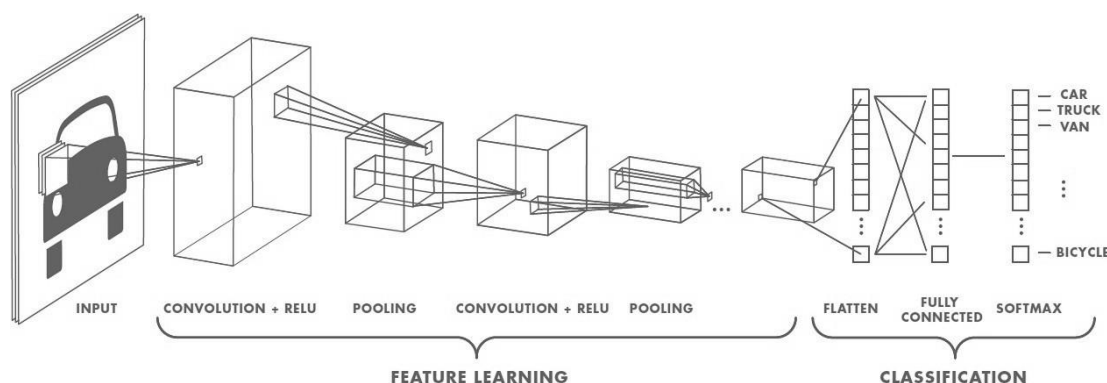




הפקולטה להנדסה

רשת נוירונים עמוקה לזיהוי הולכי רגל במערכת נהיגה אוטונומית

Deep Neural Network for Pedestrian Detection for an Autonomous Driving System



מאת:

אסף קסלסי, ת.ז. 301467098

מנחה:

פרופסור יוסף בן-עזרא

תוכן עניינים

4תקציר
4Abstract
61. מבוא
61.1. מוטיבציה
71.2. אבני דרך בציר הזמן של רשתות נוירונים ולמידת מכונה
81.3. רקע לרשתות נוירונים
81.3.1. הגדרת רשת נוירונים
81.3.2. כיצד רשת נוירונים דומה למוח ?
91.3.3. רשת נוירונים כתת מערכת בעולם הבינה המלאכותית
101.4. רשתות נוירונים מלאכותיות
101.4.1. ייצוג
111.5. סוגים של פונקציות הפעלה
111.5.1. פונקציית סף (Threshold function)
121.5.2. פונקציית סיגמוייד (Sigmoid function)
121.5.3. פונקציית המיישר (ReLU function)
141.5.4. פונקציית מיישר דולפת (Leaky ReLU function)
141.5.5. פונקציית מיישר פרמטרית (Parametric ReLU function)
151.6. ארכיטקטורות רשת
151.6.1. רשתות חד-שכבתיות ללא משוב (Feed Forward)
161.6.2. רשתות רב-שכבתיות ללא משוב (multi-layer no-feedback)
171.6.3. רשתות חוזרות (Recurrent)
181.7. סוגי בעיות
181.7.1. בעיות התאמה (Matching problem)
181.7.2. בעיות סיווג (Binary Classification problem)
181.7.3. בעיות סיווג מרובה-מחלקות: (Multi-class Classification problem)
191.8. למידה
201.8.1. למידה מפקחת (Supervised learning)
201.8.2. למידה ללא פיקוח (Unsupervised learning)
211.9. מדד ביצועים ופונקציית העלות
211.10. רשת קונבולוציה
231.11. מישקול
231.11.1. מישקול והטיה
231.11.2. אתחול משקולות
241.12. סגמנטציה
251.13. זיהוי הולכי רגל

26	אתגרים בזיהוי הולכים רגל:
26	Holistic detection 1.13.1
26	Patch-based detection 1.13.2
27	Motion-based detection 1.13.3
27	multiple cameras- based detection 1.13.4
28	data-driven CNN U-net based approach 1.13.5
29	2. ניסוי
29	2.1 כלים, סביבות תכנות ומערכות
29	2.1.1 מערך הנתונים (Data Base) -
29	2.1.2 הכנת וסידור הדאטה
29	2.1.3 בניית הרשת
29	2.1.4 ריצות ואימון הרשת
30	2.1.5 ניהול ומעקב אחרי גאנט ומשימות הפרויקט
30	2.1.6 החומרה בשימוש הפרויקט
31	2.2 תהליך ואימון
31	2.2.1 Hyper-Parameters
31	2.2.2 אימון הרשת – זמן כולל ופלטפורמת המערכת
32	2.3 ארכיטקטורת הרשת:
33	2.4 Over Fitting ודרכים למניעה
33	2.5 Batch normalization
34	2.6 Dropout
35	2.7 Pooling2d
36	2.8 Conv2d
36	2.9 Back-propagation
39	2.10 תאור הארכיטקטורה של הרשת:
41	תוצאות
41	דוגמאות למקרי קצה שלא צלחו
42	3. סיכום
43	3.1 הצעות להמשך עבודה בנושא וייעול המערכת:
44	4. ביבליוגרפיה
45	5. נספחים
45	5.1 רשימת נוסחאות:
46	5.2 ראשי תיבות
47	5.3 רשימת איורים:

תקציר

ברכבי הנהיגה אוטונומית נדרשות המערכות השונות לזיהוי רב של אובייקטים, זיהוי תמרורים, זיהוי רכבים שכנים, זיהוי מעברי חציה, רמזורים ועוד, אחד האתגרים הגדולים הוא זיהוי הולכי רגל, בפרויקט זה אציג פיתוח שלי של ארכיטקטורה חדשנית לזיהוי הולכי רגל במערכות נהיגה אוטונומית בעזרת רשת נוירונים עמוקה. השיטה המוצעת כאן, מבוססת על רשת נוירונים רב שכבתיות בתצורת

fully-connected מסוג CNN בארכיטקטורת U-Net, בגישה מבוססת דאטה.

בעזרת גישה זו של תחום הראיה ההמוחשבת ניתן לפתור את הבעיה המורכבת הזו. יש אפשרות בעזרת מצלמה אחת זיהוי של הולכי רגל רבים. אסביר כיצד 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.

דוח זה צולל לעומק התיאוריה שמאחורי כל מודול במערכת, האלגוריתמים והיישומים המעשיים שלהם, תהליך האינטגרציה של המודלים, ניתוח נתונים ומסקנות, אסקור את האתגרים העומדים בפני התחום ורעיונות והצעות לשיפור בעתיד

Abstract

In autonomous driving vehicles, the various systems are required for multiple object detection, road signs, neighboring vehicle identification, crosswalk identification, traffic lights and more. One of the major challenges is pedestrian detection.

In this project I will present my development of an innovative architecture for identifying pedestrians in autonomous driving systems with the help of a deep neural network.

The method proposed here is based on a network of multilayered neurons in a fully-connected CNN-type configuration, U-Net architecture, with a data-based approach.

With this approach in the field of computer vision this complex problem can be solved. It is possible with the help of one camera to detect many pedestrians. I will explain how the system classifies the pixels of the image as a background, when there is no certain identification in high percentages of the object being searched for (in my case a pedestrian), or indeed a pedestrian if a positive identification is obtained identified based on learning and understanding the scene from the learning process.

דוגמאות לאתגרי זיהוי הולכי רגל:



איור 1: אתגרי זיהוי הולכי רגל, דוגמאות:

(a) צבעי דמויות זהים לרקע, (b) הבדלי תאורה, (c) הסתרות, (d) הבדלי גודל, (e) גשם וערפל, (f) תנאי לילה

1. מבוא

בשביל לאפשר נהיגה אוטומטית אחד הדברים שצריך לעשות זה זיהוי סיטואציה, צריך להיות להיות מסוגלים לנתח סצנה של צומת עם רכבים שכנים, מעברי חציה רמזורים תמרורים והולכי רגל, להבין את איזה רזמור צריך לבחון מתי לסוע ואיך. בעיה לא פשוטה ומאוד מורכבת, בשנים נאחרונות בזכות השימוש של רשתות למידה עמוקה חלה התפתחות רבה ומטרת הפרויקט היא לפתור חלק מהבעיות, אתמקד בעית זיהוי הולכי הרגל

למידה חישובית (Machine Learning) הינה תחום במדע החוקר מצב בו אלגוריתמים יכולים "ללמוד" מה נכון לעשות ומה לא נכון לעשות מתוך אוסף נתון של data - מידע. למידה חישובית היא כלי חשוב בתחום ה-AI. אלגוריתמים אלו משתמשים במודל המבוסס על input 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]

1.1. מוטיבציה

למרות שתחום הלמידה החישובית עבר שינוי בתחומים מסוימים, אפליקציות רבות מבוססות רשתות נוירונים לרוב אינן מצליחות להביא את התוצאות הצפויות.^[2] 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], מקרה קיצוני אפילו הביא למוות.

מותה של איליין הרצברג (18 במרץ 2018) היה המקרה המוקלט הראשון של הולך רגל הרג 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.

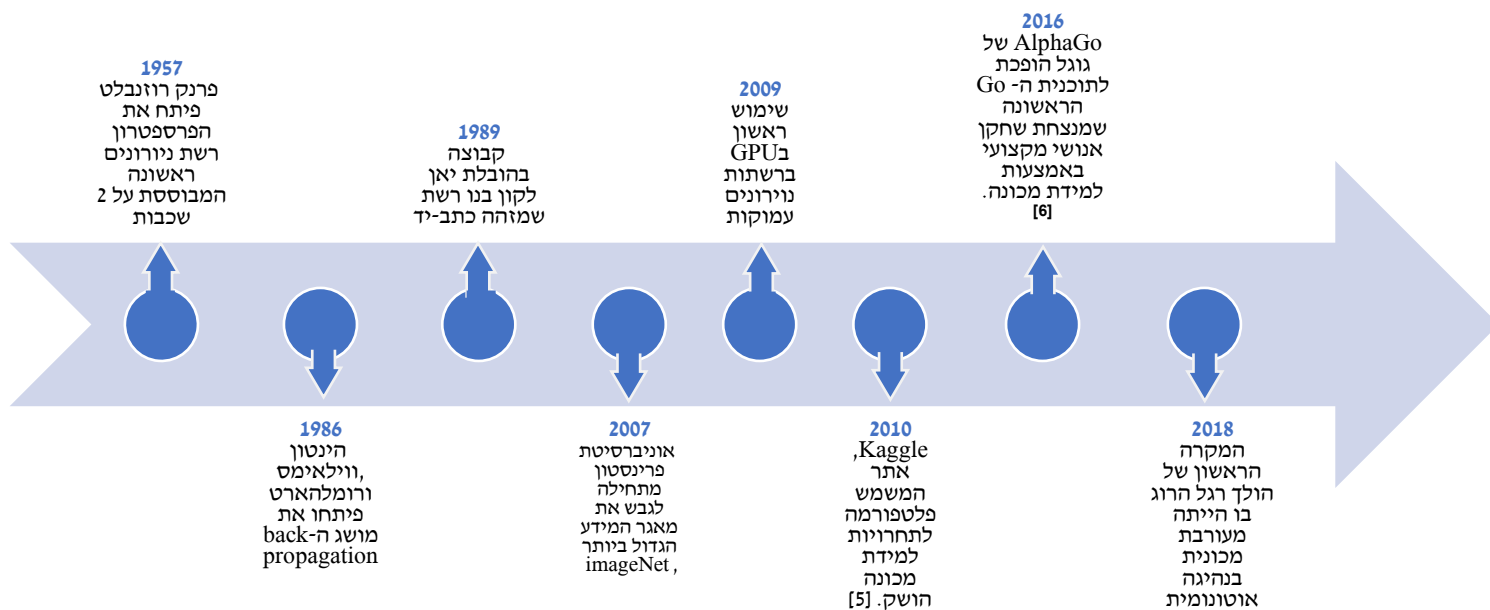
מייקל רמזי, מומחה לתחום נהיגה אוטונומית, איפיין את הסרטון המוצג כמראה של "כישלון מוחלט של המערכת בזיהוי אדם שנראה בבירור לאורך זמן לא מבוטל במספר פריימים.

