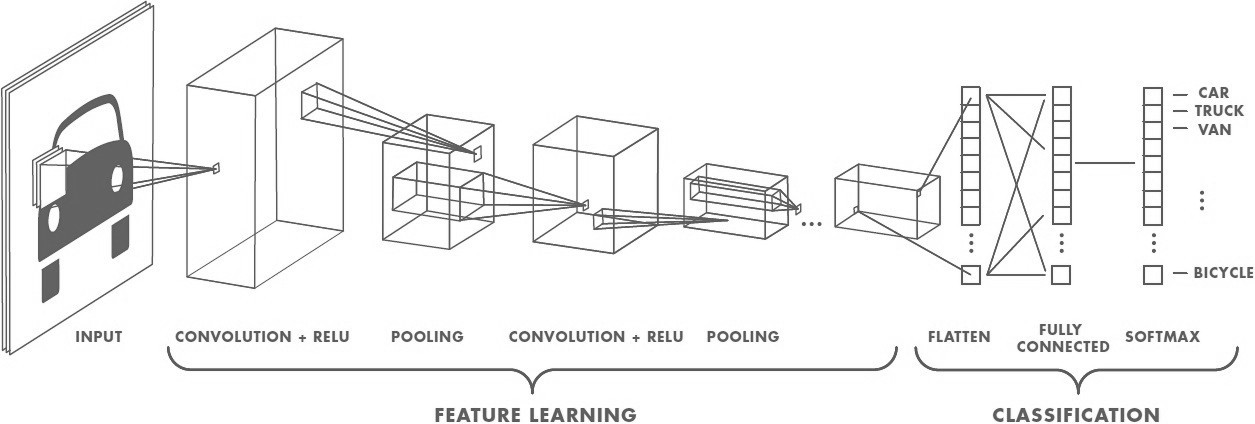
******

***הפקולטה להנדסה***

**רשת נוירונים עמוקה לזיהוי הולכי רגל במערכת נהיגה אוטונומית**

**Deep Neural Network for Pedestrian Detection for Autonomous Driving System**



**מאת:**

**אסף קסלסי , ת.ז 301467098**

**מנחה:**

**פרופסור יוסף בן-עזרא**

תוכן עניינים

[נספחים 4](#_Toc43757995)

[רשימת נוסחאות: 4](#_Toc43757996)

[ראשי תיבות 5](#_Toc43757997)

[רשימת איורים: 6](#_Toc43757998)

[1. מבוא 7](#_Toc43757999)

[1.1. מוטיבציה 7](#_Toc43758000)

[1.2 אבני דרך בציר הזמן של רשתות נוירונים ולמידת מכונה 8](#_Toc43758001)

[1.3. רקע לרשתות נוירונים 9](#_Toc43758002)

[1.3.1 הגדרת רשת נוירונים 9](#_Toc43758003)

[1.3.2 כיצד רשת נוירונים דומה למוח ? 9](#_Toc43758004)

[1.4. רשתות נוירונים מלאכותיות 10](#_Toc43758005)

[1.4.1 ייצוג 10](#_Toc43758006)

[1.5 סוגים של פונקציות הפעלה 11](#_Toc43758007)

[1.5.1. פונקצית סף function) Threshold ) 11](#_Toc43758008)

[1.5.2 פונקצית סיגמוייד ( function (Sigmoid 11](#_Toc43758009)

[1.5.3 פונקצית המיישר (ReLU function) 12](#_Toc43758010)

[1.5.4. פונקצית מיישר דולפת (Leaky ReLU function) 13](#_Toc43758011)

[1.5.5. פונקצית מיישר פרמטרית (Parametric ReLU function) 13](#_Toc43758012)

[1.6. ארכיטקטורות רשת 14](#_Toc43758013)

[6.1. 1. רשתות חד-שכבתיות ללא משוב 14](#_Toc43758014)

[6.2. 1. רשתות רב-שכבתיות ללא משוב 15](#_Toc43758015)

[1.6.3 רשתות חוזרות (Recurrent) 16](#_Toc43758016)

[7. 1. סוגי בעיות 17](#_Toc43758017)

[7.1. 17](#_Toc43758018)

[7.2. 1. בעיות סיווג 17](#_Toc43758019)

[7.3. 1. בעיות סיווג מרובה-מחלקות: 17](#_Toc43758020)

[8. 1. למידה 18](#_Toc43758021)

[1.8.1 למידה מפוקחת (Supervised learning) 18](#_Toc43758022)

[9. 1. פונקציית העלות 19](#_Toc43758023)

[10. 1. רשת קונבולוציה 19](#_Toc43758024)

[11. 1. מישקול 20](#_Toc43758025)

[1.11.1 מישקול והטיה 20](#_Toc43758026)

[1.11.2 אתחול משקולות 20](#_Toc43758027)

[12. 1. סגמנטציה 21](#_Toc43758028)

[13. 1. זיהוי הולכי רגל 22](#_Toc43758029)

[אתגרים בזיהוי הולכים רגל: 22](#_Toc43758030)

[1.13.1 Holistic detection 22](#_Toc43758031)

[1.13.2 Patch-based detection 22](#_Toc43758032)

[1.13.3 Motion-based detection 22](#_Toc43758033)

[1.13.4 multiple cameras- based detection 23](#_Toc43758034)

[1.13.5 data-driven CNN-Unet based approach 23](#_Toc43758035)

[2. ניסוי 24](#_Toc43758036)

[2.1 ארכיטקטורה ואימון: 24](#_Toc43758037)

[2.2 אימון הרשת – זמן כולל ופלטפורמת המערכת. 25](#_Toc43758038)

[כמות פרמטרים הכוללת ברשת 27](file:///C:\Users\Assaf\Desktop\פרוייקט_גמר_אסף\AsafKaslassy_DeepLearning\Project%20book_firstDraft__22-6.docx#_Toc43758039)

[ארכיטקטורת הרשת: 28](#_Toc43758040)

[Over Fitting ודרכים למניעה 29](#_Toc43758041)

[Batch normalization.2 29](#_Toc43758042)

[Dropout.3 30](#_Toc43758043)

[MaxPooling2d.4 31](#_Toc43758044)

[Conv2d 32](#_Toc43758045)

[Back-propagation 32](#_Toc43758046)

[תאור הארכיטקטורה של הרשת: 35](#_Toc43758047)

[3. סיכום 37](#_Toc43758048)

[הצעות להמשך עבודה בנושא וייעול המערכת: 38](#_Toc43758049)

[ביבליוגרפיה 39](#_Toc43758050)

# **נספחים**

## רשימת נוסחאות:

## ראשי תיבות

AI - Artificial intelligence

CNN - Convolutional Neural Network

convNet - Convolutional Neural Network

DNN - Deep Neural Network

ReLU - Rectified Linear Unit

FC - Fully Connected

FCN - Fully Convolutional Network

COCO- Common Objects in Context

ADAS - Advanced driver-assistance systems

HOG - histogram of oriented gradients

ISM - Implicit Shape Mode

POM - Probability Occupancy Map

LR - Learning Rate

BS- Batch Size

DP - Dropout

## רשימת איורים:

איור 1 - אבני דרך בציר הזמן של רשתות נוירונים ולמידת מכונה

איור 2 - השוואה בין רשת בגוף האדם לרשת נוירונים מלאכותית

איור 3 - מודל של נוירון

איור 4 - גרף פונקציית סף

איור 5 - גרף פונקציית סיגמוייד

איור 6 - גרף פונקציית המיישר

איור 7 - גרף פונקציית מיישר דולפת

איור 8 - - גרף פונקציית מיישר פרמטרית

איור 9- - רשת חד-שכבתית ללא משוב

איור 10 - רשת רב-שכבתית ללא משוב

איור 11 - רשתות חוזרות

איור 12 - רשת רב שכבתית לסיווג מרובה מחלקות

איור 13 - רשתות קונבולוציה

איור 14 -

איור 15 -

איור 16 -

איור 17 -

איור 18 -

איור 19 -

איור 20 -

איור 21-

איור 22 -

# 1. מבוא

למידה חישובית (Machine Learning ) הינה תחום במדע החוקר מצב בו אלגוריתמים יכולים "ללמוד" מה נכון לעשות ומה לא נכון לעשות מתוך אוסף נתון שלdata - מידע. למידה חישובית היא כלי חשוב בתחום הAI . אלגוריתמים אלו משתמשים במודל המבוסס עלinput אשר לפיו הם חוזים או מחליטים. זה כמובן שונה מאלגוריתמים שלהם אין את האפשרות ללמוד מנתונים אלא רק עובדים על ידי ביצוע של סט פקודות ברורות ומוגדרות מלכתחילה.

טום מ. מיטשל בסיפרו *Machine Learning* הגדיר את למידת המכונה באופן הבא:

*"תוכנית מחשב תיקרא לומדת מניסיון E ביחס למחלקת משימות T ומדד ביצועים P , אם הביצועים של משימות ב- T בהתאם למדד P ,משתפרים עם הניסיון E.*  [***[1]***](#_ביבליוגרפיה)

## 1.1. מוטיבציה

למרות שתחום הלמידה החישובית עבר שינוי בתחומים מסוימים, אפליקציות רבות מבוססות רשתות נוירונים לרוב אינן מצליחות להביא את התוצאות הצפויות. [***[2]***](#_ביבליוגרפיה) הסיבות לכך הן רבות: היעדר נתונים (מתאימים), חוסר גישה לנתונים, הטיית נתונים, בעיות פרטיות, משימות ואלגוריתמים שנבחרו בצורה לא טובה, כלים שגויים, חוסר משאבים ובעיות בהערכה.

ניסיונות להשתמש בלימוד מכונות בבריאות עם מערכת IBM Watson לא הצליחו לספק תוצאות טובות גם לאחר שנים של מחקר ומיליארדים של כסף שהושקעו[***[3]***](#_ביבליוגרפיה), מקרה קיצוני אפילו הביא למוות.

מותה של איליין הרצברג (18 במרץ 2018) היה המקרה המוקלט הראשון של הולך רגל הרוג בו הייתה מעורבת מכונית בנהיגה עצמית (אוטונומית), הרצברג דחפה את אופניה לאורך כביש בן ארבעה נתיבים בטמפה, אריזונה, ארצות הברית, כאשר נפגעה על ידי רכב מבחן אוטונומי של Uber, שפעל במצב של נהיגה עצמית עם נהג גיבוי לבטיחות ,אדם שישב במושב הנהיגה.

מייקל רמזי, מומחה לתחום נהיגה אוטונומית, איפיין את הסרטון המוצג כמראה של "כישלון מוחלט של המערכת בזיהוי אדם שנראה בבירור לאורך זמן לא מבוטל במספר פריימים.

הטלמטריה המוקלטת הראתה שהמערכת גילתה את הרצברג שש שניות לפני הפגיעה וסיווגה אותה תחילה כאובייקט לא ידוע, אחר כך כרכב, ולבסוף כאופניים, כשלכל אחד מהם היה מסלול חזוי אחר לפי היגיון האוטונומיה המתוכנת ברכב. [4]

יותר מ -6,000 הולכי רגל ו -800 רוכבי אופניים נהרגים מדי שנה בארה"ב בתאונות דרכים. יותר מ 270,000 הולכי רגל נהרגים מדי שנה בעולם

## 1.2 אבני דרך בציר הזמן של רשתות נוירונים ולמידת מכונה

***איור 1***

## 1.3. רקע לרשתות נוירונים

### 1.3.1 הגדרת רשת נוירונים

מחקרים ומאמרים רבים אודות שימוש ברשתות נוירונים לחישוב ופתרון בעיות בעזרת מחשב נבעו מההבנה שהמוח האנושי מבצע חישובים בצורה שונה מהדרך בה מתבצעים חישובים ע"י מחשב דיגיטלי קונבנציונלי. המוח מאוד מורכב , עובד בצורה אי-ליניארית ומקבילית. יכול לארגן את הנוירונים , בצורה ייחודית המאפשרת זיהוי תבניות, שליפה מזיכרון, יכולות מוטוריות וכו' במהירות גדולה בכמה סדרי גדול מהמחשב המהיר ביותר כיום.

artificial neural networks בנויות כך שהן יכולות "ללמוד" מניסיון העבר, בכל רשת יש הרבה מאוד קשרים בין נוירונים מלאכותיים (או יחידות עיבוד) שמשתנים לאורך חיי הרשת כדי להפיק ביצועים טובים וחישוב יעיל.

### 1.3.2 כיצד רשת נוירונים דומה למוח ?

רשת נוירונים מלאכותית דומה למוח בשני אופנים עיקריים:

רכישת ידע באמצעות תהליך למידה ואחסון מידע באמצעות חיבורים בין נוירונים (משקלים סינפטיים).

באמצעות אלגוריתם למידה, משתנים משקלי הרשת בצורה איטרטיבית בכדי להגיע למטרה שדרשנו מהרשת.

הקשרים עצמם קבועים אך המשקל של כל קשר, או החוזק שלו, משתנה (תהליך דומה למוח ).

