

# סמינר פרויקט מסכם

מסמך הפרויקט



שמירה על הנחיות הקורונה באמצעות ראייה ממוחשבת

## מגישים:

אביחי צרפתי 204520803

אסף אטיאס 308214899

דביר שמחון 20433166

בהנחיית פרופ' גיא שני

ינואר 2021

# תוכן עניינים

4	1 תקציר
5	2 חקר המצב הקיים
5	2.1 המצב הקיים
5	2.1.1 תיאור הלקוחות
5	2.1.2 מערכות קיימות
7	2.2 סקירה טכנולוגית
11	2.3 סקר ספרות
14	2.4 ביבליוגרפיה
16	3 חקר דרישות ואפיון ראשוני
16	3.1 מטרת המערכת
16	3.2 תהליכים עסקיים
16	Use Cases 3.2.1
19	Business flow chart 3.2.2
20	3.3 דרישות מערכת
20	3.3.1 דרישות פונקציונאליות
21	3.3.2 דרישות לא פונקציונאליות
22	3.4 מתודולוגיה
24	3.5 מאגר נתונים
25	3.6 תכנון הממשק
25	Class Diagram – ממשק לוגי 3.6.1
26	3.6.2 ממשק התצוגה – תכנון מסכים
28	3.7 הערכת מצב
28	3.7.1 אתגרים
28	3.7.2 סיכונים
30	3.8 תוצרים
30	WBS, לוח זמנים וחלוקת אחריות
30	3.8.1 אבני דרך
30	3.8.2 WBS, חלוקת אחריות ותיאור לו"ז מפורט

32	4 חלוקת אחריות
33	5 הפניות
33	6 נספחים

## תיאור

מסמך זה מספק פרטים על דרישות המערכת (SRS), תכנון והטמעה (SDD) של הפרויקט OpenCoVid.

## 1 תקציר

בדצמבר 2019, מגיפה עולמית בשם Covid-19 החלה להתפשט ברחבי העולם. על פי נתונים שהתקבלו על ידי ארגון הבריאות העולמי (WHO), נכון לכתיבת שורות אלה, יותר מ-66 מיליון אנשים ברחבי העולם נדבקו בנגיף. בעקבות כך, הוציא הארגון הנחיות התנהגות בשעת חירום כגון: ריחוק חברתי של שני מטרים בין אדם לאדם, חבישת מסכת פה ואף במקומות ציבוריים, מדידת חום בכניסה למשרדים ובתי עסק וכדומה, על מנת למנוע את התפשטות הנגיף.

הצורך בביצוע מעקב ואכיפת השמירה על ההנחיות הציב קשיים מהותיים בפני מדינות העולם. קשיים אלו הביאו רבים, וביניהם גם אותנו, לחפש אחר פתרונות יעילים, זמינים ונגישים, שיקלו על ההתמודדות ויאפשרו להאט את קצב ההדבקה.

כדי ליצור סביבה בטוחה התורמת לשלום הציבור, אנו מציעים פתרון יעיל המבוסס ראייה ממוחשבת ומתמקד בניטור אוטומטי בזמן אמת של בני אדם במרחב. באמצעות כלים טכנולוגיים מתקדמים נוכל לזהות אי-עמידה בהנחיות המהותיות ובכללן: אי-שמירה על ריחוק חברתי ואי-עטיית מסכת פה ואף.

בפתרון שלנו אנו נשתמש באלגוריתם למידה עמוקה המשלב טכניקות גיאומטריות לחישובי מרחקים ובכך לסייע לגורמי אכיפה ופיקוח בניטור, פיקוח ומעקב אחר מפירי הנחיות, בקלות, ללא עלויות גבוהות, תוך חיסכון משמעותי בכוח אדם.

בפתרון המוצע, נשתמש ברשת עצבית על מנת לנתח זרמי וידאו בזמן אמת באמצעות אלגוריתמי זיהוי אובייקטים State-of-the-Art כגון YOLO ו-Faster R-CNN בשילוב טכניקות גיאומטריה קלאסיות.

בהשוואה לפתרונות המוצעים בשוק כיום, המציעים זיהוי עטיית מסכת פה ואף וריחוק חברתי כפיצ'ר משני מתוך מערכת קיימת, אנו מציעים פתרון ייעודי עבור הבעיה המוצגת לעיל, נגיש וחינמי, המבוסס על טכנולוגיית ראייה ממוחשבת, לקבלת תוצאות מדויקות במהירות.

המערכת לא מצריכה רכישת מצלמה ייעודית או כל סנסור שהוא כחלק ממערכת גדולה יותר, אלא ביכולתה להתחבר לכל מצלמה שהיא, ולהפיק פלט מיידי ומדויק.

הפתרון אינו רק כלי ניטור וסיוע באכיפה. בעזרת המערכת ניתן יהיה להעריך סטטיסטית את סדר הגודל של שמירת ההנחיות במרחב הציבורי, ומתוך כך לגזור החלטות משמעותיות הנוגעות לניהול המגיפה - פתיחת עסקים לקהל הרחב, מופעי תרבות וספורט, חדרי כושר וכיו"ב.

אנו מאמינים כי המערכת עשויה להביא לתועלת רבה ולחסוך כמויות אדירות של משאבים וכסף בתקופה מאתגרת מבחינה כלכלית ובריאותית, עבור כלל מדינות העולם.

לפרויקט ישנה מטרה אלטרואיסטית ואינו נוצר לשם יצירת רווח כלכלי. עם זאת, נדגיש כי המערכת מהווה הוכחת היתכנות (POC - Proof Of Concept) בלבד ולא באה לפתור את הבעיה בכלליותה ולעומקה.

## 2 חקר המצב הקיים

### 2.1 המצב הקיים

#### 2.1.1 תיאור הלקוחות

כאשר מגפה מתפשטת בעולם יש לה השפעה על תחומים רבים: תחלואה, כלכלה, תנועה ועוד. בניסיונות למגר את המגפה הממשלות מטילות הנחיות ואיסורים אשר משפיעים על התנהגות כלל האוכלוסייה, העסקים והעיריות, אי ציות עלול לגרום להחמרת המגפה ומקרי מוות, סגירה והגבלת עסקים ומקומות נוספים, כיום אין אמצעי מדידה ישיר לציות האוכלוסייה להנחיות. משתמשים במדדים עקיפים כמו אחוז בדיקות חיוביות ביום וכמות חולים מאומתים, מדדים אלו נותנים הערכה על החולים רק לאחר שנדבקו כבר ואינם מצביעים על ציות בזמן אמת, בנוסף אמצעי האכיפה כמו חלוקת דוחות (ומדידת הכמות שלהם) הוא אמצעי מוגבל מאוד במשאבים, בעולם ה IOT כיום מצלמות מפוזרות במקומות רבים: עסקים, מקומות ציבוריים, מקומות ממשלתיים ועוד, המערכת שלנו פותרת את הבעיה בעזרתם ומסוגלת למדוד מדדי ציות ישירים בזמן אמת. זה באינטרסים של גופים רבים להשתמש בהם למעקב אחר שמירת ההנחיות של האוכלוסייה שלהם כדי לפתור את הבעיות שלהם, להלן מספר דוגמאות של בעיות/מקרים:

- ממשלה/משרד הבריאות - מעקב אחר ביצוע ההנחיות שהם קובעים, זיהוי ערים חריגות.
- עיריות - מעקב אחר האוכלוסייה, זיהוי נקודות תורפה על מנת לא להפוך לעיר אדומה.
- עסקים - עסק לא נסגר/נקנס בגלל תחלואה, העובדים בריאים ויכולים לייצר תפוקה.
- בתי חולים - שמירה על סטריליות ובריאות החולים והצוות הרפואי.
- שדה תעופה - זיהוי, בדיקה ובידוד אנשים שלא מקפידים על ההנחיות ועלולים להוות גורם הפצה שהגיע מחוץ למדינה.

#### 2.1.2 מערכות קיימות

- **AuraVision**, אתר: [auravision.ai/covid-solutions](https://auravision.ai/covid-solutions)

חברה זו מציעה שירות של מעקב בעזרת חיבור וניהול מצלמות אבטחה, בינה מלאכותית וראייה ממוחשבת. המערכת שלהם מיועדת לעזור לעסקים להבין את התנהגות הלקוחות והעובדים שלהם, בנוסף היא מאפשרת לבצע מעקב אחר שמירת הנחיות בידוד (לבישת מסכה וזיהוי התקהלות). המערכת היא Plug & Play למגוון רחב של מצלמות.

שונה מאיתנו: מאפשרת חלוקה וניתוח (גם לאזורים, עובדים/לקוחות..), פיצ'רים נוספים לעסקים ללא קשר למגיפה, צריך לרכוש אותה, מאפשרת דיווחים, שליחת מסרים, זיהוי ורישום בני אדם (משתמשים), ניהול מגיפה הוא כלי משני של המערכת.

• **LeewayHertz**, אתר: [eewayhertz.com/covid-19-technology-solutions](https://eewayhertz.com/covid-19-technology-solutions)

חברת הייטק בעלת מוצרים במגוון תחומים, המציעה גם פתרונות (תוכנות) לזיהוי לבישת מסכה וריחוק חברתי בעזרת חיבור למצלמות אבטחה וראייה ממוחשבת. כל שירות עומד בפני עצמו וניתן לרכוש אותו בנפרד. ישנם פתרונות נוספים שהם מציעים להתמודדות עם המגיפה כמו זיהוי מקומות שיש בהם התקהלות חוזרת בעזרת ראייה ממוחשבת.

שונה מאיתנו: לא ALL IN ONE, צריך לרכוש את השירותים, מאפשר דיווחים ושליחת מסרים, מאפשר רישום ומשתמשים.

• **AindraLabs**, אתר: [aindrallabs.com/products/pandemic-management-system](https://aindrallabs.com/products/pandemic-management-system)

חברה המפתחת מוצרי בינה מלאכותית עבור ניהול עסקים, אחד המוצרים שלה הוא מערכת לניהול מגיפה, מערכת זו משתמשת בראייה ממוחשבת וכלים נוספים על מנת לזהות את התנהגות האנשים, לבישת מסכה, ריחוק חברתי, לבישת כפפות, התנהגות חשודה, טמפרטורת גוף ועוד. במהלך ביצוע המעקב המערכת מסוגלת להתריע לרשויות על הפרות או חשד לחולים, מערכת זאת מאפשרת רישום בעזרת אפליקציה ומשתמש על מנת להתריע לאנשים על סיכונים והפרות.

שונה מאיתנו: צריך לרכוש אותה, חיבור למערכת בעזרת רישום באפליקציה ייעודית, מאפשרת זיהוי מורכב יותר חוץ משמירה על הנחיות ושולחת מסרים ודיווחים.

במהלך הסקירה נוכחנו לגלות כי פתרונות רבים הדומים לפתרון שלנו כבר מומשו בצורות שונות, נקודות השוני המרכזיות בין מערכות אלו אל המערכת שאנו מציעים הוא מטרת המערכת והעלות שלה, המערכת שלנו ממוקדת בניתוח הצילומים ומספקת מידע המאפשר שליטה במגיפה, רכיב זה ניתן להוסיף לכל מערכת ללא תלות (כמו הממשק הגרפי שלנו) לעומת מערכות בקרה וניהול עסק אשר מימשו פתרון זה כתוסף למערכת הראשית שלהם.