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 נירונים ולמידת מכונה

בהתבסס על ניתוח מאמרים וספרים בתחום [1] [8] [13] יצרתי את ציר הזמן הזה החל מ-1957 שבו היה שימוש ראשון ברשת נירונים המבוססת על 2 שכבות, דרך זיהוי כתב יד ב-1989, גיבוש מאגרי מידע לשימש הציבור ב-2007, שימושים בחומרה מתקדמת וב-GPU לצורכי אימון רשתות ועד 2018 בו המקרה הראשון של הולך רגל הרוג בו הייתה מעורבת מכונית בנהיגה אוטונומית.



איור 2 - אבני דרך בציר הזמן של רשתות נירונים ולמידת מכונה

1.3. רקע לרשתות נוירונים

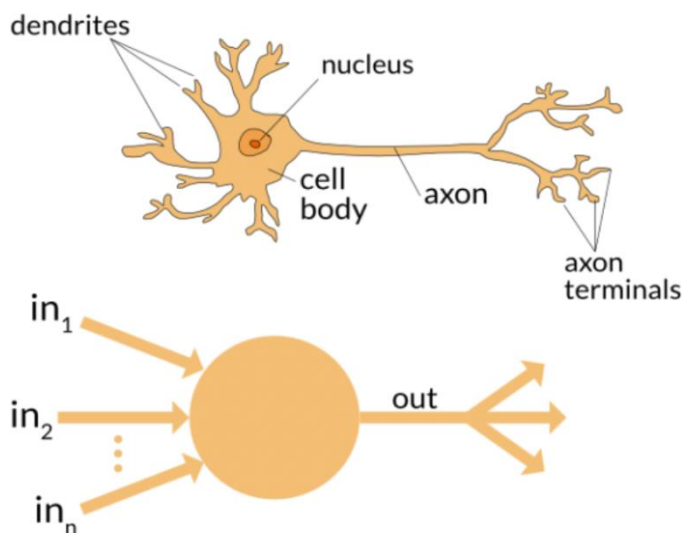
1.3.1 הגדרת רשת נוירונים

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 גודל 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).^[7]



איור 3 השוואה בין רשת בגוף האדם לרשת נוירונים מלאכותית

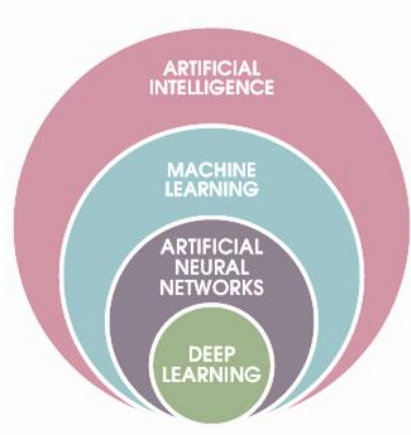
1.3.3 רשת נוירונים כתת מערכת בעולם הבינה המלאכותית.

AI עצמו הוא למעשה המעגל הגדול והחיצוני ביותר בסדרה של ארבעה מעגלים קונצנטריים. המעגל הבא המקוון

בתוך AI הוא, ML, או Machine Learning למידת מכונה באופן לא מפתיע, הוא החלק הלומד של AI

אבל ML עצמו מסתמך על המעגל הבא בתוכו ANN, או Artificial Neural Networks, המעגל הרביעי והפנימי

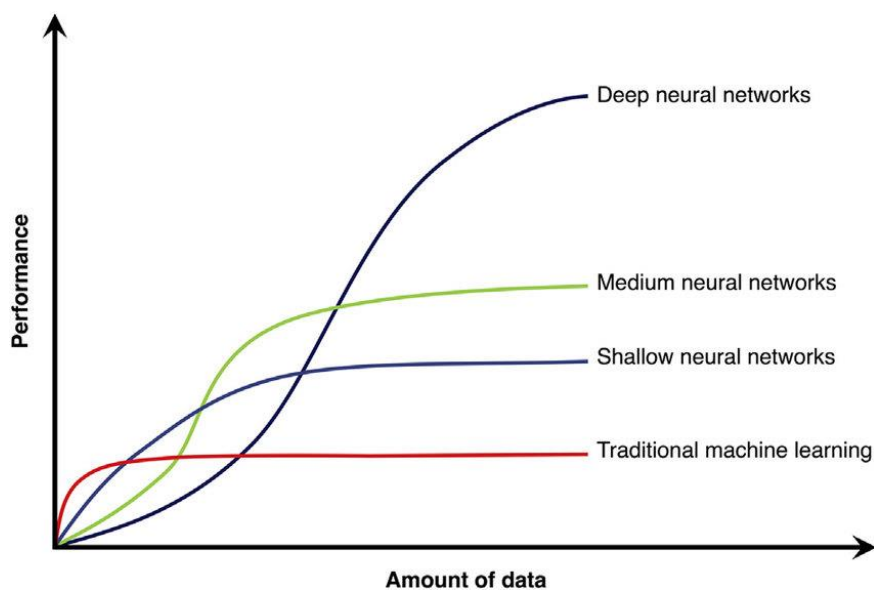
ביותר, וזה שעליו מסתמך כל מעגל אחר, הוא deep learning.



איור 4 רשת נוירונים כתת מערכת בעולם הבינה המלאכותית.

בפרייקט זה אתמקד ברשתות עצביות עמוקות או deep neural networks מכיוון שלאורך התקופה האחרונה לפי

המאמרים המובילים זאת השיטה שהוכיחה את עצמה בצורה הטובה ביותר בהשוואה לשאר הגישות.



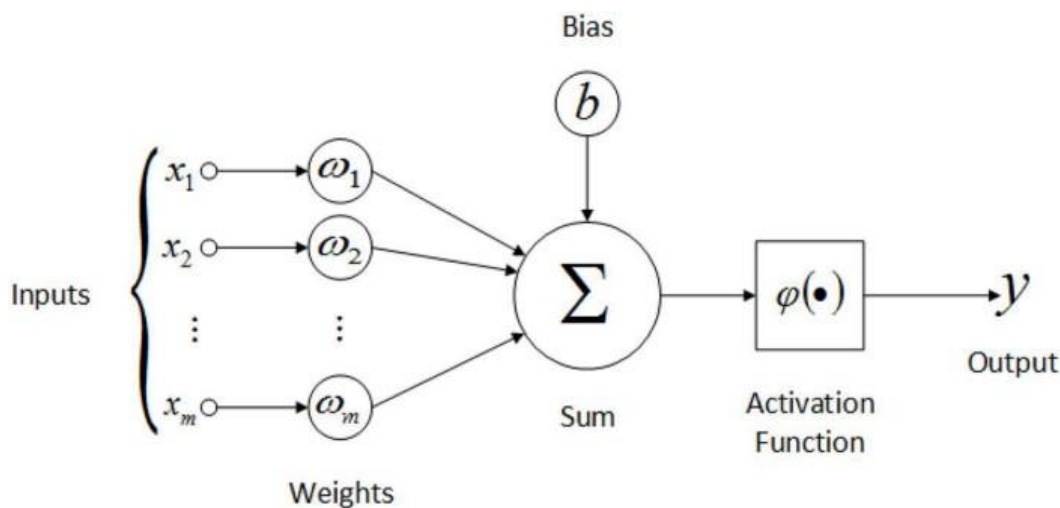
איור 5 השוואה בין הביצועים בין השיטות השונות אל מול כמות המידע.

1.4. רשתות נוירונים מלאכותיות

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 x_m , המגיע בתור קלט לסינפסה j המחוברת לנוירון k , מוכפל במשקל הסינפטי w_m
2. מחבר (adder) של אותות ה input
3. פונקציית הפעלה activation function הגורמת שפונקציית ה input של הנוירון יכולה להיות אי-ליניארית.
4. אות הטיה (bias) מסומן בסימון b מטרת אות זה היא הגברה או הנמכה של האות המסוכם טרם כניסתו לפונקציית ההפעלה.



איור 6 מודל של נוירון

1.5 סוגים של פונקציות הפעלה

פונקציות ההפעלה מגדירה את הפלט של הנוירון כתלות בפוטנציאל ההפעלה שלו ^[21]

כמה מהתכונות שעשויות להיות שימושיות לפונקציית אקטיבציה :

לא ליניארית - כאשר הפונקציה אינה ליניארית, ניתן להוכיח שרשת עצבית של שתי שכבות היא משערך אוניברסלי. פונקציית הזהות אינה מקיימת את תכונה זו. כאשר מספר שכבות משתמשות באקטיבציה של פונקציית זהות, הרשת שקולה למודל של שכבה אחת.

גזירה ברציפות – זו תכונה רצויה המסייעת לאופטימיזציה מבוססת גרדיאנט. פונקציה מדרגה בינארית אינה דיפרנציאבילית ב-0, והנגזרת שלה היא 0 עבור ערכים אחרים, ולכן שיטות מבוססות גרדיאנט עשויות להיתקע.

טווח – כאשר הטווח של פונקציית האקטיבציה סופי, שיטות אימון מבוססות גרדיאנט נוטות להיות יציבות יותר, כי ייצוג התבנית מושפע מאד ממשקולות מוגבלים. כאשר הטווח אינסופי, האימון בדרך כלל יעיל יותר כיוון שייצוג תבניות משפיע על רוב המשקולות.

