



**זיהוי ועקיבה אחר תנועות חשודות בווידיאו**

# **Identifying and Tracking Suspicious Movements on Video**

**פרויקט הנדסי**

**דו"ח מסכם פרויקט גמר**

**הוכן לשם השלמת הדרישות לקבלת**

**תואר ראשון בהנדסה B.Sc**

**מאת**

**אמיר בן אלישע**

**ת.ז. : 302719760**

**בהנחיית**

**ד"ר תום טריגנו**

**הוגש למחלקה להנדסת חשמל ואלקטרוניקה**

**המכללה האקדמית להנדסה סמי שמעון**

**אשדוד**

**יוני 2016**

**סיון תשע"ו**



## **זיהוי ועקיבה אחר תנועות חשודות בווידיאו**

# **Identifying and Tracking Suspicious Movements on Video**

**פרויקט הנדסי**

**דו"ח מסכם פרויקט גמר**

**הוכן לשם השלמת הדרישות לקבלת**

**תואר ראשון בהנדסה B.Sc**

**מאת**

**אמיר בן אלישע**

**ת.ז. : 302719760**

**בהנחיית**

**ד"ר תום טריגנו**

**הוגש למחלקה להנדסת חשמל ואלקטרוניקה**

**המכללה האקדמית להנדסה סמי שמעון**

**אשדוד**

\_\_\_\_\_ **תאריך :**

\_\_\_\_\_ **חתימת הסטודנט :**

\_\_\_\_\_ **תאריך :**

\_\_\_\_\_ **חתימת המנחה :**

\_\_\_\_\_ **תאריך :**

\_\_\_\_\_ **אישור ועדת הפרויקטים :**

## תקציר

כיום, הצרכים והשימושים במצלמות אבטחה במקומות הומי אדם ומסחר וכן, הצבת תצפיות בגבולות המדינה שכיחים וחשובים לאין שעור. באמצעות צילומים אלו ניתן למנוע מלכתחילה אירוע בטחוני או ביצוע תחקור לאירוע שהתרחש בעבר. בתחילה, פרויקט זה יציג את הקשיים והאילוצים הקיימים בעיבוד קטעי וידיאו בזמן אמת כמו האבחנה האם ההתרחשות המתוארת בקטע הינה חשודה או לא, וכן אילוצים הנדסיים בעיבוד תמונה כמו שינויי אור וצל.

על מנת לפתח מערכת אשר תתריע בהתרחשות אשר מוגדרת כחשודה יש צורך בהפעלת אלגוריתמים שונים לזיהוי אובייקטים, מעקב אחריהם וסיווג ההתרחשות כחשודה או לא. בהמשך יוצגו שיטות שונות למימוש אלגוריתמים אלו.

כמו כן, תוצג סכימה למימוש המערכת תוך מתן הסבר מפורט על אופן פעולת המערכת, על האלגוריתמים השונים המשולבים בה על מנת לספק זיהוי והתרעה במקרה של ריב בין אנשים אשר נתפס במצלמות האבטחה.

בהמשך, יוצגו כלל הבדיקות והניסויים שבוצעו למערכת, אופן בחירת הפרמטרים השונים מסקנות ותוצאות.

המערכת מספקת את היכולת לזהות ולהתריע כאשר מתרחש אירוע חריג כדוגמת ריב בין אנשים עם אחוזי הצלחה גבוהים אך עם כמה הגבלות שיפורטו בהמשך.

## **Abstract**

Nowadays, the need and use of security cameras in crowded places and near country borders is very common. With those photos we can prevent security incidents and investigate incidents that occurred in the past.

Firstly, this report will present the difficulties and the constraints of processing video clips in real time, as the diagnosis is whether the events described in the video section are suspicious or not. In order to develop a system that will alert at any suspicious occurrence, we need to run a few algorithms in order to identify objects, track them and classify as suspect or not. Various methods for realizing these algorithms are shown in this report.

A scheme of system implementation is shown, it includes a detailed explanation about the operational mechanism of the system, and which algorithms are combined in it, in order to provide movement detection and also alert in cases of people caught fighting in security cameras.

We present in the last part of this report tests and experiments performed on the system, including a discussion on the choice of the different parameters. The system provides the ability to identify people fighting, and displays an alert in that case. Performances and limitations are discussed.

## תוכן עניינים

1.....	<b>1. מבוא</b>
1.....	1.1. צילומי אבטחה : צרכים ושימושים
1.....	1.2. סקירת תנועות החשודות
3.....	1.3. הגדרת הבעיה ההנדסית
3.....	1.4. אילוצים וקשיים במימוש המערכת
4.....	<b>2. סקר ספרות</b>
4.....	2.1. זיהוי אובייקטים מתוך פריים בודד בוידאו
4.....	2.1.1. קביעת ערך סף מסתגל ליצירת mask [4]
5.....	2.1.2. פעולות מורפולוגיות לזיהוי אובייקטים [1],[3]
7.....	2.2. עקיבה אחר אובייקטים בקטע וידאו
7.....	2.2.1. עקיבה על בסיס אלגוריתם חישוב קורלציה בין היסטוגרמות [1]
7.....	2.2.2. עקיבה על-בסיס אלגוריתם חיזוי תנועת גופים [3]
8.....	2.3. סיווג סצנות שונות בקטע וידאו [2]
11.....	2.4. ספריית OPENCV
12.....	<b>3. מבנה המערכת</b>
12.....	3.1. אלגוריתם זיהוי אנשים בפריים
12.....	3.1.1. הוצאת מאפייני HOG – Histogram Of Oriented Gradients
15.....	3.1.2. מנגנון SVM – Support Vector Machine
18.....	3.2. סיווג סצנה כריב בין אנשים
22.....	<b>4. מערך בדיקות סופיות</b>
22.....	4.1. הקדמה
22.....	4.2. מערך הבדיקות
22.....	4.2.1. כלי הבדיקות ותיאורן
22.....	4.2.2. בדיקות אופטימיזציה למערכת :
24.....	4.2.3. מימוש ב- Real Time
24.....	4.2.4. בדיקות אימות למערכת

26.....	4.3 דיון בתוצאות
29.....	5. סיכום
30.....	מקורות
31.....	נספח א' – קוד המערכת

## רשימת איורים

- איור 1.1 דוגמא לפעילות חשודה המתוארת כריב בין אנשים.....2
- איור 1.2 דוגמא לפעילות חשודה המתוארת כחפץ חשוד.....2
- איור 1.3 דוגמא לפעילות חשודה המתוארת כשוטטות.....3
- איור 2.1 תיאור תהליך משותף לעקיבה וסיווג סצנות.....4
- איור 2.2 ערך סף מסתגל ליצירת mask.....5
- איור 2.3 המחשה של פעולת הרחבה.....6
- איור 2.4 המחשה של פעולת הסגירה.....6
- איור 2.5 וקטור שינוי המיקום והמיקום החזוי של קבוצת אובייקטים.....8
- איור 2.6 דיאגרמת מצבים לסיווג אובייקטים.....8
- איור 2.7 המחשת סיווג סצנה כריב בין אנשים.....9
- איור 2.8 סיווג מטען נטוש כחפץ חשוד.....10
- איור 3.1 סכימה מלבנית למימוש המערכת.....10
- איור 3.2 תיאור הוצאת מאפייני HOG.....12
- איור 3.3 פירמידת תמונה בחלוקה לרמות.....14
- איור 3.4 נתוני מאפייני HOG.....15
- איור 3.5 בחירת מישור אופטימלי בין שתי מחלקות.....16
- איור 3.6 מרחק בין שתי מחלקות מסווגות.....17
- איור 3.7 זיהוי שני אובייקטים במהלך מריבה כאובייקט אחד.....19
- איור 3.8 תרשים זרימה אופן פעולת סיווג התרחשות בפריים.....20
- איור 4.1 פעולת האלגוריתם במרכז קניות.....25
- איור 4.2 דוגמא להתרעת שווא.....25
- איור 4.3 עקומי סיכום בדיקות.....26
- איור 4.4 השוואה בין ערכי padding שונים באותו פריים.....27

27.....איור 4.5 השוואה בין ערכי scale שונים באותו פריים

28.....איור 4.6 המחשת פעילות תקינה של המערכת



## 1. מבוא

בפרק זה יוצגו השימושים והצרכים השונים בצילומי מצלמות אבטחה. אחד הצרכים העיקריים הינו לזהות התרחשויות חשודות במקומות הומי אדם. כמו כן, תוצג הבעיה ההנדסית עימה אתמודד והקשיים והאילוצים בפתרונה.

### 1.1. צילומי אבטחה : צרכים ושימושים

כיום, עקב המצב הביטחוני האזרחי הלא יציב בארצנו ובעקבות התפתחות וזמינות הטכנולוגיה, נפרסות מצלמות אבטחה ומעקב במקומות מסחריים, ברחובות ובכבישים. מבחינת שמירת הביטחון השוטף בגבולות המדינה, גם הצבא פורס מצלמות וסנסורי תצפית מתקדמים לאורך גבולות המדינה ובאזורי חיכוך ידועים.

על-ידי מצלמות אלו ניתן לאתר מבעוד מועד אירוע בטחוני או לתחקר אירוע שהתרחש בעבר. ישנם מערכות רבות בטלויזיה במעגל סגור (CCTV) אשר מאפשרות תחקור של אירועים ביטחוניים שהתרחשו בגזרת הצילום ואיתור חשודים, וכן הצבת מצלמות בשטחי מסחר הומים ומסופי תחבורה ציבורית מרכזיים אשר מטרתם שמירה על בטחון המבקרים במקומות אלו וכן, תחקור לאחור במקרי שוד וגניבה.

במערכות הצבאיות ישנם תצפיתניות אשר סורקות שטח מוגדר באמצעות מצלמות אלו 24/7 על-מנת לאתר מפגעים וחדירות לשטח המדינה.

אחת המטרות העיקריות שלשמן מבוצעים פיתוחים של אלגוריתמים של עיבוד תמונה וראייה ממוחשבת בתחום זה הינו הפקת מקסימום מהמידע המוכל בצילומי האבטחה וכן הורדת התלות בגורם האנושי הנדרש לצפות בצילומים אלו שכן, בניגוד למערכת טכנולוגית, זמני התגובה ורמת הריכוז של הצופה בתמונות האבטחה לא נשמרת לאורך זמן.

