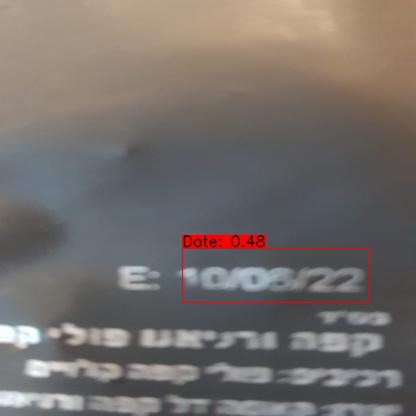
**yoloV4Tiny**

****

****

**דוגמאות ממודל התאריכים**

**yoloV4Tiny**

**עם אחוז ודאות**

**ללא אחוז ודאות**

****

**פיתוח אפליקציה** (חזרה ל[מחקר ופיתוח](#מופ))

**Backend**

פיתחנו את האפליקציה בשפה Java בסביבה המוכרת, Android Studio. התשתית של האפליקציה היא של Tensor Flow.

כפי שציינו לעיל, שתי הארכיטקטורות שבהן השתמשנו, SSD ו-yolov4tiny, אלו ארכיטקטורות שניתן להריץ על טלפונים חכמים. בכדי להטמיע את המודל באפליקציה, המרנו את המודל שיצרנו בארכיטקטורת yolov4tiny לפורמט TFLite.

בתחילת השימוש באפליקציה, המשתמש מקבל חיווי קולי שמתריע על שתי האופציות שהוא יכול לבחור לסרוק – מדבקות או תאריכים. החיווי הקולי נוסף לאפליקציה ע"י ספריה מוכרת של אנדרואיד.

בהמשך, המשתמש מתבקש לבצע סריקה של המוצר ע"מ לזהות את אחד משני הפרטים הבאים –

1. **תווית סימון תזונתי**

בבחירת אופציה זו, תיפתח המצלמה לסריקה שבה כל סימון תזונתי שזוהה ע"י המודל שיצרנו נכנס למבנה נתונים מסוג ConcurrentLinkedQueue. במידה והתבצע זיהוי של סימון אחד, המצלמה תאפשר המשך סריקה למשך 5 שניות נוספות על מנת לאפשר למודל לזהות סימונים נוספים שמופיעים על המוצר במידה וקיימים. כמו כן, לאחר זיהוי סימון אחד או מספר סימונים, המשתמש יקבל חיווי קולי עם שמות הסימונים התזונתיים שזוהו. במידה ולא התבצע זיהוי כלשהו המצלמה תישאר פתוחה עד לזיהוי של סימון תזונתי אחד לפחות. לבסוף, למשתמש מופיעות שתי אפשרויות – לבצע סריקה חוזרת או לחזור לתפריט הראשי.

1. **תאריכים**

באפשרות זו, בנוסף למודל שיצרנו, נעשה שימוש במנוע OCR.

ביצענו מחקר כדי להבין באיזה מנוע OCR אנחנו יכולים להשתמש ע"מ לקרוא את התאריך מתוך התבנית שמתקבלת כתוצאה ממודל התאריכים.

מנוע ה-OCR הראשון שמצאנו הוא Tesseract\_OCR. לאחר מספר ניסיונות הבנו שהמנוע מסובך לתפעול ולכן החלטנו לחפש מנוע OCR פשוט יותר שיוכל לפתור את אותה הבעיה. מבדיקה נוספת, עלה מנוע ה-OCR של גוגל – Google Vision.

בבחירת אופציה זו, תיפתח המצלמה לסריקה עד לזיהוי תבנית (תמונה מוקטנת) של תאריך ע"י המודל שיצרנו. התבנית נשלחת למנוע OCR, Google Vision, שיפענח את הטקסט שנמצא בתבנית. המשתמש מקבל חיווי קולי עם התאריך תפוגה שנמצא. לבסוף, למשתמש מופיעות שתי אפשרויות – לבצע סריקה חוזרת או לחזור לתפריט הראשי.