מונוטונית – כאשר פונקציית האקטיבציה היא מונוטונית, שטח השגיאה הקשור במודל עם שכבה יחידה מובטח להיות קמור.

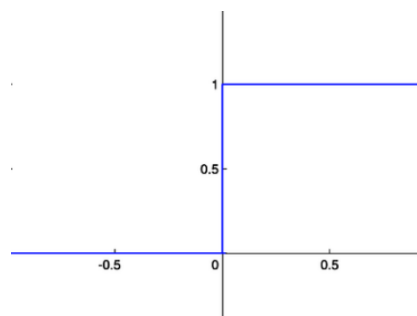
קרובות לזהות ליד נקודת הראשית – כאשר פונקציית האקטיבציה ניחנת בתכונה זו, הרשת לומדת ביעילות כאשר המשקולות מאותחלים לערכים אקראיים קטנים. כאשר תכונה זו לא מתקיימת בפונקציית האקטיבציה, נדרשת תשומת לב באתחול ערכי המשקולות.

כדי שהרשת תוכל ללמוד בצורה מיטבית, אנו נצטרך להשתמש בפונקציות אקטיבציה לא ליניאריות. זאת מפני שהרכבה של שתי פונקציות ליניאריות נותנת פונקציה ליניארית, למעשה לא משנה כמה שכבות יש לנו ברשת, אם לא מוסיפים אי-ליניאריות לרשת, הרשת תוכל ללמוד רק פונקציות ליניאריות. לכן שימוש בפונקציות לא ליניאריות מאפשר לנו ללמוד יחסים מורכבים יותר.

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

1.5.1. פונקציית סף (Threshold function)

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



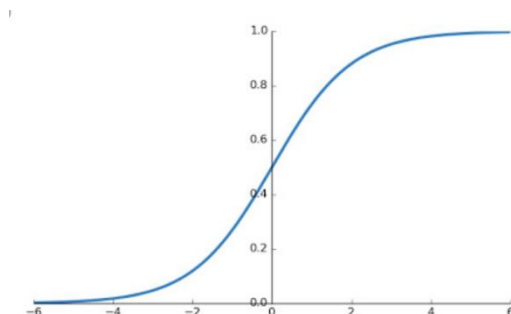
$$(1) \quad \varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

איור 7 - גרף פונקציית סף

1.5.2 פונקצית סיגמוייד (Sigmoid function)

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).

$$(2) \quad \varphi(v) = g(v) = \frac{1}{1 + \exp(-av)}$$

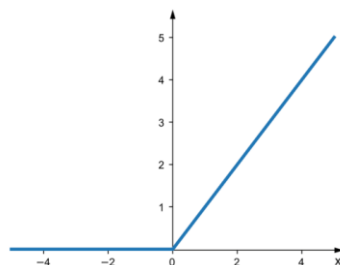


איור 8 - גרף פונקציית

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.

$$(3) \quad g(v) = \max(0, v)$$



איור 9 - גרף פונקציית המיישר

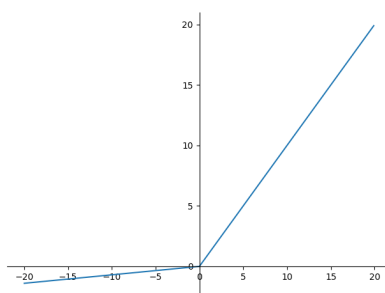
למרות מספר חסרונות שאפרט בהמשך, ReLU (נכון לשנת 2020) היא פונקצית ההפעלה הפופולארית ביותר עבור רשתות עצביות עמוקות.

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 $x < 0$.

1.5.4. פונקציית מיישר דולפת (Leaky ReLU function)

פונקציית הפעלה מאוד דומה ל ReLU הרגילה אך הבדל משמעותי קורה בערכים השליליים: היא מאפשרת 0.01 של גרדיאנט חיובי כשהיחידה לא אקטיבית [11]. 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.

$$(4) \quad \varphi(v) = \begin{cases} v & \text{if } v > 0 \\ 0.01v & \text{if } v \leq 0 \end{cases}$$

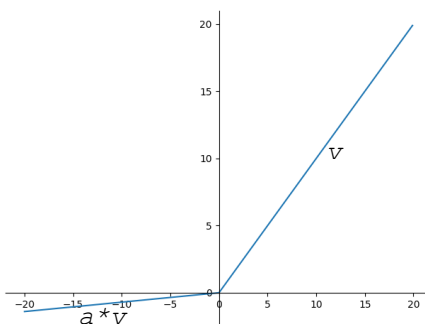


איור 10 -גרף פונקציית מיישר

1.5.5. פונקציית מיישר פרמטרית (Parametric ReLU function)

פונקציית הפעלה מאוד דומה ל ReLU הרגילה אך לוקחת את הזליגה צעד אחד קדימה על ידי הגדרת כמות הזליגה בפרמטר שנלמד כמו שאר הפרמטרים של רשת הנוירונים [12].

$$(5) \quad \varphi(v) = \begin{cases} v & \text{if } v > 0 \\ av & \text{if } v \leq 0 \end{cases}$$



איור 11 - גרף פונקציית מיישר פרמטרית

1.5.6. פונקציית סופטמקס (Softmax function)

פונקציית 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 בפונקציית *softmax* לבעיות סיווג רב-כיתיות. (*multiclass classification problems*) פונקציית זו מחזירה את ההסתברות לדאטא מסוים השייך לכל מחלקה בנפרד. בעת בניית רשת לבעיה רב-כיתית, 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:

$$(6) \quad \sigma(z)_j = f \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad , \quad \text{for } j=1, \dots, K$$

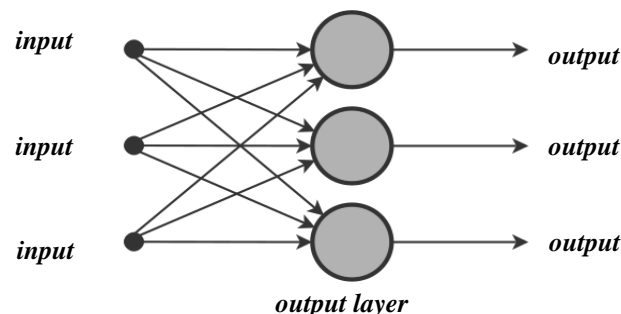
1.6. ארכיטקטורות רשת

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:

1.6.1. רשתות חד-שכבתיות ללא משוב (Feed Forward)

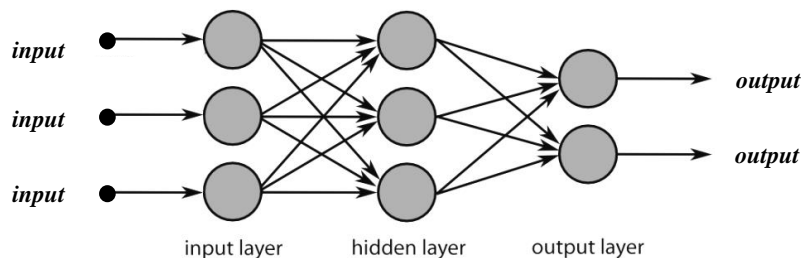
רשת בעלת שכבות היא רשת שבה הנוירונים מסודרים בצורת שכבות קשורות. אם אנסה לפשט את התהליך בשביל ההבנה הצורה הכי פשוטה של רשת שכבות מכילה רק שכבה אחת, שכבת-קלט של מקור, שכבה זו מעבירה מידע לפלט של נוירונים, אבל חשוב לשים לב שברשתות מהסוג הזה המידע לא יכול לזרום לצד השני כלומר אין משוב. ובאנגלית נקראת גם רשת מזינה קדימה (*Feed-Forward*) [13] באיור מוצגת רשת בעלת 3 מקורות ו3 נוירונים רשת כזאת נקראת גם "חד-שכבתית" מכיוון שבנויה משכבת חישוב אחת בלבד – שכבת הפלט.



איור 12 - רשת חד-שכבתית ללא משוב

6.2.1. רשתות רב-שכבתיות ללא משוב (multi-layer no-feedback)

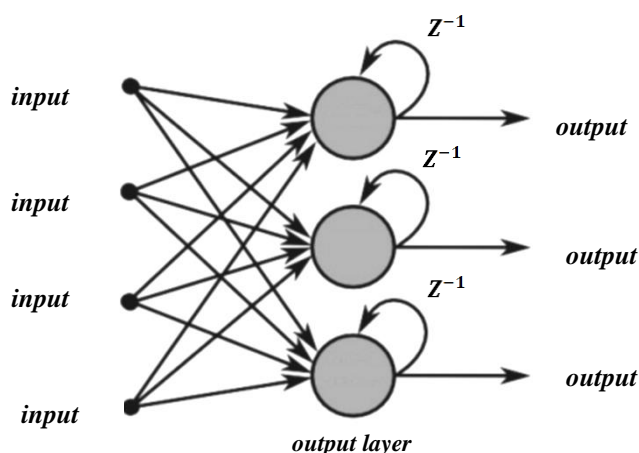
רשתות מהסוג הזה מכילות שכבות חישוב נוספות הנקראות שכבות חבויות או נסתרות (*hidden layers*) לכן גם הנוירונים בשכבה הזו נקראים נוירונים חבויים (*hidden neurons*). [13] תפקיד השכבות החבויות הוא "להתערב" בין שכבת הקלט ושכבת הפלט. בעזרת הוספת שכבות חבויות הרשת מסוגלת להפיק סטטיסטיקות מסדר גבוה יותר. זו כמובן 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.



איור 13- רשת רב-שכבתית ללא משוב

1.6.3 רשתות חוזרות (Recurrent)

רשתות חוזרות שונות במיוחד מהרשתות חסרות המשוב בגלל העובדה שלהן יש לולאות משוב (*feedback loops*). באיור 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 Z^{-1} , יחידות אלה תורמות לאופי האי-ליניארי של רשתות הנוירונים [13].