**להלן טבלת השוואת תכונות מרכזיות בין המערכות שמסכמת את השוני בין המערכות:**

מערכת	זיהוי מסכה	זיהוי מרחק	Real Time	ALL IN ONE	מערכת ייעודית לניהול מגיפה	דורש רישום משתמשים	עלות כספית
AuraVision	V	V	V	V	X	V	כן
LeewayHertz	V	V	V	X	X	V	כן
AindraLabs	V	V	V	V	V	V	כן
OpenCovid	V	V	V	V	V	X	לא

## 2.2 סקירה טכנולוגית

- Technologies We use:
  - Development Tools
    - PyCharm
    - Google Colab
  - Programming Languages
    - Python 3
  - Packages and Modules
    - OpenCV 3
  - DL Frameworks
    - Tensorflow 2
    - sklearn
  - Algorithms
    - YOLOv4/v5
    - Faster R-CNN
  - Source and Management
    - Git and Github
    - Trello

### DEVELOPMENT TOOLS

#### - Pycharm

סביבות פיתוח משולבת לפיתוח תוכנות, בעיקר בשפת Python.

#### *השוואה למול הטכנולוגיה המתחרה Visual Studio Code.*

##### יתרונות

- **עריכה וניתוח קוד** - Pycharm מאפשר עריכה נוחה ופשוטה של קוד באמצעות עזרה סינטקטית. כמו כן, הוא כולל תמיכה מובנית להשלמת קטעי קוד, ניווט קל בקוד ושחזור היסטוריה בצורה קלה ומהירה בעת הצורך. על אף שגם VS Code מכיל את מרבית התכונות הללו, Pycharm מראה ביצועים טובים יותר.
- **ביצוע Debugging** - גם Pycharm וגם VS Code מציעות מגוון תכונות שמאפשר ביצוע Debugging נוח ויעיל. אולם, Pycharm מציעה תכונות נוספות שהופכות את ביצוע ה- Debugging לנוח אף יותר.

##### חסרונות

- **שימוש בזיכרון** - בהשוואה ל- Pycharm, סביבת הפיתוח VS Code קלה מאוד ואף מאפשרת launching מהיר מאוד, בעוד Pycharm כבדה ואיטית בהעלאה ראשונית.

לסיכום, 2 סביבות הפיתוח, Pycharm ו- VS Code, מאפשרות פיתוח יעיל ונוח, אך לאור העובדה שיתרונותיה של Pycharm גוברים על חסרונותיה למול VS Code בחרנו להשתמש בסביבת פיתוח זו.

## - Google Colab

כלי המאפשר לכתוב ולהריץ קוד פייתון דרך הדפדפן. מתאים במיוחד ללמידת מכונה וניתוח נתונים.

### השוואה למול הטכנולוגיה המתחרה Jupyter Notebook.

#### יתרונות

- **הרצה מכל מקום** - Google Colab מאפשר הרצה מכל מחשב שהוא, כל המחברות נשמרות ב - Google Drive, ללא צורך בהתקנות קודמות.
- **GPU חופשי** - הפרויקט שלנו מצריך שימוש ב - GPU's בעלי כח חישוב גבוה מאוד. Google Colab מאפשר הרצה מרוחקת על מכונות בעלות GPU's איכותיים, ללא עלות.

#### חסרונות

- **הוספת תוכן לתאים בהרצה** - Jupyter Notebook מאפשרת הוספת תוכן עשיר לכל תא קוד, בהשוואה ל - Google Colab.

לסיכום, Google Colab הינו הכלי המוביל בשוק בתחומו. בעיקר האופציה לשימוש ב - GPU's חזקים בצורה מרוחקת, שכל כך נדרשת לביצוע הפרויקט, היא זו שהובילה אותנו להשתמש בכלי.

## PROGRAMMING LANGUAGE

### - Python

פייתון היא שפה מרובת-פרדיגמות, המאפשרת תכנות מונחה-עצמים, תכנות פרוצדורלי, ובמידה מסוימת גם תכנות פונקציונלי (מתוך: ויקיפדיה).

### השוואה למול Java

#### יתרונות

- **תחביר** - Python שפה קלה מאוד ללמידה ושימוש ומאפשרת ביצוע פעולות מורכבות באמצעות Syntax פשוט ובהיר, לעומתה Java מורכבת יותר ופחות אינטואיטיבית וכן אינה מאפשרת scripting.
- **זמן פיתוח** - משך זמן פיתוח ממוצע ב - Python קצר משמעותית מאשר ב - Java.

#### חסרונות

- **ביצועים** - באופן כללי, Java מהירה יותר מ - Python. העובדה ש - Python כה אינטואיטיבית עבור מפתחים מצריכה ביצוע פעולות רבות נוספות מאחורי הקלעים, שמתרחשות בזמן הריצה ולכן מאריכות אותו.

לסיכום, גם Python וגם Java שתיהן שפות נפוצות מאוד ובעלות מגוון רחב של ספריות, אך יתרונותיה של Python, בוודאי בהקשרי פיתוח בעולמות ה - Machine Learning וה - Computer Vision, הביאו אותנו לבחור בשפה זו לפיתוח הפרויקט.



## COMPUTER VISION PACKAGES

### - OpenCV

היא חבילת תוכנה שנועדה לעזור לפתח יישומים של ראייה ממוחשבת.

#### *השוואה למול Scikit-Image*

#### יתרונות

- **Real-Time Video** - ל-OpenCV קיימת אופציה לביצוע ניתוח ב-Real-Time על וידאו ואילו ל-Scikit-Image אין את היכולת הזו.
- **ביצועים** - OpenCV נכתב ב-C++ ולכן מציג ביצועים טובים יותר מאשר Scikit-Image שכתוב ב-Python.

#### חסרונות

- **קלות שימוש** - לדעת רבים, Scikit-Image קלה ואינטואיטיבית יותר לשימוש מאשר OpenCV.
- לסיכום, 2 החבילות נמצאות בשימוש רחב מאוד בקרב מפתחים ושתייהן מציגות ביצועים טובים ודיקן יחסית מיטבי, אך ל-OpenCV קיימים מספר יתרונות על Scikit-Image, בעיקר בהיבטי הביצועים.

## DL FRAMEWORKS

### - TensorFlow

ספריית קוד פתוח ללמידת מכונה לבניית ואימון רשתות נוירונים.

#### *השוואה למול PyTorch*

#### יתרונות

- ל-TensorFlow יש **קהילת משתמשים** גדולה בהשוואה ל-PyTorch. לאור עובדה זו, קיים מידע רב וכך ניתן למצוא משאבים בקלות יתרה, פתרונות לבעיות וכדומה.
- TensorFlow כדאי יותר עבור **Production models** ו-**Scalability**.
- **TensorBoard** - כלי שקיים רק ב-TensorFlow ומאפשר לבצע ויזואליזציות של המודלים ישירות בדפדפן.

#### חסרונות

- **עדכון גרפים דינאמי** - PyTorch מאפשר ביצוע מניפולציות ועדכון על גרפים קיימים (כלומר כאלו שנבנו ואומנו) ללא צורך בהרצה מחודשת. רוב הזמן, הרצת מודלים הינה פעולה ארוכה שדורשת זמן רב, ולכן יש ערך רב בביצוע מניפולציות שכאלה on-the-fly.

לסיכום, החלטנו לבחור ב-TensorFlow לאור הכלים המגוונים והייחודיים שמכיל, התמיכה הרחבה שניתן למצוא לשאלות וכדומה.

## - SKlearn

ספריית תוכנה ללמידת מכונה.

### השוואה למול Keras

#### יתרונות

- **מגוון וקהילת משתמשים** - SKlearn מכילה כלים יעילים רבים ללמידת מכונה כולל סיווג, רגרסיה, אשכולות, הפחתת מימדים וכן מבוססת ע"י קהילה גדולה של מפתחים מקצועיים, כך שעדיפה על Keras בהיבטים אלה.

#### חסרונות

- **תמיכה בשימוש ברשתות נוירונים** - ב-SKlearn קשה מאוד למצוא תמיכה בשימוש ברשתות נוירונים מאשר ב-Keras, עובדה די משמעותית עבורנו לאור העובדה שנרצה להשתמש לא מעט באימון ע"י רשתות.

לסיכום, לאור ההיכרות שלנו עם SKlearn ולאור העובדה ש-SKlearn ספרייה מבוססת וניתן למצוא בה, ברוב המקרים, את התמיכה הרחבה ביותר, החלטנו לבחור בספרייה זו.

## ALGORITHMS

### - YOLO

אלגוריתם לזיהוי אובייקטים ב-real-time.

### השוואה למול SSD

#### יתרונות

- **מהירות** - YOLO הינו מהיר יותר מ-SSD. למעשה בעוד YOLO מגיע ל - 60 frames לשנייה בממוצע, SSD מגיע לכל היותר ל - 22 frames לשנייה בממוצע.

#### חסרונות

- **דיוק** - YOLO משיג אחוזי דיוק נמוכים יותר מאשר SSD.

לסיכום, לאור העובדה ש-YOLO הינו אחד האלגוריתמים המהירים ביותר (אם לא המהיר ביותר) לביצוע Object Detection ושאיפתנו היא לממש זיהוי אובייקטים ב-real-time, החלטנו להעדיף ביצועים על פני דיוק.

### - Faster R-CNN

אלגוריתם לזיהוי אובייקטים.

### השוואה למול SPP

#### יתרונות

- **דיוק** - Faster R-CNN מגיע ל-70% אחוזי דיוק בממוצע ואילו SPP מגיע ל-59% דיוק בממוצע.

#### חסרונות

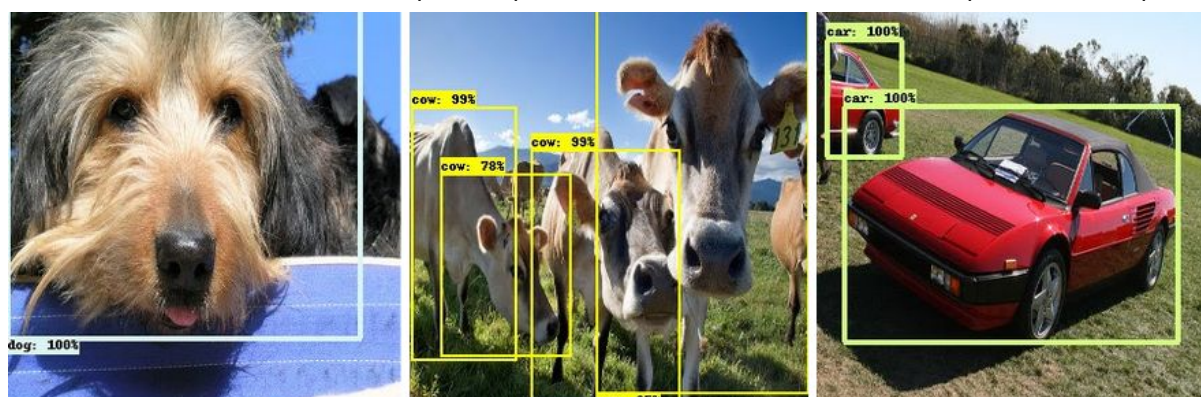
- **ביצועים** - Faster R-CNN מצליח להגיע ל-21 frames לשנייה בממוצע, ואילו SPP מגיע ל-56 frames לשנייה בממוצע.

לסיכום, נרצה לבצע השוואה בין YOLO לבין Faster-R-CNN כדי להשוות ביצועים מול דיוק, ולכן כאן נבחר כאן באלגוריתם בעל דיוק גבוה יחסית.