**Frontend**

עיצוב האפליקציה נעשה תוך מחשבה על קהל היעד שישתמש בה. כלומר, מכיוון שמדובר בכבדי ראייה, העיצוב צריך להיות מחולק כך שהרקע כהה והכיתוב בהיר (בשביל ליצור את הניגודיות המוחלטת ובכך להקל על המשתמשים). בנוסף לכך, באפליקציה ישנם כפתורים גדולים ובתוכם כיתוב גדול בכדי למזער טעויות של משתמשים ולהנגיש להם את השימוש בצורה הטובה, הפשוטה והקלה ביותר.

כמו כן, קיים פער גדול בין תפעול האפליקציה (פשוט ונוח למשתמש) לבין האלגוריתמיקה של האפליקציה (מורכבת מאוד). פער זה בא ללמד כי עם מספר לחיצות מועט בטלפון החכם, ניתן לבצע דברים שעבור האחר הם עולם ומלואו.

**קשיים**

במהלך העבודה על הפרויקט נתקלנו במספר קשיים –

1. **Data** – ראינו כי לא קיים data באינטרנט שמכיל מוצרים עם הסימון התזונתי או מוצרים עם תאריכי התפוגה שלהם, לכן, נאצלנו לאסוף את ה-data בעצמנו. נעזרנו בחברי הקבוצה ובסביבתם כדי לאסוף כמות גדולה יותר של data.
2. **זיהוי תאריכים** – כשניגשנו לתחום התאריכים, נוכחנו לגלות שמדובר בדבר מורכב מכמה סיבות –
   1. זיהוי התאריכים תלוי בשני דברים –
      * 1. מודל שניצור שיזהה תבנית של תאריך.
        2. מנוע OCR שיקרא את התאריך שנמצא על אותה התבנית.
   2. קיים ריבוי של תבניות של תאריכים –
      * 1. dd/mm/yyyy.
        2. dd/mm.
        3. dd/mm/name of month/yyyy.

כשניגשנו לפתור את הבעיה, החלטנו לצמצם אותה ולהתרכז בזיהוי של תאריכים מהתבנית – dd/mm/yyyy.

**שימוש במנוע OCR** – ביצענו מחקר כדי להבין באיזה מנוע OCR אנחנו יכולים להשתמש ע"מ לקרוא את התאריך מתוך התבנית שמתקבלת כתוצאה ממודל התאריכים.

מנוע ה-OCR הראשון שמצאנו הוא Tesseract\_OCR. לאחר מספר ניסיונות הבנו שהמנוע מסובך לתפעול ולכן החלטנו לחפש מנוע OCR פשוט יותר שיוכל לפתור את אותה הבעיה. מבדיקה נוספת, עלה מנוע ה-OCR של גוגל – Google Vision. נוכחנו לגלות כי המנוע עונה על הדרישות שלנו.

**שיפורים**

מפאת קוצר הזמן והמיעוט ביכולתנו להשתמש במנועים מתקדמים יותר, עלו לנו מספר שיפורים שניתן לבצע –

1. **זיהוי תאריכים** – השלב הבא הוא לבצע זיהוי מקיף של כלל תבניות התאריכים (שהוזכרו לעיל) ולשפר את תוצאות הזיהוי ע"י איסוף נרחב של data מתאים.
2. **זיהוי תוויות סימון תזונתי** – ביצוע זיהוי הסימונים התזונתיים המופיעים על גבי המוצר הספציפי שאנו מחזיקים **ורק על גבי אותו מוצר**. כרגע, האפליקציה מזהה את הסימונים התזונתיים שנמצאים על גבי המוצר שהמשתמש מחזיק ובמידה ויש מוצרים נוספים מאחוריו עם סימון תזונתי הוא יזהה גם אותם. אנו רוצים להימנע מסיטואציה זו.

**תוצרי הפרויקט**

כאשר המשתמש נכנס לאפליקציה הוא יכול לבחור האם לבצע סריקה לתאריך תפוגה או האם לסרוק תווית, במקביל הוראות קוליות לתפעול האפליקציה מושמעות להגברת הנגישות.

**בעת לחיצה על סריקת התוויות**, נפתחת מצלמת הטלפון החכם והמשתמש יכול "לסרוק" (לסובב את המוצר ב-360 מעלות מסביב למצלמה בכדי שהתוויות אכן יופנו כלפי המצלמה) את המוצר אשר ברצונו לבדוק. בעת הסריקה, רשת הנוירונים תזהה את התוויות שעל המוצר. לבסוף, בסיום הסריקה, ייפלט חיווי קולי אודות התוויות/ התוויות שעל המוצר הרלוונטי.