איור 14 - רשתות חוזרות

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 (Matching problem)

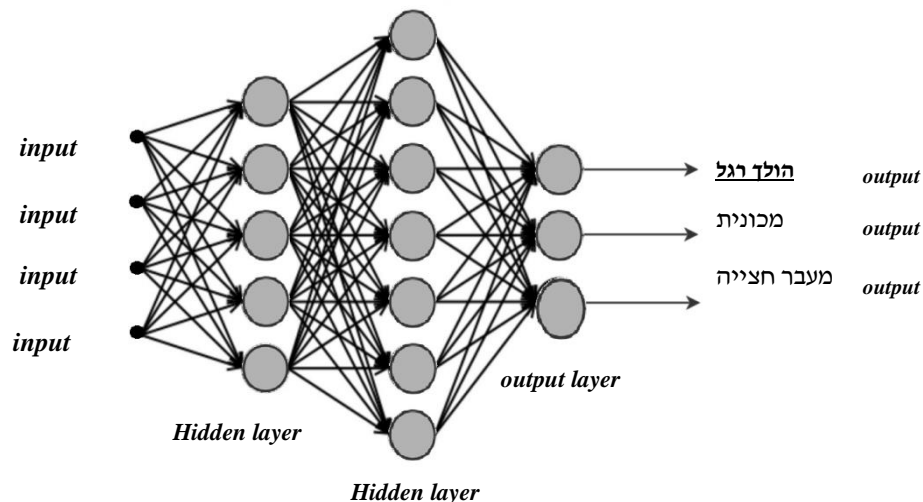
בעיות מסוג "התאמה" יש לבסס החלטה מושכלת, בהינתן הקלט, מהו המספר שהכי "מתאים" לקלט (המשמעות שלמספר זה תלויה בבעיה המדוברת), לדוגמה, בעיית מחיר רכב: נתונים מאפיינים שלרכב כמו למשל שנה, מודל, יצרן, מספר יד, נפח תא מטען, נצילות דלק וכו' והפלט הוא המחיר שבו יימכר הרכב הספציפי הזה.

7.2.1. בעיות סיווג (Binary Classification problem)

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: $y \in \{0, 1\}$

7.3.1. בעיות סיווג מרובה-מחלקות: (Multi-class Classification problem)

בבעיות מהסוג "בעיות סיווג מרובה-מחלקות" האתגר הוא לקבל החלטה מתוך קטגוריות קיימות לאיזו קטגוריה, שייך הקלט. 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:



איור 15- רשת רב שכבתית לסיווג מרובה מחלקות

ברשת הזו לדוגמא ניתן לראות שהמודל בנוי מ3 נוירונים-פלט כשה $output$ שלהם בינארי (0 או 1) זוהי דוגמא לבעיית סיווג, ליד כל נוירון פלט מצוינת המחלקה אותה "מייצג" הנוירון. במקרה זה $y \in \{0, 1\}^3$

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

0	0	1
0	1	0
1	0	0
מייצג מעבר חצייה	מייצג מכונית	מייצג <u>הולך-רגל</u>

1.8. 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 שמשוגלות להמשיך ללמוד במהלך הפרודקשן ידועות כמערכות דינמיות.

בהינתן קלט x לרשת, הרשת מפיקה ערך y^* שהוא קירוב לערך $y=f(x)$ בתהליך שנקרא התפשטות קדימה (forward propagation) אם נתונים אסוף של inputs נחשב לכל x input את שגיאת הקירוב בין y^* ל- y ונגדיר את השגיאה הכוללת באופן הבא:

$$(6) J(\theta) = \frac{1}{N} \sum_i^N \text{loss}(f(x^i; \theta), y^i)$$

כאשר N הוא מספר הפעמים שנחזור על האימון, y הוא ערכו של y באיטרציה i , $f(x^i; \theta)$ הוא הקירוב של y באיטרציה i , ו- $\text{loss}(f(x^i; \theta), y^i)$ הוא ערך הפונקציה לחישוב השגיאה בין הערך האמיתי של y לבין הקירוב שלו על ידי הרשת.

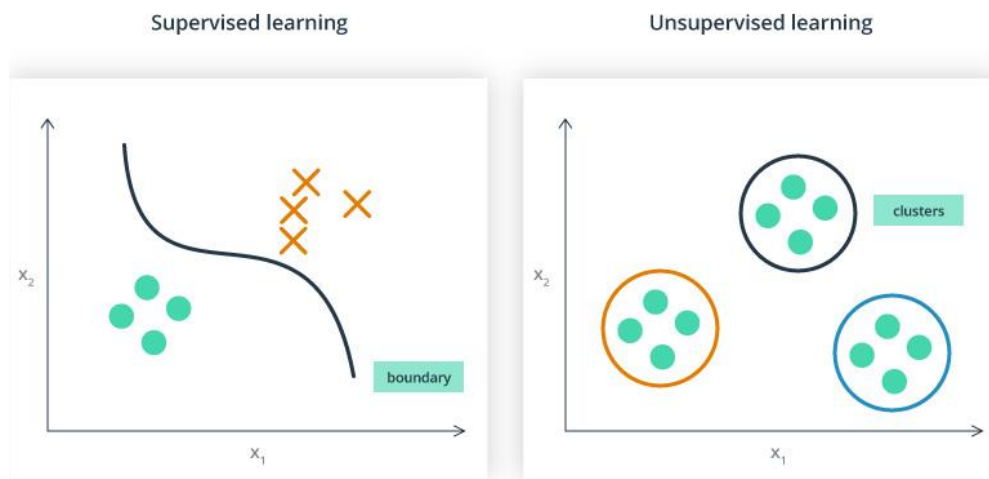
ישנן מספר שיטות למידה שונות, שיטת למידה מפקחת, ללא פיקוח או השילוב (מצב היברידי) של השניים. ככלל למידה היא מודל לסוגי השיטות המשמשות להדרכת המערכת לאילו סוגי תוצאות היא אמורות להפיק. אלגוריתם הלמידה הוא השיטה המתמטית הספציפית המשמשת לעדכון המשקולות הסינפטיים הבין-עצביים במהלך כל איטרציה של אימון. ניתן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

1.8.1 למידה מפקחת (Supervised learning)

למידה מפקחת ממפה קלט לפלט על בסיס דוגמאות של זוגות קלט-פלט. למידה מפקחת 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 מפקחת 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^[10].

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 תוויות אנושיות, 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 מפקחת למחצה, וריאציה קשורה, עושה שימוש בטכניקות מפקחות ולא מפקחות^[9].



איור 16 - למידה מפקחת לעומת למידה ללא פיקוח

9.1. מדד ביצועים ופונקציית העלות

פונקציית העלות או פונקציית הפסד (Loss Function) מעריכה באיזו מידה אלגוריתם הלמידה הולם את המידע עליו הוא לומד, אם אלגוריתם הלמידה מנבא בצורה שגויה אז פונקציית ההפסד תקבל ערך גבוה, בעוד שאם הניבוי נכון

הרי שהיא תקבל ערך נמוך פונקציית ההפסד מחושבת ביחס למדד ביצועים כלשהו לדוגמא, במשימות סיווג ניתן להשתמש במדד דיוק (Accuracy) אשר מחשב את אחוז הסיווגים הנכונים ביחס לכל הסיווגים. המדידה נעשית על חלק מאוסף המידע שהמודל לא התאמן עליו כלומר אוסף המידע מחולק ל2 חלקים, קבוצת אימון וקבוצת מבחן, שגיאת האימון מוגדרת להיות השגיאה של פונקציית ההפסד על קבוצת האימון, שגיאת המבחן מוגדרת כשגיאה על קבוצת המבחן. מטרת אלגוריתם הלמידה היא בעצם למזער ככל שניתן את שגיאת האימון ואת הפער שבין שגיאות האימון לשגיאת המבחן,

פונקציית אלו 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].

בהינתן X כמרחב הווקטורי של כל הכניסות האפשריות,

$Y = \{-1, 1\}$ כמרחב הווקטורי של כל היציאות האפשריות,

אנו מעוניינים למצוא פונקציה $f: X \rightarrow \mathbb{R}$ (7)

אשר ממפה בצורה הטובה ביותר את $\vec{x} \rightarrow y$ (8)

אך בגלל מידע לא שלם, רעש במדידה או רכיבים הסתברותיים בתהליך הבסיסי, יתכן שאותו \vec{x} ינבא y שונה

כתוצאה מכך, מטרת בעיית הלמידה היא למזער את הסיכון הצפוי, המוגדר כ- $I[f]$ (9)

$$\int_{X \times Y} V(f(\vec{x}), y) p(\vec{x}, y) d\vec{x} dy$$

כאשר $V(f(\vec{x}), y)$ היא פונקציית העלות ו $p(\vec{x}, y)$ היא פונקציית הצפיפות או PDF של התהליך שיצר את

הדאטה.

לפונקציית 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]

10.1. רשת קונבולוציה

פעולת קונבולוציה מסומנת ב* ומוגדרת עבור פונקציית מקור x ופונקציית מסנן w באופן הבא:

$$s(t) = (x * w)(t) = \int x(a)w(t - a)da \quad (10)$$

אפשר כמובן גם להגדיר את פעולת הקונבולוציה במישור הדיסקרטי:

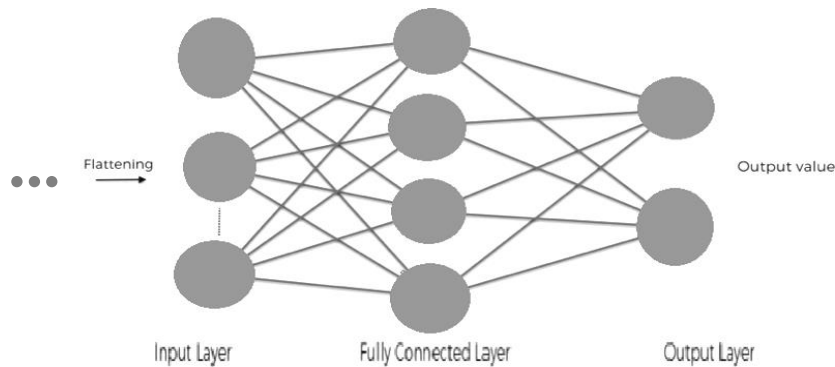
$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (11)$$

אפשר גם לחשב עבור מספר מימדים, למשל עבור 2 מימדים:

$$s(i, j) = (I * W)(i, j) = \sum_m \sum_n I(m, n)W(i - m, j - n) \quad (12)$$

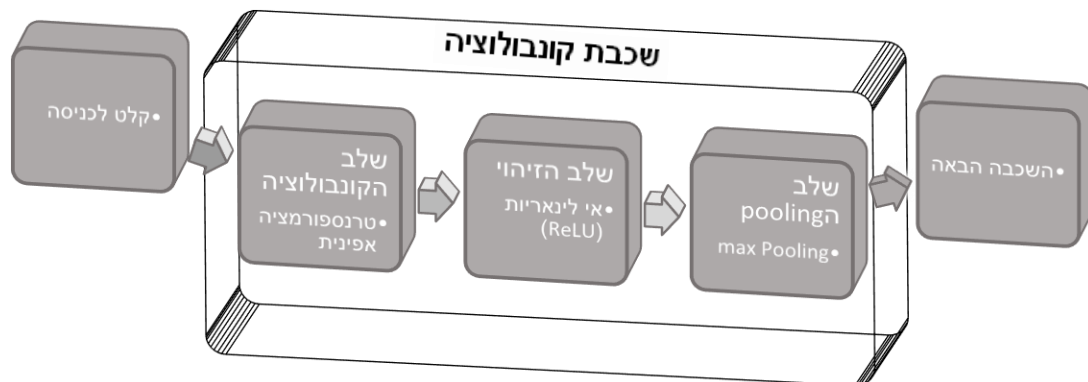
בלמידה עמוקה, רשת עצבית מבוססת קונבולוציה (CNN, או ConvNet) היא מחלקה של רשתות עצביות עמוקות, המיושמת לרוב לניתוח דימויים חזותיים.