קיימות גם רשתות שיכולות לשנות את הארכיטקטורה והמבנה של עצמן על ידי יצירת קשרים חדשים

)תהליך נוסף הדומה למוח).[*[5]*](#_ביבליוגרפיה)

## 1.4. רשתות נוירונים מלאכותיות

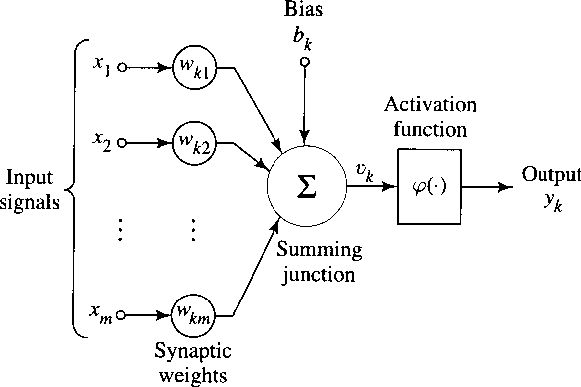
***איור* 2**

### 1.4.1 ייצוג

הנוירון הוא יחידת עיבוד המידע הבסיסית הנחוצה לתפקוד רשת הנוירונים. באיור ניתן לראות מודל של נוירון, היוצר את הבסיס לעיצוב רשת נוירונים מלאכותית. ניתן לזהות 4 אלמנטים בסיסיים במודל

המוצג:

1. סט של סינפסות אשר כל אחת מהן מאופיינת במשקל משלה. כלומר אות כלשהו ,xm המגיע בתור קלט לסינפסה j המחוברת לנוירון ,k מוכפל במשקל הסינפטי wkm
2. מחבר (adder) של אותות ה input
3. פונקצית הפעלה function activation הגורמת שפונקצית הinput של הנוירון יכולה להיות אי-ליניארית.
4. אות הטיה (bias) מסומן בסימון bk מטרת אות זה היא הגברה או הנמכה של האות המסוכם טרם כניסתו לפונקצית ההפעלה.



***איור* 3**

## 1.5 סוגים של פונקציות הפעלה

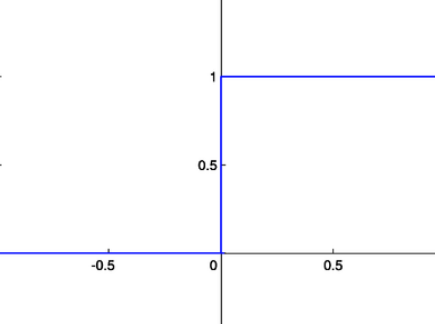
פונקציות ההפעלה מגדירה את הפלט של הנוירון כתלות בפוטנציאל ההפעלה שלו . ישנם סוגים

רבים של פונקציות הפעלה, אפרט על הבסיסים ועל פונקצית ההפעלה שבחרתי בפרויקט.

### 1.5.1. פונקצית סף function) Threshold )

פונקציה זו עוזרת בהחלטה האם יש להפעיל את הנוירון או לא על סמך הערך מהטרנספורמציה הליניארית. במילים אחרות, אם הקלט לפונקצית ההפעלה גדול מסף מסוים, אז הנוירון מופעל, אחרת הוא מבוטל, כלומר הפלט שלו לא נחשב לשכבה הסמויה הבאה.

מוגדרת בצורה הבאה:



def threshold \_function(x):

if x<0:

return 0

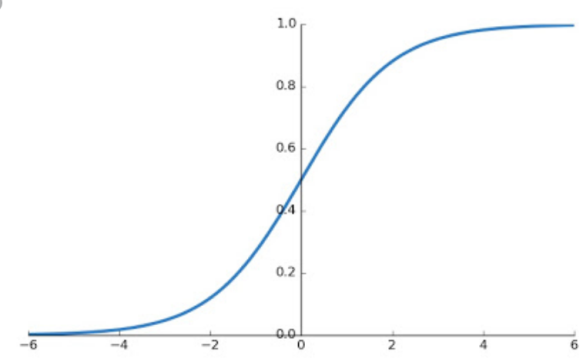
else:

return 1

***איור* 4**

### 1.5.2 פונקצית סיגמוייד ( function (Sigmoid

הגרף שלה דומה וויזואלית לאות ,S אחת הפונקציות הנפוצות ביותר בבניית רשתות. מוגדרת כפונקציה עולה-ממש, ואחת התכונות המיוחדות שלה שבתחום מוגדר אפשר להגיד שמתפקדת כפונקציה ליניארית ובאזורים אחרים כפונקציה אי-ליניארית. (אחד ההבדלים בין פונקציית סיגמוייד לבין פונקצית אקטיבציה מסוג "threshold" היא תכונת הגזירות שכן סיגמוייד גזירה ופונקצית סף לא גזירה.(

****

def sigmoid\_function(x):

z = (1/(1 + np.exp(-x)))

return z

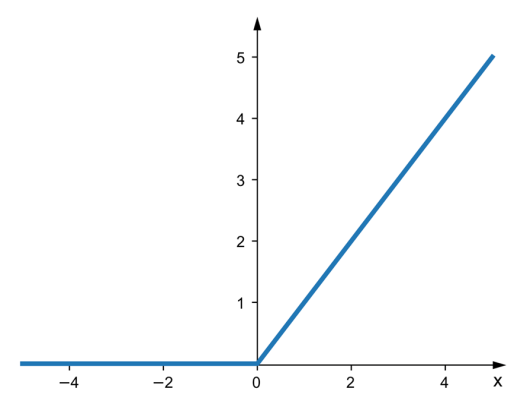
***איור* 5**

### 1.5.3 פונקצית המיישר (ReLU function)

### 

פונקצית הפעלה המוגדרת כהחלק החיובי בארגומנט שלה , כאשר x הוא הקלט לנוירון, ידועה גם כפונקצית רמפה (Ramp function) והיא מקבילה לתיקון חצי גל בעולם הנדסת החשמל.

אחד היתרונות של פונקציה זו הוא היכולת להעלים את כל החלק השלילי ביעילות על ידי הגדרתו כאפס - מה שמגביר את תכונות האי-ליניאריות שפירטתי בפרק הקודם.



def relu\_function(x):

if x<0:

return 0

else:

return x

***איור 6***

למרות מספר חסרונות שאפרט בהמשך, ReLU (נכון לשנת 2020) היא פונקצית ההפעלה הפופולארית ביותר עבור רשתות עצביות עמוקות.

חסרונות ReLU והסיבה שרבים משתמשים בגרסאות שונות שלה  
לא ניתנת להבחנה באפס, אך עם זאת, ניתן להבחין בה בכל מקום אחר, וניתן לבחור באופן שרירותי את הערך של הנגזרת באפס להיות 0 או 1, ללא גבול,

בעיית ReLU גוססת: לעיתים ניתן לדחוף נוירונים של ReLU למצבים בהם הם הופכים לא פעילים. במצב זה הנוירון נתקע במצב בלתי פעיל כל הזמן ו"מת ". זוהי בעיית Vanishing gradient problem . בחלק מהמקרים, מספר גדול של נוירונים ברשת יכול להיתקע במצבים מתים, ובכך להפחית למעשה את יכולת המודל. בעיה זו מתעוררת בדרך כלל כאשר קצב הלמידה מוגדר גבוה מדי. במקום זאת ניתן להקל על ידי שימוש ב- ReLUs דולפים,

המקצים שיפוע חיובי קטן עבור x < 0

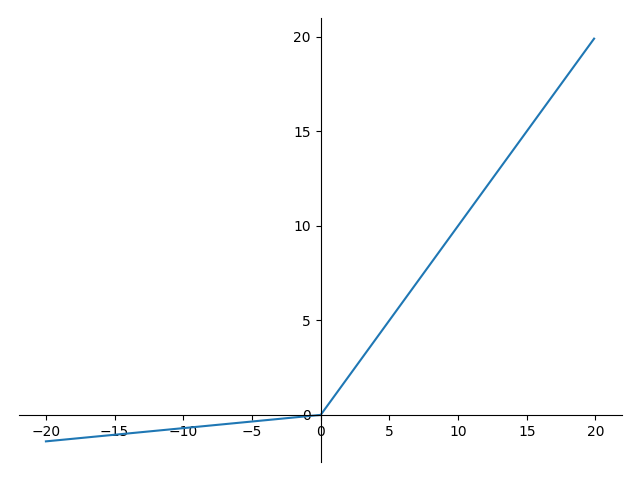
11

### 1.5.4. פונקצית מיישר דולפת (Leaky ReLU function)

פונקצית הפעלה מאוד דומה לReLU הרגילה אך הבדל משמעותי קורה בערכים השליליים:

היא מאפשרת 0.01 של גרדיאנט חיובי כשהיחידה לא אקטיבית *.*[*[11]*](#_ביבליוגרפיה)

אחד היתרונות של פונקציה זו הוא היכולת להעלים את כל החלק השלילי ביעילות על ידי הגדרתו כאפס - מה שמגביר את תכונות האי-ליניאריות שפירטתי בפרק הקודם.



***איור 7***

*def leaky\_relu\_function(x):*

*if x<0:*

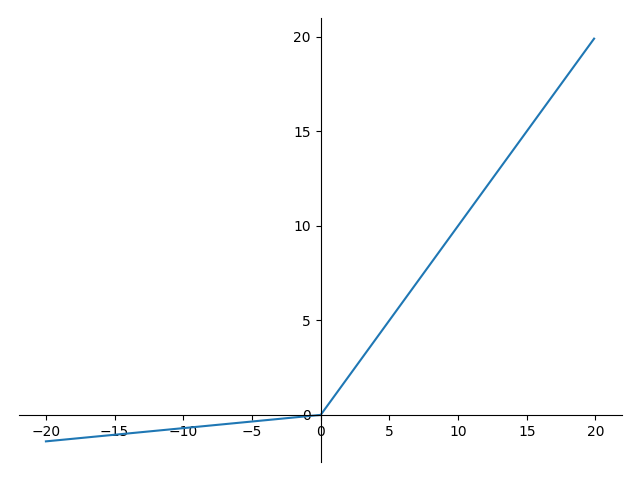
*return 0.01\*x*

*else:*

*return x*

### 1.5.5. פונקצית מיישר פרמטרית (Parametric ReLU function)

*פונקצית הפעלה מאוד דומה לReLU הרגילה אך לוקחת את הזליגה צעד אחד קדימה על ידי הגדרת כמות הזליגה בפרמטר שנלמד כמו שאר הפרמטרים של רשת הנוירונים* .[[12]](#_ביבליוגרפיה)

**

*def PReLU\_function(x):*

*if x<0:*

*return a\*x*

*else:*

*return x*

***איור 8***

*a\*v*

*v*

**1.5.6. פונקצית סופטמקס (Softmax function)**

*פונקצית Softmax מתוארת לעתים קרובות כשילוב של מספר sigmoids מרובים. ידוע כי sigmoid מחזיר ערכים בין 0 ל -1, שניתן להתייחס אליהם כאל הסתברות לנתונים השייכים למחלקה מסוימת. לכן משתמשים בסיגמואיד באופן נרחב בבעיות סיווג בינאריות. ניתן להשתמש בפונקציה softmax לבעיות סיווג רב-כיתתיות. (multiclass classification problems) פונקציה זו מחזירה את ההסתברות לדאטא מסויים השייך לכל מחלקה בנפרד . בעת בניית רשת לבעיה רב-כיתתית, בשכבת הפלט יהיו מספר נוירונים רבים כמספר הכיתות ביעד(target) לדוגמה, אם יש לך שלוש כיתות, יהיו שלושה נוירונים בשכבת הפלט*

*הביטוי*המתמטי של סופטמקס נראה כך:

*def softmax\_function(x):*

*z = np.exp(x)*

*z\_**soft = z/z.sum()*

*return z\_soft*

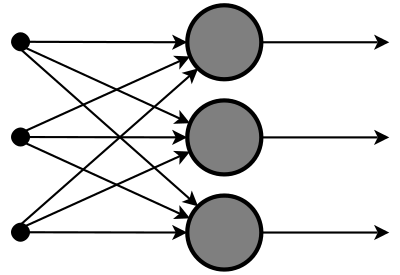
## 1.6. ארכיטקטורות רשת

### צורת הבנייה והגדרת הקשר בין נוירון לנוירון תלוי במספר גורמים- אחד מהם הוא האלגוריתם הלמידה בו הוחלט להשתמש לאימון הרשת.

באופן כללי, ישנם 3 סוגים בסיסיים של ארכיטקטורות רשת:

### 6.1. 1. רשתות חד-שכבתיות ללא משוב

רשת בעלת שכבות היא רשת שבה הנוירונים מסודרים בצורת שכבות קשורות. אם אנסה לפשט את התהליך בשביל ההבנה הצורה הכי פשוטה של רשת שכבות מכילה רק שכבה אחת, שכבת-קלט של מקור , שכבה זו מעבירה מידע לפלט של נוירונים , אבל חשוב לשים לב שברשתות מהסוג הזה המידע לא יכול לזרום לצד השני כלומר אין משוב. ובאנגלית נקראת גם רשת מזינה קדימה ( *Feed-Forward* )[***[5]***](#_ביבליוגרפיה) באיור 7 מוצגת רשת בעלת 3 מקורות ו3 נוירונים רשת כזאת נקראת גם "חד-שכבתית" מכוון שבנויה משכבת חישוב אחת בלבד– שכבת הפלט.