## 2.3 סקר ספרות

מבוא

בשנים האחרונות, טכניקות זיהוי אובייקטים (Object Detection) המשתמשות במודלים עמוקים [1] (Deep Learning) מוצלחים יותר ממודלים רדודים בטיפול במשימות מורכבות והן השיגו התקדמות מרהיבה בראייה ממוחשבת [2] (Computer Vision). מודלים עמוקים לזיהוי אדם מתמקדים בלמידת מידע קונטקסטואלי על תכונות [3] המרכיבות את אובייקט המטרה. במשימות איתור אובייקטים, המודל מכוון לשרטט תיבות תחום (Bounding box), הדוקות סביב האובייקטים הרצויים בתמונה, לצידם תיוג כל אובייקט ואובייקט.



איור 1: דוגמה לזיהוי אובייקטים, מערך נתונים VOC-2012

מודלי למידה עמוקה לזיהוי אובייקטים [4] יכולים להיות מחולקים בעיקר לשתי משפחות: (1) מודלים דו-שלביים כגון SSD, YOLO [8] ו-R-CNN [5], Faster R-CNN [7] וגירסותיהם (2) מודלים חד-שלביים כגון SSD ו-YOLO. מודל לזיהוי דו-שלבי כולל שני שלבים: הצעת אזור ואז סיווג אזורים אלה ושכלול חיזוי המיקום. זיהוי חד-שלבי מדגל על שלב ההצעה לאזור ומניב לוקליזציה סופית וחיזוי תוכן בבת אחת. גרסאות מהירות יותר של RCNN הן הבחירה הפופולארית לשימוש במודלים דו-שלביים, ואילו גלאי אובייקטים Single-Shot-Detector או בקיצור SSD ו-YOLO הם הגישה הפופולרית לחד-שלבי. ארכיטקטורת YOLO, אם כי מהירה יותר מ-SSD, היא פחות מדויקת.

## עבודות קשורות

**R-FCN** היא עוד מטא-ארכיטקטורה פופולארית דו-שלבית, בהשראת Faster-RCNN. בגישה זו, רשת הצעות אזור (RPN) מציעה מועמדים (Rols - Region of Interests), אשר מיושמים לאחר מכן על מפות ציון. כל השכבות הניתנות ללמידה הן קונבנציונליות והם מחושבות על התמונה כולה. שם, כמעט כל חישוב האזורים המוצעים השונים משותף. העלות החישובית לפי RoI זניחה בהשוואה ל-R-FCN. Fast-RCNN הוא מעין הכלאה בין הגישה החד-שלבית ודו-שלבית. כשאתה באמת בוחן את זה, אתה רואה שבעצם מדובר בגישה דו-שלבית עם כמה מהיתרונות והחסרונות של החד-שלבי.

בעוד שמודלים לגילוי דו-שלביים משיגים ביצועים טובים יותר, זיהוי חד-שלבי נמצא במקום טוב יותר ביחס לביצועים ומהירות / משאבים. בנוסף, SSD ניתן לאימון מהיר יותר ויש לו הסקה מהירה יותר ממודל דו-שלבי. אימון מהיר יותר מאפשר לבצע אב טיפוס וניסויי ביעילות מבלי לגזול הוצאות ניכרות למחשוב וחומרה. חשוב מכך, חיזוי מהיר הוא בדרך כלל דרישה בכל הנוגע ליישומים בזמן אמת.

מכיוון שהוא כרוך בפחות חישוב הוא צורך הרבה פחות אנרגיה לחיזוי. מודלים אלו פותחים דלתות חדשות למגוון רחב של שימושים, במיוחד במכשירי קצה ושמים את SSD כגישת איתור האובייקטים המועדפת על שימושים רבים.

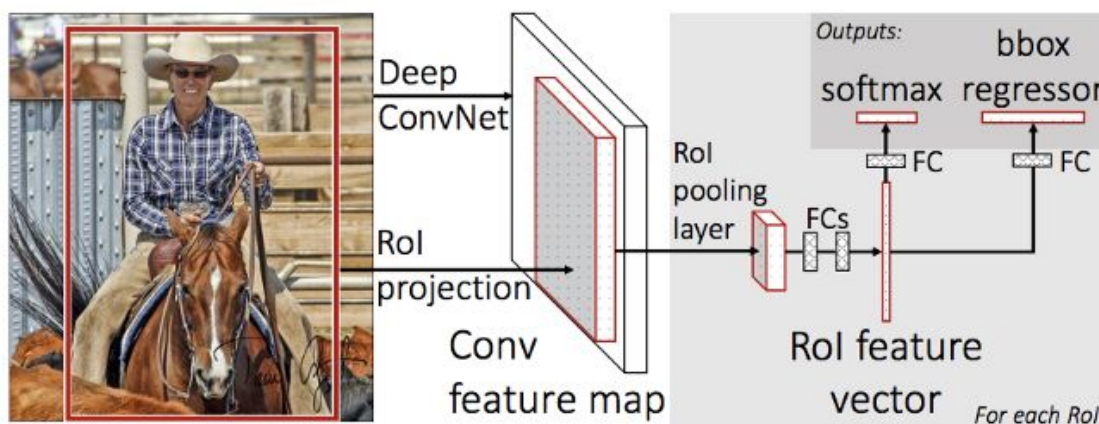
## חסרונות בשימוש R-CNN

- זה עדיין לוקח זמן גדול מאוד לאמן את הרשת, כיוון שנצטרך לסווג 2000 הצעות אזוריות לתמונה.
- לא ניתן ליישמו בזמן אמת מכיוון שלוקח כ- 47 שניות לעיבוד תמונה בודדת.
- אלגוריתם החיפוש הסלקטיבי הוא אלגוריתם קבוע. לכן, שום למידה לא מתרחשת באותו שלב. זה יכול להוביל ליצירת הצעות גרועות לאזור המועמד.

## Faster-RCNN

Faster-RCNN קורה בשני שלבים. השלב הראשון נקרא הצעת אזור. תמונות מעובדות על ידי חילוץ תכונות, כגון ResNet50, עד לשכבת רשת ביניים שנבחרה. לאחר מכן, רשת קטנה המחוברת לחלוטין (FC - fully connected) עוברת על שכבת התכונות כדי לחזות הצעות לחיזוי האובייקט, ביחס לרשת העוגנים הממופים בתמונה וביחס בין אורך לרוחב.

בשלב השני, ההצעות לתיבות אלה משמשות לחיתוך תכונות ממפת תכונות הביניים שכבר חושבה בשלב הראשון. התיבות המוצעות מוזנות לשארית מחלץ התכונות בעל ראשי חיזוי ורגרסיה, שם מחושבים תיבות ספציפיות לכל אובייקט עבור כל הצעה.



איור 2: אזורי עניין ב Faster-RCNN מחולצים על ידי שכבת FC

Faster R-CNN מתגבר על חסרונות רבים של שיטות קודמות כמו אלה שהצגנו לעיל עבור R-CNN ומשפר את מהירות והדיוק מולם.

## יתרונות בשימוש Faster R-CNN:

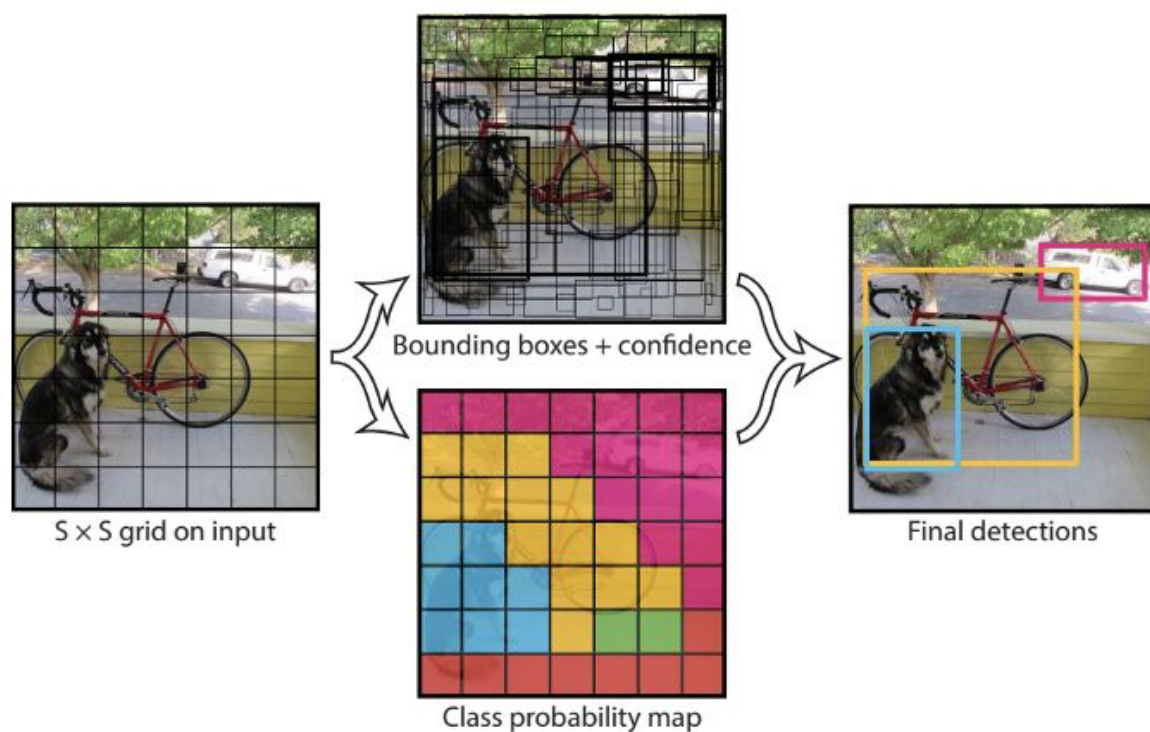
- תוצאות זיהוי טובות יותר לפי mAp על פני R-CNN
- האימון הוא חד-שלבי, תוך שימוש בעיבוד רב-משימתי
- תוך כדי אימון ניתן לעדכן את כל שכבות הרשת
- אין צורך באחסון זמני של התכונות המוחלצות בזמן האימון

:SSD

SSD היא מטא-ארכיטקטורה [9] המחשבת את הלוקליזציה במעבר רשת יחיד ורציף ובך משיג תוצאות מהירות יותר [10] על פני ארכיטקטורת R-CNN. בדומה ל-Fast-RCNN, אלגוריתם ה-SSD קובע רשת עוגנים על התמונה. בניגוד לשיטות הדו-שלביות, המודל מניב וקטור תחזיות לכל אחת מהקופסאות במעבר רשת עוקב. וקטור זה מחזיק ציון עבור מידת הביטחון שהמודל חזה עבור כל אובייקט ואובייקט.

## YOLO — You Only Look Once

כל האלגוריתמים הקודמים לזיהוי אובייקטים משתמשים באזורים כדי למקם את האובייקט בתוך התמונה. הרשת לא מסתכלת על התמונה המלאה. במקום זאת, מסתכלת על חלקים בתמונה בעלי סבירות גבוהה להכיל את האובייקט. YOLO או [11] You Only Look Once הוא אלגוריתם לגילוי אובייקטים שונה בהרבה מן אלגוריתמים מבוססי אזור שהראינו לעיל. ב-YOLO רשת קונבולוציונית אחת מנבאת את תיבות התחום ואת ההסתברויות עבור תיבות אלה.



איור 3: כל תמונה מחולקת לתיבות בגודל שווה ותיבות התחום משורטטות כאשר כל עובי קו מציין את מידת הביטחון לחיזוי.