רשתות נוירונים רב שכבתיות בד"כ מרביתן בתצורת fully-connected, כלומר כל נוירון בשכבה אחת מחובר לכל הנוירונים בשכבה הבאה (ראה איור). 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 ה-Loss. 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 17 - רשת קונבולוציה פולי קונקטד.

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, (שלב זה משמש להקטנת מספר המימדים של הקלט באמצעות דגימה מקטינה (down-sampling) של התוצאה מהשכבה הקודמת, אחת השיטות לעשות זאת היא MAX_POOLING שעליה אדבר בפרק הניסוי). לאחר מכן שכבות מחוברות לחלוטין (fully connected) ושכבות נורמליזציה, המכונות שכבות נסתרות מכיוון שהכניסות והתפוקות שלהן מוסוות על ידי פונקציית ההפעלה והפיתרון הסופי.



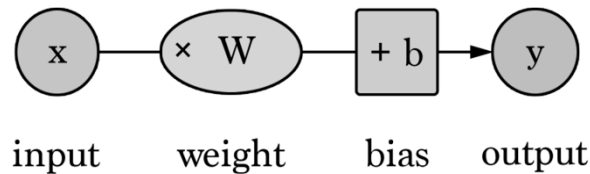
איור 18 - פירוט שכבות הקונבולוציה

1.11. 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^[13].

1.11.1 מישקול והטיה

משקולות (Weights) והטיה (Bias) הם שני פרמטרי למידה בתוך הרשת. רשת עצבית תגדיל 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 19- משקל והטיה

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^[14].

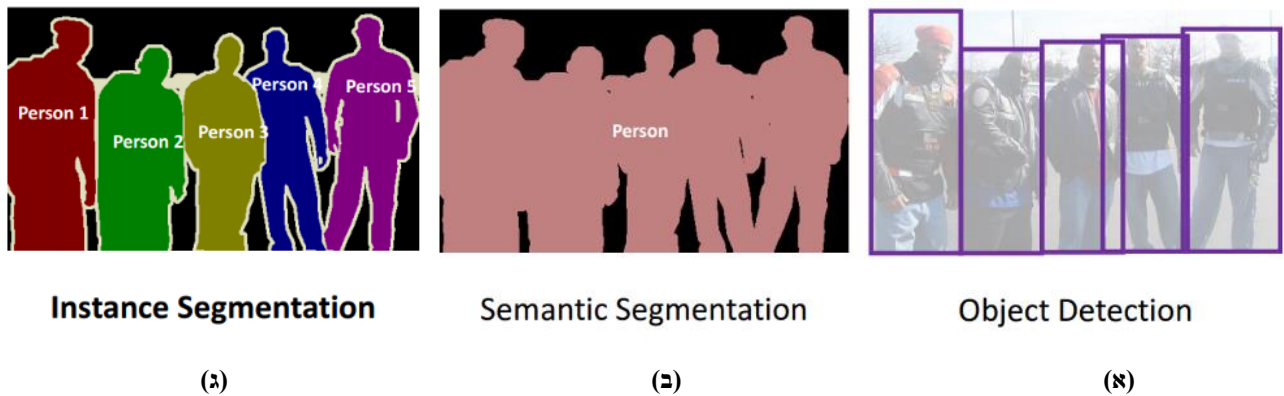
$$(10) \quad np.random.randn([matrix\ shape]) * \sqrt{\frac{2}{([input\ layer\ size]) + ([output\ layer\ size])}}$$

1.12. סגמנטציה

בכדי לנתח את הסצנה המלאה עלינו להבין מה מופיע בתמונה ולחלק אותה למקטעים מופרדים וברורים. 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^[15]. 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.

תחילה אסביר את ההבדל בין זיהוי אובייקט, פילוח (סגמנטציה) סמנטי וסגמנטציה לפי מופע (instance):



איור 20 - הבדלים בסוגי זיהוי אובייקטים.

בתמונה א' ניתן לראות אובייקטים מאותו הקלאס (אנשים) ולכולם יש bounding box שנותן לוקליזציה של האובייקט במרחב התמונה.

בתמונה ב' ניתן לראות שישנם מספר אובייקטים מאותו הקלאס כשלכל אובייקט סגמנטציה הדוקה. בתמונה ג' ניתן לראות שישנם מספר אובייקטים מאותו קלאס אך מזוהים כאובייקטים שונים ונפרדים אחד מהשני, כשלכל אובייקט סגמנטציה הדוקה, וצבע אחר המתאר אובייקט 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 ניוונים מבוססת CNN בארכיטקטורת מקודד מפענח (Unet) המשתמשת כ database במאגר תמונות COCO לטובת משימת זיהוי.

1.13. 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 הווידאו. יש לה שימוש רחב גם ביישומי רכב בגלל הפוטנציאל לשיפור מערכות הבטיחות. יצרני רכב רבים (למשל וולוו, פורד, GM, ניסאן, טויוטה, ססלה, יונדאי ועוד) מציעים זאת כאופציה של ADAS (Advanced driver-assistance systems) כלומר מערכות סיוע מתקדמות לנהג.

כיום ישנם 5 דרגות לטכנולוגיית נהיגה אוטונומית:

דרגה 0 – אף פונקציית נהיגה לא אוטומטית.

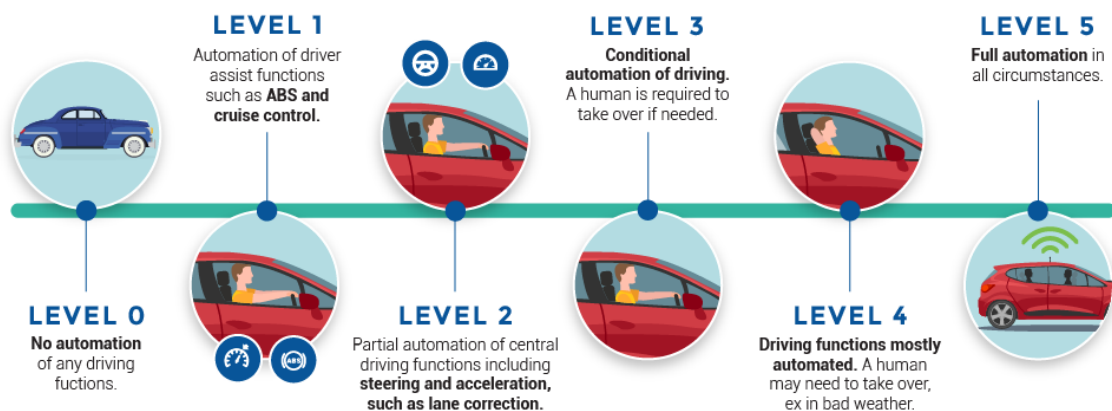
דרגה 1 – אוטומציה של פונקציות עזרה לנהג כמו שיוט וABS

דרגה 2 – אוטומציה חלקית של פונקציות נהיגה מרכזית כמו היגוי ותיקון סטייה מנתיב

דרגה 3 – נהיגה אוטונומית חלקית, הנהג עדיין נדרש להשתלט אם יש צורך

דרגה 4 – רוב הנהיגה אוטונומית, הנהג נדרש להשתלט במצבי מזג האוויר קשים

דרגה 5 – נהיגה אוטונומית מלאה בכל התרחישים.



איור 21 – חמשת הדרגות לטכנולוגיית נהיגה אוטונומית

אתגרים בזיהוי הולכים רגל:

- סגנון לבוש שונה במראה, גודל שונה, תאורה שונה.
- מגוון תנוחות אפשריות שונות, אין פאות מוגדרות.
- מהירות זיהוי לצורך תגובה
- נוכחות של אביזרים מסתירים, חסימות והסתרות תכופות

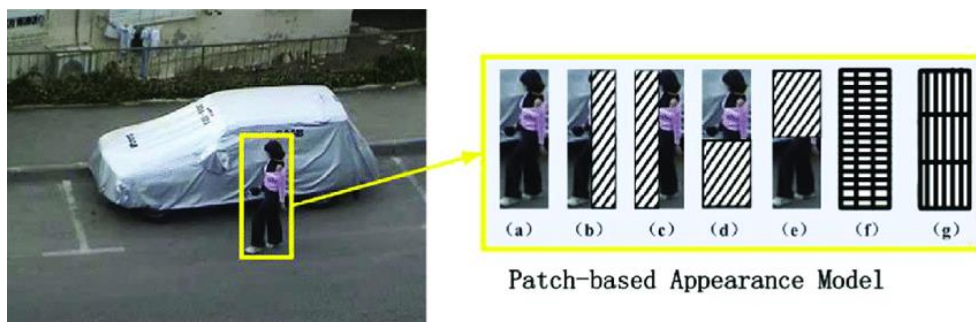
למרות האתגרים, כפי שראינו [באיור 1](#) זיהוי הולכי הרגל עדיין נותר תחום מחקר פעיל ב computer vision בשנים האחרונות. הוצעו מספר רב של גישות:

Holistic detection 1.13.1

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, אחרות משתמשות בתכונות מקומיות כמו HOG (histogram of oriented gradients) היסטוגרמה של גראדיאנט מונחה ^[16]. החיסרון של גישה זו הוא שאפשר להשפיע בקלות על הביצועים כתוצאה מעומס רעש ברקע והסתרות רבות.