***איור 9***

***input***

***output***

***output***

***output***

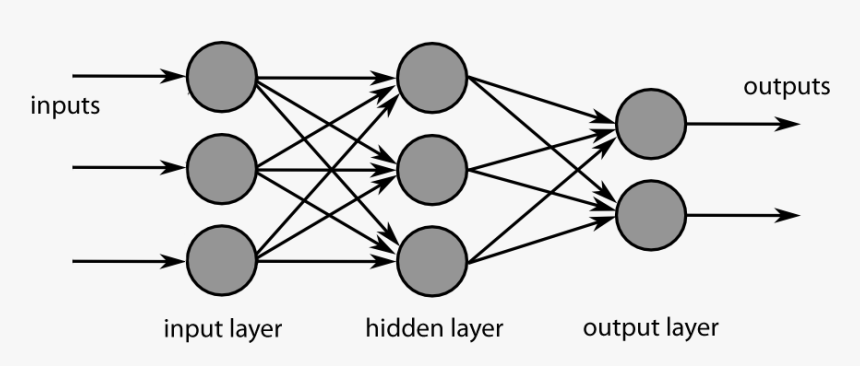
***input***

***input***

***output layer***

### 6.2. 1. רשתות רב-שכבתיות ללא משוב

רשתות מהסוג הזה מכילות שכבות חישובנוספות הנקראות שכבות חבויות או נסתרות ( *layers hidden)* לכן גם הנוירונים בשכבה הזו נקראים נוירונים חבויים *neurons)* (*hidden* . תפקיד השכבות החבויות הוא "להתערב" בין שכבת הקלט ושכבת הפלט. בעזרת הוספת שכבות חבויות הרשת מסוגלת להפיק סטטיסטיקות מסדר גבוה יותר. זו כמובן תכונה חשובה מאוד כששכבת הקלט מכילה הרבה נוירונים. אותות הפלט של הנוירונים בשכבה השנייה הופכים לאותות הפלט של השכבה השלישית, וכך הלאה. בדרך כלל, הנוירונים בכל שכבה מקבלים את אותות הקלט שלהם רק מהשכבה הקודמת להם ולכן זוהי רשת ללא משוב.   
אותות הפלט של השכבה האחרונה ,מהווים את התגובה הכוללת של הרשת לאותות הקלט אשר התקבלו בעזרת השכבה הראשונה. איור מדגים את הפריסה של רשת רב-שכבתית בעלת שכבה חבויה אחת. רשת זו מחוברת בצורה מלאה (*fully connected* ) כלומר כל נוירון מחובר לכל אחד מהנוירונים בשכבה הבאה. אם לרשת אין את התכונה הזו, היא נקראת רשת מחוברת-חלקית.



***output***

***input***

***input***

***input***

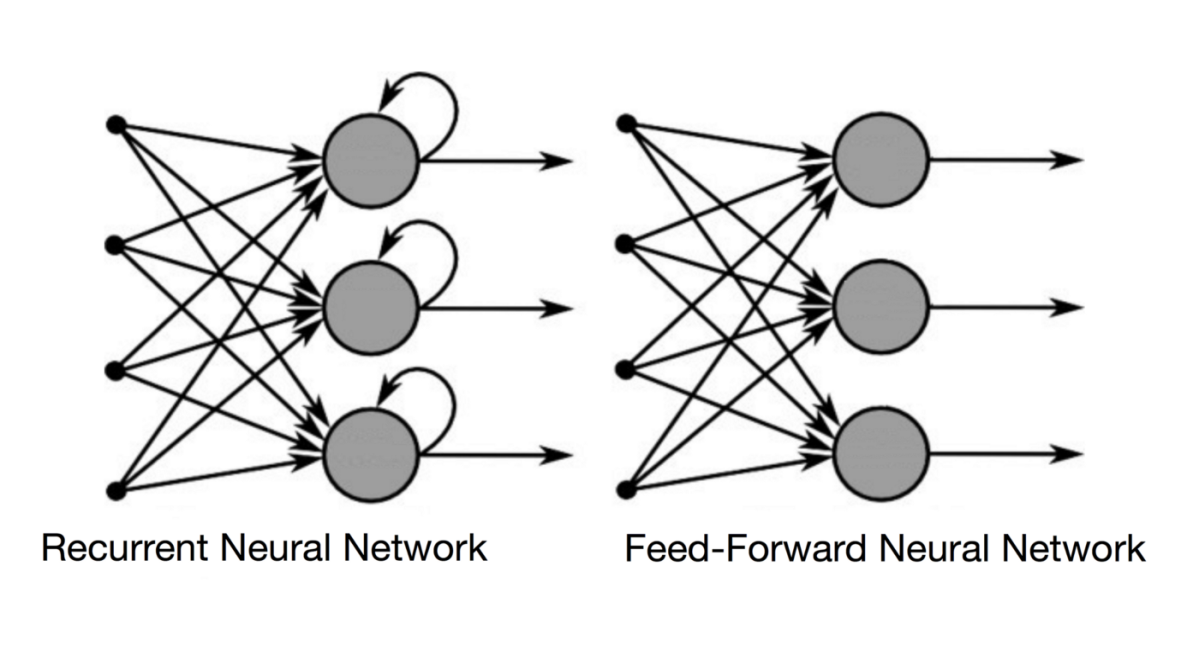
***output***

***input***

***איור 10***

### . 1.6.3 רשתות חוזרות (Recurrent)

רשות חוזרות שונות במיוחד מהרשתות חסרות המשוב בגלל העובדה שלהן יש לולאות משוב   
*( loops .( feedback* באיור 7 ניתן לראות רשת חוזרת. ברשת זו יש משוב-עצמי: הפלט של כל נוירון מועבר בתור קלט לעצמו. ניתן גם לתכנן רשת ללא משוב עצמי בה הפלט של כל נוירון מועבר לנוירון אחר באותה שכבה. לולאות משוב משפיעות מאוד על יכולות הלמידה של הרשת ועל ביצועיה. בלולאות משוב ניתן למצוא ענפים (קשרים סינפטיים) מיוחדים המכילים יחידות מעכבות מידע אשר מטרתן לעכב את האות לפני שהוא מגיע לנוירון הבא בד"כ נסמן אותן כמו ב , יחידות אלה תורמות לאופי האי-ליניארי של רשתות הנוירונים.



***איור 11***

***input***

***input***

***input***

***output***

***output***

***output***

***input***

***output layer***

## 7. 1. סוגי בעיות

רשתות נוירונים יכולות לפתור בקלות יחסית מספר רב של בעיות אני אסווג את סוגי הבעיות ל3 סוגים עיקריים:

7.1. **1.** בעיות התאמה

בעיות מסוג "התאמה" יש לבסס החלטה מושכלת , בהינתן הקלט, מהו המספר שהכי "מתאים" לקלט (המשמעות של מספר זה תלויה בבעיה המדוברת), לדוגמה, בעיית מחיר רכב: נתונים מאפיינים של רכב כמו למשל שנה , מודל, יצרן , מספר יד , נפח תא מטען, נצילות דלק וכו' והפלט הוא המחיר שבו יימכר הרכב הספציפי הזה.

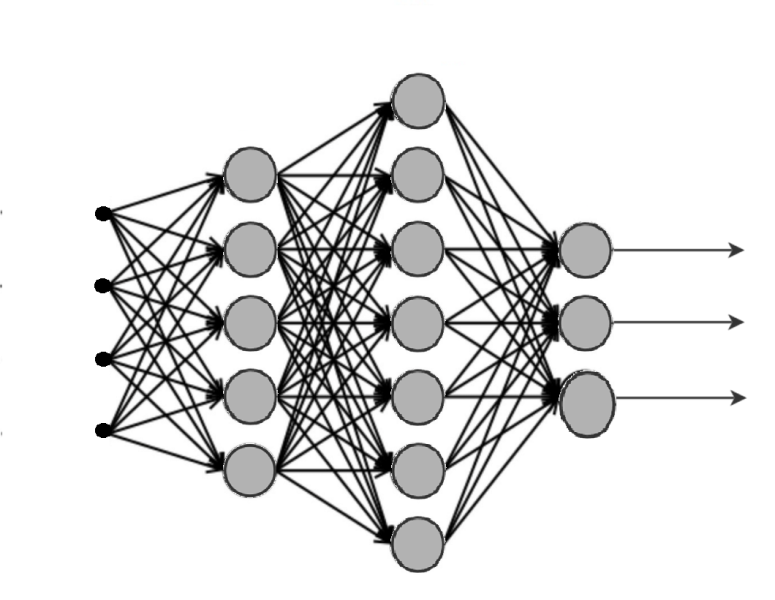
### 7.2. 1. בעיות סיווג

בעיות שבהן יש להחליט האם הקלט הנתון שייך לקטגוריה (מחלקה) מסוימת או לא.

לדוגמה, בעיית זיהוי תמונה: נתונה תמונה ויש לקבוע האם בתמונה קיים אובייקט מסוג "הולך רגל" או לא.   
במקרה זה

### 7.3. 1. בעיות סיווג מרובה-מחלקות:

בבעיות מהסוג "בעיות סיווג מרובה-מחלקות" האתגר הוא לקבל החלטה מתוך קטגוריות קיימות לאיזו קטגוריה, שייךהקלט. לדוגמה, בבעיית זיהוי תמונה: נתונה תמונה מסוימת מתוך מאגר נתונים והמטרה היא לקבוע האם התמונה הינה תמונה של מכונית, של הולך-רגל, או של מעבר חצייה. רשת אפשרית לפתרון הבעיה הזו:



***input***

***output***

***הולך רגל***

***input***

***output***

*מכונית*

***input***

*מעבר חצייה*

***output***

***input***

***output layer***

***Hidden layer***

***איור 12***

***Hidden layer***

ברשת הזו לדוגמא ניתן לראות שהמודל בנוי מ3 נוירוני-פלט כשה*output* שלהם בינארי ( 1 או 0 ( זוהי דוגמא לבעיית סיווג , ליד כל נוירון פלט מצוינת המחלקה אותה "מייצג" הנוירון. במקרה זה

סט האימון יורכב מזוגות y) (x, כאשר x הוא תמונה מטריצת פיקסלים ו-y הוא וקטור בינארי בגודל ,3

שנראה כך לדוגמא:

*מייצג*

***הולך-רגל***

*מייצג*

*מעבר חצייה*

*מייצג*

*מכונית*

## 8. 1. למידה

למידה היא מרכיב בסיסי ומאוד חשוב ברשת נוירונים, אם כי קשה לייצר הגדרה מדויקת של למידה. מבחינת רשת עצבית מלאכותית, הלמידה מתרחשת בדרך כלל בשלב אימונים ספציפי. לאחר הכשרת הרשת היא נכנסת לשלב הפרודקשן בו היא מניבה תוצאות באופן עצמאי. אימון יכול לתפוס צורות רבות ושונות, תוך שילוב של שיטות למידה, כללי למידה ואלגוריתמי למידה שונים. מערכת בעלת שלבי למידה וייצור שונים ידועה כרשת סטטית. רשתות שמסוגלות להמשיך ללמוד במהלך הפרודקשן ידועות כמערכות דינמיות.

ישנן מספר שיטות למידה שונות, שיטת למידה מפוקחת, ללא פיקוח או השילוב (מצב היברידי) של השניים. ככלל למידה היא מודל לסוגי השיטות המשמשות להדרכת המערכת לאילו סוגי תוצאות היא אמורות להפיק. אלגוריתם הלמידה הוא השיטה המתמטית הספציפית המשמשת לעדכון המשקולות הסינפטיים הבין-עצביים במהלך כל איטרציה של אימון. ניתן להשתמש בכללי למידה ואלגוריתמי למידה עם שיטות למידה מפוקחות או לא מפוקחות, וכל אחת מהן תביא לאפקט שונה.

### 1.8.1 למידה מפוקחת (Supervised learning)

למידה מפוקחת ממפה קלט לפלט על בסיס דוגמאות של זוגות קלט-פלט. למידה מפוקחת היא למידה שמבוססת על מערכת של סט של דוגמאות. בלמידה מסוג זה , כל דוגמה שניתנת היא זוג המורכב מאובייקט קלט (בדרך כלל וקטור) וערך פלט רצוי (נקרא גם אות הפיקוח). אלגוריתם למידה מפוקחת מנתח את נתוני הדוגמאות שניתנו לו ומפיק מהן פונקציה מתאימה שבעזרתם ניתן להשתמש בה למיפוי דוגמאות חדשות. תרחיש אופטימאלי יאפשר לאלגוריתם לקבוע נכון את תוויות המחלקה עבור מקרים שעוד לא "נתקל" בהם.[[8]](#_ביבליוגרפיה)

**1.**8.2 למידה ללא פיקוח (Unsupervised learning)

למידה ללא פיקוח היא סוג של למידה המחפשת דפוסים שלא נתגלו בעבר במערך נתונים ללא תוויות קיימות ועם מינימום פיקוח והתערבות אנושית. בניגוד ללמידה בפיקוח העושה שימוש בדרך כלל בנתונים עם תווית אנושית, למידה ללא פיקוח, הידועה גם כארגון עצמי, מאפשרת דוגמנות של צפיפות הסתברות על פני תשומות. זה מהווה אחת משלוש הקטגוריות העיקריות של למידת מכונות, יחד עם למידת פיקוח וחיזוק. למידה מפוקחת למחצה, וריאציה קשורה, עושה שימוש בטכניקות מפוקחות ולא מפוקחות.[[9]](#_ביבליוגרפיה)

## 9. 1. פונקציית העלות

פונקציית עלות או בשמה האחר פונקציות אובדן הן מושגים מעולם האופטימיזציה , שימושה העיקרי בלמידת מכונה הוא לסיווג או קלאסיפיקציה, פונקציות שמאפשרות חישוב ה"מחיר" ששולם עבור אי דיוק של תחזיות בבעיות סיווג (בעיות בזיהוי לאיזו קטגוריה תצפית מסוימת שייכת) .[[10]](#_ביבליוגרפיה)

בהינתן ***X*** כמרחב הווקטורי של כל **הכניסות** האפשריות ,   
 ***Y*** = {–1,1} כמרחב הווקטורי של כל **היציאות** האפשריות,

אנו מעוניינים למצוא פונקציה

אשר ממפה בצורה הטובה ביותר את

אך בגלל מידע לא שלם, רעש במדידה או רכיבים הסתברותיים בתהליך הבסיסי, יתכן **שאותו**  ינבא ***שונה***

כתוצאה מכך, מטרת בעיית הלמידה היא למזער את הסיכון הצפוי, המוגדר כ-

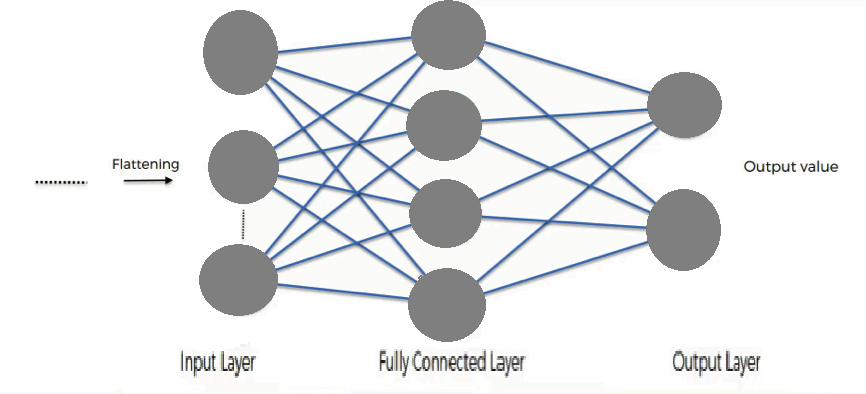
כאשר היא פונקצית העלות ו היא פונקציית הצפיפות או **PDF** של התהליך שיצר את הדאטה.