YOLO עובד בכך שהוא לוקח תמונה ומפצל אותה לרשת  $S \times S$ , בתוך כל תא ברשת אנו לוקחים תיבות תחום. עבור כל אחת מתיבות התחום, הרשת מפיקה ערכי הסתברות עבור חיזוי לאובייקט ותיבות התחום בעלות ההסתברות לאובייקט מעל ערך סף מסוים, נבחרות ומשמשות לאיתור האובייקט בתוך התמונה.

## חיסרונות בשימוש YOLO

YOLO מטיל מגבלות מרחביות חזקות על חיזוי תיבות תוחמות שכן כל תא מנבא רק שתי תיבות ויכול לקיים אובייקט אחד בלבד. אילוץ מרחבי זה מגביל את מספר האובייקטים הסמוכים שהמודל יכול לחזות. המודל נאבק עם חפצים קטנים המופיעים בקבוצות, כמו להקות ציפורים.

## יתרונות בשימוש YOLO

- YOLO מהיר במיוחד - מסוגל לזהות אובייקטים בקצב של 45 פריימים לשנייה.
- YOLO רואה את התמונה כולה בזמן האימון ובמבחן כך שהיא מקודדת באופן מרוזם מידע קונטקסטואלי על שיעורים כמו גם על המראה שלהם.
- YOLO לומד ייצוגים הכללתיים של אובייקטים כך שכאשר הם מאומנים על תמונות טבעיות ונבדקים על תמונות גרפיקאיות, האלגוריתם עולה על שיטות זיהוי מובילות אחרות.

## סיכום

באמצעות טכנולוגיות ואלגוריתמים אלו נוכל להבטיח את שלומם של האנשים במקומות ציבוריים על ידי ניטור אוטומטי שלהם אם הם שומרים על ריחוק חברתי [12] בטוח לצד איתור מסכת פנים בקרב הציבור. סקירה זו מתארת בקצרה את ארכיטקטורת המודלים לפתרון וכיצד מערכת יכולה באופן אוטומטי למנוע את התפשטות נגיף הקורונה [13] באמצעות יכולות למידה עמוקה בשילוב טכנולוגיות מתקדמות של ראייה ממוחשבת.

## 2.4 ביבליוגרפיה

- [1] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Type Pyramid Networks for Object Detection," IEEE Conference Proceedings on Computer Vision and Pattern Recognition, pp. 2117-2125. 2017.
- [2] Chen, S., Zhang, C., Dong, M., Le, J., R., M., 2017b. Using rankingcnn for age estimates, in: IEEE Conference on Computer Vision and Pattern Recognition
- [3] GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. (2016). Deep learning process. Chapter 6.
- [4] S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face recognition using deep convolutional neural networks. In ACM ICMR, pp 643–650, 2015.
- [5] Masita, K. L., Hasan, A. N., and Satyakama, P., 2018. Pedestrian identification by mean of R-CNN object detector. IEEE Latin American Conference on Computational Intelligence, 2018.
- [6] R. Girshick, "Fast R-CNN," in Proc. International Conference Computer Vision, pp. 1440–1448. 2015
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with area proposal networks," in Proc. Adv. Neural Inf. Process. Syst., pp. 91–99, 2015.
- [8] Liu, H., Chen, Z., Li, Z., and Hu, W. 2018. An powerful method of pedestrian detection based on YOLOv2. Mathematical Engineering Issues, 2018.
- [9] Wei Liu<sup>1</sup>, Dragomir Anguelov<sup>2</sup>, Dumitru Erhan<sup>3</sup>, Christian Szegedy<sup>3</sup>, Scott Reed<sup>4</sup>, Cheng-Yang Fu<sup>1</sup>, Alexander C. Berg<sup>1</sup> "SSD: Single Shot MultiBox Detector", pp. 1-3, 2016.
- [10] Jonathan Huang Vivek Rathod Chen Sun Menglong Zhu Anoop Korattikara, Speed/accuracy trade-offs for modern convolutional object detectors, pp.7314-7315, 2017.

- [11] Joseph Redmon , Santosh Divvala, Ross Girshick, Ali Farhadi You Only Look Once: Unified, Real-Time Object Detection, pp. 1-3, 2016.
- [12] Matrajt L, Leung T. Evaluating the efficacy of social distancing strategies to postpone or flatten the curve of coronavirus disease. Emerg Infect Dis, man. 2020:
- [13] S. Wang Chen, Horby Peter W, Hayden Frederick G, Gao George F. A novel coronavirus epidemic of global concern for health. It's the Lancet, pp. 375. 2020



### 3 חקר דרישות ואפיון ראשוני

#### 3.1 מטרת המערכת

מגפת הקורונה (Covid-19) הציפה קשיים מהותיים בפני מדינות העולם בנושא מעקב וביצוע אכיפה אחר מפירי הנחיות השמירה על הבריאות. קשיים אלו הביאו רבים, וביניהם גם אותנו, לחפש אחר פתרונות יעילים, זמינים ונגישים, שיקלו על ההתמודדות ויאפשרו להאט את קצב ההדבקה.

מערכת OpenCoVid הינה מערכת לזיהוי מפרי הנחיות עטיית מסכה ושמירה על ריחוק חברתי, בזמן אמת.

מטרת המערכת היא לסייע לגורמי אכיפה ופיקוח בניטור, פיקוח ומעקב אחר מפירי הנחיות, בקלות, ללא עלויות גבוהות ותוך חיסכון משמעותי בכוח אדם.

נדגיש כי פרויקט זה מהווה הוכחת היתכנות (POC - Proof Of Concept) בלבד ולא בא לפתור את הבעיה בכלליותה ולעומקה.

#### 3.2 תהליכים עסקיים

##### Use Cases 3.2.1

תיאור Use Cases שהמערכת צריכה לבצע:

מרכיב	הסבר
שם	שינוי הגדרות
תנאי קדם	המערכת מאותחלת ובמסך ההגדרות.
תנאי סיום	מקור הוידאו מחובר למערכת, אותחל אובייקט אבסטרקציה עבור, אלגוריתם ה-CV שונה, ערך הרגישות שונה, אומדן המרחק שונה, יעד קובץ ה-LOG שונה, המסך שונה למסך הראשי.
תרחיש הצלחה ראשי	<ol style="list-style-type: none"> <li>1. המשתמש בוחר את מקור הוידאו לניתוח.</li> <li>2. המשתמש בוחר את רמת הרגישות של הניתוח.</li> <li>3. המשתמש בוחר את אומדן המרחק.</li> <li>4. המשתמש בוחר את יעד כתיבת קובץ ה-LOG.</li> <li>5. המשתמש מאשר את ההגדרות.</li> <li>6. המערכת מוודאה שההגדרות חוקיות</li> <li>7. המערכת יוצרת אובייקט אבסטרקציה עבור מקור הוידאו, קובעת את רמת הרגישות, אומדן המרחק והאלגוריתם לפי בחירת המשתמש ומשנה את יעד הכתיבה.</li> <li>8. המערכת עוברת למסך הראשי.</li> </ol>
תרחישים אלטרנטיביים	<p>צעד 6 - לא ניתן להתחבר למקור הוידאו/יעד לכתיבת LOG לא קיים.</p> <p>a. המערכת מודיעה על התקלה למשתמש.</p> <p>b. חזרה לצעד 1.</p>



מרכיב	הסבר
שם	התחלת ניתוח הוידאו
תנאי קדם	מקור וידאו מחובר, ניתוח לא התחיל, קיימים פריימים לניתוח ונמצאים במסך הראשי.
תנאי סיום	תהליך ניתוח הפריימים נוצר והתחיל, תוצאות הניתוח מופיעות במסך עם כל פריים שנותח.
תרחיש הצלחה ראשי	<ol style="list-style-type: none"> <li>המשתמש לוחץ על כפתור התחלת הניתוח.</li> <li>המערכת יוצרת תהליך חדש של ניתוח זרם של פריימים.</li> <li>זרם הפריימים שנותחו מופיעים על המסך (מתעדכן) עם ויזואליזציה של התוצאות שלהם.</li> </ol>
תרחישים אלטרנטיביים	<p>צעד 3 - זרם הפריימים נגמר</p> <ul style="list-style-type: none"> <li>תהליך הניתוח מופסק.</li> <li>המערכת מודיעה למשתמש על סיום הניתוח.</li> </ul> <p>צעד 3 - זרם הפריימים מתנתק</p> <ol style="list-style-type: none"> <li>המערכת מודיעה למשתמש על שגיאה.</li> <li>המערכת מנסה להתחבר מחדש.</li> <li>אם לא מצליחה או עובר פרק זמן מוגדר <ol style="list-style-type: none"> <li>תהליך הניתוח מופסק.</li> <li>המערכת מודיעה למשתמש על סיום הניתוח.</li> </ol> </li> </ol>

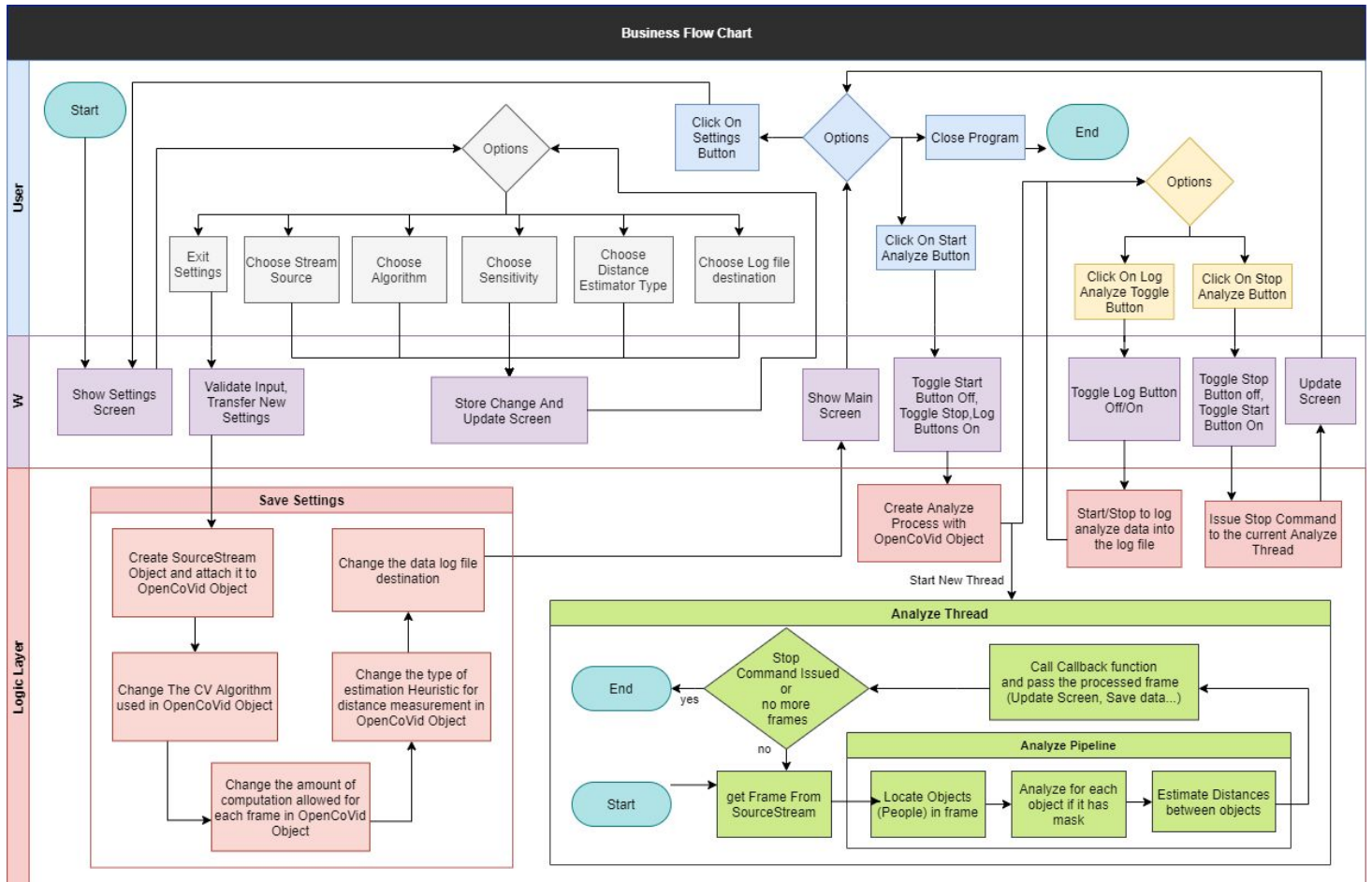
מרכיב	הסבר
שם	הפסקת ניתוח הוידאו
תנאי קדם	קיים תהליך ניתוח פעיל, נמצאים במסך הראשי.
תנאי סיום	תהליך ניתוח הפריימים הופסק ונמחק, תצוגת הפריימים הופסקה.
תרחיש הצלחה ראשי	<ol style="list-style-type: none"> <li>המשתמש לוחץ על כפתור הפסקת הניתוח.</li> <li>המערכת מודיעה לתהליך הניתוח להפסיק.</li> <li>תהליך הניתוח מופסק ונמחק.</li> <li>המערכת מעדכנת את המסך.</li> </ol>
תרחישים אלטרנטיביים	