### 1.2 סקירת תנועות החשודות

במסגרת פרויקט זה נחקרו מספר התרחשויות אשר מקבלות את התיאור כחשודות. כלל ההתרחשויות בעלות מאפיינים שונים אשר ניתן לתרגם אותם לתנאים ולביטויים מתמטיים, כפי שיוצג בפרק הבא. בפרק זה יוצגו מאפיינים והגדרת התנועות החשודות תוך שימת דגש לפן החברתי-מוסרי ולא לפן הטכני.

#### 1.2.1 תגרה בין אנשים

אחת ההתרחשויות הנפוצות ביותר הינו תגרה או ריב בין אנשים, בין אם מדובר בפעילות טרור ובין אם זה מתרחש בפארק הציבורי או ברחוב הומה אדם. דוגמאות לסצנות אשר מסווגות כריב שנקלטו במצלמות אבטחה שונות מוצגות באיור 1.1 :



**איור 1.1 דוגמאות לפעילות חשודה המתוארת כריב בין אנשים**

## 1.2.2 חפץ חשוד

סוג נוסף של התרחשויות חשודה הינה הוצאת התרעה על חפץ נטוש ללא השגחה במרחב הקרוב אליו, וברוב המקרים החפץ מונח במקומות הומי אדם כדוגמת תחנת רכבת או רחוב מרכזי וכדומה. דוגמאות לסצנות אשר מסווגות כחפץ חשוד שנקלטו במצלמות אבטחה שונות מוצגות באיור 1.2:



**איור 1.2 דוגמאות לפעילות חשודה המתוארת כחפץ חשודה**

## 1.2.3 שוטטות

הגדרת ההתרחשויות הינה נוכחות של אדם באזור מסוים בפרק זמן ארוך יותר מהנדרש לביצוע פעולה מסוימת. אמנם זמני התגובה משתנים מפעולה לפעולה אך מתוך כך אם המערכת תדע לבצע הבדלה בין סוגי הפעולות השונות ולהבחין בין זמני תגובה שונים הרלוונטיים לכל פעולה יעלה את יוקרתה וחשיבותה באבטחת המרחב הציבורי. דוגמאות לסצנות אשר מסווגות כחפץ חשוד שנקלטו במצלמות אבטחה שונות מוצגות באיור 1.3:



**איור 1.3 דוגמאות לפעילות חשודה המתוארת כשוטטות**

### 1.3 הגדרת הבעיה ההנדסית

רבות עסקו בנושא עיבוד תמונה ווידאו המתקבל ממצלמות אבטחה אלו על מנת לבצע עיבוד בזמן אמת, real-time, ולהציג התראות על התרחשויות לא סטנדרטיות וחשודה כפי שתוארו לעיל.

אחת הבעיות המרכזיות עליהן המערכת צריכה להתגבר הינו מתן התרעה נכונה וביצוע זיהוי נכון למתרחש בתמונה הנכנסת למערכת. כמובן שישנו אחוז סטייה שבו המערכת תיתן התרעה שווא, אך נדרש כי אחוז זה יהיה נמוך ככל האפשר. ניתן לתאר בפשטות לדוגמא את הבעיה על ידי לקיחת שתי התרחשויות בעלות מאפיינים זהים, ריב בין אנשים וחיבוק בין אנשים רגילים ברחוב, המאפיינים זהים כמעט לגמרי בשתי ההתרחשויות הללו, בשניהם האנשים בתמונה מתקרבים אחד לשני ואף נצמדים אחד לשני אך עם מטרות קירבה שונות, וכאן נכנס האתגר הגדול בפיתוח מערכת כזו בביצוע זיהוי והבנה נכונה למתרחש ובהתאם הוצאה או אי הוצאת התרעה למנטר.

### 1.4 אילוצים וקשיים במימוש המערכת

במימוש מערכת כזו קיימים מספר קשיים ואילוצים שנדרש להתחשב בהם. ראשית, על מנת שהמערכת תספק התרעה בזמני תגובה קצרים מאוד, near real time, נדרשות יכולות עיבוד והיתוך מידע גבוהות לכמות מידע גדולה אשר יכולות להגיע דרך חומרה חזקה, קרוב לוודאי באמצעות שרתים עם מעבדים חזקים ומעבדים גרפיים.

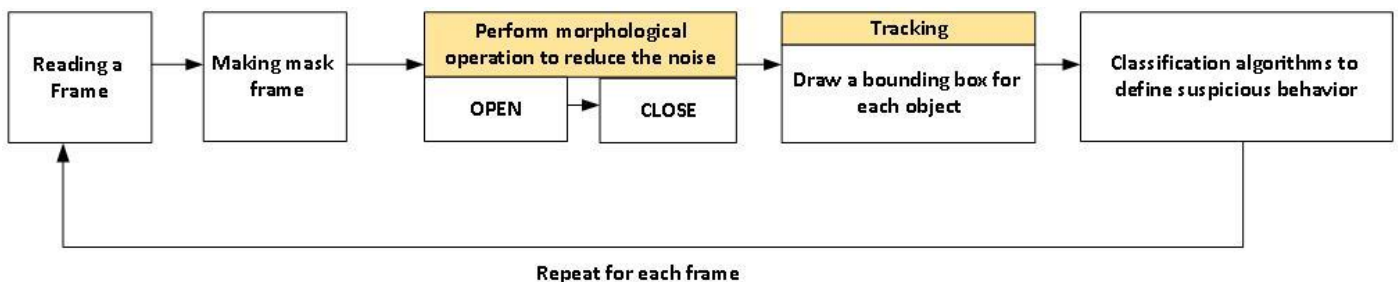
שנית, במסגרת הזמן המועט והקצוב למימוש המערכת לא ניתן לספק מענה במסגרת פרויקט זה לכלל התנועות וההתרחשויות החשודות שהוצגו בתחילת פרק זה, ולכן ההתמקדות תהיה על סוג אחד של התרחשויות והינה ריב בין אנשים.

## 2. סקר ספרות

בפרק זה יוצגו השיטות השונות הקיימות לזיהוי ועקיבה אחר אובייקטים בוידאו, וכן שיטות ושנות לסיווג תנועות המוגדרות כחשודות.

הסכימה הבאה, המוצגת באיור 2.1, מתארת את התהליך המשותף ברובו לאלגוריתמים הרבים המתארים עקיבה אחר אובייקטים בתמונה וסיווג הסצנה כחשודה או לא.

עבור כל פריים בקטע הוידאו ניצור mask – תמונה לוגית המכילה ערכי 0 או 1 על-פי קריטריונים שונים לפי השיטות השונות, לאחר מכן יבוצעו על כל פריים פעולות מתמטיות מורפולוגיות אשר יפורטו בהמשך, מתוך הפעולות הללו ניתן לזהות ולהפריד אובייקטים מתוך הפריים הבודד. לאחר שהאובייקטים זוהו, יופעל אלגוריתם לסיווג התנועה או הסצנה המתרחשת כחשודה או לא.



### איור 2.1 תיאור תהליך משותף לעקיבה וסיווג סצנות

2.1 זיהוי אובייקטים מתוך פריים בודד בוידאו  
 על מנת לזהות אובייקטים בפריים בודד, מתבצעות סדר הפעולות הראשוניות כפי שמתואר לעיל.

תחילה, יבוטלו רעשים בפריים וניצור mask (תמונה לוגית המורכבת מערכי 0 או 1), כדי שנוכל לבצע בעזרתה את הפעולות המורפולוגיות המתוארות בהמשך.

#### 2.1.1 קביעת ערך סף מסתגל ליצירת mask [4]

במידה ונגדיר ערך סף יחיד גלובלי כדי ליצור תמונת mask, נוכל לאבד מידע הקשור לאובייקטים קיימים בתמונה אשר ממוקמים בפינות כתוצאה משינויים של אור וצל. השיטה המוצגת הינה קביעת ערך סף מסתגלים לפי המידע הקיים בתמונה הנתונה, כאשר התמונה תחולק למקרו-בלוקים (פרמטר בר הגדרה) ויבוצע חישוב של מינימום ומקסימום מקומי בתוך המקרו-בלוק ויופעל תנאי, האם ערך החיסור בין המקסימום והמינימום גדול או קטן מערך הסף שנקבע.

$$dst(x, y) = \begin{cases} 1, & M_{xy} - m_{xy} > T \\ 0, & M_{xy} - m_{xy} < T \end{cases} \quad (2.1)$$

$M_{xy}$  – ערך מקסימום מקומי.

$m_{xy}$  – ערך מינימום מקומי.

$dst(x, y)$  – ערך פיקסל התמונה במוצא.

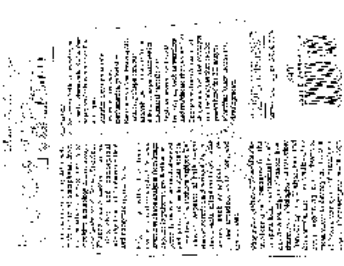
האיור 2.2 ממחיש את פעולת קביעת ערך סף מסתגל לעומת קביעת ערך סף גלובלי לתמונה.

פעולת ערך הסף המסתגל חילקה את התמונה למקרו-בלוקים בגודל של  $5 \times 5$ .

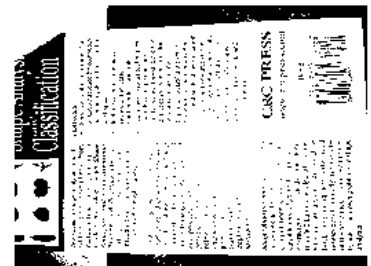
The original gray picture



The Adaptive mask



The Global mask



### איור 2.2 ערך סף מסתגל ליצירת mask

ניתן לראות כי בתמונה עם ערך הסף המסתגל קיים מידע במרכז התמונה שאינו נמצא בתמונה עם ערך הסף הגלובלי וזאת בשל שינויי אור בתמונה.

2.1.2. פעולות מורפולוגיות לזיהוי אובייקטים [1],[3]

לאחר יצירת mask לכל פריים בקטע הוידאו, יבוצעו 2 פעולות מורפולוגיות מתמטיות על התמונה:

Open – פתיחה של התמונה על מנת לצמצם אזורים בתמונה עם רעש.

$$A \circ B = (A \ominus B) \oplus B \quad (2.2)$$

האיור 2.3 מדגים את פעולת ההרחבה על תמונה:

The original gray picture



The logical mask



Morphology OPEN operation output



### איור 2.3 המחשה של פעולת הרחבה

ניתן לראות כי על-ידי פעולת ההרחבה האזורים השחורים הורחבו לעומת האזורים השחורים שהיו בתמונת ה-mask המקורית.