**בעת לחיצה על סריקת תאריכים**, נפתחת מצלמת הטלפון החכם ורשת הנוירונים מחפשת אחר תבנית של תאריך. ברגע שהתבנית זוהתה, התמונה תיחתך סביב גבולות התאריך ותישלח למנוע OCR אשר ימיר את התמונה החתוכה לטקסט בליווי חיווי קולי של התאריך.

VisiFood מעניקה ללקוח את האפשרות לזהות את הפרטים הנ"ל ללא כל צורך בראייה כלל. בעזרת סיבוב המוצר ב-360 מעלות מול המצלמה, הנתונים ייאספו וברגע שהזיהוי יתבצע התוצאה תיאמר בקול.

**חלוקת עבודה**

גיבוש הרעיון, מחקר ובניית הרשתות – עבודה משותפת של כל הקבוצה.  
TTS ו- backendשל תוויות סימון תזונתי – גיא ויניב.  
Frontend והטמעת המודלים של תוויות הסימון והתאריכים באפליקציה – ניר, אסף ומעיין.

**ביבליוגרפיה**

* Training a TensorFlow MobileNet Object Detection Model

<https://blog.roboflow.com/training-a-tensorflow-object-detection-model-with-a-custom-dataset/>

* Train a Custom Mobile Object Detection Model (with YOLOv4 Tiny and TensorFlow Lite)

<https://blog.roboflow.com/how-to-train-a-custom-mobile-object-detection-model/>

* Expiry date recognition using deep neural networks (Vlad Florea, Traian Rebedea University POLITEHNICA of Bucharest)

<http://rochi.utcluj.ro/ijusi/articles/IJUSI-13-1-Florea.pdf>

* TensorFlow Lite converter

<https://www.tensorflow.org/lite/convert>

* tensorflow-yolov4-tflite

<https://github.com/hunglc007/tensorflow-yolov4-tflite>

* Run apps on a hardware device

<https://developer.android.com/studio/run/device>

* Darknet: Open Source Neural Networks in C

<https://pjreddie.com/darknet/>

* Detect text in images

<https://cloud.google.com/vision/docs/ocr>

* Tesseract OCR

<https://github.com/tesseract-ocr/tesseract>

* Documentation for app developers

<https://developer.android.com/docs>

**Summary of VisiFood – Project**

More than 250 million people are blind or visually impaired at some level in the world.

Due to the disabilities, during their daily life they encounter many difficulties in various fields. One of the most prominent areas is the retail field.

Today, a visually impaired person uses various means when shopping at the supermarket, such as -

1. Using their sense of touch to identify the product they have selected.
2. Getting accompanied by another person to the grocery store.
3. Asking for help from people in the store.

These measures, excluding #1, reduce that person's sense of independence.

Also, there are various symbols providing info on the product that are printed on the product so that the sense of touch will not help in this case, such as the labels of the Ministry of Health (sugar, sodium, saturated fat and the green sticker) and expiration dates.

VisiFood wants to improve the supermarket shopping experience for visually impaired people.

Using an app, with a minimal design tailored for those people, VisiFood gives them the ability to be more independent by "seeing" in their place the details about the products.

The application allows the user to select one of the two options at a given time -

1. Marking label identification.
2. Date identification.

The identification process of each of the options is performed by two models built for the application.

Also, using the app for the user is simple: the user has to select one of the options, aim the camera at the product and from there the app - using the models mentioned above - sends the user a voice indication detailing the relevant results obtained.

VisiFood opens doors for visually impaired people, creates a friendly shopping experience for them and reduces their dependency on the environment.

**Our Repository:**

[**https://github.com/AsafWeitzman/FinalProjectVisiFood**](https://github.com/AsafWeitzman/FinalProjectVisiFood)

1. <https://he.wikipedia.org/wiki/%D7%9C%D7%A7%D7%95%D7%AA_%D7%A8%D7%90%D7%99%D7%99%D7%94> [↑](#footnote-ref-1)
2. <https://www.orcam.com/he/> [↑](#footnote-ref-2)