מרכיב	הסבר
שם	התחלה/הפסקת הקלטת תוצאות
תנאי קדם	קיים תהליך ניתוח פעיל, נמצאים במסך הראשי, סופק נתיב לתקייה קיימת.
תנאי סיום	המערכת תכתוב/תפסיק לכתוב את תוצאות סיכום הניתוח אל קובץ LOG
תרחיש הצלחה ראשי	<ol style="list-style-type: none"> <li>1. המשתמש לוחץ על כפתור ההקלטה/הפסקה.</li> <li>2. המערכת מעדכנת את דגל הכתיבה.</li> <li>3. המערכת מחליפה את כפתור ההקלטה/הפסקה בשני.</li> <li>4. אם הדגל כתיבה הורם, עבור כל פריים שנוחח <ol style="list-style-type: none"> <li>a. המערכת מסכמת את תוצאות הניתוח.</li> <li>b. המערכת כותבת את הסיכום אל הקובץ LOG בנתיב שסופק.</li> </ol> </li> </ol>
תרחישים אלטרנטיביים	<p>צעד 3b – לא ניתן לכתוב לקובץ</p> <ol style="list-style-type: none"> <li>a. המערכת מעדכנת שדגל הכתיבה יורד.</li> <li>b. המערכת מודיעה על שגיאה ומעדכנת את הכפתור.</li> </ol>

## Business flow chart 3.2.2

תיאור זרימת הפעולות אשר ימומשו במסגרת הפרויקט בחלוקה לפעולות משתמש, השכבה

הלוגית, שכבת התצוגה והקשר ביניהם:



### 3.3 דרישות מערכת

דרישה היא בעצם תכונה או יכולת של המוצר, הנדרשת למשתמש לצורך פתרון בעיה או השגת מטרה. היא גם יכולה להיות תכונה או יכולת של המוצר הנדרשת לצורך עמידה בחוזה, בתקן או במסמך מחייב אחר.

#### 3.3.1 דרישות פונקציונאליות

דרישות בחלק זה מגדירות את תכולת הפתרון – והן מקבלות מענה ספציפי ישיר בתוך הפתרון המימוש ובעצם מתארות מה המערכת צריכה לעשות ובאיזה מידע עליה לטפל. דרישות פונקציונאליות לרוב מחולקות בכלליות לשתי קטגוריות: דרישות תפעוליות ודרישות מידע.

כללי:

1. אתחול המערכת הכולל את טעינת המודל ותלויותיו יקרה באופן אוטומטי עם טעינת המערכת
2. המערכת תאפשר ניתוח תמונות וידאו על פי פרמטרים מוגדרים מראש
  - a. זיהוי וספירת בני אדם
  - b. זיהוי מסכת פה ואף עבור אובייקטים
  - c. זיהוי ריחוק חברתי בין אובייקטים
3. המערכת תאפשר חיבור למצלמה חיה/ וידאו
4. המערכת תוכל להחליף את האלגוריתם שהוגדר לניתוח (Faster R-CNN or YOLO)
5. המשתמש יוכל לבחור רמת רגישות זיהוי אובייקטים בניתוח ה - frame
6. המשתמש יוכל להגדיר יעד לכתיבת קובץ ה - log של התוצאות
7. המשתמש יוכל להגדיר את המרחק החברתי שברצונו לנטר
8. המשתמש יוכל להגדיר את האומדן לחישוב המרחק בו המערכת תשתמש (מבנה אנטומי/גובה ממוצע)
9. המערכת תספק ממשק משתמש לניטור וידאו חי בזמן אמת
10. המערכת תתחיל/תפסיק ניתוח לפי דרישה
11. המערכת תאפשר לייצא את התוצאות לקובץ חיצוני
12. המערכת תאפשר שליטה דרך מערכת CLI
13. המערכת תספק אינדיקציה ברורה אודות חריגות מהנחיות, בצורה פשוטה ונוחה

### 3.3.2 דרישות לא פונקציונאליות

דרישות בחלק זה מגדירות תכונות נוספות של הפתרון שצריכות להתמלא תוך כדי מילוי הדרישות הפונקציונליות ומחולקות לדרישות ביצועים ולמאפייני איכות, דרישות אלה גם מאופיינות באילוצים במגוון תחומים כגון חומרה מימוש וניהול.

1. המערכת תעבד וידאו בקצב של מינימום **4 frames לשנייה**
2. במקרה של קריסת המערכת, המערכת תתאושש תוך 30 שניות לכל היותר
3. המערכת תוכל לקבל נתוני וידאו מכל מצלמה שאיכות הצילום בה לא יורדת מ - 416X416 resolution.
4. המערכת תפותח בשפת Python.
5. המערכת תאפשר זיהוי חריגות ברמת ודאות של 60% לכל הפחות.
6. המערכת תתמוך בניתוח וידאו לפחות ב-2 הפורמטים הבאים: .avi, .mp4.
7. ניסיון חיבור נוסף למצלמת רשת עד timeout יהיה לכל היותר 10 שניות.

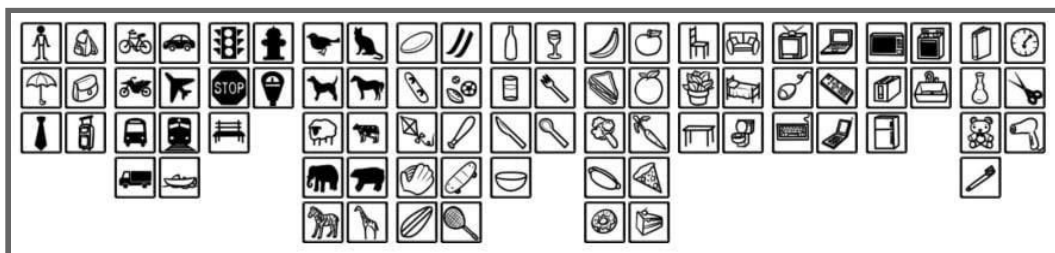
## 3.4 מתודולוגיה

### הקדמה

תקופה לא קלה זו בה אנו נמצאים גרמה לנו לחשוב על פיתח מודל AI שיוכל לזהות האם אנשים מקפידים על הנחיות הקורונה הכוללות ריחוק חברתי ובאותה עת חבישת מסכה. בעזרת למידה עמוקה ו- OpenCV נוכל להפיק תובנות מעניינות מתמונות ווידאו. תיבות תחום אדומות מעידות על כך שהאדם נמצא בקרבת אדם אחר ותיבה ירוקה מציינת כי האדם שומר על מרחק תקין. בנוסף נשתמש בטכניקה נוספת לזיהוי אם האדם חובש מסכה או לא ונאכוף גם זאת.

### סקירת השלבים

ממשק ה-API לזיהוי אובייקטים TensorFlow הוא הבסיס ליצירת רשת למידה עמוקה הפותרת בעיות זיהוי אובייקטים. ה-API מספק מודלים לזיהוי אובייקטים שאומנו מראש כ- Model Zoo, מודלים אלו אומנו על בסיס הנתונים של [COCO](https://arxiv.org/pdf/1405.0312.pdf). במאגר הנתונים קיימים שלל אובייקטים נפוצים (COCO - Common Objects In Context) וישנם 200,000 תמונות עם יותר מ- 500,000 תיוגים עבור אובייקטים ב- 90 מחלקות שונות. ראה איור 1 מציג חלק מהאובייקטים שנמצאים ב-COCO.



איור 1 : קטגוריות אובייקט ב-COCO. התמונה מ- <https://arxiv.org/pdf/1405.0312.pdf>

במקרה זה, אנו מתמקדים במחלקה 'אדם' בלבד המהווה חלק ממערך הנתונים של COCO. ל-API של TensorFlow יש גם סט גדול של מודלים שהוא תומך בהם. עיין בטבלה למטה לצורך העניין.

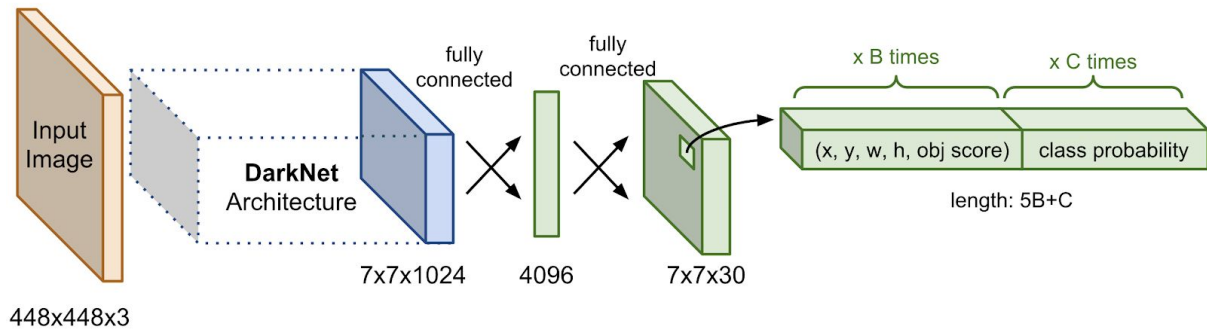
#### COCO-trained models

Model name	Speed (ms)	COCO mAP <sup>[1]</sup>	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_fpn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rcnn_resnet101_coco	92	30	Boxes

המודלים נבדלים ביניהם על היחס בין מהירות לדיוק. אנו בחרנו `faster_rcnn_resnet50_coco` שבאמצעותו נוכל לזהות 'אדם'. ברגע שזיהינו את 'האדם' באמצעות ממשק ה-API לזיהוי אובייקטים, כדי לחזות אם האדם שומר על ריחוק חברתי נוכל להשתמש ב- OpenCV שהיא ספרייה עוצמתית לעיבוד תמונות יחד עם אלגוריתם שנפתח. ברגע שזוהה ריחוק חברתי, נשתמש ב-API, כדי לזהות אם האדם חובש מסכה או לא.