לפונקצית העלות או ההפסד יש תפקיד חשוב בכך שהיא חייבת לזקק בצורה נאמנה את כל היבטי המודל למספר בודד באופן כזה שכל שיפור בערך הזה יגרום לשיפור ולמודל טוב יותר. [[11]](#_ביבליוגרפיה)

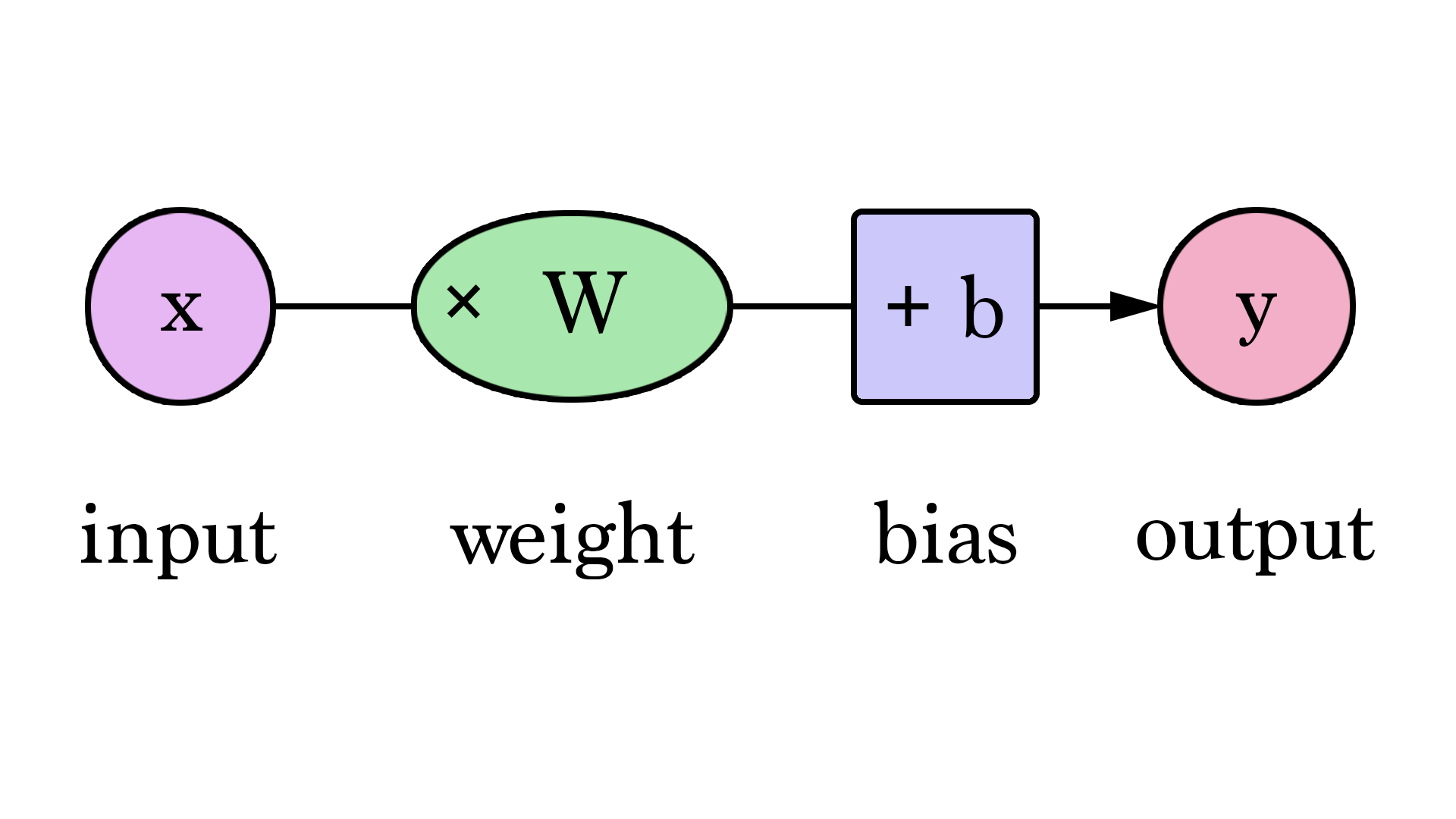
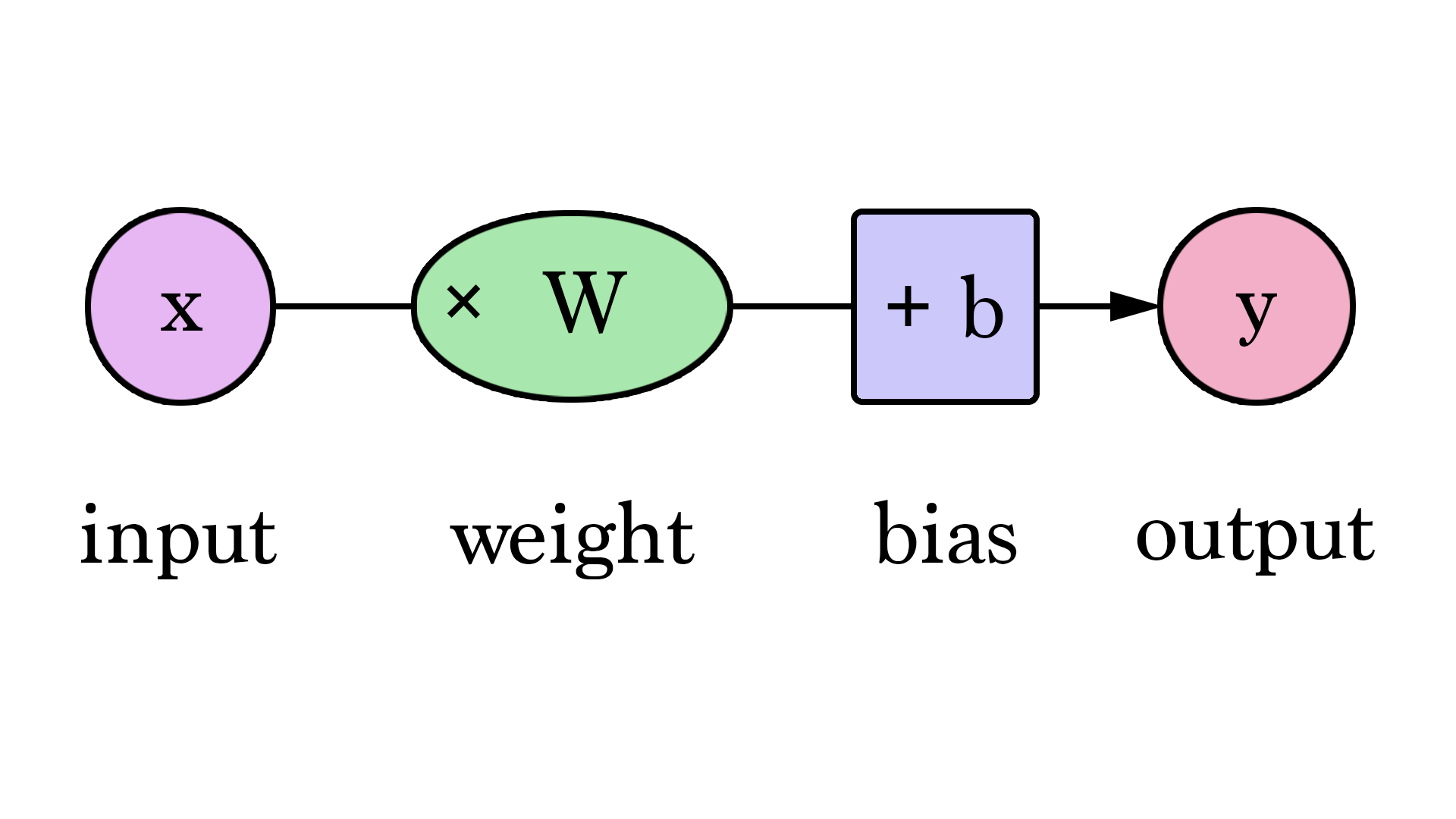
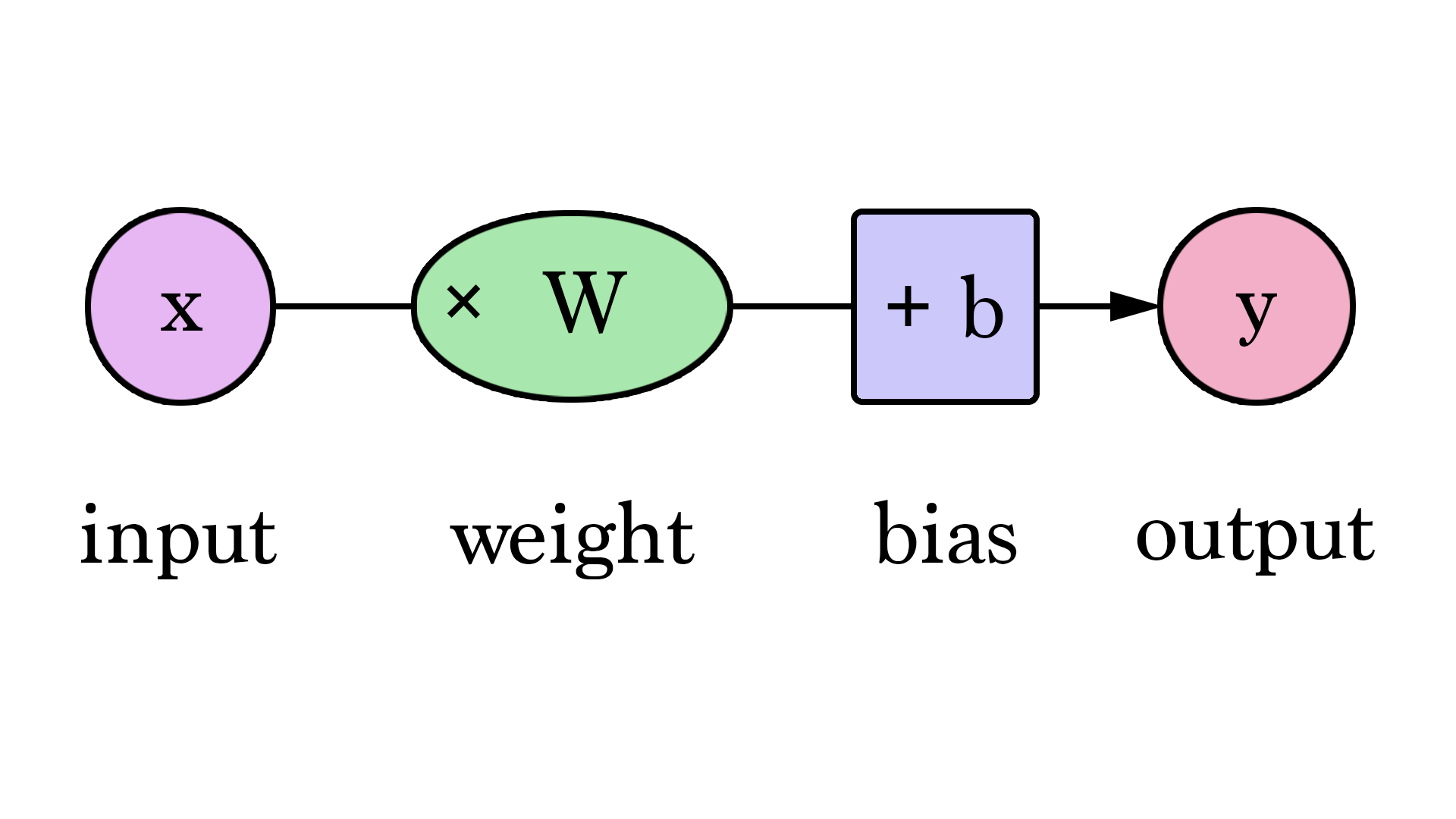
## 10. 1. רשת קונבולוציה

בלמידה עמוקה , רשת עצבית מבוססת קונבולוציה (CNN, או ConvNet) היא מחלקה של רשתות עצביות עמוקות, המיושמת לרוב לניתוח דימויים חזותיים. [1]

רשתות נוירונים רב שכבתיות בד"כ מרביתן בתצורת fully-connected, כלומר כל נוירון בשכבה אחת מחובר לכל הנוירונים בשכבה הבאה (ראה איור 7) . ה"התחברות המלאה "של רשתות אלה גורמת להם להיות מועדים להתאמת יתר של הנתונים. דרכים רגילות של רגולריזציה כוללות הוספת סוג כלשהו של מדידת גודל של משקולות לתפקוד הLOSS. בארכיטקטורת CNN נוקטים גישה שונה בנושא רגולריזציה: פה מנצלים את התבנית ההיררכית בנתונים ומרכיבים דפוסים מורכבים יותר תוך שימוש בתבניות קטנות ופשוטות יותר. לכן, בסדר גודל של חיבור ומורכבות, רשתות ה- CNN נמצאות בקצה התחתון.