Close – סגירה, צמצום של החורים בתמונה לאחר שהורדנו את רכיבי הרעש כדי להשלים את המידע שאבד.

$$A \bullet B = (A \oplus B) \ominus B \quad (2.3)$$

האיור 2.4 מדגים את פעולת הסגירה על תמונה :

The original gray picture



The logical mask



Morphology CLOSE operation output



### איור 2.4 המחשה של פעולת סגירה

ניתן לראות כי על-ידי פעולת הסגירה חורים שחורים קטנים נסגרו, אזורים עם שיוניים מהירים בין שחור ללבן נעלמו ונהיו לבנים, וכן חורים גדולים נשארו כמו בתמונת ה-mask המקורית.

לאחר ביצוע הפעולות המורפולוגיות, נתחום את האובייקטים שקיבלנו על ידי מסגרת וכך בעצם איתרנו את האובייקטים בתמונה הבודדת.

כלל הפעולות שתוארו יופעלו על כלל הפריימים בוידאו.



## 2.2 עקיבה אחר אובייקטים בקטע וידאו

2.2.1. עקיבה על בסיס אלגוריתם חישוב קורלציה בין היסטוגרמות [1]  
 על-פי שיטה זו, לכל פריים בוידאו ולפריים הקודם שלו, נחשב היסטוגרמה לכל אחד  
 מרכיבי הצבע: אדום, ירוק וכחול (RGB).

לאחר קבלת ההיסטוגרמות נחשב קורלציה בין היסטוגרמת הפריים הנוכחי לפריים  
 הקודם לכל אחד מרכיבי הצבע, את התוצאה של כל קורלציה נסכום למשתנה אחד.

במידה וקיבלנו כי ערך סכום הקורלציות המקסימלי הוא 3 (מכיוון שערך מקסימאלי של  
 קורלציה בין 2 היסטוגרמות הוא 1), ניתן לומר בוודאות כי האובייקט שהופיע בפריים  
 הקודם מופיע גם בפריים הנוכחי, וכן הלאה.

לחישוב הקורלציה בין 2 היסטוגרמות במרכיב צבע זהה קיים ערך סף, במידה וערך  
 חישוב הקורלציה שהתקבל נמוך מערך סף זה, ניתן לומר כי מדובר באובייקט חדש.

2.2.2. עקיבה על-בסיס אלגוריתם חיזוי תנועת גופים [3]  
 על-פי שיטה זו, לאחר שזיהינו את האובייקטים בפריים, נחלק אותם לקבוצות ונתחום  
 אותם במסגרת, נבצע חישוב ושערוך של מיקום קבוצת האובייקטים בפריים הבא על-  
 בסיס נתוני מיקום האובייקט בפריים הנוכחי ובפריים הקודם. כל קבוצת אובייקטים  
 מקבלת מזהה ייחודי (ID), כך יהיה ניתן לעקוב אחר תנועת הקבוצה לאורך כל קטע  
 הוידאו. הנוסחאות מטה מתארות את תהליך החישוב וחיזוי התנועה:

$$\begin{aligned} PG_j^k[x] &= IG_j^k[x] + \cos(IGD_j^k[\theta])IGD_j^k[v] \\ PG_j^k[y] &= IG_j^k[y] + \sin(IGD_j^k[\theta])IGD_j^k[v] \\ IGD_j^k[v] &= \frac{\|IG_j^k - IG_{j-1}^k\|}{1 / \text{FrameRate}} \end{aligned} \quad (2.4)$$

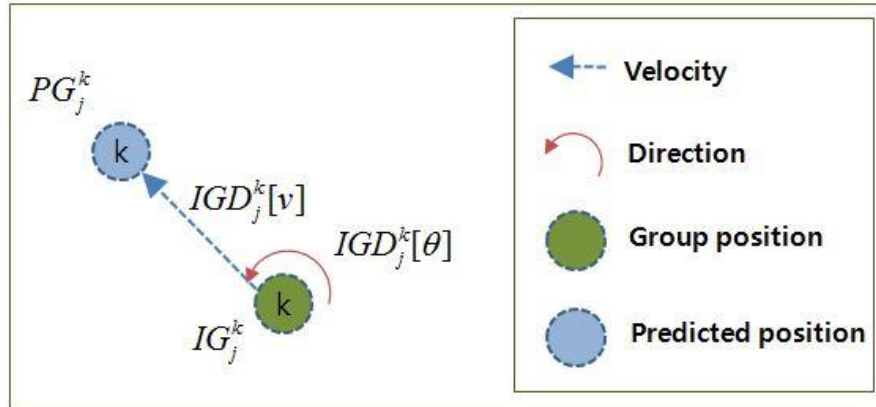
$$IGD_j^k[\theta] = \begin{cases} \cos^{-1}\left(\frac{IG_j^k[x] - IG_{j-1}^k[x]}{IG_j^k[y] - IG_{j-1}^k[y]}\right), & IG_j^k[y] - PG_j^k[y] \geq 0 \\ \cos^{-1}\left(\frac{IG_j^k[x] - IG_{j-1}^k[x]}{IG_j^k[y] - IG_{j-1}^k[y]}\right) + \pi, & \text{else} \end{cases}$$

$IG_j^k[x], IG_j^k[y]$  - מיקום קבוצת אובייקטים k בפריים j הנוכחי לפי ערכי x ו-y.

$PG_j^k[x], PG_j^k[y]$  - מיקום חזוי של קבוצת אובייקטים k לפי הנתונים בפריים j.

$IDG_j^k[\theta], IDG_j^k[v]$  - מציין את מהירות שינויי הכיוון של קבוצה k.

האיור 2.5 ממחיש את הנוסחאות (2.4):

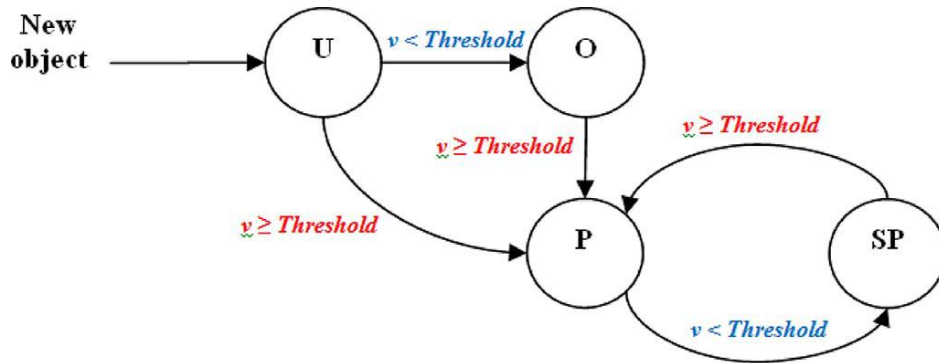


איור 2.5 וקטור שינוי המיקום והמיקום החזוי של קבוצת האובייקטים. מופיע ב- [3]

2.3 סיווג סצנות שונות בקטע וידיאו [2]

בהתאם לנאמר בפרק המבוא, בחלק זה יוצגו קריטריונים לקביעת התרחשויות בקטע וידיאו כחשודות או לא, לגבי 3 מצבים שונים.

ההנחה בשיטה זו היא, שבפריימים הקודמים התבצע זיהוי ועקיבה לאובייקטים הנתונים בקטע. בתחילה, כלל האובייקטים יסווגו לפי הקטגוריות הבאות: בן-אדם (בתנועה), בן-אדם עומד, חפץ ועצם לא מזוהה. דיאגרמת המצבים הבאה ממחישה את אופן סיווג כל אובייקט:



$U = \text{unknown}, P = \text{person}, SP = \text{still person}, O = \text{inanimate object}, v = \text{velocity}.$

איור 2.6 דיאגרמת מצבים לסיווג אובייקטים. מופיע ב- [2]

ההבחנה בסיווג האובייקטים הינה על-ידי בדיקת מהירות האובייקט והשוואתו לערך הסף שנקבע:  $v_{\text{Threshold}} = 0.5 \text{ m/s}$ . חישוב מהירות האובייקט יתבצע כך:

$$v_i^t = \frac{\|p_i^t - p_i^{t-\Delta t}\|}{\Delta t} \quad (2.5)$$

$$p_i^t = (x_i^t, y_i^t)$$



$v_i^t$  – מהירות האובייקט  $i$  בזמן  $t$ .

$p_i^t$  – מיקום אובייקט  $i$  בזמן  $t$  לפי קורדינאטות  $x, y$ .

סיווג התרחשויות חשודות אשר בעקבותיהם תוצג התראה: [1-2]  
 קיימים סוגים רבים של התרחשויות חשודות שעליהן ניתן להציג התראה, אך בפרויקט זה  
 יפורטו 3 סוגים של התרחשויות חשודות: ריב בין אנשים, זיהוי חפץ חשוד ושוטטות  
 והשתהות במרחב מוגדר.

ריב בין אנשים – סצנת ריב היא אחת הקשות והמאתגרות באפיון שלה מכיוון שישנן  
 אפשרויות שונות להגדיר ריב.

על-מנת לסווג התרחשויות כלשהי כריב בין אנשים, השיטה המוצעת הינה לבדוק את תדירות  
 היפרדות והתמזגות של אובייקטים במקטע הוידאו. בדיקה זו מתבצעת על-ידי חישוב שינויי  
 מהירות האובייקטים ביחס לערכי סף שנקבעו מראש, כאשר התנאי המקדים הינו סיווג  
 האובייקטים כ-P (לפי דיאגרמת המצבים באיור 2.6), אובייקטים נעים ולא אובייקטים  
 נייחים או חפצים.

איור 2.7 ממחיש סיווג סצנה כריב לאור התמזגות גופים רבים ביחד, מופיע ב [2].



**איור 2.7 המחשת סיווג סצנה כריב בין אנשים**

התנאי שיש לקיים על-מנת לסווג ריב:

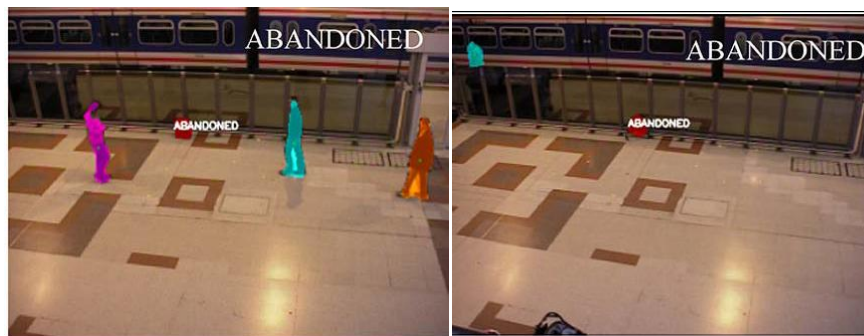
$$\begin{aligned}
 &v_{obj1} > v_{fight_{min}} \\
 &\& \\
 &v_{obj2} > v_{fight_{min}}
 \end{aligned}
 \tag{2.6}$$

$v_{obj1}, v_{obj2}$  – מהירות אובייקטים 1 ו-2.