איור 2: תת-קבוצה קטנה של מודלים הנתמכים על ידי tensorflow

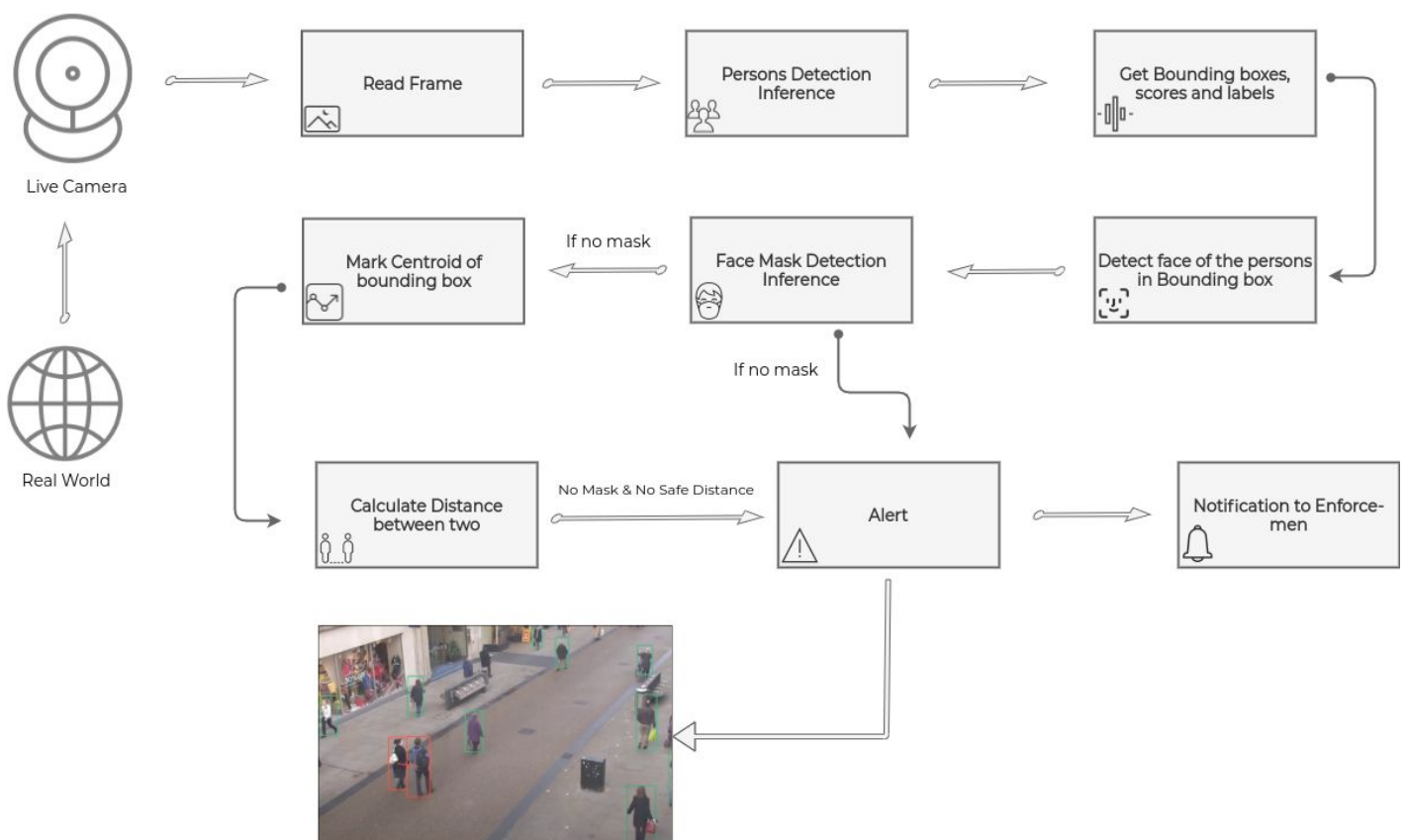
בנוסף עבור חקר ביצועים אנו משתמשים במערכת המוצעת גם ב-YOLO שנראה כי זו בחירה טובה לזיהוי אובייקטים בזמן אמת והאיזון בין דיוק לביצועים הוא טוב. YOLO משתמש במודל DarkNet בחלק התחתון ובחלק העליון שלו, מוסיף כמה שכבות FC על מנת לבצע חישובים פנימיים ולהחזיר את ערכי החיזוי שהמודל צפה.



איור 3: ארכיטקטורה של מודל YOLO.

## נצלול אל הצעדים העיקריים

המטרת המרכזית שלנו בשימוש במודלים היא לאתר פרצופים של אנשים בתמונה על מנת לאבחן אם הם עם מסכה או ללא מסכה. נשתמש בגישה של fine tuning לאופטימיזציה של ביצועים על אלגוריתם למידה עמוקה וראייה ממוחשבת והמודל הנבחר יעשה את עבודת חילוץ התכונות עבור משימת איתור המסכה.



איור 4: ארכיטקטורת הפתרון של המערכת המוצעת

השלבים העיקריים אותם אנחנו נבצע על פי ארכיטקטורת הפתרון (איור 4):

- טען את מודל yolov4/ faster\_rcnn\_resnet50 המאומן על COCO יחד עם המשקולות לגרף.
- טען וידאו או פתח מצלמת רשת וקרא כל פריים אחד אחרי השני וכתוב אותו לנתיב.
- עבור כל פריים בצע איתור אובייקטים באמצעות המודל הטעון.
- התוצאה שחוזרת מהמודל היא כל מחלקה מזוהה יחד עם ציון הביטחון שלה וחיזוי תיבת התחום.
- עבור ציון ביטחון  $> 0.60$ , גלה את מספר האנשים בפריים.
- צייר תיבת תחום על סמך חיזוי התיבה התוחמת שקיבלנו מעלה ומצא את נקודת הסינטרואיד.
- חשב מרחק בין נקודות הסינטרואיד בפריים על-ידי האלגוריתם שלנו.
- לאחר חישוב המרחק, מצא את תיבות התחום שהמרחק שלהן נמוך מ-2 מטרים וצבע את התיבה לאדום.
- שמור את כל התיבות שזוהה בהן ריחוק חברתי.
- עבור כל פריים שזוהה בו ריחוק חברתי בצע איתור אובייקטים של אנשים שחובשים מסכה.
- לבסוף מזג את כל הפריימים שהושגו והצג אותם למשתמש.

במערכת המוצעת מתבצעים ארבעה שלבי על, והם:

- (1) איסוף נתונים ועיבוד מקדים
- (2) פיתוח ואימון מודל
- (3) בדיקת המודל
- (4) יישום המודל

על כל אחד משלבים אלה ועל תוצאותיהם נרחיב במסמכים ודו"חות עתידיים היות וטרם בוצעו.

ריחוק חברתי, חבישת מסכה יחד עם אמצעי הגנה בסיסיים אחרים חשובים מאוד בכדי להאט את התפשטות הנגיף ככל הניתן. פרויקט זה מהווה הוכחת היתכנות (POC - Proof Of Concept) בלבד ולא בא לפתור את הבעיה בכלליותה ולעומקה.

### 3.5 מאגר נתונים

לצורך אימון המודל, אנו נדרשים לכמות תמונות רבה הכוללת מגוון רחב של סיטואציות: אנשים בודדים ללא/עם מסכה, קבוצות אנשים שחלקם עוטים מסכה וחלקם לא, תמונות שצולמו בתנאי ראות קשים (חושך, גשם, שמש חזקה), קבוצות אנשים שביניהם מרחק/קרבה וכו'.

התמונות שתוארו לעיל צולמו על ידינו או נאספו ממקורות שונים באינטרנט.

סך התמונות שישמשו לצורך אימון המודל ובדיקתו: כ-800 תמונות המייצגות את המגוון הרחב שתואר לעיל.

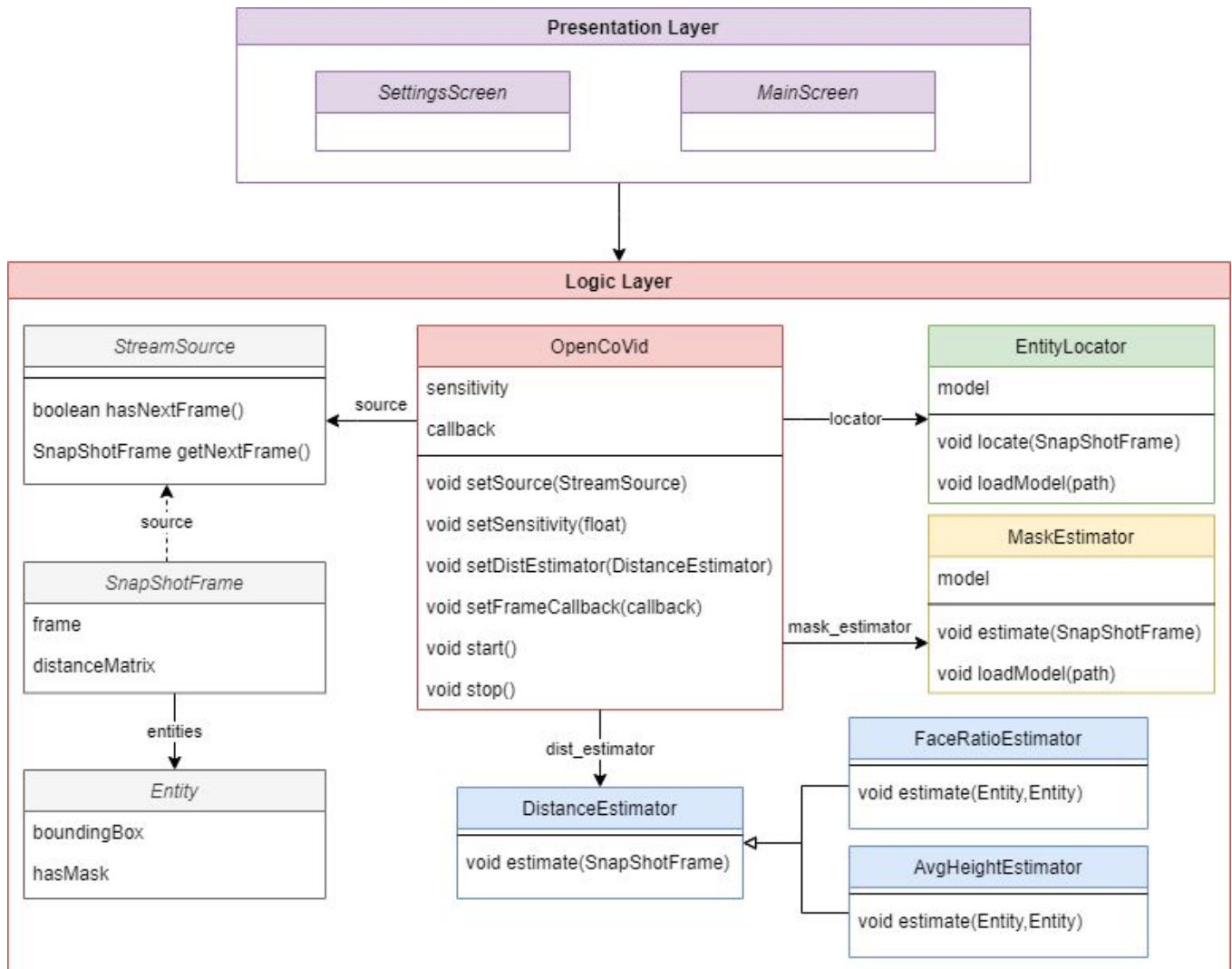
בשלב הרצת המערכת, נקבל וידאו ב-real-time ממצלמות המחוברות למערכת ועל בסיס הנתונים שהוסרטו נבצע ניתוח בזמן אמת.