Patch-based detection 1.13.2

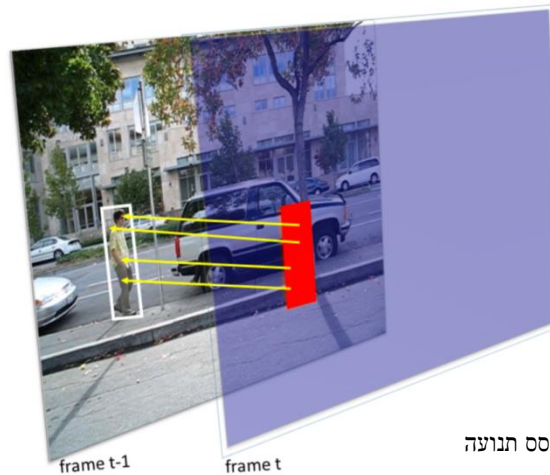
גישה המשלבת גם זיהוי וגם סגמנטציה (Implicit Shape Model (ISM). מערך נתונים של מראה מקומי נלמד במהלך תהליך האימון. בתהליך הזיהוי משתמשים בפיצורים מקומיים שנמצאו כדי להתאים זה לזה כנגד ערכי מערך הנתונים שנלמדו. ניתן להשיג תוצאות איתור סופי על ידי שיפור נוסף של השערות אלה. היתרון בגישה זו הוא שרק מספר מצומצם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



איור 22 - זיהוי אובייקטים מבוסס חלקים

Motion-based detection 1.13.3

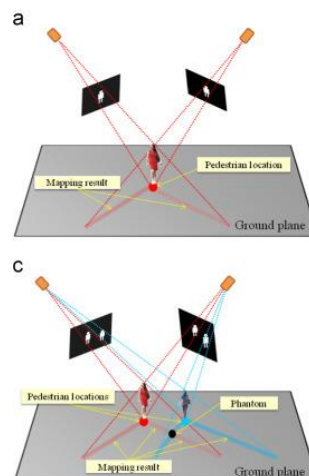
כאשר יש תנאים מתאימים מאוד ספציפיים (מצלמה קבועה, תנאי תאורה טובים וכו'), שיטת חיסור רקע (**Background subtraction**) יכולה לסייע בזיהוי הולכי רגל. חיסור רקע מסווג את הפיקסלים של קטעי הווידאו כרקע, כאשר לא מזוהה שום תנועה, או הולכי רגל אם מתגלה תנועה. הליך זה מדגיש את הצלליות של כל אלמנט נע בסצנה, כולל אנשים. בשיטה זו פותח אלגוריתם [17] 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.



איור 23 - זיהוי אובייקטים מבוסס תנועה

multiple cameras- based detection 1.13.4

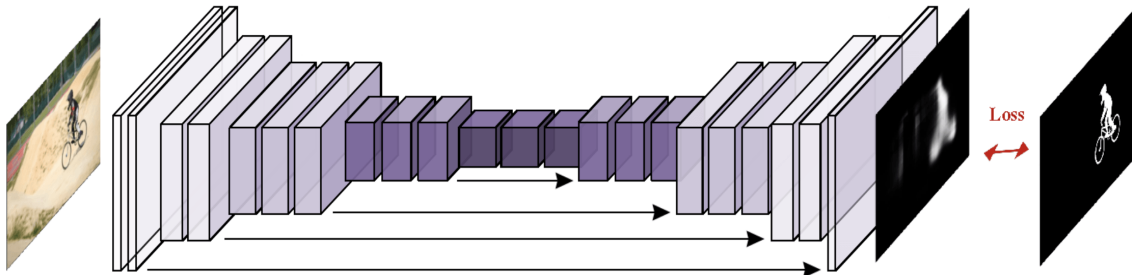
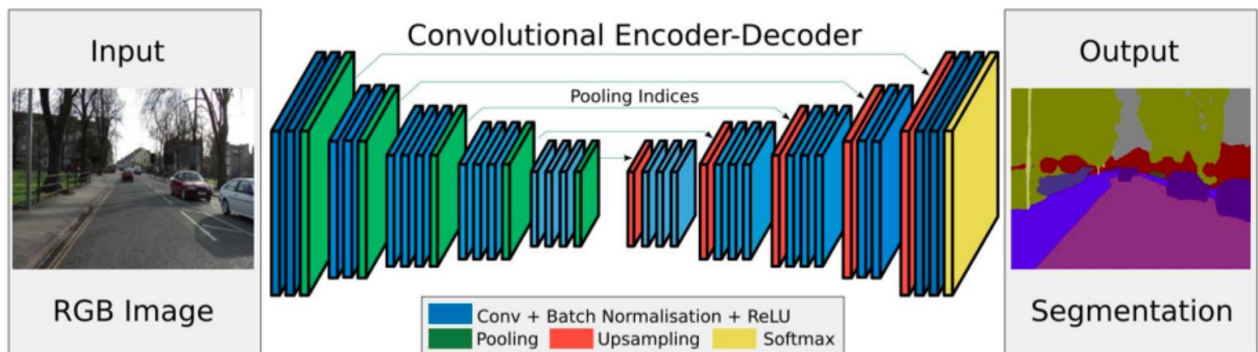
שיטה זו מבוססת על מספר מצלמות מכוילות לזיהוי הולכי רגל רבים. בגישה זו, מישור הכביש מחולק לתאי רשת אחידים ולא חופפים, בדרך כלל בגודל 25 על 25 (ס"מ). הגלאי מייצר Probability Occupancy Map (POM) מפת הסתברות התפוסה, הוא מספק הערכה של ההסתברות של כל תא רשת שיתפס על ידי אדם. בהינתן שניים עד ארבעה קטעי וידאו 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.



איור 24 - זיהוי אובייקטים מבוסס מספר מצלמות

data-driven CNN U-net based approach 1.13.5

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.



איור 25 – דוגמאות לארכיטקטורת רשת מקודד-מפענח (UNET)

2. ניסוי

2.1 כלים, סביבות תכנות ומערכות

בחלק זה אפרט אודות הכלים, התוכנות, המערכות וסביבת העבודה שבחרתי עבור הפרויקט,

2.1.1 מערך הנתונים (Data Base) - Microsoft COCO RGB images

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 ^[17], במאגר זה משתמשים בקבצי json לתאור האנוטציות בתמונות. מאגר התמונות שאיתו אני עובד מחולק ל2 תיקיות מסודרות:

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

2.1.2 הכנת וסידור הדאטה python using PyCharm IDE

עבור הכנת וסידור הדאטה אני בחרתי להשתמש בשפת פייטון עקב הנוחות שלה והידע המקדים שלי בשפה, כמו-כן השימוש ב IDE של פייצ'ארס מאוד נוח לדיבוג הקוד בצורה גראפית וויזואלית.

2.1.3 בניית הרשת python using TensorFlow and Keras

עבור בניית הרשת בחרתי את השילוב של טנסור פלו וקארס,

TensorFlow היא ספריית קוד פתוח ללמידת מכונה, המפותחת על ידי חברת גוגל לבנייה ואימון רשתות עצביות ^[18]

Keras היא ספריית רשת עצבית עם קוד פתוח שנכתבה בפיתון. ספרייה זו מסוגלת לרוץ על גבי TensorFlow, שנועד לאפשר ניסויים מהירים עם רשתות עצביות עמוקות, קאראס מתמקד בהיותו ידידותי למשתמש, מודולרי וניתן להרחבה. ^[19]

2.1.4 ריצות ואימון הרשת Online Jupyter notebook and google colab

את הרצות ואימון הרשת עשיתי גם לוקאלית על המחשב שלי וגם ONLINE על שירותי ענן של גוגל colab מחברת **Jupyter** (בעבר נקראה מחברות IPython Notebooks) היא סביבת חישוב אינטראקטיבית מבוססת-אינטרנט ליצירת מסמכי מחברת של Jupyter. ^[20]

Colaboratory, או "Colab" בקיצור, הוא מוצר מקבוצת המחקר של גוגל. Colab מאפשר לכל אחד לכתוב ולהריץ קוד פיתון שרירותי באמצעות הדפדפן ומתאים במיוחד ללימוד מכונה וניתוח נתונים. מבחינה טכנית יותר, Colab הוא שירות שדומה מאוד לשל Jupyter שפירטתי קודם, שפשוט לא דורש שום הגדרה לשימוש, תוך מתן גישה חופשית למשאבי מחשב כולל GPUs של חוות שרתים של גוגל (עקב סיבוכיות אימון הרשת נדרשים משאבי מחשב GPU)



2.1.5 ניהול ומעקב אחרי גאנט ומשימות הפרויקט

עבור ניהול ומעקב אחרי משימות הפרויקט בחרתי להשתמש בפלטפורמת Trello, עקב הנוחות שלה והידע המקדים שלי בתוכנה הרשת בחרתי את השילוב
טרלו הוא יישום על גבי דף אינטרנט המבוסס רשימת Kanban, המהווה חברה בת של אטלאסיאן.
משתמשים יכולים ליצור לוחות משימות, עם עמודות שונות ולהעביר את המשימות מעמודה לעמודה בהתאם לסטאטוס. בדרך כלל עמודות כוללות סטטוסים של משימות כמו to do, בתהליך, בוצע.



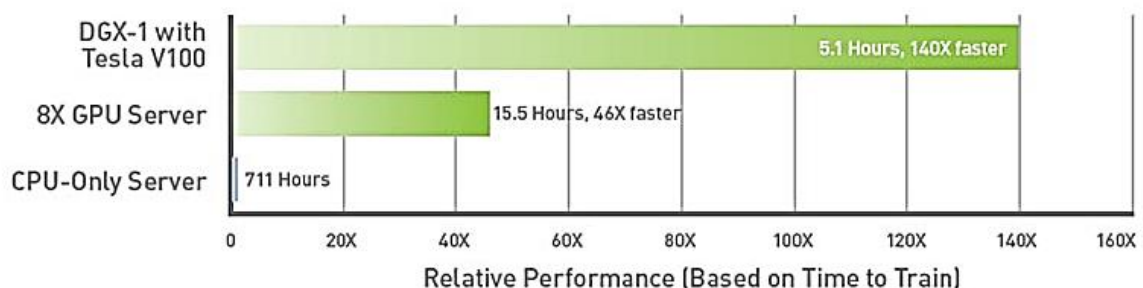
2.1.6 החומרה בשימוש הפרויקט

מבחינת חומרה לשימוש הפרויקט השתמשתי ב2 סביבות.
הסביבה הלוקאלית שלי וסביבת שירותי הענן של גוגל. (google Colab)
כמו שפירטתי קודם, שירותי הענן של גוגל קולאב מאפשר גישה חופשית למשאבי מחשב כולל GPUs של חוות שרתים של גוגל (עקב סיבוכיות אימון הרשת נדרשים משאבי מחשב GPU)

בסביבה הלוקאלית שלי :
CPU: Intel i7 (9thGen) 9700k
GPU: nvidia geforce 1050ti
RAM: 32GB DDR4
Disk: SSD kingston

בסביבת הענן של גוגל :
CPU: Intel i9 (10thGen) 9700k
GPU: 2 x nvidia TitanX
RAM: 64GB DDR4
Disk: SSD kingston

NVIDIA DGX-1 Delivers 140X Faster Deep Learning Training



איור 26 – זמן אימון רשת בעזרת GPU מסוג TESLA לעומת זמן אימון רשת בעזרת CPU

2.2 תהליך 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 ורכבים אז לכל אובייקט יש מספר סידורי שמתאר את הקלאס, למשל הערך שמאפיין אנשים הוא 1 והערך שמאפיין רכבים הוא 7 אז יצרתי מסכה שבכל מקום שיש בן אדם שם יהיה ערך 1 ובמקום שיש רכב יהיה ערך 2 כך אני יכול לסווג את המידע שלי בצורה נוחה למודל. בגלל מגבלות של זיכרון נאלצתי לשנות את הרזולוציה של התמונות ל-64x64 כך יכולתי לאמן את המודל על כל המאגר

Hyper-Parameters 2.2.1

בלימוד מכונה hyper-parameters הם פרמטרים שערכם משמשים לשליטה בתהליך הלמידה. לעומת זאת, הערכים של פרמטרים אחרים (בדרך כלל משקולות צומת) נגזרים באמצעות תהליך הלמידה עצמו. הזמן הדרוש ללמידה ובדיקת מודל יכול להיות תלוי בבחירת הפרמטרים שלו. [\[23\]](#) היפר-פרמטר הוא בדרך כלל מספר רציף או מספר שלם, מה שמוביל לבעיות אופטימיזציה כאשר יש סוגים מעורבים. קיומם של כמה פרמטרים מותנה בערכם של אחרים, למשל גודל כל שכבה מוסתרת ברשת יכול להיות מותנה במספר השכבות שהוגדרו.