$v_{fight_{min}} = [0.5, 1] m/s$  – מהירות הייחוס שנקבע לתרחיש זה.

חפץ חשוד – אחת המטרות במערכות צילומי אבטחה הינה למנוע מעשי טרור ואחת הדוגמאות לכך הינה הנחת מטען במקום מסחרי הומה אדם או באזורי ריכוז גדולים כמו תחנת רכבת. אחד האתגרים בגילוי חפץ חשוד הינו ההבחנה בין חפץ נטוש המעורר חשד לבין אדם סטציונרי העומד במקום. יש לבצע לאובייקט סיווג ראשוני כחפץ ( O לפי דיאגרמת המצבים באיור 2.6), בנוסף, ייקבע ערך סף של זמן מינימאלי שלא נגעו בחפץ זה, ופרמטר של מרחק מינימאלי של אדם העומד בקרבת החפץ

איור 2.8 ממחיש סיווג סצנה מטען נטוש כחשוד בתחנת רכבת, מופיע ב [2].



**איור 2.8 סיווג מטען נטוש כחפץ חשוד**

כאשר 2 התנאים הבאים יתקיימו ניתן לקבוע כי החפץ מוגדר כנטוש וחשוד:

$$\begin{aligned} t &> t_{abandoned} \\ d_{obj1} &> d_{abandoned} \end{aligned} \quad (2.7)$$

$t$  – זמן שלא נגעו בחפץ המתואר.

$t_{abandoned}$  – ערך סף לקביעת זמן נגיעה בחפץ המתואר, נקבע ל- 30sec.

$d_{obj1}$  – מרחק מינימאלי של אובייקט 1 מהחפץ המתואר.

$d_{abandoned}$  – מרחק מינימאלי של אובייקט מהחפץ, נקבע ל- 3m.

שוטטות והשתהות במרחב מוגדר - העיקרון המנחה בהגדרה זו היא, שאובייקט שזוהה כאדם או כאדם עומד (P או SP לפי דיאגרמת המצבים באיור 2.6), נמצא במקום מסוים מעבר לזמן הסביר לשם ביצוע הפעולה הנדרשת. לכן, מוגדר ערך סף של זמן סביר לביצוע הפעולה והימצאות של האובייקט מעל זמן זה תגדירו כחשוד. לפי השימוש ומיקום התקנת המערכת יוגדר ערך הסף, שכן, זמן הוצאת כסף מכספומט או זמן מעבר בידוק בטחוני בשדה תעופה שונים לגמרי אחד מהשני.

התנאי שצריך להתקיים הינו:

$$lifetime(obj1) \geq t_{loitering} \quad (2.8)$$

$lifetime(obj1)$  - הזמן שלוקח לאובייקט לבצע את הפעולה הנדרשת.

$t_{loitering}$  - הינו פרמטר אשר משתנה, אינו קבוע ותלוי בכל תרחיש שבו נדרש השימוש בו, שכן כפי שהוזכר לעיל, זמני הפעולה הממוצעים משתנים בין כל סוג פעילות נדרשת ואין זה ערך קבוע וגלובלי.

#### 2.4 ספריית OPENCV

Opencv (Open Source Computer Vision) - זוהי ספריית תוכנה שימושית ונפוצה לעיבוד תמונה ולראייה ממוחשבת. הספרייה מכילה למעלה מ 2500 אלגוריתמים בעיבוד תמונה כמו אלגוריתמים לזיהוי פנים, עקיבה, שחזור אובייקטים בתלת מימד וכן מכילה מנגנוני machine learning.

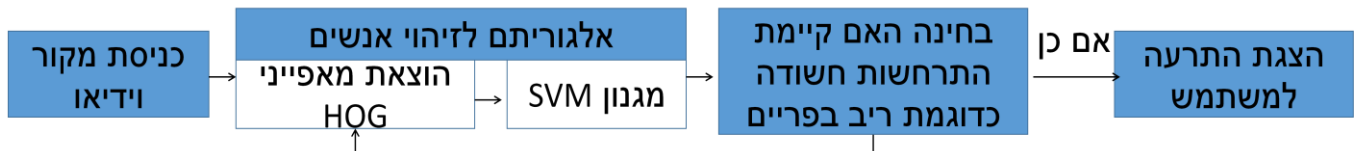
בנוסף, קיימת היכולת לשימוש בספרייה זו על מנת לפתח מערכות בזמן אמת, Real Time, תוך ניצול המעבד הגרפי (gpu) במידה וקיים.

ספרייה זו הינה קוד פתוח ומונה קהילה של כ-50 אלף משתמשים ברחבי העולם אשר מעדכנים את התכנים של הספרייה ומוסיפים אלגוריתמים לשימוש הכלל.

ניתן לפתח ולשלב את הספרייה הזו במגוון רחב של שפות תכנות כמו C, C++, python, וכו' וניתנת לשילוב בכל מערכת הפעלה Windows, Linux, או Mac.

### 3. מבנה המערכת

בפרק זה תוצג מבנה המערכת אשר תממש זיהוי של תנועה חשודה בקטע וידיאו, כאשר ההתמקדות בפרויקט הינו זיהוי ריב וסצנת דקירה בין אנשים. בתחילה, תוצג סכימת מלבנים, איור 3.1, אשר תתאר את אלגוריתם המערכת ובהמשך יפורטו האלגוריתמים.



מתבצע עבור כל פריים

#### איור 3.1 סכמה מלבנית למימוש המערכת

3.1 אלגוריתם זיהוי אנשים בפריים

כפי שמתואר בסכימת בלוקים לעיל, המערכת תבדוק כל פריים בקטע הוידאו נכנס אליה על מנת לזהות את האנשים תחילה ולאחר מכן תסווג האם מדובר בסצנת ריב או לא. זיהוי האנשים בפריים מתבצע בשני שלבים על ידי חיבור שני אלגוריתמים קיימים:

3.1.1 הוצאת מאפייני HOG – Histogram Of Oriented Gradients

מטרתו של שימוש במאפייני HOG היא לרכז בוקטור אחד את כל הנתונים על האובייקטים הנעים בפריים הנכנס למערכת.

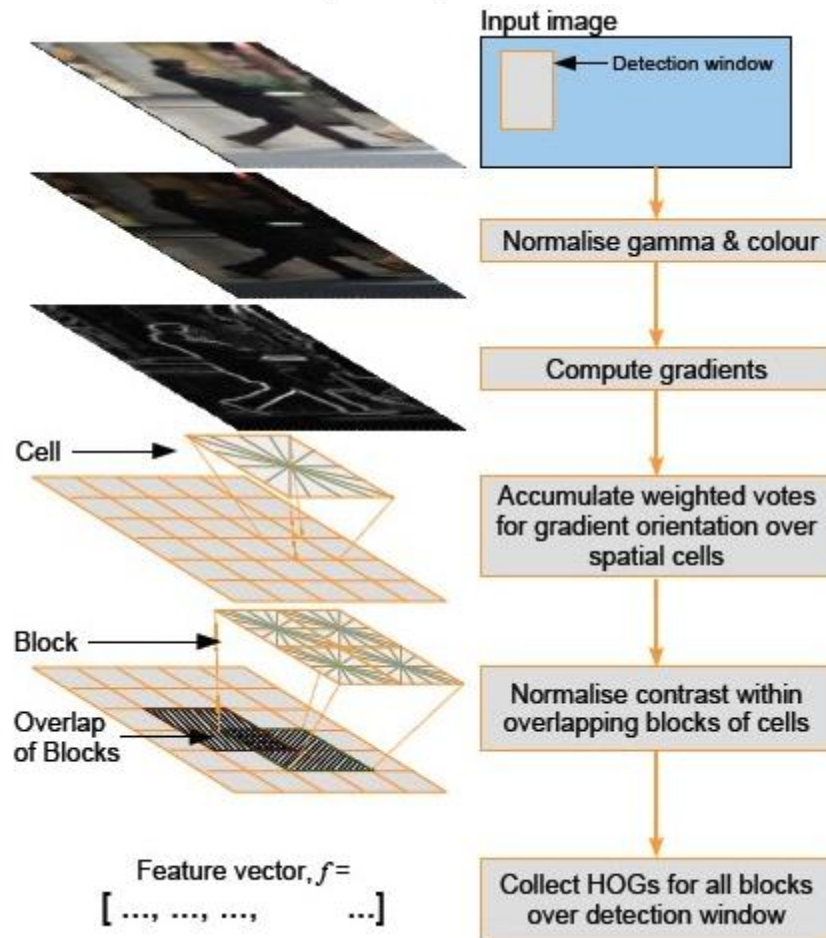
אופן הפקת המאפיינים:

בתחילה יבוצע נרמול של מרכיבי הצבע בפריים, חישוב גרדיאנט כללי לתמונה, לאחר מכן, התמונה תחולק ותיסרק לפי בלוקים וכל בלוק מחולק למספר תאים (cells). בכל תא יצטבר ערכי הגרדיאנט ובהמשך בסריקה הבאה במסגרת סריקת הבלוק יתבצע נרמול לתוצאה שהתקבלה, כך נוצר מצב שישנן סריקות כפולות לאותו תא על מנת למקסם את התוצאות.

כל מכלול התוצאות של הפריים מרוכזות בוקטור אחד והוא נשלח למנגנון ה-SVM לביצוע קלסיפיקציה – אימות האם הזיהוי הינו בן אדם או לא.