## 3.6 תכנון הממשק

### 3.6.1 ממשק לוגי – Class Diagram

תיאור המחלקות והשיטות המרכזיות אשר ימומשו והקשרים ביניהם



**Entity** – תיאור של יישות בפריים, מתואר ע"י bounding box של מיקום בפריים וערכים המעודכנים בניתוח.

**SnapshotFrame** – מבנה נתונים המתאר פריים ובו יועדן כל תוצאות הניתוח שלו (רשימת ישויות שזוהו, מטריצת מרחקים בין כל הישויות).

**StreamSource** – אבסטרקציה של מקור זרם פריימים (וידאו), מאפשר ממשק אחיד מול הרכיבים.

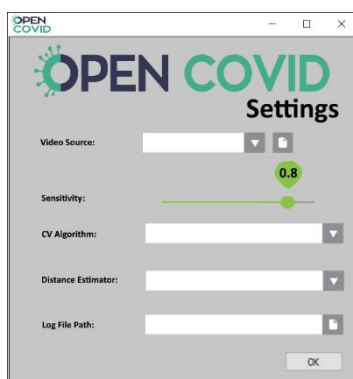
- `boolean hasNextFrame()` – מחזיר TRUE אם קיימים פריימים נוספים בזרם הנתונים.
- `SnapshotFrame getNextFrame()` – מחזיר SnapshotFrame המייצג את הפריים הבא בזרם.

**EntityLocator** – מחלקה המספקת ממשק לאיתור ישויות (בני אדם) בפריים בעזרת מודל CV.

- `(void locate(SnapShotFrame` – מאתר בני אדם בפריים נתון, יוצר מהם ישויות ומעדכן את הרשימה בפריים.
- `(void loadModel(Path` – טעינת מודל ה-CV לאובייקט.
- **MaskEstimator** – מחלקה המספקת ממשק לזיהוי לבישת מסכה עבור ישויות בפריים בעזרת מודל CV.
- `(void estimate(SnapShotFrame` – מעריך עבור הישויות בפריים נתון האם הם לובשים מסכה או לא ומעדכן את המידע בהם.
- `(void loadModel(Path` – טעינת מודל ה-CV לאובייקט.
- **DistanceEstimator** – מחלקה אבסטרקטית המגדירה אומדן למדידת מרחקים בין כל שני ישויות בפריים.
- `(void estimate(SnapShotFrame` – מעריך עבור כל זוג ישויות בפריים נתון את המרחק ביניהם, יוצר מטריצת מרחקים ומעדכן את הפריים.
- **AvgHeightEstimator** – מחלקה יורשת המממשת אומדן מרחק בעזרת גובה ממוצע
- `estimate(Entity, Entity) void` – שיטה המקבלת שני ישויות ומעריכה את המרחק האמיתי ביניהם
- **FaceRatioEstimator** – מחלקה יורשת המממשת אומדן מרחק בעזרת מבנה אנטומי של הפנים
- `(void estimate(Entity, Entity` – שיטה המקבלת שני ישויות ומעריכה את המרחק האמיתי ביניהם
- **OpenCoVid** – אובייקט העוטף את כל הרכיבים ומנהל אותם, מאפשר גישה נוחה ומסודרת ליכולות הספרייה.
- `(void setSource(StreamSource` – מאפשר הזרקה של מקור זרם פריימים אשר יעברו ניתוח.
- `(void setSensitivity(float` – הגדרת ערך בטווח [0,1] המאפשר אופטימיזציה של עומק הניתוח מול המשאבים שמושקעים בכל ניתוח של פריים.
- `(void setDistEstimator(DistanceEstimator` – מאפשר הזרקה של אמד מרחק אמיתי בפריים שיופעל במהלך pipeline של ניתוח הפריים.
- `(void setFrameCallback(callback` – מאפשר הגדרה של פונקציית callback המקבלת SnapShotFrame אשר תהליך הניתוח יפעיל אותה עבור כל פריים שסיים לעבור ניתוח ובכך יעביר את המידע המנותח אליו.
- `(void start` – התחלת ניתוח מקור זרם הנתונים שהוגדר עד לסיום הזרם או הפסקה יזומה של הניתוח, יוצר הליך חדש שבו ינותחו הפריימים.
- `(void stop` – הפסקת תהליך ניתוח פעיל אם קיים כזה.

### 3.6.2 ממשק התצוגה – תכנון מסכים

#### מסך הגדרות:

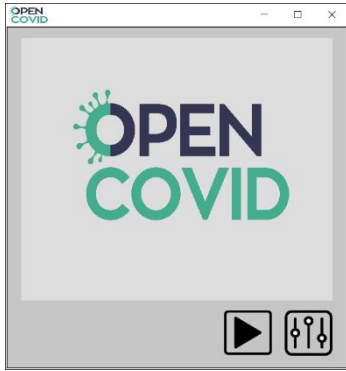


מסך המאפשר להגדיר את הגדרות המערכת, בנוסף המסך הראשוני שיופיע כאשר התוכנית תעלה.

מאפשר להגדיר מקור זרם נתונים (וידאו/מצלמה), ערך של רגישות הניתוח, סוג האלגוריתם CV שיעמוד בבסיס ועוד.

במידה ויהיו שגיאות בבחירה בעת לחיצה על אישור יופיע הודעת שגיאה, במידה ולא יהיו שגיאות והמסך יתעדכן להיות המסך הראשי ללא הפעלת ניתוח.

## מסך ראשי:



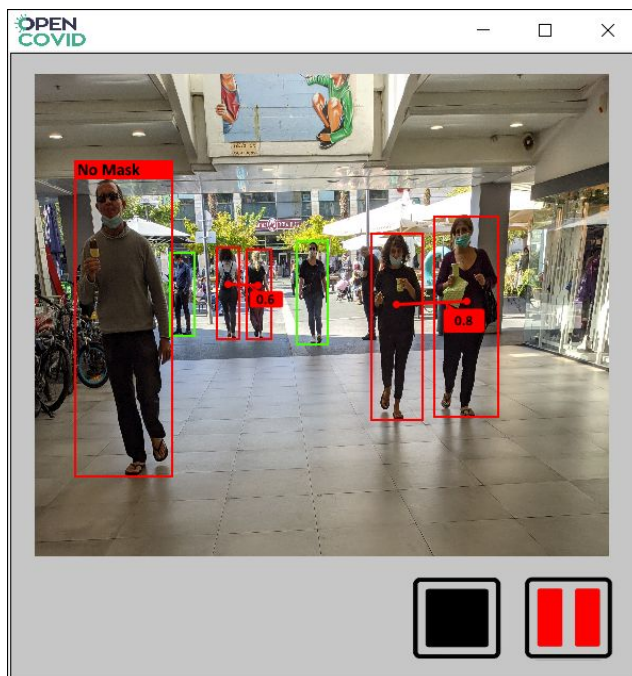
### • מצב 1: ללא הפעלת הניתוח

במצב זה ניתן ללחוץ על שני כפתורים, הכפתור הימני יחזיר את המשתמש למסך ההגדרות, הכפתור השמאלי יתחיל את ניתוח הזרם ויעבור למצב 2.

### • מצב 2: ניתוח מופעל

במצב זה ניתן להפסיק את הקלטת התוצאות (הכפתור הימני) או להפסיק את זרם הפריימים וניתוחם (הכפתור השמאלי).

בנוסף ניתן לראות את זרם הפריימים שנותרו וויזואליזציה של התוצאות:



o ישויות המוקפות ב bounding Box ירוק, עומדים בהנחיות.

o ישויות המוקפות ב bounding Box אדום, לא עומדים בהנחיות.

• אם לא לובשים מסכה יופיע מעל ה bounding Box אינדיקציה לכך.

• אם לא שומרים מרחק חברתי יצויר קו בין האובייקטים עם ערך המרחק.

## 3.7 הערכת מצב

### 3.7.1 אתגרים

בפרויקט שלפנינו קיימים מספר אתגרים במגוון תחומי אחריות, אנו נציג חלק מהם.

- יצירת מאגר נתונים
  - כמות - מודל למידה עמוקה מצריך כמות גדולה של דאטה לאימון
  - איכות - תמונות בתחום ההגדרה המתאים
  - תיוג - עבור על תמונה שנאספה לבצע תיוג בהתאם לסיווג
- איזון ביצועים (דיוק מול זמן עיבוד FPS)
  - מודלים שונים נבדלים בביצועים שונים, בדרך כלל דיוק גבוה משמעותו זמן עיבוד ארוך
- דיוק בתנאים מאתגרים
  - חשכה
  - רזולוציה נמוכה
  - ריבוי אובייקטים
- עיבוד בזמן אמת
- טכנולוגיה ותחום עם היכרות בסיסית
- מדידת מרחק - קיים קושי בהערכת מרחק בפריים שהרי המודל קבל את הפריים בפיקסלים והמעבר ליחידות מידה מפיקסלים הוא מאתגר כי הרי אין נקודת ייחוס בפריים. הנה שני דרכים שהצענו להתמודדות:
  - דרך 1 - דיוק בהערכת גובה (ממוצע)
  - דרך 2 - זיהוי הפנים (זוויות משתנות) זיהוי פרופורציות (נק מנחות לחישוב יחסים)

### 3.7.2 סיכונים

סיכון פירושו כל גורם העלול למנוע מהפרויקט להגיע לסיומו המוצלח ולעמוד במטרותיו.

קיימים גורמים רבים המהווים סיכון להצלחת הפרויקט:

- חוסר ידע (פערי ידע טכנולוגים, מקצועיים, מסחריים...)
  - חוסר זמן
  - חוסר בכ"א
  - שינויים בתכולת הפרויקט תוך כדי עבודה
- גורמים אלה ישפיעו באופן ישיר על הצלחתו של הפרויקט
- אי מימוש התכולה
  - אי עמידה בלוח הזמנים
  - חוסר שביעות רצון לקוח
  - אי עמידה במטרות נוספות שהוגדרו לפרויקט

## סבירות דירוג ועוצמות:

ערך	חומרה	הסתברות	רמה
1	זניח	מינימום	ארגוני
2	מקומי	יכול לקרות	תכן הנדסי
3	פגיעה קלה	סביר שיקרה	תכולה, טכנולוגי
4	פגיעה גבוהה	סבירות גבוהה	ניהול הפרויקט
5	חוסר תפקוד	כמעט ודאי	צד שלישי

#	סיכון	דירוג			התמודדות	גורם אחראי
		חומרה	הסתברות	רמה		
1	אי עמידה בלוח הזמנים	3	4	4	יצירת אבני דרך קטנות וריאליות	PM
2	חוסר שביעות רצון הלקוח	4	3	5	פגישות לעיתים תקופות עם הלקוח ותיאום ציפיות	PM
3	איכות ירודה של מערך הנתונים	4	3	2	הכנה של מערך נתונים לגיבוי ממקורות אחרים	DEVS
4	אי מימוש התכולה	4	1	3	אם מפאת חוסר ידע יש לקרוא חומר פרקטי בנושא והתייעצות נרחבת עם מנחה הפרויקט	DEVS
5	חוויה משתמש רעה	3	2	2	התייעצות עם גורמים חיצוניים ואפשרות למשוב ממשתמשים פוטנציאליים	DEVS