הגדרתי את hyper-parameters שלאחר הרבה איטרציות של ניסוי וטעייה מצאתי שמניבים את עקומת הלמידה הטובה ביותר, ה accuracy הגבוה ביותר וה loss הנמוך ביותר למודל ולארכיטקטורה שבניתי:

lr =	1e-4	# Learning rate
beta_1 =	0.9	# beta 1 - for adam optimizer
beta_2 =	0.95	# beta 2 - for adam optimizer
epsilon =	1e-7	# epsilon - for adam optimizer
epochs =	50	# number of epochs
bs =	32	# batch size
dp =	0.5	# dropout probability

2.2.2 אימון הרשת – זמן כולל ופלטפורמת המערכת.

Train time

2ms per sample

173s per Epoch

2.5 hours of training (total of 50 epochs)

2.3 ארכיטקטורת הרשת:

בארכיטקטורה שבנית ישרו של שכבות קונבולוציה עם ReLU ובסוף שכבת פולי-קונקט שמאפשרת טיפול במקרים אי-ליניארים ומשמרת את ה dimensions וכך משפרת את ה robustness 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.

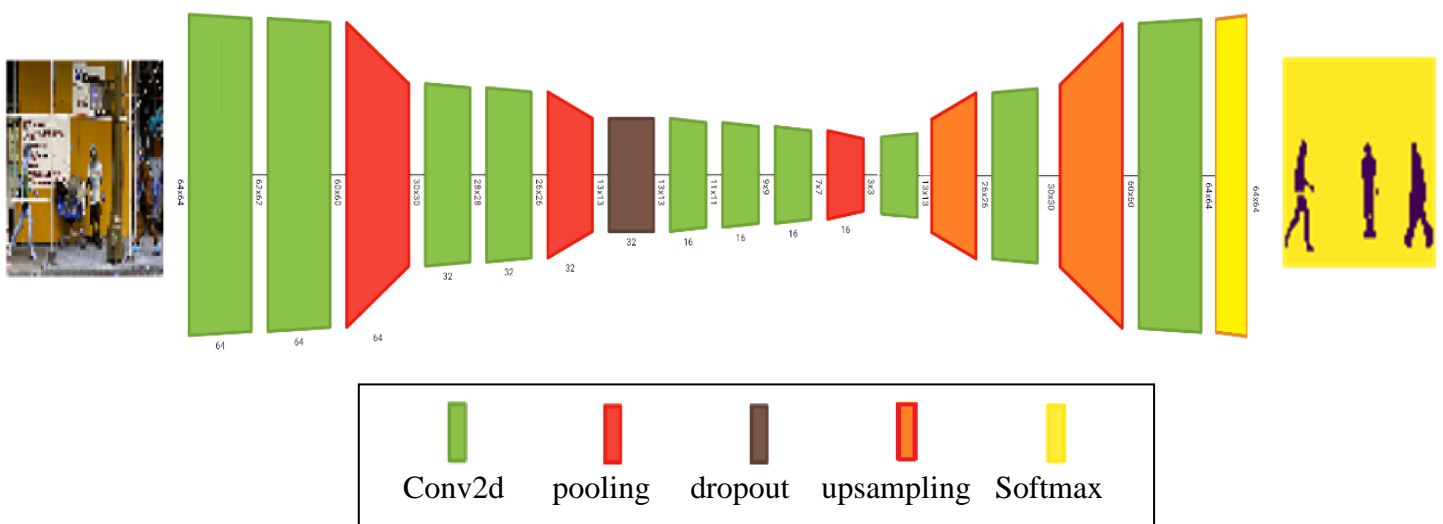
רשתות אלו בנויות בארכיטקטורה כזו שגודל השכבות החבויות יורד והולך ואחר כך, הוא נפתח חזרה, בדרך כלל, לגודל המקורי, כלומר, רשת בארכיטקטורה הכוללת "צוואר בקבוק". במצב זה, החלק של הרשת מהשכבה של כניסת הנתונים ועד לצוואר בקבוק מהווה "מקודד" של המידע ואילו השכבה מצוואר הבקבוק ועד לשכבת יציאת הנתונים מהווה "מפענח".

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.



איור 27 – ארכיטקטורת הרשת (מקודד-מפענה – *UNET*)

2.4 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 ניוונים ברשת ויכולת החישוב עולה על הממדיה 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

2.5 Batch normalization

טכניקה שהייתה אצלי בשימוש נרחב נקראת batch normalization או בעברית, נירמול חלקים, שיטה זו מאפשרת לזרז ולייצב את תהליך הלמידה, זה מאפשר להשתמש בממד למידה גבוה יותר כך גם הלמידה מהירה יותר, אחרי פונקציית אקטיבציה אנחנו מנרמלים את המידע לשכבה הבאה לדוגמה כאשר יש שיכבה חבויה שמורכבת מ

$$(11) \quad Z^1, Z^2, \dots, Z^m \rightarrow Z^{L(i)}$$

אחשב את המיצוע והשונות

$$(12) \quad \mu^2 = \frac{1}{m} \sum_i (Z^i)$$

$$(13) \quad \sigma^2 = \frac{1}{m} \sum_i (Z_i - \mu)^2$$

כאשר נוסחאת הנרמול נראית כך:

$$(14) \quad Z_{norm}^i = \frac{Z^i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

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

$$(15) \quad \tilde{z}^i = \gamma * Z_{norm}^i + \beta$$

במידה ו

$$(16) \quad \gamma = \sqrt{\sigma^2 + \varepsilon}; \quad \beta = \mu$$

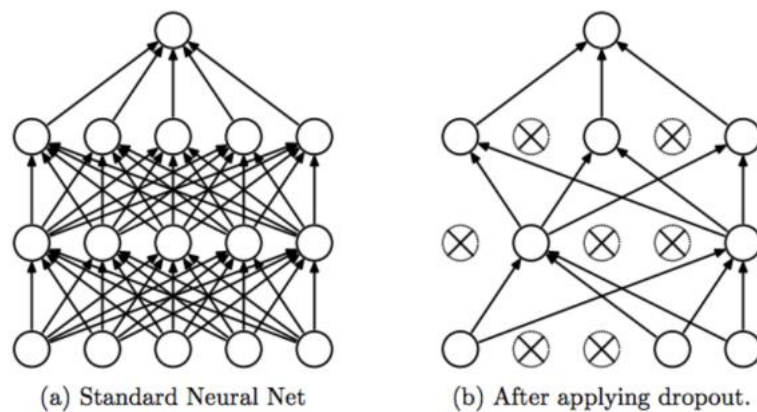
$$\Rightarrow \tilde{z}^i = z^i$$

כעת ניתן לשחק עם הפרמטרים של גמא ובטא ואז המיצוע והשונות יכולים להיות מנורמלים סביב איזה אזור שנרצה במודל. במידה ולא סיפקתי פרמטרים - נירמול המיצוע קרוב לאפס ואת השונות קרוב ל1

שיטת נוספת שבחרתי להשתמש בה למניעת overfitting היא דרופ-אאוט

Dropout.2.6

טכניקת רגולריזציה שמאפשרת להתגבר על הנושא של תאימות יתר (over fitting) של המודל שבזמן הלימוד בהסתברות מסוימת מאפסים 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 28 - רשת סטנדרטית (a) לעומת רשת לאחר dropout (b)

2.7. Pooling2d

Pooling מתבצע ברשתות עצביות כדי להפחית את השונות ואת מורכבות החישוב.

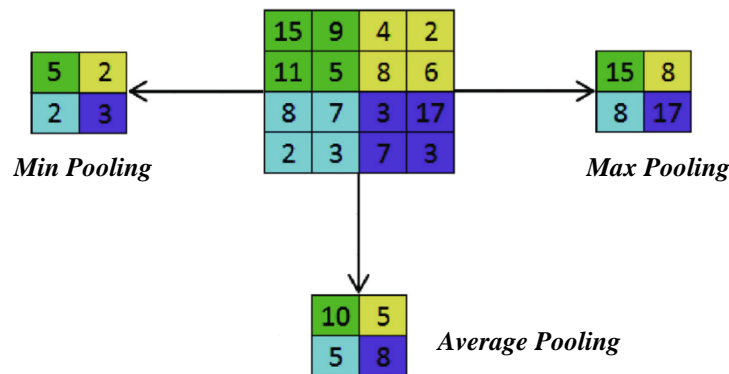
ישנן שלוש שיטות נפוצות ובסיסיות ל pooling הנמצאות בשימוש נרחב:

Max Pooling: נבחר הפיקסל בעל הערך המרבי של החלון.

Min Pooling: נבחר הפיקסל בעל הערך המינימלי של החלון.

Average Pooling: נבחר הערך הממוצע של כל הפיקסלים בחלון.