תיאור ויזואלי של תהליך הפעולה מתואר באיור 3.2:

# SCE



## איור 3.2 תיאור הוצאת מאפייני HOG – Histogram Of Oriented Gradients

בספריות הקוד של Opencv הפונקציה העיקרית בה תשמש המערכת הינה:

Hog.detectMultiscale (input, vector location, hitTreshold, winStride, padding, scale, finalTreshold)

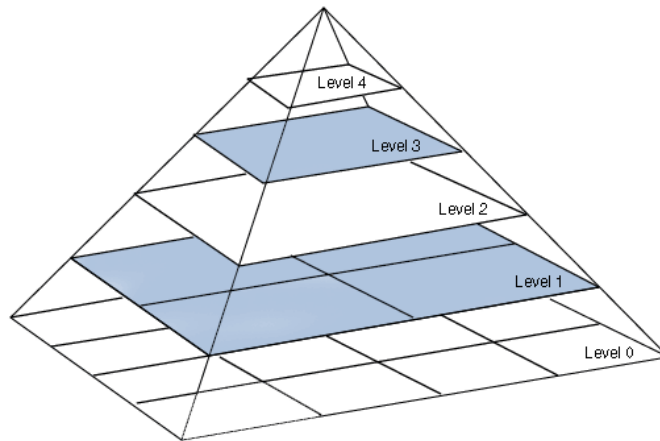
הסבר פרמטרי הכניסה של הפונקציה:

hitTreshold – פרמטר אשר אחראי על המרחק האוקלידי של מנגנון הזיהוי ב-SVM (מוגדר ל-0 כברירת מחדל).

winstride – פרמטר אשר שולט על גודל החלון שרץ וסורק את התמונה בכניסה לפונקציה. בכל עצירה של סריקת החלון נשלחים ערכים למנגנון ה-SVM לביצוע קלסיפיקציה, ככל שגודל החלון יהיה קטן יותר כך ישלחו פחות נתונים לעיבוד בכל רגע נתון וכך הקוד יתקרב לזמן אמת (real-time). במידה וגודל החלון יהיה גדול מדי, יוכל להיווצר מצב שנאבד זיהוי של עצמים שכן נרצה לזהות.

**Padding** – פרמטר המציין הוספת מרווח בין דגימה לדגימה, על פי המודל יש לשים מרווחים בין הדגימות השונות של החלון שרץ בסריקות כדי לקבל דיוק גבוהה יותר בתוצאות. ערכים אפשריים:  $8 \times 8$ ,  $16 \times 16$ ,  $24 \times 24$ ,  $32 \times 32$ .

**Scale** – בעיבוד תמונה ובראייה ממוחשבת כל תמונה מחולקת למספר רמות רזולוציה שניתן לגשת אליהן ולעבוד איתן – image pyramid. פרמטר זה מציין את מספר הרמות שהקוד יבחן בפריים הנכנס אליו, ככל שהפרמטר יותר נמוך כך מספר השכבות גבוהה יותר וכן זמן הריצה יהיה ארוך יותר. במידה והפרמטר יהיה נמוך מדי (1.01) נקבל דגימות יתר (ריבוע על ריבוע על אובייקט יחיד), יחד עם זאת, ערך גבוה לפרמטר זה יוכל לגרום לאיבוד זיהוי אובייקטים שנרצה לזהות. תחום ערכי הפרמטר: [1.01-1.5].



### איור 3.3 פירמידת תמונה בחלוקת לרמות

פרמטר זה ופרמטר winstride קשורים זה לזה באופן מהותי ויש להגיע למצב האופטימאלי בו לא נאבד מידע בזיהוי וזמני או שיהיו דגימות יתר, בפרק הבא יתואר תהליך קביעת הפרמטרים המתאימים בהתאם לתוצאות הניסויים.

הוקטור הסופי המתקבל במוצא האלגוריתם מכיל את הפרמטרים של המסגרת התוחמת של אובייקט שזוהה: מיקום  $(x,y)$  של הפינה השמאלית העליונה של המסגרת, רוחב המסגרת והגובה. איור 3.4 מביא דוגמא לנתוני מוצא מאפייני HOG.

HOG:  $\langle (x_1, y_1, w_1, h_1), (x_2, y_2, w_2, h_2), (x_3, y_3, w_3, h_3), \dots \rangle$



**איור 3.4 נתוני מאפייני HOG**

### 3.1.2 מנגנון SVM – Support Vector Machine

#### 3.1.2.1 רקע

SVM היא טכניקה של למידה מכונת וקטורים תומכים מונחית. אלגוריתם למידה חישובית זה הוצג על ידי ולדימיר ופניק בשנת 1963, ומאז מהווה כלי מרכזי בפתרון בעיות באמצעים סטטיסטיים. כנהוג בתחום זה, דוגמאות האימון מיוצגות כווקטורים במרחב לינארי. תכליתו של שלב האימון היא בניית מסווג (classifier) אשר מפריד בצורה מדויקת ככל האפשר בין דוגמאות אימון חיוביות ושליליות. המסווג שנוצר ב SVM הוא מפריד לינארי אשר יוצר מרווח גדול ככל האפשר בינו לבין הדוגמאות הקרובות לו ביותר מתוך קבוצות האימון.

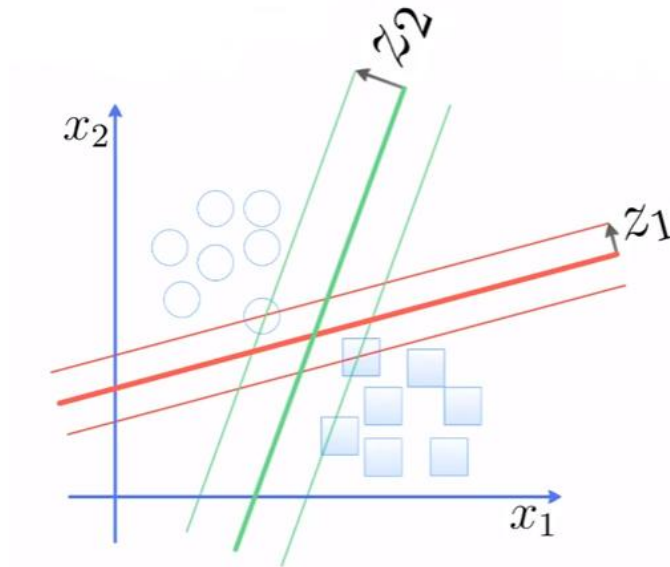
למנגנון ה-SVM קיימים שימושים רבים, לאו דווקא בזיהוי אנשים, יכולות האימות של מנגנון זה תלויות במידע שמאמנים אותו טרם הפעלתו במערכת הכללית. בפרויקט זה מנגנון ה SVM נועד לאמת את המידע הנכנס אליו האם זה בן אדם או לא, המידע המוקדם עליו מתבסס המנגנון הינו תמונות מזהות של אנשים בגדלים וזוויות שונות כחלק מפונקציה מובנית בתוך ספריות ה-OpenCv.

#### 3.1.2.2 מטרת מנגנון ה- SVM

המטרה העיקרית ב- SVM היא ליצור על-מישור מפריד (Separating Hyper plane), ב- SVM בינארי או מספר על-מישורים מפרידים בין וקטורי האלמנטים השונים למספר מחלקות. יכולות להיות מספר אפשרויות למיקום המישור המפריד, אך הבחירה הטובה ביותר תהיה כאשר המישור אשר יספק את המרחק הגדול ביותר בין שני המחלקות הוא ייבחר. איור 3.5



מטה מתאר המחשת בחירת מישור מפריד במתאר בינארי, כמו כן באמצעות דוגמא זו יוסבר אופן פעולת ה-SVM הבינארי בסעיף הבא.



### איור 3.5 בחירת מישור מפריד אופטימלי בין שתי מחלקות

על פי האיור מעלה, ניתן לראות כי מישור Z2 בעל מרחק גדול יותר בין המחלקות מאשר מישור Z1 ולכן הוא ייבחר כמישור הפרדה.

3.1.2.3 אופן פעולת ה-SVM

בהמשך לדוגמא באיור 3.5, למישור הנבחר תינתן משוואה אשר תתאר אותו ומתוך כך ייגזרו תנאי החלוקה:

$$g(\vec{x}) = \omega^T \vec{x} + \omega_0 \quad (3.1)$$

$g(\vec{x})$  - משוואת המישור המפריד.

$\vec{x}$  - וקטור הכניסה.

$\vec{\omega}$  - וקטור רוחב, המרחק בין שני אובייקטים מקבוצות סיווג שונות.

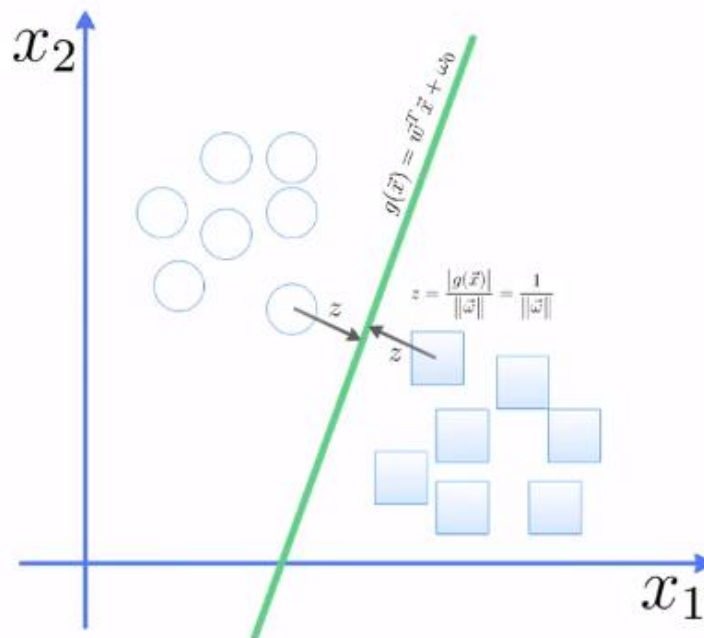
$\omega_0$  - משתנה חופשי.

תנאי הסיווג והחלוקה לקבוצות:

$$\begin{aligned} g(\vec{x}) &\geq 1, \forall \vec{x} \in \text{class1} \\ g(\vec{x}) &\leq -1, \forall \vec{x} \in \text{class2} \end{aligned} \quad (3.2)$$



בהמשך לדוגמא, class1 מציינת את קבוצת העיגולים ו-class2 מציינת את קבוצת הריבועים. הנחת העבודה היא כי המרחק בין המחלקות השונות לאחר ההפרדה הינו 1. לכן, ניתן לתאר את המרחק של האובייקט הקרוב למישור המפריד מכל קבוצת סיווג על ידי הביטוי  $\frac{1}{\|\vec{w}\|}$  ולכן את המרחק בין שני המחלקות ניתן לתאר כ-  $\frac{2}{\|\vec{w}\|}$ . המחשת המרחקים מתוארת באיור 3.6 מטה.



**איור 3.6 מרחק בין מחלקות מסווגות**

מתוך הביטוי למרחק בין המחלקות, המטרה היא לבצע מינימיזציה לוקטור  $\|\vec{w}\|$  כדי שנקבל מרחק מקסימלי בין שני המחלקות השונות וכך אחוזי הצלחת סיווג האובייקטים בצורה נכונה תעלה.