****

**...**

****

***איור 13***

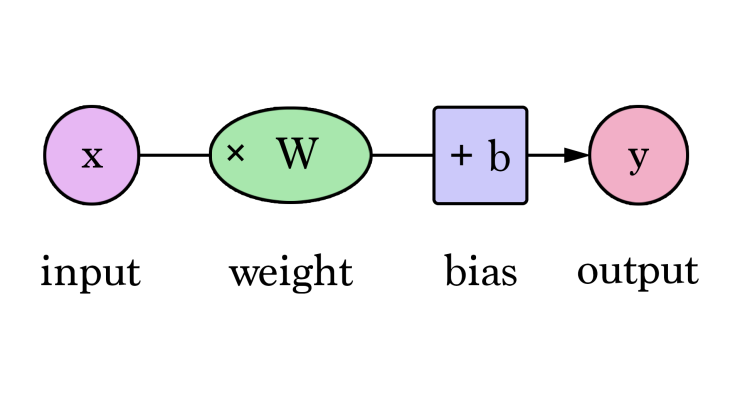
רשת קונבולוציה עמוקה מורכבת משכבת כניסה ושכבת פלט או יציאה, כמו כן שכבות נסתרות מרובות. השכבות הנסתרות של CNN מורכבות בדרך כלל מסדרה של שכבות קונבולוציה שמבוססות על כפל או dot product. פונקציית ההפעלה היא בדרך כלל שכבת ReLU, ואחריה שכבות קונבולוציה נוספות כמו שכבות pooling, שכבות מחוברות לחלוטין

(fully connected) ושכבות נורמליזציה, המכונות שכבות נסתרות מכיוון שהכניסות והתפוקות שלהן מוסוות על ידי פונקציית ההפעלה והפיתרון הסופי.

## 11. 1. מישקול

כל נוירון ברשת עצבית מחשב ערך פלט על ידי יישום פונקציה ספציפית לערכי הקלט המגיעים מהשדה הקולט בשכבה הקודמת. הפונקציה המיושמת על ערכי הקלט נקבעת על ידי וקטור משקולות והטיה (בדרך כלל מספרים אמיתיים). הלמידה, ברשת עצבית, מתקדמת על ידי ביצוע התאמות איטרטיביות להטיות ומשקולות אלה. וקטור המשקולות וההטיה נקראים פילטרים ומייצגים תכונות מסוימות של הקלט (למשל, צורה מסוימת). מאפיין המייחד של רשתות CNN הוא שהרבה נוירונים יכולים לחלוק את אותו פילטר [[12]](#_ביבליוגרפיה).

1.11.1 מישקול והטיה  
משקולות (Weights) והטיה (Bias) הם שני פרמטרי למידה בתוך הרשת. רשת עצבית תגריל באופן אקראי את ערכי המשקל וגם את ההטיה לפני תחילת הלמידה. ככל שנמשך האימון, שני הפרמטרים מותאמים לערכים הרצויים ולפלט הנכון. שני הפרמטרים נבדלים זה מזה במידת השפעתם על נתוני הקלט. ההטיה מייצגת את המרחק מהתחזיות מהערך המיועד. הטיה מהווה את ההבדל בין תפוקת הפונקציה לפלט המיועד לה. הטיה נמוכה מרמזת על כך שהרשת עושה יותר הנחות לגבי צורת הפלט, ואילו ערך הטיה גבוה מניח פחות הנחות לגבי צורת הפלט. משקולות, לעומת זאת, יכולים להיחשב כחוזק הקשר. משקל משפיע על כמות ההשפעה שיש לשינוי בקלט על הפלט. ערך משקל נמוך לא ישנה את הכניסה, לעומת זאת ערך משקל גדול יותר ישנה משמעותית יותר את הפלט.



***איור 14***

### 1.11.2 אתחול משקולות

כשעובדים עם רשתות עצביות עמוקות, אתחול הרשת עם המשקולות הנכונות יכול להיות קשה מכיוון שרשתות עצביות עמוקות סובלות מבעיות שנקראות Vanishing/Exploding Gradients.

לכן צריך שהאות יזרום כראוי לשני הכיוונים: בכיוון קדימה בעת ביצוע תחזיות, ובכיוון ההפוך עבור Back Propagating gradients

כמובן שלא רוצים שהאות ימות, וגם לא רוצים שהוא יתפוצץ או יגיע לסאטורציה (רוויה). לכן צריכים שהמדרגות יהיו שונות שווה לפני ואחרי שזורמים בשכבה בכיוון ההפוך.

אתחולים מגדירים את הדרך להגדרת המשקולות האקראיות הראשוניות של שכבות keras.

ברשת שלי אני השתמשתי באתחול משקולות אחיד מסוג XAVIER ידוע גם בשם GLOROT [[13]](#_ביבליוגרפיה).

## 12. 1. סגמנטציה

תהליך של חלוקת תמונה דיגיטלית לקטעים מרובים (קבוצות של פיקסלים, המכונים גם אובייקטים של תמונות). מטרת הסגמנטציה היא לפשט או לשנות את ייצוג התמונה למשהו שהוא יותר משמעותי וקל יותר לניתוח[[14]](#_ביבליוגרפיה).

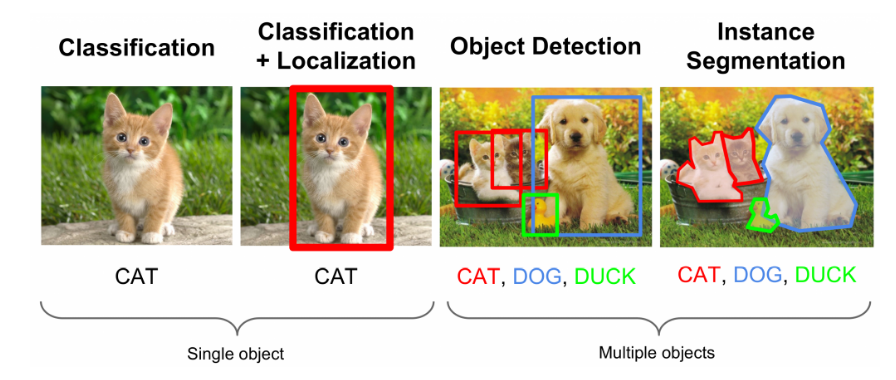
סגמנטציה של תמונות משמש בדרך כלל לאיתור אובייקטים וגבולות (קווים, עקומות וכו') בתמונות. ליתר דיוק, סגמנטציית תמונות הוא תהליך הקצאת תווית לכל פיקסל בתמונה כך שפיקסלים עם אותה תווית חולקים מאפיינים מסוימים

בעזרת מערכת לומדת ניתן לזהות אובייקטים בתמונות במידה ויש מאגר תמונות רבות שמסווגות בצורה טובה.

מערכת לומדת זאת אפשרות מאוד טובה לזיהוי אובייקטים ואפשר לראות מהתעשייה שיש הרבה יישומים שמשתמשים במערכות לומדות לזיהוי אובייקטים

תחילה אסביר את ההבדל בין קלאסיפיקציה, זיהוי אובייקט, ופילוח (סגמנטציה) סמנטי:

***ד' ג' ב' א'***



***איור 13***

בתמונה **א**' ניתן לראות שישנו אובקייט אחד והוא זוהה כחתול

בתמונה **ב**' ניתן לראות שישנו אובקייט שזוהה כחתול וסביבו ממורכז bounding-box שנותן לוקאליזציה של האובייקט במרחב התמונה.

בתמונה **ג**' ניתן לראות שישנם מספר אובקייטים מקאלסים שונים ולכולם boxes ותיוגים

בתמונה **ד'** ניתן לראות שישנם מספר אובקייטים מקאלסים שונים כשלכל אובייקט סגמנטציה הדוקה, צבע אחר המתאר קלאס שונה.

בעולם הראייה הממוחשבת לאחרונה ישנה התקדמות מהירה בנושא הסגמנטציה, כאשר המשימה היא להקצות כל פיקסל תיוג מסוים וכך לקבל הבנה של מה נמצא בתמונה. עד כה נעזרו לרוב בשיטות בסיסיות פשוטות אך חזקות, כולל רשתות Fully Convolutional Networks (FCN) ו - MASK\_R-CNN

שיטות אלה פשוטות באופן רעיוני, מהירות וגמישות. בעבודה אציג ארכיטקטורה משלי למשימת זיהוי אובייקטים בעזרת רשת ניורנים מבוססת CNN המשתמשת כdatabase במאגר תמונות COCO בתצורת זיהוי

## 13. 1. זיהוי הולכי רגל

זיהוי הולכי רגל הוא משימה חיונית ומשמעותית בכל מערכת מעקב וידאו חכמה שכן היא מספקת את המידע הבסיסי להבנה סמנטית של צילומי הווידיאו. יש לה שימוש רחב גם ביישומי רכב בגלל הפוטנציאל לשיפור מערכות הבטיחות. יצרני רכב רבים (למשל וולוו, פורד, GM, ניסאן,טויוטה, טסלה,יונדאי ועוד) מציעים זאת כאופציה של ADAS (Advanced driver-assistance systems) כלומר מערכות סיוע מתקדמות לנהג.

### אתגרים בזיהוי הולכים רגל:

* סגנון לבוש שונה במראה
* מגוון תנוחות אפשריות שונות
* מהירות זיהוי לצורך תגובה
* נוכחות של אביזרים מסתירים ,חסימות והסתרות תכופות

למרות האתגרים, זיהוי הולכי הרגל עדיין נותר תחום מחקר פעיל בcomputer vision בשנים האחרונות. הוצעו מספר רב של גישות:

### 1.13.1 Holistic detection

גלאים מאומנים לחפש הולכי רגל במסגרת הווידיאו על ידי סריקת כל הפריים. הגלאי "יורה" אם התכונות בתמונה בתוך חלון החיפוש המקומי עומדות בקריטריונים מסוימים. שיטות מסוימות משתמשות בתכונות גלובליות כגון מציאת קצוות, [1] אחרות משתמשות בתכונות מקומיות כמוHOG (histogram of oriented gradients) היסטוגרמה של גראדיאנט מונחה .[[14]](#_ביבליוגרפיה) החיסרון של גישה זו הוא שאפשר להשפיע בקלות על הביצועים כתוצאה מעומס רעש ברקע והסתרות רבות.

### 1.13.2 Patch-based detection

גישה המשלבת גם זיהוי וגם סגמנטציה (Implicit Shape Model (ISM. מערך נתונים של מראה מקומי נלמד במהלך תהליך האימון. בתהליך הזיהוי משתמשים בפיצ'רים מקומיים שנמצאו כדי להתאים זה לזה כנגד ערכי מערך הנתונים שנלמדו. ניתן להשיג תוצאות איתור סופי על ידי שיפור נוסף של השערות אלה. היתרון בגישה זו הוא שרק מספר מצומצם של תמונות אימונים הכרחי לשיטה.

### 1.13.3 Motion-based detection

כאשר יש תנאים מתאימים מאוד ספציפיים (מצלמה קבועה, תנאי תאורה טובים וכו'), שיטת חיסור רקע **(Background subtraction)** יכולה לסייע בזיהוי הולכי רגל. חיסור רקע מסווג את הפיקסלים של קטעי הווידיאו כרקע, כאשר לא מזוהה שום תנועה, או הולכי רגל אם מתגלה תנועה. הליך זה מדגיש את הצלליות של כל אלמנט נע בסצנה, כולל אנשים. בשיטה זו פותח אלגוריתם [[15]](#_ביבליוגרפיה) שמנתח את צורת הצלליות הללו כדי לאתר את בני האדם. מכיוון שהשיטות הרואות את הצללית בכללותה ומבצעות סיווג יחיד הן בד"כ רגישות מאוד לפגמי צורה, לכן הוצעה שיטה שמחלקת את הצלליות לאזורים קטנים יותר כדי להפחית את השפעת הפגמים.

### 1.13.4 multiple cameras- based detection

שיטה זו מבוססת על מספר מצלמות מכוילות לזיהוי הולכי רגל רבים. בגישה זו, מישור הכביש מחולק לתאי רשת אחידים ולא חופפים, בדרך כלל בגודל 25 על 25 (ס"מ). הגלאי מייצר (POM) Probability Occupancy Map

מפת הסתברות התפוסה , הוא מספק הערכה של ההסתברות של כל תא רשת שיתפס על ידי אדם. בהינתן שניים עד ארבעה קטעי וידיאו מסונכרנים שצולמו בגובה העיניים ומזוויות שונות, שיטה זו יכולה לשלב ביעילות מודל דינמי שעוקב במדויק אחר שישה אנשים על פני אלפי פריימים, למרות הסתרות ושינויים משמעותיים בתאורה. שיטה זו יכולה גם לשערך את מסלולי הדרך המדויקים של כל אחד מהולכי הרגל.