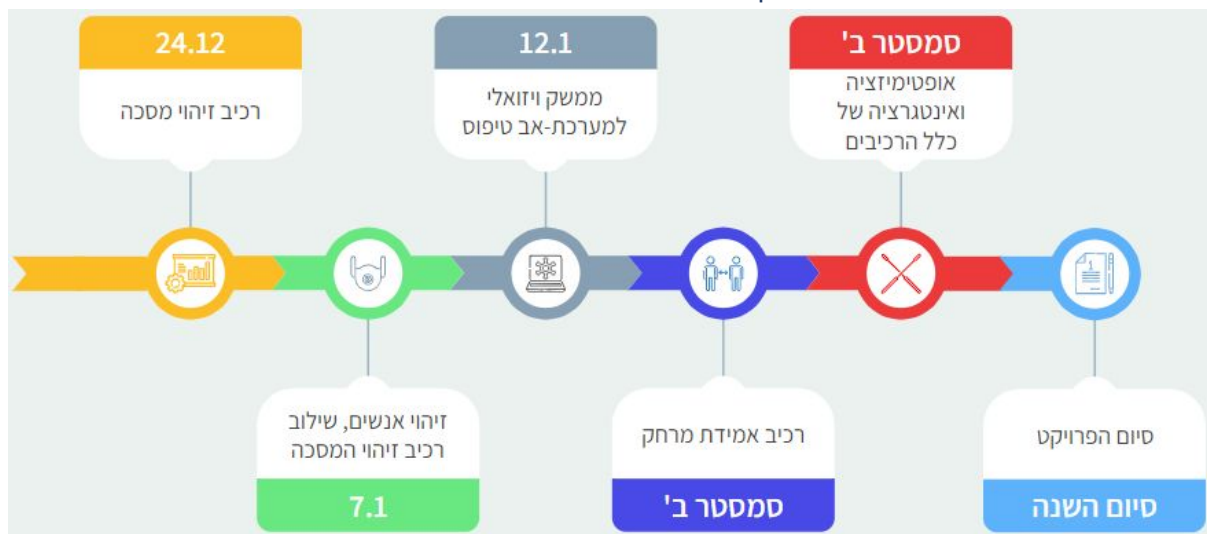
### 3.8 תוצרים

הפרויקט יניב 4 תוצרים עיקריים:

1. מודל Computer-Vision לפיקוח על ההנחיות - מודל למידת מכונה, שמטרתו לזהות ב- real-time בני אדם ופרצופים, עטיית מסכת פה ואף.
2. אלגוריתם לאמידת מרחק בין בני אדם - חישוב ריחוק חברתי בין אדם לאדם במרחב הציבורי.
3. מערכת שו"ב - ממשק משתמש (GUI) לטובת הצגה ויזואלית של תוצאות הניטור.
4. מסמך חקר ביצועים - השוואה בין מודלים state-of-the-art, בה נבחן ביצועים מול דיוק (שימוש במודלים YOLO, Faster R-CNN).

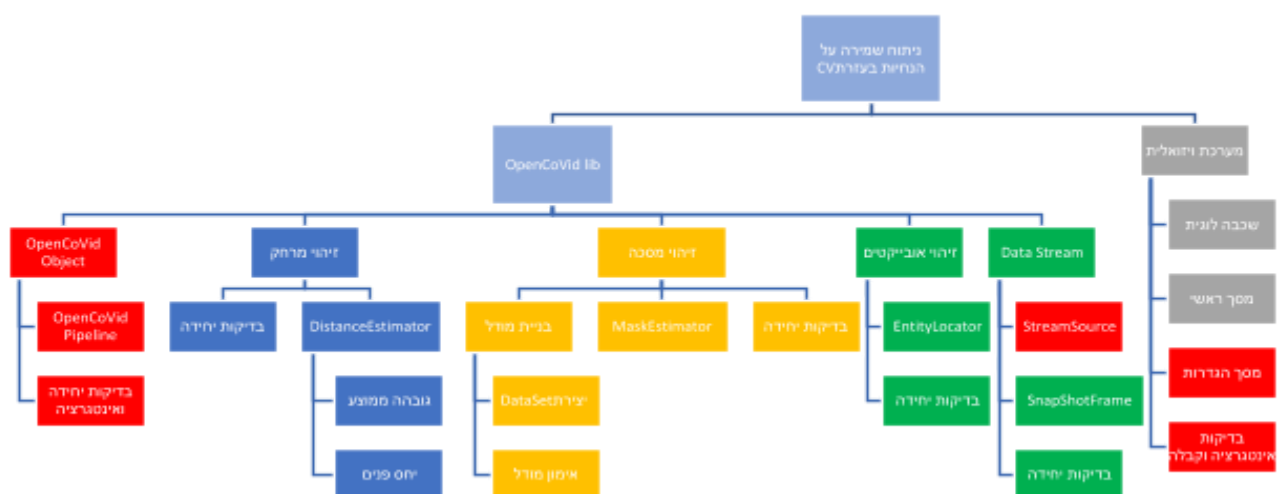
### 3.9 WBS, לוח זמנים וחלוקת אחריות

3.8.1 אבני דרך



תיאור אבני הדרך של הפרויקט, תאריכים משוערים עבורם והסדר הכרונולוגי שלהם

### 3.8.2 WBS, חלוקת אחריות ותיאור לו"ז מפורט



תיאור גרפי מפורט של אבני הדרך (חלוקה לצבעים) והרחבת התוכן שלהם כרכיבי פרויקט (WBS)

להלן פירוט של רכיבי המערכת, זמני הביצוע המשוערים ואחריות חברי הצוות בפרויקט

מרכיב WBS	תיאור	אחראי	תאריך ההתחלה	תאריך סיום
ניתוח שמירה על הנחיות בעזרת CV	הפרויקט המפותח			
1. OpenCoVid lib	ספרייה המכילה את כל מרכיבי הניתוח			
1.1 OpenCoVid Object	רכיב המאחד את יכולות הספרייה ומספק ממשק נוח לשימוש בהם	אביחי	TBA	TBA
1.1.1 OpenCoVid Pipeline	מימוש pipeline לניתוח פריים ושילוב הרכיבים, בנוסף מאפשר התאמה ושינוי פרמטרים לתהליך	אביחי	TBA	TBA
1.1.2 בדיקות	בדיקות יחידה ואינטגרציה לרכיבים	אביחי	TBA	TBA
1.2 Data Stream	רכיב המגדיר זרם של פריימים (וידאו)	דביר	TBA	TBA
1.2.1 StreamSource	Adapter למקורות וידאו המאפשר ממשק אחיד לקבלת פריימים	דביר	TBA	TBA
1.2.2 SnapShotFrame	אבסטרקציה של פריימים והמידע עליו (התוצאות המנותחות)	דביר	TBA	TBA
1.2.3 בדיקות	ביצוע בדיקות יחידה	דביר	TBA	TBA
1.3 זיהוי אובייקטים	רכיב המאפשר זיהוי בני אדם בפריימים	דביר	24/12	7/1
1.3.1 EntityLocator	ניתוח פריימים ויצירת bounding box סביב האובייקטים הנמצאים בו	דביר	24/12	5/1
1.3.2 בדיקות	ביצוע בדיקות יחידה	דביר	5/1	7/1
1.4 זיהוי מסכה	רכיב המאפשר זיהוי לבישת מסיכה בפריימים	אביחי	01/11	24/12
1.4.1 MaskEstimator	טעינת המודל וסיפוק הממשק לסימון אובייקטים חריגים בפריימים	אביחי	23/12	24/12
1.4.2 בניית מודל	מודל למידת מכונה לזיהוי מסיכה	אביחי	01/11	22/12
1.4.2.1 יצירת DataSet	איסוף, תיוג ויצירת datasets של תמונות	אביחי	01/11	17/12
1.4.2.2 אימון מודל CV לזיהוי מסיכה	אימון המודל והערכה שלו, אופטימיזציה	אביחי	17/12	23/12

24/12	22/12	אביחי	ביצוע בדיקות יחידה	1.4.3. בדיקות
TBA	TBA	אסף	רכיב המאפשר האמידה של מרחק בין שני אובייקטים בפריים	1.5. זיהוי מרחק
TBA	TBA	אסף	רכיב המאפשר אמידה של מרחק בין אובייקטים בפריים בעזרת היוריסטיקה וסימון חריגים	1.5.1. DistanceEstimator
TBA	TBA	אסף	מימוש היוריסטיקה לאמידה של מרחק (יחס פיקסלים עבור מרחק בעולם) בעזרת גובה ממוצע באוכלוסיה	1.5.1.1. אמידה - גובה ממוצע
TBA	TBA	אסף	מימוש היוריסטיקה לאמידה של מרחק (יחס פיקסלים עבור מרחק בעולם) בעזרת יחס נק' נבחרות בפנים	1.5.1.2. אמידה - יחס פנים
TBA	TBA	אסף	ביצוע בדיקות יחידה	1.5.2. בדיקות
12/1	7/1	אסף	רכיב המאפשר ויזואליזציה של ניתוח הפריים	2. מערכת ויזואלית
10/1	7/1	אסף	שילוב ספריית OpenCoVid ותרגום הפלט שלה לתצוגה ויזואלית	2.1. שכבה הלוגית
10/1	7/1	אסף	מימוש עיצוב המסך והממשק בין המשתמש למערכת	2.2. מסך ראשי
TBA	TBA	אסף	מימוש עיצוב המסך והממשק בין המשתמש למערכת	2.3. מסך הגדרות
12/1	11/1	אסף	ביצוע בדיקה לשיטות הממשק הגרפי ושילוב השכבה הלוגית	2.4. בדיקות אינטגרציה וקבלה

## 4 חלוקת אחריות

אחת ממטרות העל האישיות שלנו בפרויקט, היא שלכל אחד מחברי הצוות תהיה נגיעה בכל אחת מנק' הפרויקט - הכנת תשתית הפרויקט (מסמך הפרויקט), איסוף ותיוג נתונים, רכישת ידע מקצועי, פיתוח, בדיקות וכיו"ב.

לאור האמור לעיל בחרנו לבצע **חלוקה רוחבית** של משימות הפרויקט, כך שכל אחד מחברי הצוות מקבל נגיעה משמעותית בכל שלב - במסמך הפרויקט, באיסוף ותיוג הנתונים, בקריאה ולמידת המודלים, בפיתוח ולמעשה בכל שלב בביצוע הפרויקט.

כל אחד מחברי הצוות שותף שווה בכל משימת-על, המפורקת על ידינו לתתי-משימות, וחלוקת האחריות מתבצעת כך שכל חבר צוות עובד על תת-משימה באופן עצמאי, כשבסוף כל החלקים שבוצעו בנפרד מאוחדים לכדי משימת-העל הסופית, בפגישות חתך המבוצעות על ידינו באופן תדיר (אחת לשבוע).

במידה ואחד מחברי הצוות הספיק לבצע את משימותיו, באפשרותו לשלוח תת-משימה שטרם בוצעה מתוך מאגר תתי המשימות שקיים לכל אבן-דרך בתהליך.

החל משלב הפיתוח נשתמש במערכת "Trello" לניהול משימות וחלוקתן בין חברי הצוות.



## 5 הפניות

- IEEE Recommended Practice for Software Requirements Specifications. 1994.
- IEEE Standard for Information Technology — Systems Design — Software Design Descriptions. 2009
- Donn Le Vie, Jr. "Writing Software Requirements Specifications (SRS)". 2010.
- Femmer, Henning; Méndez Fernández, Daniel; Wagner, Stefan; Eder, Sebastian (2017). "Rapid quality assurance with Requirements Smells". Journal of Systems and Software.
- *IEEE SRS - Template - Github*

## 6 נספחים

נספח א: מצגת פתיחה - [OpenCoVid](#)