וקטור  $\vec{w}$  אינו ליניארי ולכן על מנת לבצע לו מינימיזציה יתבצע שימוש בכופלי לגראנז' :

$$\vec{w} = \sum_{i=0}^N \lambda_i y_i \vec{x}_i \quad (3.3)$$

לבסוף, תוצאת וקטור  $\vec{w}$  תיקרא support vector שבאמצעותה תבוטא משוואת המישור מפריד לצורך ביצוע סיווג בין אובייקטים.

#### 3.1.2.4 יתרונות וחסרונות של ה-SVM

שימוש במנגנון ה-SVM נובע מכמה יתרונות בולטים, הראשון שבהם הוא הנוחות והפשטות לשימוש בו, בנוסף המנגנון נפוץ ושימושי מאוד בקרב קהל המהנדסים העוסקים בעיבוד תמונה ובראייה ממוחשבת ולכן קיימת ספרות ענפה תוך מתן הסברים ודוגמאות רבות. יחד עם זאת, למנגנון זה גם מספר חסרונות, במקרה של זיהוי אנשים, המנגנון יוכל לספק זיהוי שגוי על אובייקטים בעלי מאפיינים זהים לאדם, כמו עץ. הפתרון לחסרון זה הוא יצירת מאגר גדול מאוד של דוגמאות כדי לאמן את כושר החלטת ה-SVM, כדי ליצור מאגר מידע גדול יש צורך בהקמת מסד נתונים (Database) המשולב יחד עם חומרות מחשוב בעלות כושר עיבוד גבוה.

#### 3.1.2.5 שילוב SVM במערכת

מוצא מאפייני HOG מספק למנגנון ה-SVM ווקטור אשר מכיל את המידע המבוקש לגבי אובייקטים בפריים המתקבל בכניסה למערכת. ייעודו של מנגנון ה-SVM במערכת הוא לבחון את המידע הקיים בווקטור המידע הנכנס אליו ולהחליט האם האובייקטים שזוהו הינם אנשים או לא.

#### 3.2 סיווג סצנה כריב בין אנשים

על מנת לסווג סצנה כריב קיימות כמה שיטות, אחת מהן הוזכרה בסקר ספרות כמדידת מהירות בין אובייקטים, שיטה נוספת היא בדיקה התמזגות והיצמדות אובייקטים אחד לשני. הרעיון שמומש במערכת זו הוא בחינה של היצמדות של אובייקטים קרובים אחד לשני. הרעיון מומש על ידי הפעלת תנאי האם בפריים שנכנס למערכת קיימים לפחות שני אובייקטים אשר זוהו בוודאות כאנשים והמסגרות התוחמות אותם קרובות מאוד אחת לשנייה ובתוך כך אפילו בעלות שטח כיסוי משותף או שמתוך התקרבות האובייקטים המערכת מזהה אותם כאובייקט אחד גדול. לצורך המחשה האיור מטה מתאר את זיהוי האובייקטים תוך כדי מריבה כאובייקט אחד, טרם הפעלת התנאי לסיווג הסצנה.

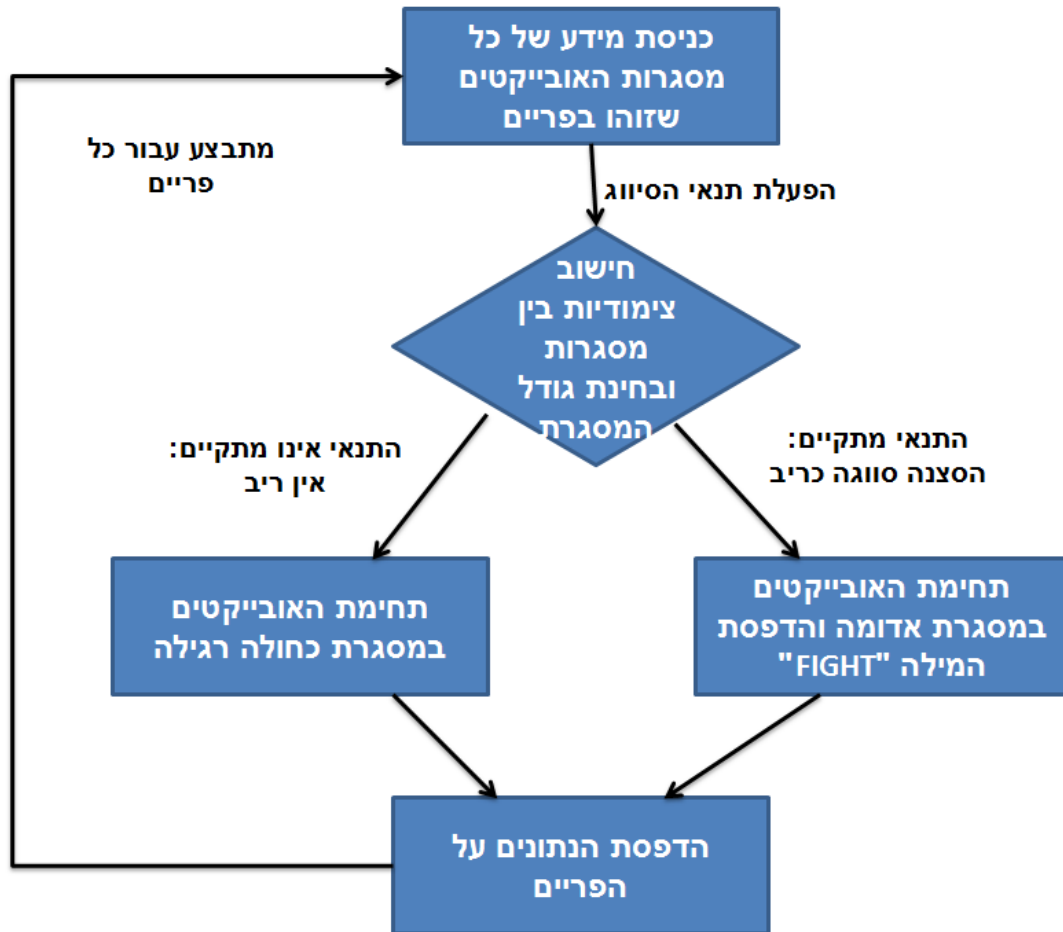


### איור 3.7 זיהוי שני אובייקטים במהלך מריבה כאובייקט אחד

על מנת להפעיל את תנאי סיווג הסצנה נשתמש במידע המתקבל לאחר סיווג האובייקטים מה- SVM, המידע שיתקבל הוא המסגרת התוחמת את האובייקט, משתנה מסוג Rect בעל ארבע פרמטרים שהוזכרו בתחילת פרק זה, קורדינאטות הפינה השמאלית העליונה של המסגרת  $(x,y)$ , רוחב המסגרת וגובה המסגרת.

פעולת הסיווג תבצע השוואה בין גודלי המסגרות של כלל האובייקטים שזוהו בפריים, האם יש מרחב משותף ביניהם או האם גודל המסגרת גדול כך שזוהו 2 אובייקטים כאחד גדול.

תרשים הזרימה המופיע באיור 3.8 מתאר את אופן פעולה הסיווג:



איור 3.8 תרשים זרימה אופן פעולת סיווג התרחשות בפריים

המשוואה הבאה מתארת את אופן פעולת הסיווג לפי אלגוריתם המערכת.

```

for(i >= 1){
  if ((abs(b[i] - b[i-1]) < 30) || b[i] > 95), FIGHT
}
  
```

(3.4)

$b$  - וקטור מסוג int אשר מכיל את ערכי רוחב המסגרות של כלל האובייקטים בפריים הנוכחי.

$b[i], b[i-1]$  - הינו אינדקס המציין את מספר האובייקט הנבחן, כאשר  $i$  מספר החל מ-0. ההשוואה נעשית בין מסגרת האובייקט הנוכחי ( $b[i]$ ) לעומת מסגרת האובייקט הקודם ( $b[i-1]$ ).

הערך המופיע בשורת התנאי, 30 פיקסלים, לבדיקת צימודיות נבחר לאחר מספר הרצות עם ערכים שונים עד לקבלת תוצאות מתאימות לצורכי המערכת בהיבט של התרעות נכונות.

כמו כן, ערך גודל מסגרת שנבחר, 95 פיקסלים, נבחר לאחר מספר בדיקות זיהוי אובייקטים שהראה כי רוחב מסגרת סטנדרטית לבן אם הינה בין 65-75 פיקסלים, וממספר דגימות של איחוד אובייקטים הערך הממוצע לרוחב מסגרת הינו בין 95-110 פיקסלים.

על פי תרשים הזרימה לעיל, לאחר שכלל האובייקטים נבדקו, המערכת תעבור לבחון את הפריים הבא כאשר כל הוקטורים יתאפסו.

#### 4. מערך בדיקות סופיות

##### 4.1 הקדמה

קוד המערכת נכתב בשפת C++ ובשימוש בספריות של OpenCV המתמקדות בכל נושא עיבוד תמונה ווידאו. בבדיקות שיתוארו בהמשך נבחנו מספר פרמטרים על מנת לקבל את התוצאה הסופית הטובה ביותר תוך התחשבות במספר פרמטרים כמו: זמני ריצת הקוד, רמת דיוק בגילוי האובייקטים תוך חישוב אחוזי הצלחה ואחוזי התרעות שווא בכל סט הגדרות. כמו כן, בוצעו מספר בדיקות על מנת להביא את המערכת לזמני תגובה קצרים אשר מתאימים למערכת התרעה ב- Real Time.

##### 4.2 מערך הבדיקות

פרק זה מכיל מספר בדיקות עיקריות בשלושה תחומים, החלק הראשון מתעסק במציאת סט הגדרות מתאים למערכת בפונקציות הקוד השונות, החלק השני יתאר את ניסיון מימוש המערכת בזמני תגובה קצרים ואילו בחלק האחרון תוצג יכולת המערכת בהתמודדות עם סטעי וידאו בעלי מאפיינים דומים למאפייני ריב.