### 1.13.5 data-driven CNN-Unet based approach

השיטה המוצעת כאן, שאותה פיתחתי בפרויקט שלי, מבוססת על רשת נוירונים רב שכבתיות בתצורת fully-connected מסוג CNN בארכיטקטורת U-Net, בגישה מבוססת דאטה יש אפשרות בעזרת מצלמה אחת זיהוי הולכי רגל רבים. בגישה זו המערכת מסווגת את הפיקסלים של התמונה כרקע, כאשר אין זיהוי וודאי באחוזים גבוהים של האובייקט אותו מחפשים (במקרה שלי הולך רגל) , או אכן הולך רגל אם אכן התקבל זיהוי חיובי שזוהה על סמך הלמידה והבנה של הסצנה מתוך תהליך הלמידה של הדוגמאות שניתנו לרשת.

# 2. ניסוי

בפרויקט זה בחרתי את מאגר הנתונים COCO המכיל תמונות RGB בגדלים שונים.

COCO הוא מערך נתונים לזיהוי חזותי בשימוש נרחב, שנועד לדרבן מחקר לגילוי אובייקטים תוך התמקדות בהבנת הסצנה המלאה. בפרט: איתור תצוגות לא רגילות של אובייקטים, לוקליזציה של אובייקטים בתמונות בדיוק ברמת הפיקסלים וגילוי אובייקטים בסצנות מורכבות. מערך הנתונים של COCO כולל 330 אלף תמונות של סצנות מורכבות המקוטלגות ומתויגות באופן מוחלט עם 80 קטגוריות אובייקט עם מסכות סגמנטציה

מערך הנתונים הניתן להורדה מכיל **118**,**287** תמונות לאימון, **5000** תמונות לוולידציה ו-**40**,**670** תמונות לבחינה.

את המאגר ניתן להוריד מהאתר של קהילת COCO [[16]](#_ביבליוגרפיה), במאגר זה משתמשים בקבצי json לתאור האנוטציות בתמונות.



***איור 13***

## 2.1 ארכיטקטורה ואימון:

לאחר ייבוא כל הספריות הנדרשות , טענתי את מאגר התמונות למחשב,

השתמשתי בקובץ Instances כדי לקבל את המסכות של האובייקט שאני מנסה לזהות בתמונות.

לאחר מכן יצרתי מסיכות בינאריות לכל התמונות

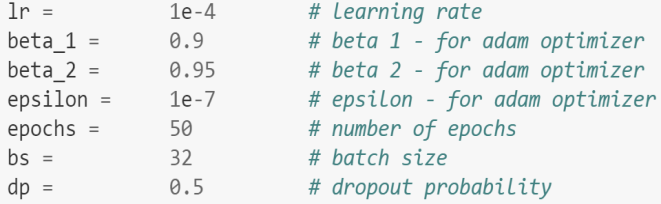
במידה וקיימת אנוטציה של האובייקט שמעניין אותי בתמונה אז יצרתי מסיכה עם ערך המאפיין את האובייקט

לדוגמה אם אני מנסה לזהות אנשים וחתולים אז לכל אובייקט יש מספר סידורי שמתאר את הקלאס , למשל הערך שמאפיין אנשים הוא 1 והערך שמאפיין חתולים הוא 17

אז יצרתי מסכה שבכל מקום שיש בן אדם שם יהיה ערך 1 ובמקום שיש חתול יהיה ערך 2 כך אני יכול לסווג את המידע שלי בצורה נוחה למודל.

בגלל מגבלות של זכרון נאלצתי לשנות את הרזולוציה של התמונות ל64x64 כך יכולתי לאמן את המודל על כל המאגר

הגדרתי את hyper-parameters שלאחר הרבה איטרציות של ניסוי וטעייה מצאתי שהכי מתאימים למודל ולארכיטקטורה שבניתי:



מאגר התמונות COCO שאיתו אני עובד מחולק לתיקיות מסודרות:

* החלק המרכזי שעליו אימנתי את המודל (train)
* החלק שעליו אעשה test וכך אוכל לבחון את הביצועים של המודל עבור מידע שעדיין הוא לא ראה ואומן עליו.

## 2.2 אימון הרשת – זמן כולל ופלטפורמת המערכת.

***Train time***

2ms per sample

173s per Epoch   
2.5 hours of training (total of 50 epochs)

***PC platform:***

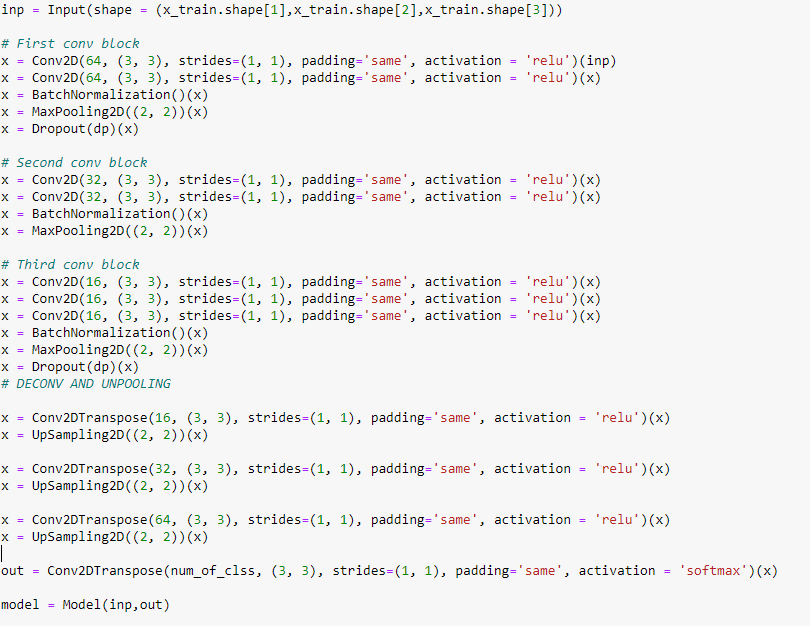
CPU: Intel i7 9700k

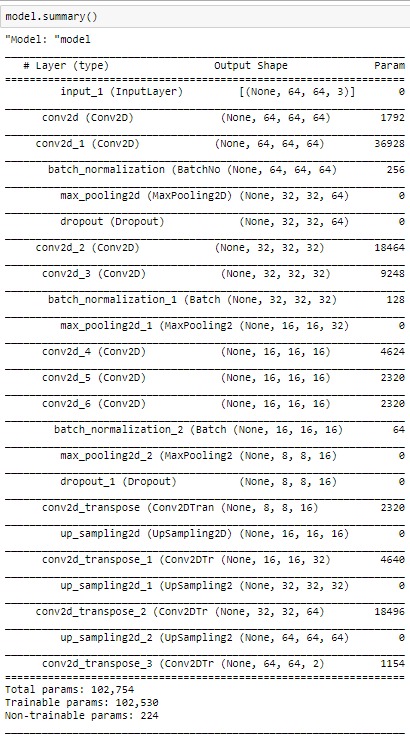
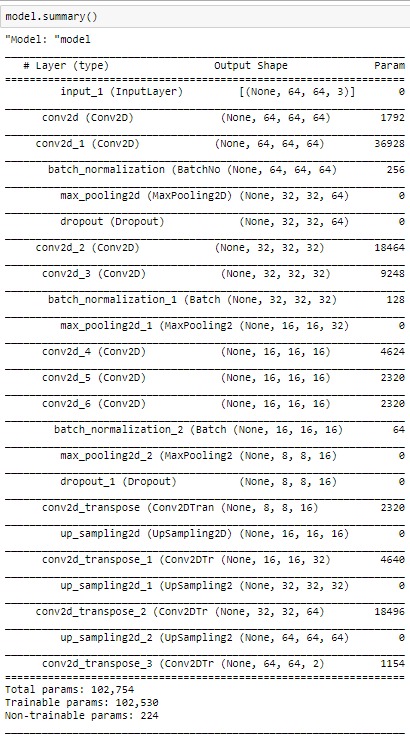
GPU: nvidia geforce 1050ti

RAM: 32GB DDR4

Disk: SSD kingston

המודל שבניתי בנוי מהשכבות הבאות:





***איור 13***

### כמות פרמטרים הכוללת ברשת

### ארכיטקטורת הרשת:

שרשור של שכבות קונבולוציה עם ReLU ובסוף שכבה פולי-קונקטד שמאפשרת טיפול במקרים אי-לינארים ומשמרת את הdimensions וכך משפרת את הrobustness של הרשת.

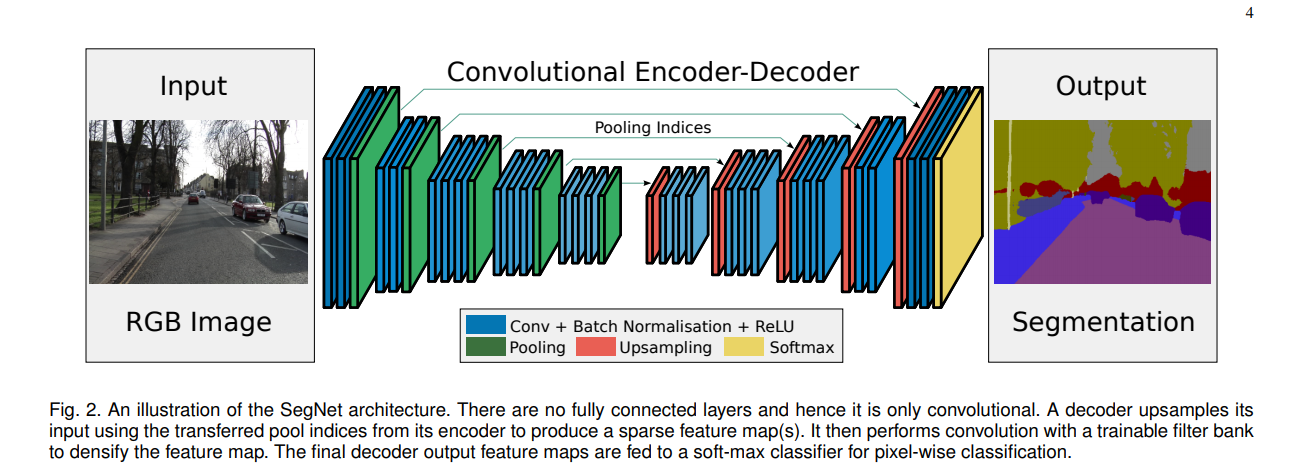
ניתן לראות שברשת שלי בחרתי בארכיטקטורת encoder-decoder או בעברית מפענח-מקודד   
ארכיטקטורה זו היא תבנית ידועה של רשתות שנוח להשתמש בה בעיקר לsegmentation.

הארכיטקטורה מחולקת לשני חלקים, המקודד והמפענח. תפקיד המקודד הוא לקודד את הכניסות למצב.

ואז המצב מועבר למפענח כדי לייצר את היציאות.

המפענח הוא בלוק ברשת (בדרך כלל אותה מבנה רשת כמו המקודד) שלוקח את וקטור הפיצ'רים מהמקודד ומעניק את ההתאמה הטובה ביותר לקלט או לפלט המיועד.

לאחר שאימנתי את המודל נותר כעת לבחון אותו על מידע חדש שהוא לא ראה מימיו כדי לראות את היעילות שלו בפועל.

******

### Over Fitting ודרכים למניעה

בסטטיסטיקה, התאמת יתר היא תוצר של ניתוח שנותן תוצאות קרוב מדי או מדויק עם מערך נתונים מסוים, ולכן עשוי שלא להתאים לנתונים נוספים או לחזות תצפיות עתידיות באופן אמין". מודל עם התאמת-יתר הוא מודל סטטיסטי המכיל יותר פרמטרים שיכולים להיות מוצדקים על ידי הנתונים.

בתור דוגמה קיצונית להתאמת יתר, אם מספר הפרמטרים המשמשים להגדרת המודל זהה או גדול יותר מאשר מספר התצפיות, אזי המודל יוכל לחזות באופן מדויק את התצפיות פשוט על ידי זכירתן במלואן.

התאמת יתר מתרחשת במקרים בהם חסרה תאוריה להנחיית האנליזה של הנתונים. במצב כזה יש צורך לבחון מספר רב של מודלים אפשריים ונדרשים יותר פרמטרים כדי להגדירם[13].

התאמת יתר היא בעיה שמתעוררת כאשר ניתנים יותר מדי דוגמאות לאימונים, והמערכת הופכת לא מסוגלת ליצור הכללה מועילה. זה יכול להתרחש גם כשיש יותר מדי נוירונים ברשת ויכולת החישוב עולה על הממדיות של מרחב הקלט. במהלך האימונים, יש להקפיד שלא לספק יותר מדי דוגמאות קלט ומספרים שונים של דוגמאות אימונים עשויים להניב תוצאות שונות מאוד באיכות וברמת החוסן של הרשת

ישנן כמה שיטות מקובלות להפחתת הסיכון בהתאמת יתר, בכולם השתמשתי בפרויקט שלי: רגולריזציה (regularization), עצירה מוקדמת (early stopping) , עיוות הדאטה (Data augmentation וכן drop-out

הבסיס לשיטות אלה הוא:

* מחיר מפורש למורכבות המודל (למשל ברגולריזציה) ובכך לתת לתהליך האופטימיזציה להעדיף פתרונות פשוטים. מורכבות המודל אינה נבחנת רק לפי מספר הפרמטרים אלא גם לפי מידת הביטוי של הפרמטרים.
* עיוות המידע והזנתו למאגר כמידע נוסף , ושימוש במספר שינויים קטנים ברשת כדי ליצור אנסמבל של רשתות אקראיות וזאת כדי למנוע תאימות יתר

### Batch normalization.2

טכניקה שמאפשרת לזרז ולייצב את תהליך הלמידה זה מאפשר להשתמש במדד למידה גבוה יותר כך גם הלמידה מהירה יותר, אחרי פונקצית אקטיבציה אנחנו מנרמלים את המידע לשיכבה הבאה

לדוגמה כאשר יש שיכבה חבוייה שמורכבת מ

*אחשב את המיצוע והשונות*

*כאשר נוסחאת הנרמול נראית כך:*

*מטרת האפסילון היא למקרה שסיגמה מתאפסת ואחרי הנרמול עכשיו לכל יחדה של השכבה יש עכשיו מיצוע סביב האפס ושונות השווה לאחד ולכן אעדיף שיהיה לכל נוירון ערכים שונים אז אפשר לכתוב את זה בצורה כזאת כך שהמידע יהיה מנורמל בצורה אחרת*

*במידה ו*

*כך שאפשר לשחק עם הפרמטרים של גמא ובטא ואז המיצוע והשונות יכולים להיות מנורמלים סביב איזה אזור שנרצה במודל כאשר לא סיפקתי פרמטרים אז הוא נרמל את המיצוע קרוב לאפס ואת השונות קרוב ל1*

שיטות נוספת שבחרתי להשתמש בה למניעתoverfitting היא דרופ-אוט

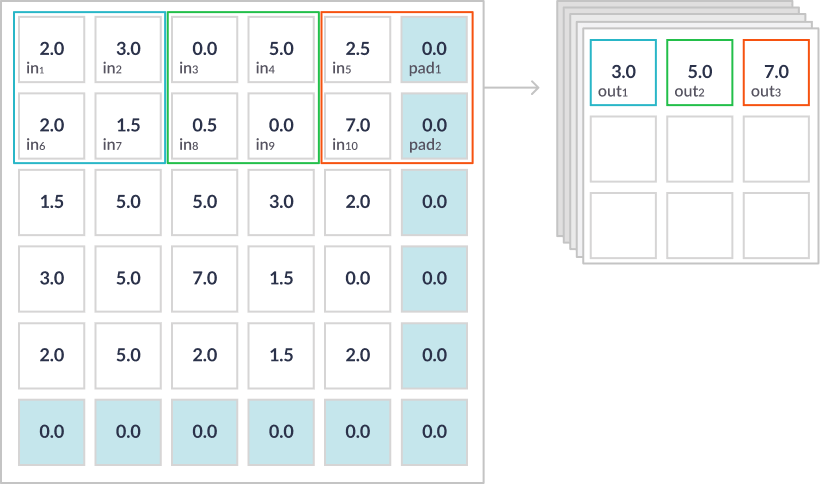
### Dropout.3

טכניקת רגולריזציה שמאפשרת להתגבר על הנושא של תאימות יתר (over fitting) של המודל שבזמן הלימוד בהסתברות מסוימת מתעלמים מחלק הנוירונים בשכבה, בצורה זו אני מכניס אקראיות למודל - שכן כל ריצה רצף נוירונים אחרים "מתים" וכך המודל חייב להתאים את עצמו לנוירונים אחרים.

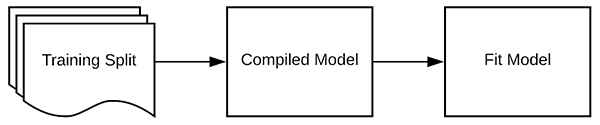
***איור 13***

### MaxPooling2d.4

מטרת שכבת הפולינג היא להוריד את גודל הכניסה שכבה זו מרדדת את הרעש ומרדדת שינויים קלים כמו סיבוב או הזזה של התמונה גם שכבה זו מבוססת על פילטור של חלון רץ השכבה הזו לוקחת את התוצאה הגבוהה ביותר בכל חלון דוגמה בתמונה: (כפי שהשתמשתי במודל שלי חלון 2X2) בכל חלון ערך המקסימום נלקח למפת הפיצ'רים במילים אחרות כל ערך מקסימום הוא מייצג את החלון של 2X2 במפת הפיצ'רים שכבה זו בתמונות קטנות יכולה לעזור בפתרון האובר-פיטינג



***איור 13***



השתמשתי לאחר מכן ב adam optimizer

***איור 13***

### Conv2d

שכבת קונבולוציה דו ממדית מקבלת בכניסה מערך תלת מימדי בדרך כלל תמונה עם שלושה ערוצי צבע,

השכבה רצה עם חלון (קרנל) קורולציה קטן (בדרך כלל 3x3) על כל התמונה כך למעשה הוא מוצא פיצ'רים (אזורי עניין) בתמונה החלון מאותחל עם משקלים שמתעדכנים בכל איטרציה בזמן הלימוד עד שמגיעים לתוצאות אופטימאליות למודל.

החלון נקרא 2D בגלל שתזוזת החלון היא למעשה על התמונה שהיא דו ממדית שלכל פיקסל יש שלושה ערכי צבע אדום ירוק וכחול

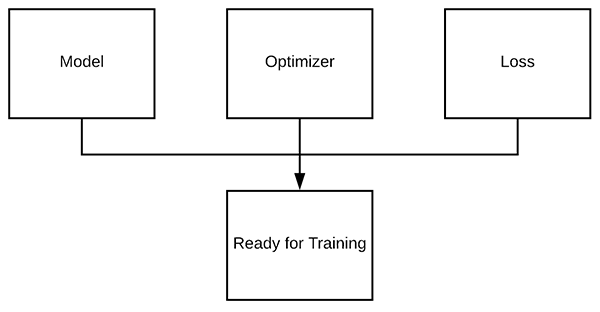
הפילטר רץ על התמונה שלוש פעמים על כל ערוץ צבע.

אפשר להתייחס לזה בצורה כזאת ששכבת הקונבולוציה הראשונה תופסת פיצ'רים בתמונה המקורית אם נקח לדוגמה חתולים היא מזהה פיצ'רים כגון עיניים , אוזניים, ראש, זנב וכדומה לאחר מכאן היא מזהה פיצ'רים באזורי העניין של הראש של האוזניים, של העיניים וכדומה בסופו של דבר המטרה היא לנסות לזהות את כל הפיצ'רים שאיתם אפשר לסווג את האובייקט ולשייך אותו לקבוצה מסוימת

### Back-propagation

הרעיון הוא לעדכן את המשקלים, דוגמה אחר דוגמה, עד סוף תקופה מסוימת (EPOCH)

תקופה זו מייצגת את מספר המעברים הנדרש להגיע להצגה מלאה של כל סט האימון לרשת הנוירונים. עדכון המשקלים נעשה ביחס לשגיאות המחושבות עבור כל אחת מהדוגמאות בסט האימון. הממוצע של שינויי המשקל תוך כדי מעבר על סט האימון הוא הערכה של השינוי האמתי שיקרה כתוצאה משינוי משקלים



***איור 13***

לאחר מכן הוצאתי גרף למידה כדי לוודא התכנסות,והצגתי את התוצאות בעזרת מטריצת מבוכה

לבסוף הוצאתי עקומת ROC

עקומת ROC, היא גרף המציג את הביצועים של מסווג, לאור סף ההחלטה שנקבע לו. העקומה נוצרת על ידי התוויית שיעור החיוביים האמיתיים (TPR) מול שיעור החיוביים הכוזבים (FPR) תחת ספי קבלה שונים. שיעור החיוביים האמיתיים ידוע גם ככיסוי בלמידת מכונה





***איור 13***

אחת הדרכים למדוד את טיב הבחירה היא להשתמש במדד F-score , מדד סטטיסטי לבדיקת דיוק שמתחשב

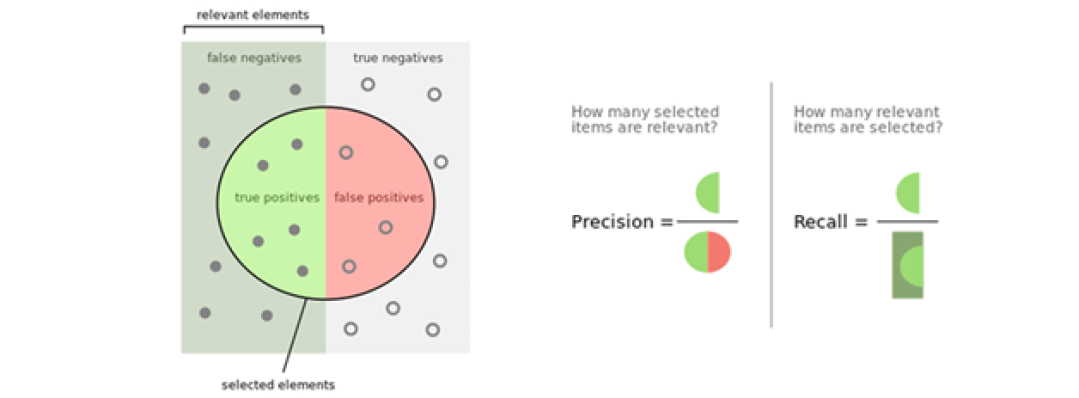
ב- Recall = TP/(TP+FN)

 וב- Precision = TP/(TP+FP)

כאשר TP=True Positive זה כשהאלגוריתם חזה שהקטע מעניין וצדק

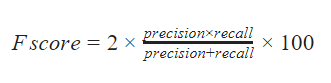
ו FN=False Negative זה כשהאלגוריתם חזה שהקטע לא מעניין וטעה

ו FP=False Positive זה כשהאלגוריתם חזה שהקטע מעניין וטעה.

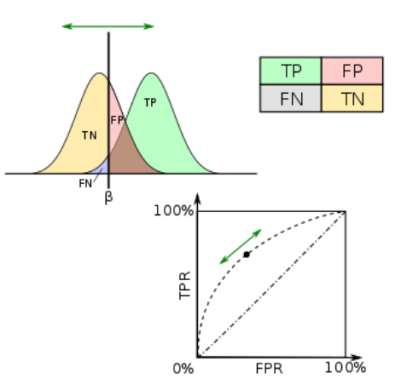
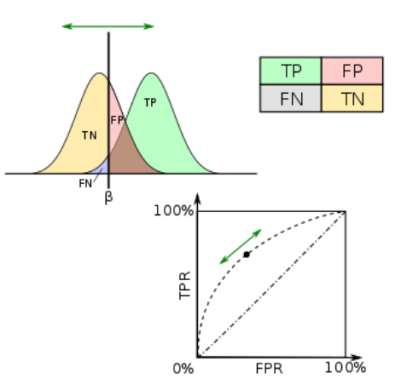


***איור 13***

 מדד  F-score  מוגדר באופן הבא:

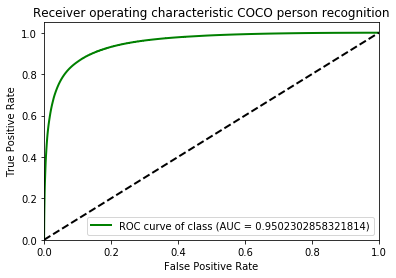


ניתן לראות שהגעתי למודל טוב שמצליח לזהות את הקטגוריות השונות בצורה דיי טובה, לפי מטריצת המבוכה ולפי גרף הROC ,