##### 4.2.1 כלי הבדיקות ותיאורן

סטעי הווידאו הנכנס למערכת הינו סרטון אשר נלקח מצילומי מערכות אבטחה שבו מתרחשת דקירת חייל על רקע לאומני. סרטון צבעוני באורך של 4 שניות, 120 פריימים. כלל הבדיקות בוצעו על מחשב בעל מעבד Intel Core i3 2.4GHz עם זיכרון RAM של 4GB. חישוב השגיאה הכללי מתבצע על ידי בדיקת מספר הפריימים שבהם היה זיהוי של האנשים הדומיננטיים בסרטון על ידי המערכת לעומת כלל הפריימים של הסרטון. Missing true alarm – בדיקה שמתבצעת לאחר סיום ריצת המערכת, ובה כל פריים נבדק האם היה צורך להתריע בפריים הזה והמערכת לא התריעה. False alarms - בדיקה שמתבצעת לאחר סיום ריצת המערכת, ובה כל פריים נבדק האם הייתה בו התרעה ללא צורך.

##### 4.2.2 בדיקות אופטימיזציה למערכת:

בחלק זה של הבדיקות, המטרה הינה למצוא את סט ההגדרות המתאים ביותר לפונקצית גילוי האנשים. הפרמטרים העיקריים שנבדקו הם:

Scale -I Padding, Winstride

בדיקה #1				
Input	Output	Winstride	Padding	Scale
ref_knife.mpg	test1_knife.mpg	(4,4)	(16,16)	1.3
False alarms	Missing true alarm	אחוז הצלחה : 52% = 63/120		שגיאות
12/120=10%	45/120=37%			

כחמש דקות				זמן ריצה
אחוז הצלחת המערכת נמוך מאוד וכן אחוז missing true alarm גבוה מאוד, דבר שפוגע באמינות המערכת.				מסקנה עיקרית
בדיקה #2 – שינוי ערך פרמטר ה- scale				
Input	Output	winstride	Padding	Scale
ref_knife.mpg	Test2_knife.mpg	(4,4)	(16,16)	1.1
False alarms		Missing true alarm		שגיאות
10% =13/120		19% =23/120		
כתשע דקות				זמן ריצה
פרמטר scale בעל ערך נמוך יותר מה שגרם לאחוזי הצלחת המערכת לעלות אך גרם לעליה בזמן ריצת המערכת				מסקנה עיקרית
בדיקה #3 – שינוי ערך פרמטר ה- padding				
Input	Output	winstride	Padding	Scale
ref_knife.mpg	Test3_knife.mpg	(4,4)	(24, 24)	1.1
False alarms		Missing true alarm		שגיאות
8.3% =10/120		10% =12/120		
כשמונה דקות				זמן ריצה
שינוי ערך ה padding ושמירה על ערך פרמטר scale העלה את אחוז הצלחת המערכת והקטין את אחוז אי התרעה במקרה אמת משמעותית				מסקנה עיקרית
בדיקה #4 – שינוי ערך פרמטר ה- padding				
Input	Output	winstride	Padding	Scale
ref_knife.mpg	Test4_knife.mpg	(4,4)	(32,32)	1.1
False alarms		Missing true alarm		שגיאות
8.3% =10/120		8.3% =10/120		
כשמונה דקות				זמן ריצה
אחוזי הצלחת המערכת הינו הגבוה ביותר, אחוז אי התרעה במקרי אמת ירד והינו הנמוך ביותר				מסקנה עיקרית

### 4.2.3 מימוש ב- Real Time

אחד האתגרים במימוש מערכת לעיבוד תמונה ווידאו הינו לממשה לפעולה בזמן אמת, לפי תוצאות הבדיקות מהחלק הקודם, זמן הריצה הממוצע של המערכת היה כשמונה דקות – פרמטר שאינו נחשב לעיבוד בזמן אמת. בחלק הזה של הבדיקות התבצע ניסיון לממש את המערכת בזמני קרובים לזמן אמת על ידי כך שקוד המערכת לא יבדוק כל פריים בסרטון הכניסה אלא יפעיל את האלגוריתם המערכת באופן שונה לפי פירוט הבדיקות הבא.

הבדיקות הבאות התקיימו לפי סט הפרמטרים האחרון והסופי שהוצג בסעיף הקודם.

בדיקה 1 – הפעלת האלגוריתם על כל פריים עשירי				
Input	Output	winstride	Padding	Scale
ref_knife.mpg	Test5_knife.mpg	(4,4)	(32,32)	1.1
62 שניות				זמן ריצה
זמן הריצה נמוך מאוד אך חווית המשתמש נפגעה ואינה נוחה לצפייה				מסקנה עיקרית
בדיקה 2 – הפעלת האלגוריתם על כל פריים חמישי				
Input	Output	winstride	Padding	Scale
ref_knife.mpg	Test6_knife.mpg	(4,4)	(32,32)	1.1
114 שניות				זמן ריצה
זמן הריצה עלה יחסית לבדיקה הקודמת אך עדיין חווית המשתמש נפגעה ואינה נוחה לצפייה				מסקנה עיקרית

### 4.2.4 בדיקות אימות למערכת

על מנת לוודא כי המערכת והאלגוריתם פועלים באופן נכון, הוכנסו למערכת סרטוני בדיקה אשר אינם מכילים סצנות של ריב בין אנשים, כגון מצלמת אבטחה ממרכז קניות ומצלמת רחוב המתעדת אנשים מתחבקים.

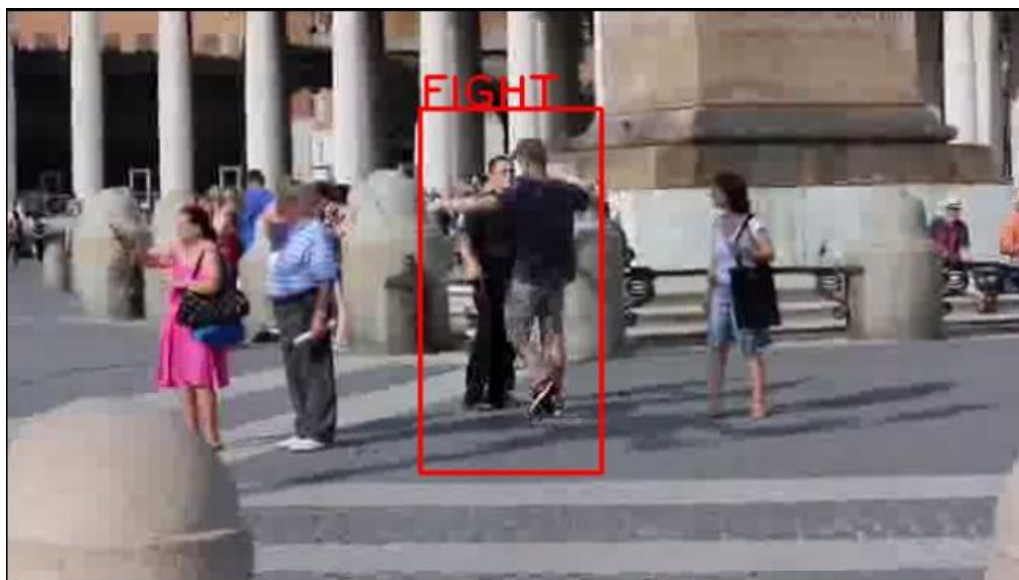
כפי שניתן לראות באיור 4.1, המערכת הצליחה להתמודד עם סרטון האבטחה ממרכז הקניות שכן המרחק בין האובייקטים היה יחסית גדול ולכן המערכת לא זיהתה היצמדות והיפרדות של אובייקטים כסצנה של ריב:





## איור 4.1 פעילות האלגוריתם במרכז קניות

בסרטון אחר שהוכנס למערכת, שבו מצולמת דרך עירונית ובה שני אנשים מתחבקים, המערכת כן מספקת התרעת ריב וזאת בשל העובדה כי האובייקטים היו קרובים מאוד אחד שני ולכן המערכת התריעה.

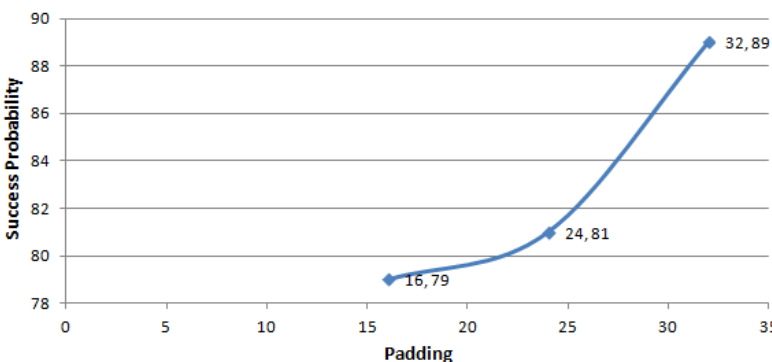


## איור 4.2 דוגמא להתרעת שווא

### 4.3 דיון בתוצאות

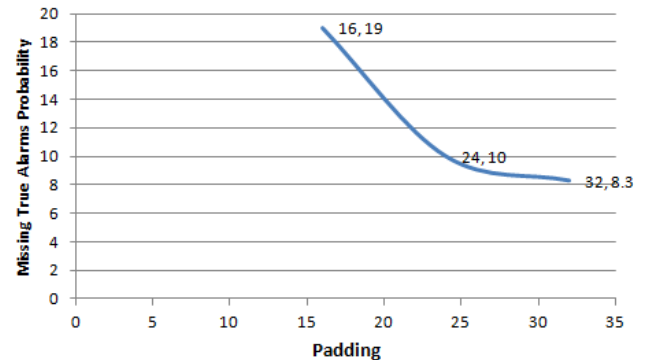
מתוך כלל הבדיקות שתוארו בסעיפים הקודמים ניתן להסיק מספק מסקנות על הפרמטרים השונים של המערכת ואופן השפעתם הישיר. המסקנות יוסקו מתוך הגרפים המסכמים הבאים:

Success Probability VS Padding



(a)

Missing True Alarms Probability VS Padding



(b)

### איור 4.3 עקומי סיכום בדיקות

ראשית, כלל התוצאות המתוארות בגרפים בוצעו כאשר ערך פרמטר scale הינו 1.1 וערך פרמטר winstride הינו (4,4).

מתוך הגרפים לעיל ניתן לראות כי שינוי פרמטר ה- padding גורם לשינוי מהותי באחוזי הצלחת המערכת. בפן הכללי, העלאת ערך פרמטר זה למקסימום האפשרי (32,32) סיפק את אחוז הצלחת המערכת לכ- 89%, ניתן לראות את מגמת עליית אחוזי ההצלחה כפונקציה של העלאת ערך זה.

כמו כן, בצורה דומה העלאת ערך ה- padding העלה את אמינות המערכת מבחינת התרעות כאשר אחוז פספוס התרעות אמת בסרטון הבדיקה הנכנס למערכת הציג מגמת ירידה, עד לכ- 8.3% מכלל הסרטון כתוצאה מהעלאת ערך פרמטר ה- padding.

יחד עם זאת, למרות שערך גבוה של פרמטר ה- padding מספק אחוזי הצלחה גבוהים, אחד החסרונות של כך הוא שבמוצא המערכת קיימים מספר זיהויי שווא בפריים שאינם קיימים בפריים מקביל עם ערך נמוך יותר לפרמטר זה, דבר אשר מעכב במידה את זמני תגובת המערכת וזאת בשל הצורך לאמת את הזיהוי השגוי כאובייקט או לא ולבחון האם יש בו צורך. באיור 4.4 מתואר פריים זהה במוצא המערכת עם פרמטרים זהים מלבד פרמטר ה- padding, ניתן לראות את זיהוי השווא הקיים בצדדי הפריים במוצא המערכת עם הערך הגבוה יותר.



Frame 70



Frame 70

#### איור 4.4 השוואה בין ערכי padding שונים באותו פריים

בנוסף, ערך פרמטר scale נבחר ל-1.1, על פי התוצאות המובאות בתחילת הפרק, סיפק רמת דיוק והבחנה גבוהה יותר למערכת לעומת ערך אחר שנבדק 1.3, ניתן לראות המחשה באיור 4.5, כי בהרצת המערכת עם ערך שונה לפרמטר זה כאשר כל שאר הפרמטרים זהים קיים אי זיהוי אובייקטים חשובים בפריים כאשר ערך ה- scale גבוה. עובדה זו מתאימה לרקע התיאורטי שהוצג בפרק השלישי בו פרמטר זה אחראי על אופן חקירת אלגוריתם HOG של התמונה בכניסה למערכת, לאיזו רמת רזולוציה האלגוריתם יעמיק בבדיקה, וכן, כאשר פרמטר זה הינו נמוך רמת הבדיקה תהיה מעמיקה ויסודית יותר. יחד עם זאת, ערך נמוך לפרמטר זה העלה את זמן ריצת המערכת מכ-חמש דקות לסביבות 8 וחצי דקות.



Frame 35



Frame 35

#### איור 4.5 השוואה בין ערכי scale שונים באותו פריים

אופן פעולה תקין של המערכת תוך זיהוי ריב בין אנשים יראה כך :



Frame 22



Frame 85

איור 4.6 המחשת פעילות תקינה של המערכת

## 5. סיכום

במציאות הביטחונית המורכבת בימים אלו רמת החשיבות של השימוש בצילומי אבטחה שבעזרתם ניתן לתחקר אירועים ביטחוניים ואף למנוע אותם.

בתחום עיבוד התמונה קיימות מגוון שיטות לזיהוי אובייקטים ועקיבה אחריהם כפי שהוצג בתחילת הספר, וכן אלגוריתם לסיווג אובייקטים וסיווג סצנות כהתרחשות חשודה או לא.

כל אלגוריתם שהוצג ניסה להתמודד עם מספר בעיות הקיימות בעיבוד תמונה כגון, הפרעות של שיוניי אור וצל וסיווג אובייקטים לא נכון בסצנה כלשהי.

בפרויקט זה מטרת המערכת שמומשה הייתה להתריע לקבל צילומי אבטחה ממקומות שונים ולהתריע על גבי המסך כאשר מתרחשת מריבה בין אנשים.

המערכת שתוארה בספר זה חיברה בין אלגוריתמים קיימים בעולם כדוגמת HOG אשר מבצע סריקה על כל פריים הנכנס למערכת ומחפש את השינויים ומחשב גרדיאנטים בתמונה, כאשר הסריקה מתבצעת ברמות שונות והתמונה מתחלקת לבלוקים ולתאים כדי להפיק את המיטב מסריקת התמונה, בהמשך כלל התוצאות יגיעו למנגנון האימות, SVM, אשר יחליט האם המידע שהתקבל מאלגוריתם HOG הינו בן אדם או לא. בהתאם להחלטת מנגנון האימות הופעלו תנאים לבדיקת ההתרחשות בתמונה האם יש להתריע שמתקיימת מריבה או לא.

במסגרת הבדיקות הוכח כי ישנם פרמטרים חשובים אשר השפיעו על התוצאות באופן ישיר כדוגמת שילוב ערכי ה-scale וה-padding אשר קבעו באיזו רזולוציה של התמונה האלגוריתם יעבוד, ומה המרווח בין הסריקות בפריים, ככל שערך ה-scale היה נמוך יותר וערך ה-padding היה גבוה יותר, התקבלו תוצאות איכותיות יותר מבחינת אחוזי שגיאה נמוכים, ואחוזי אי התרעה במקרי אמת נמוכים, אך החיסרון הינו זמני עיבוד ארוכים של המערכת.

במסגרת הבדיקות היה ניסיון לממש את המערכת בזמני תגובה קצרים, אך התוצאות שהתקבלו לא היו דיי איכותיות מבחינת המשתמש, על מנת לממש את המערכת בזמן אמת יש צורך בחומרה ייעודית חזקה כדוגמת שרתים בעלי יכולות עיבוד וזיכרון גבוהות.

מבחינת תוצאות הבדיקות עולה כי המערכת פעלה היטב בסרטון שהתרחש בו סצנת ריב, אך יחד עם זאת המערכת לוקה בחסר בסרטונים המתארים התקרבות של אנשים בפעילות רגילה, כדוגמת חיבוק ומבצעת התרעה שגויה על המסך.

על מנת למקסם את יכולות המערכת הטכנולוגיה היום בעולם הראייה הממוחשבת פונה בעיקר לתחום הבינה המלאכותית ומנגנוני המערכות ה"לומדות" האוטומטיות אשר בונות מאגרי מידע (DB) שמרכזים ומתייגים בו כל התרחשות שקרתה בין אם חשודה או לאו, ומתוך כל מבצעות את הפעולות שאופיינו לכל מערכת על מנת לעמוד בדרישות הלקוחות.

### מקורות

- [1] Sandesh Patil and Kiran Talele, "Suspicious Movement Detection and Tracking based on Color Histogram", International Conference on Communication, Information & Computing Technology, Mumbai, India, 2015.
- [2] Mohannad Elhamod and Martin D. Levine, "Automated Real-Time Detection of Potentially Suspicious Behavior in Public Transport Areas", *IEEE Transaction On Intelligent Transportation Systems*, vol.. 14, NO.2, 2013.
- [3] Jong Sun Kim, Dong Hae Yeom and Young Hoon Joo, "Fast and Robust Algorithm of Tracking Multiple Moving Objects for Intelligent Video Surveillance Systems", *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 3, 2011.
- [4] Rafael C. Gonzalez, *Digital Image Processing Using MATLAB*, pp.404-406, 2002.
- [5] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection", Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.



## נספח א' – קוד המערכת

```
#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
#include<fstream>
#include <time.h>

#define CLOCKS_PER_SEC 1000;
using namespace std;
using namespace cv;

int main(int argc, char *argv[])
{
    clock_t tStart=clock();
    cv::Mat frame;

    // input and output movies directions for the system
    char*input = "D:\\study\\final project\\demos\\ref_knife.mpg";
    char*outputv= "D:\\study\\final project\\demos\\test_new.mpg";
    std::cerr << "opening " << input << std::endl;
    cv::VideoCapture cap(input);
    // cv::BackgroundSubtractorMOG2 bg;

    // saving the output video parameters
    cv::VideoWriter output;
    //int ex = static_cast<int>(cap.get(CV_CAP_PROP_FOURCC));
    int ex = CV_FOURCC('P','I','M','1');
    cv::Size size = cv::Size((int) cap.get(CV_CAP_PROP_FRAME_WIDTH),
                             (int) cap.get(CV_CAP_PROP_FRAME_HEIGHT));
    std::cerr << "saving to " << outputv << std::endl;
    output.open(outputv, ex, cap.get(CV_CAP_PROP_FPS), size, true);

    std::vector<std::vector<cv::Point> > contours;

    cv::namedWindow("Frame");
    cv::namedWindow("Fore");
    cv::namedWindow("Background");

    // location of log file
    ofstream file( "D:\\study\\final project\\demos\\new.txt" );
    cv::HOGDescriptor hog;

    // define the SVM classifier
    hog.setSVMDetector(cv::HOGDescriptor::getDefaultPeopleDetector());
    int foundCounter = 0, frameCounter = 0;
    //CvLatentSvmDetector* cvLoadLatentSvmDetector(const char* filename);

    for(;;)
    {
        frameCounter++;

        cap >> frame;
        //if ((frameCounter % 3) ==0)    {
```

```

        // HOG algorithm
std::vector<cv::Rect> found, found_filtered;
hog.detectMultiScale(frame, found, 0, cv::Size(4,4), cv::Size(32,32), 1.1, 2);

for (int i = 0; i < found.size(); ++i) {
    cv::Rect r = found[i];
    int j = 0;
    for (; j < found.size(); ++j) {
        if (j != i && (r & found[j]) == r) {
            break;
        }
    }
    if (j == found.size()) {
        found_filtered.push_back(r);
    }
}

    if(found_filtered.size()>0)
        foundCounter++;
    std::vector<int> a (found_filtered.size());
    std::vector<int> b (found_filtered.size());

    // check the scine if there are fight on it
for (int i = 0; i < found_filtered.size(); ++i) {
    cv::Rect r = found_filtered[i];
        a.at(i) = r.height;
        b.at(i) = r.width;
        if (r.x>158){
            if (i>=1 && (((std::abs(b[i] - b[i-1])<30)) || b[i]>=94))
                cv::rectangle(frame, r.tl(), r.br(),
cv::Scalar(0,0,255), 2);

                cv::putText(frame,"FIGHT",r.tl(),FONT_HERSHEY_PLAIN,2,Scalar(0,0,255),2);
            }
            else cv::rectangle(frame, r.tl(), r.br(),
cv::Scalar(255,0,0), 1);
        }
    }
    output << frame;

cv::resize(frame, frame, cv::Size(1280, 720));
cv::imshow("Frame", frame);

    // calculate the runing time of the system
double time;
time = (clock() - tStart)/ CLOCKS_PER_SEC;

    // calculate the success rate of the system
int foundPrc = 100*foundCounter/frameCounter;

//    printing runing time and success rate
cout << "Found in " << foundCounter << " out of " << frameCounter << " frames." <<
endl;

```



```
file << "Found in " << foundCounter << " out of " << frameCounter << " frames and the  
error is" << foundPrc <<"Running Time " << time << endl;  
  
if(cv::waitKey(30) >= 0) break;  
  
}  
  
return 0;  
}
```