******

***איור 13***

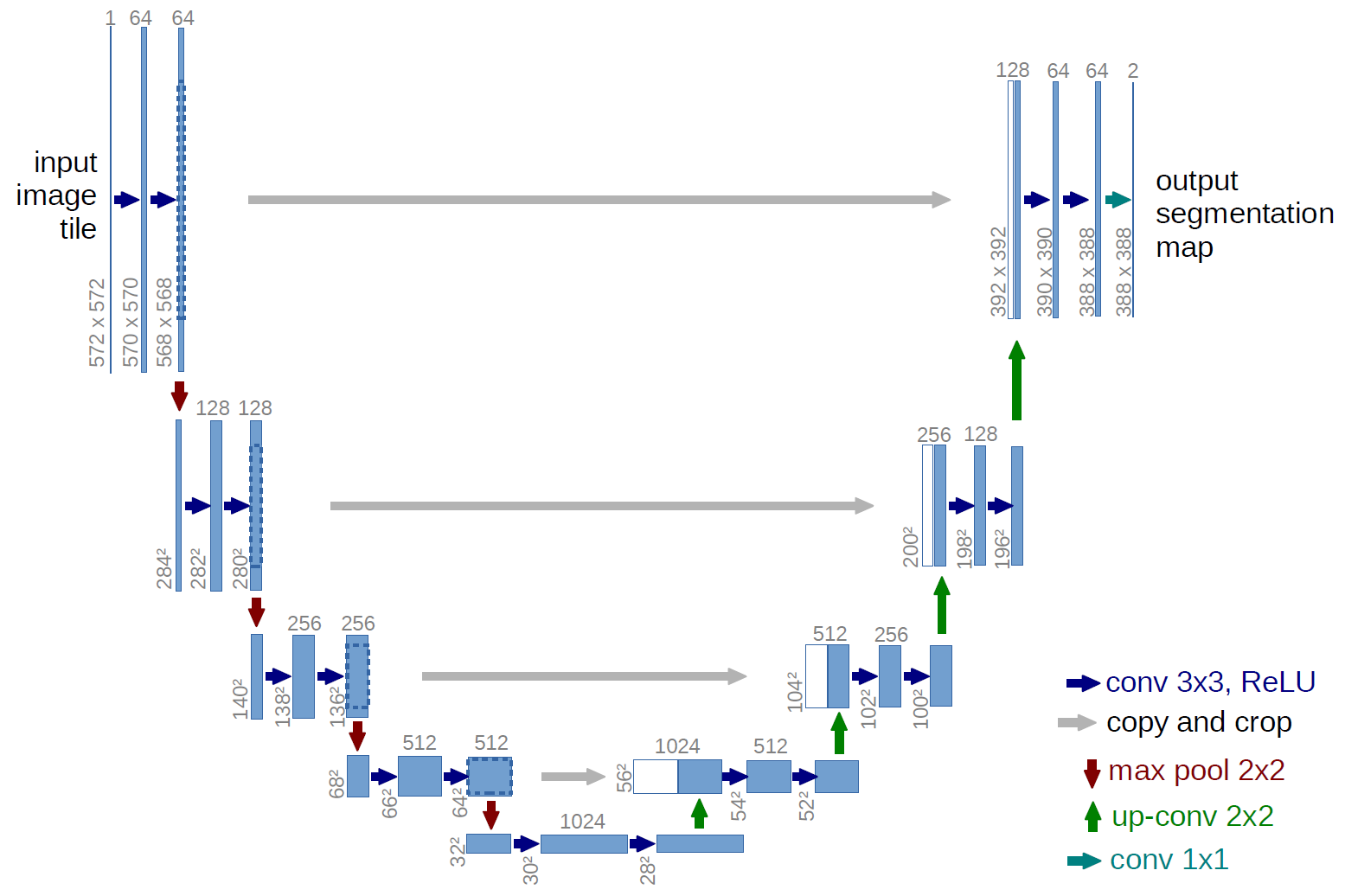
***איור 13***



### תאור הארכיטקטורה של הרשת:

הרעיון המרכזי בשימוש בשכבות קונבולוציה זה לזהות פיצ'רים בתמונה ולמפות אותם וזה עובד ממש טוב בפתרון בעיות קטלוג לקטגוריות,למעשה מתמונה מייצרים ווקטור שבוא משתמשים לקטלוג אבל כאשר מנסים לייצר מערכת שממפה תמונה למסכה של אובייקט בתמונה לא רק צריכים לייצר ווקטור אנחנו גם צריכים לייצר תמונה שזאת משימה מורכבת מאוד, מהרעיון הזה למעשה נוצר הרעיון להשתמש במודל Unet לאחר שלקחתי תמונה ויצרתי ממנה ווקטור, מיפיתי את הפיצ'רים בתמונה למעשה זה המתכון למערכת מפה ואז אפשר לעשות את התהליך בחזרה ולייצר תמונה

דוגמה לארכיטקטורת Unet



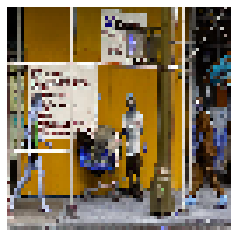
***איור 13***

הארכיטקטורה נראת בצורת האות U מפה השם

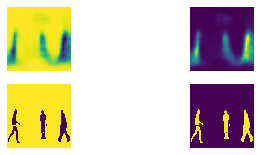
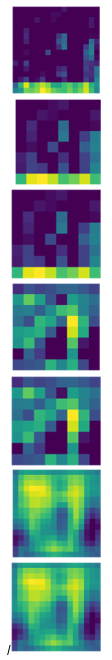
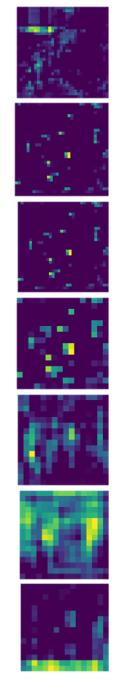
לשם הבנה יותר טובה של **מה הרשת למדה** ניתן להתבונן במשקולות בשכבה הראשונה, להכניס

תמונות בוחן ולראות מה קורה ברשת

הכנסתי את התמונה הזאת:

******

וכאן רואים וויזואליזציה של איך הרשת שלי טיפלה בתמונה כל שכבה בתמונה:



***איור 13***

# 3. סיכום

כמו שצייתי בהתחלה, המטרה הייתה ליצור רשת CNN שמסווגת בצורה מיטבית ואכן הצלחתי ולמדתי מכך המון.

בעיקר למדתי ש deployment של מודל הוא הרבה יותר פשוט ממה שחשבתי בהתחלה, אך אימון מודל הוא עדיין תהליך ארוך ומאתגר – שדורש תכנון, הבנה, תיעוד ותחקור משמעותי.

**ביצועי הרשת** שלי דיי טובים יחסית למידע שסיפקתי למודל למרות חוסר זמן וכוח מחשוב חזק מספיק הרשת הצליחה במשימה בסופו של דבר.

**האתגרים** שהיו לי הם בעיקר כח עיבוד וחוסר זמן לכן הורדתי את הרזולוציה של התמונות לרזולוציה נמוכה ((64\*64 וחיפשתי מודל שיעבד את המידע בצורה המהירה ביותר כך שבתחילת הדרך המודל "למד" במשך יממה וקצרתי את זה לשעתיים וחצי אימון והתוצאות כמעט ולא נפגעו.

אתגר נוסף היה סידור המידע שהייתי צריך ללמוד איך לעבוד עם מאגר המידע COCO ולטעון את המידע בצורה מסודרת לזכרון כך שיהיה מסודר בצורה הגיונית (מבחינת פורמט, רזולוציה, כמות ערוצים וכו) גם בכניסה למודל.

## הצעות להמשך עבודה בנושא וייעול המערכת:

1. בניית חבילה יעילה יותר וספציפית יותר המתאימה לעבודה עם תמונות הלקוחות מרכב אוטונומי (בניגוד לחבילת COCO המותאמת לעבודה לאובייקטים רבים כללים). חבילה כזו תאפשר להגדיל את אוסף המאפיינים ובכך לשפר את תוצאות המערכת. בנוסף, חבילה יעילה יותר תאפשר לקצר את זמן אימון המודל וכן תאפשר הגדלת קבוצת האימון.
2. הגדלת אוסף המאפיינים: שילוב מאפיינים נוספים במודל. המודל המוצע בעבודה זו הינו מודל בסיסי בלבד המתבסס על מספר מאפיינים מצומצם שמבוסס על כמות השכבות שבניתי. ניתן להרחיב את קבוצת המאפיינים (בהינתן התשתית המתאימה כמובן) ובכך להגיע לתוצאות טובות יותר.
3. בעבודה עם רזולוציה גבוהה יותר ניתן לאסוף יותר מידע המבדיל בין סגמנטים.
4. בעבודה עם הAPI של COCO ניתן גם לשערך את הPOSE של האדם. כלומר ללמוד ממפות הסגמנטציה שהתקבלו את צורת העמידה או ההליכה של הולך הרגל. יתכן והפיצ'ר הזה יתן למודל ערך מוסף ועל פיו אוכל לסנן ולקבל תוצאות מדויקות יותר.
5. ניתן לחלק את התמונות לקטגוריות למשל הולכי רגל יחידים, קבוצת הולכי רגל, הולכי רגל עם אופניים, הולכי רגל במעבר חצייה , הולכי רגל על מדרכה. לפי כל קטגוריה נבנה מודל נפרד. בהינתן תמונה שהקטגוריה שלה ידועה, נשתמש במודל שהוגדר ע"י אותה קטגוריה.
6. כמובן שגם את ארכיטקטורת ומבנה הרשת ניתן לשפר - תהליך שדורש הרבה זמן מחקר:
   * + 1. אפשר לנסות כלי אופטימיזציה שונים אני השתמשתי ב"אדם" אבל לפי מחקרים ומאמרים עדכניים ישנם עוד אלגוריתמים מהירים יותר ומדויקים יותר שניתן להשתמש בהם,
       2. הרשת שבניתי דיי קטנה מבחינת כמות שכבות - אפשר להוסיף עוד שכבות וכמובן להגדיל את רזולוציית התמונות.
       3. במידה והמבנה החדש יגיע למצב של התאמת יתר יהיה אפשר להתגבר על over Fitting בעזרת שיטות שלא השתמשתי כמוdata Augmentation או early Stopping שתי שיטות שבחרתי שלא להשתמש בהם, כמו כן, שכן ראיתי שיש צורך באיזון בין ניסוי כיוונים חדשים לבין מיצוי המיטב מגישות קיימות

# **4. ביבליוגרפיה**

1. Mitchell, T. (1997). *Machine Learning*, McGraw Hill. [ISBN](https://he.wikipedia.org/wiki/%D7%9E%D7%A1%D7%AA%22%D7%91) [0-07-042807-7](https://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%95%D7%97%D7%93:%D7%9E%D7%A9%D7%90%D7%91%D7%99_%D7%A1%D7%A4%D7%A8%D7%95%D7%AA/0-07-042807-7), p.2.
2. [*"The First Wave of Corporate AI Is Doomed to Fail"*](https://hbr.org/2017/04/the-first-wave-of-corporate-ai-is-doomed-to-fail)*. Harvard Business Review. 2017-04-18.*
3. *Hernandez, Daniela; Greenwald, Ted (2018-08-11).*[*"IBM Has a Watson Dilemma"*](https://www.wsj.com/articles/ibm-bet-billions-that-watson-could-improve-cancer-treatment-it-hasnt-worked-1533961147)*. Wall Street Journal.*[*ISSN*](https://en.wikipedia.org/wiki/ISSN_(identifier))[*0099- 9660*](https://www.worldcat.org/issn/0099-9660)*. Retrieved 2018-08-21.*
4. ["PRELIMINARY REPORT – HIGHWAY – HWY18MH010"](https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf) (PDF) . [National Transportation Safety Board](https://en.wikipedia.org/wiki/National_Transportation_Safety_Board). May 24, 2018.
5. [*" Kaggle "*](https://www.kaggle.com/docs/competitions)*. Kaggle competitions, Kaggle Inc.*
6. [*"Google achieves AI 'breakthrough' by beating Go champion"*](https://www.bbc.com/news/technology-35420579)*. BBC News. BBC. 27 January 2016.*
7. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436 (2015).
8. ["Deep learning in neural networks: An overview"](https://www.sciencedirect.com/science/article/pii/S0893608014002135). *Neural Networks 61: 85–117. 1 January 2015.* [ISSN](https://he.wikipedia.org/wiki/International_Standard_Serial_Number) [0893- 6080](https://www.worldcat.org/issn/0893-6080). [arXiv](https://he.wikipedia.org/wiki/ArXiv):[1404.7828](https://arxiv.org/abs/1404.7828). [doi](https://he.wikipedia.org/wiki/Digital_object_identifier):[10.1016/j.neunet.2014.09.003](https://dx.doi.org/10.1016%2Fj.neunet.2014.09.003)
9. Le Cun Y., Boser B., Denker J. S., Henderson D., Howard R.E., Hubbard W., and Jackel L.D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in neural information processing systems,* volume II, pages 396-404. Morgan Kaufmann Publishers Inc. San Francisco, CA.
10. [Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks](https://amzn.to/2S8qRdI), 1999, page 155
11. Stuart J. Russell, Peter Norvig (2010) [*Artificial Intelligence: A Modern Approach*](https://en.wikipedia.org/wiki/Artificial_Intelligence:_A_Modern_Approach)*, Third Edition*, Prentice Hall [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [9780136042594](https://en.wikipedia.org/wiki/Special:BookSources/9780136042594).
12. *LeCun, Yann.*[*"LeNet-5, convolutional neural networks"*](http://yann.lecun.com/exdb/lenet/)*.*
13. Yoshua Bengio , Xavier Glorot , DIRO, Universite de Montr ´ eal, Montr ´ eal, Qu ´ ebec, Canada, *"Understanding the difficulty of training deep feedforward neural networks"*
14. N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection”, *IEEE Computer Society*[*Conference on Computer Vision and Pattern Recognition*](https://en.wikipedia.org/wiki/Conference_on_Computer_Vision_and_Pattern_Recognition) (CVPR), pages 1:886–893, 2005
15. S. Piérard, A. Lejeune, and M. Van Droogenbroeck. "A probabilistic pixel-based approach to detect humans in video streams" *IEEE International Conference on Acoustics, Speech and Signal Processing*(ICASSP), pages 921– 924, 2011
16. Coco, Common Objects in Context, <http://cocodataset.org/#download>
17. [Linda G. Shapiro](https://en.wikipedia.org/wiki/Linda_Shapiro) and George C. Stockman (2001): “Computer Vision”, pp 279–325, New Jersey, Prentice- Hall, [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [0-13-030796-3](https://en.wikipedia.org/wiki/Special:BookSources/0-13-030796-3)
18. Hinton, Geoffrey; Sejnowski, Terrence (1999). *Unsupervised Learning: Foundations of Neural Computation*. MIT Press. [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [978-0262581684](https://en.wikipedia.org/wiki/Special:BookSources/978-0262581684).
19. Rosasco, L.; De Vito, E. D.; Caponnetto, A.; Piana, M.; Verri, A. (2004). ["Are Loss Functions All the Same?"](http://web.mit.edu/lrosasco/www/publications/loss.pdf) (PDF). *Neural Computation*. **16** (5): 1063–1076.  [CiteSeerX](https://en.wikipedia.org/wiki/CiteSeerX_(identifier)" \o "CiteSeerX (identifier)) [10.1.1.109.6786](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.109.6786).  [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1162/089976604773135104](https://doi.org/10.1162%2F089976604773135104).  [PMID](https://en.wikipedia.org/wiki/PMID_(identifier)) [15070510](https://pubmed.ncbi.nlm.nih.gov/15070510).
20. Pomerleau, D.A. (1990) Rapidly adapting artificial neural networks for autonomous navigation. In *Advances in Neural Information Processing Systems 3*
21. Ng, A. (2011) Machine Learning. <https://www.coursera.org/course/ml>
22. AWS Machine Learning Foundations Course <https://classroom.udacity.com/courses/ud090>
23. CS231n: Convolutional Neural Networks for Visual Recognition <http://cs231n.stanford.edu/>