כמו שציינתי קודם מטרת שכבת הפולינג היא להוריד את גודל הכניסה, שכבה זו מרדדת את הרעש ומרדדת שינויים קלים כמו סיבוב או הזזה של התמונה, כמו כן שכבה זו מבוססת על פילטור של חלון רף. במקרה של מקס-פולינג בשכבה זו נקח את התוצאה הגבוהה ביותר בכל חלון דוגמה בתמונה בצד ימין, (כפי שהשתמשתי במודל שלי חלון 2×2) בכל חלון ערך המקסימום נלקח למפת הפיצ'רים במילים אחרות כל ערך מקסימום הוא מייצג את החלון של 2×2 במפת הפיצ'רים. שכבה זו בתמונות קטנות יכולה לעזור בפתרון האובר-פיטינג



איור 29 - הדגמת מספרית של הבדל בתהליכי ה pooling בחלון של 2×2 על מטריצה בגודל 4×4

אי אפשר לומר ששיטת pooling מסוימת עדיפה על פני אחרת. הבחירה בפעולת pooling נעשית על סמך הנתונים העומדים לרשותך. שיטת פולינג ממוצעת מחליקה את התמונה ומכאן שלא ניתן לזהות את הפיצ'רים החדים כשמשתמשים בממוצע. Pooling מקסימלי למשל בוחר את הפיקסלים הבהירים יותר מהתמונה. זה שימושי כאשר רקע התמונה כהה ואנחנו מעוניינים רק בפיקסלים הבהירים יותר של התמונה. לדוגמה: במערך נתונים של MNIST הספרות מיוצגות בצבע לבן והרקע שחור - במקרה כזה נעשה שימוש ב max pooling. באופן דומה, במקרה הפוך נשתמש ב min pooling.

Conv2d 2.8

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 3×3) 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 הולכי רגל היא מזהה פיצ'רים כגון כתפיים, רגליים, ראש, ידיים וכדומה לאחר מכאן היא מזהה פיצ'רים באזורי העניין של הראש כמו עיניים וכדומה בסופו של דבר המטרה היא לנסות לזהות את כל הפיצ'רים שאיתם אפשר לסווג את האובייקט ולשייך אותו לקבוצה מסוימת

Back-propagation 2.9

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 הנוירונית. 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 האמיתי שיקרה כתוצאה משינוי משקלים

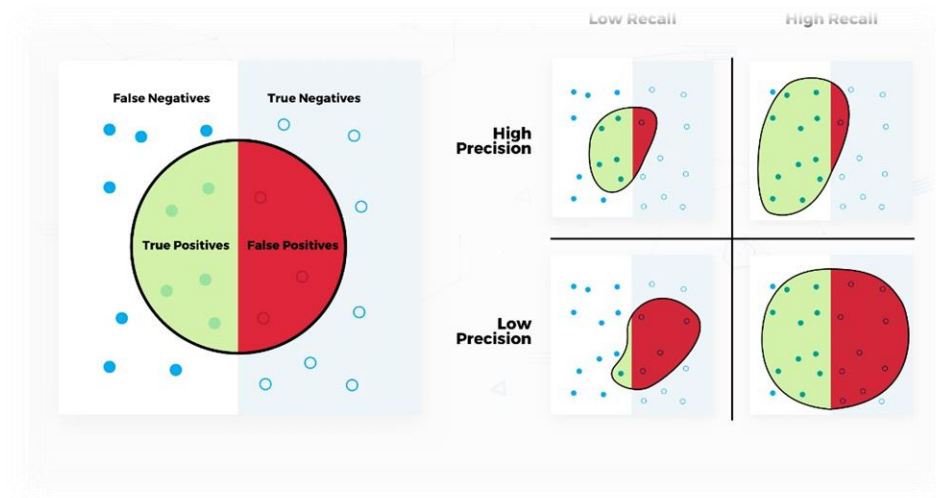
לאחר מכן הוצאתי גרף למידה כדי לוודא התכנסות, והצגתי את התוצאות בעזרת מטריצת מבוכה

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

לטובת הגדרת הדיוק נגדיר תחילה את הסיווגים השונים :

האלגוריתם חזה שהקטע מעניין וצדק	TP = True Positive כאשר
האלגוריתם חזה שהקטע לא מעניין וטעה	FN = False Negative
האלגוריתם חזה שהקטע מעניין וטעה.	FP = False Positive



איור 30 - סיווג חיוביים האמיתיים (TP) מול סיווג החיוביים הכוזבים (FP) והשליליים

אחת הדרכים למדוד את טיב הבחירה היא להשתמש בממד F-score , ממד סטטיסטי לבדיקת דיוק שמתחשב בשתי הנוסחאות הבאות :

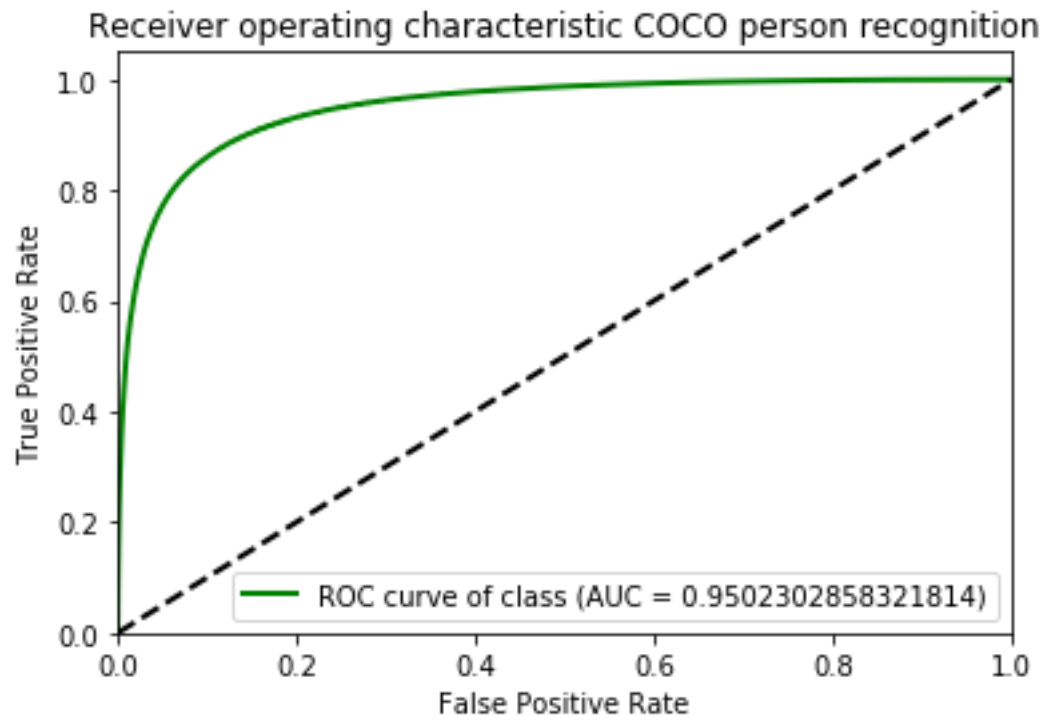
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

מדד F-score מוגדר באופן הבא:

$$Fscore = \frac{precision \times recall}{precision + recall} \times 100$$

ניתן לראות שהגעתי למודל טוב שמצליח לזהות את הקטגוריות השונות בצורה דיי טובה לפי גרף ה ROC ,

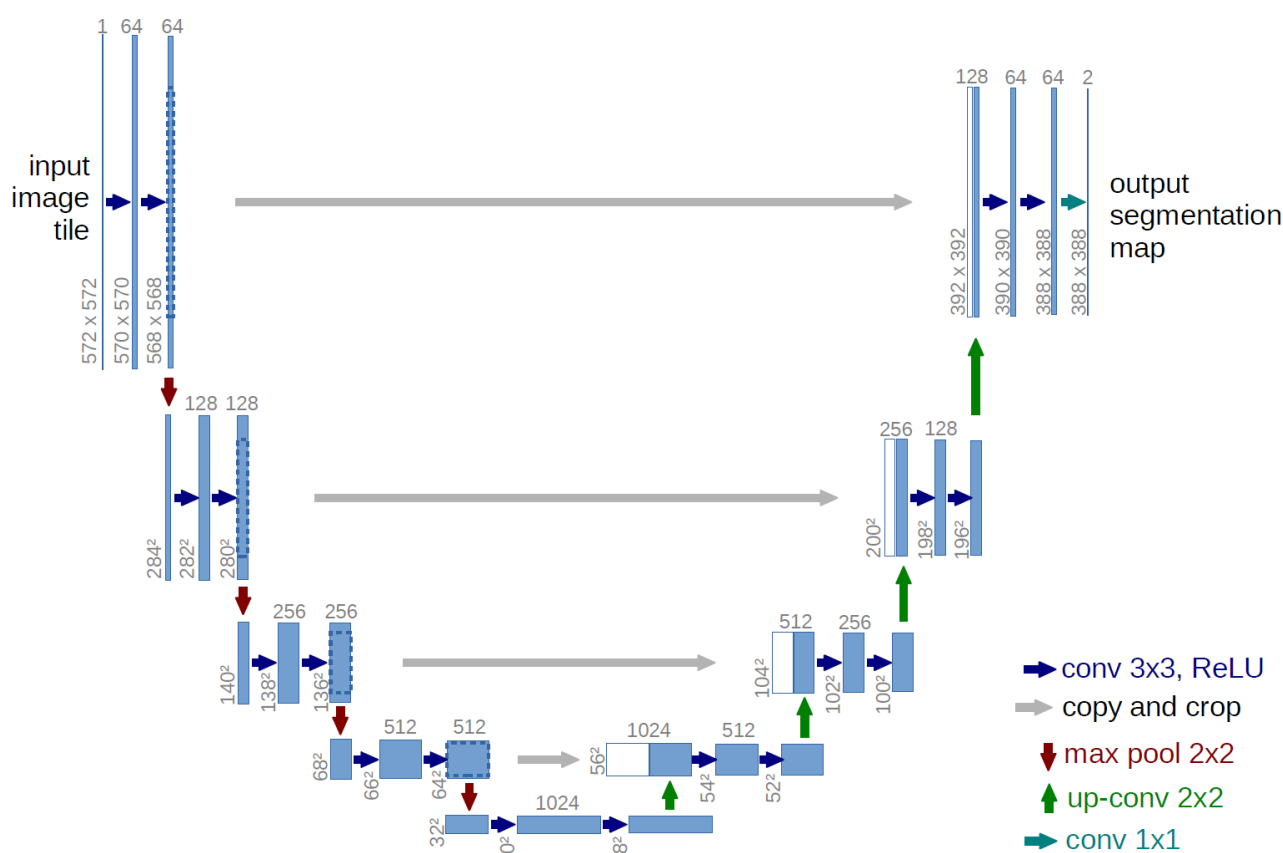


איור 31- גרף ה ROC עבור זיהוי הולך רגל.

2.10 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

הארכיטקטורה נראת בצורת האות U מפה השם (דוגמה לארכיטקטורת Unet)

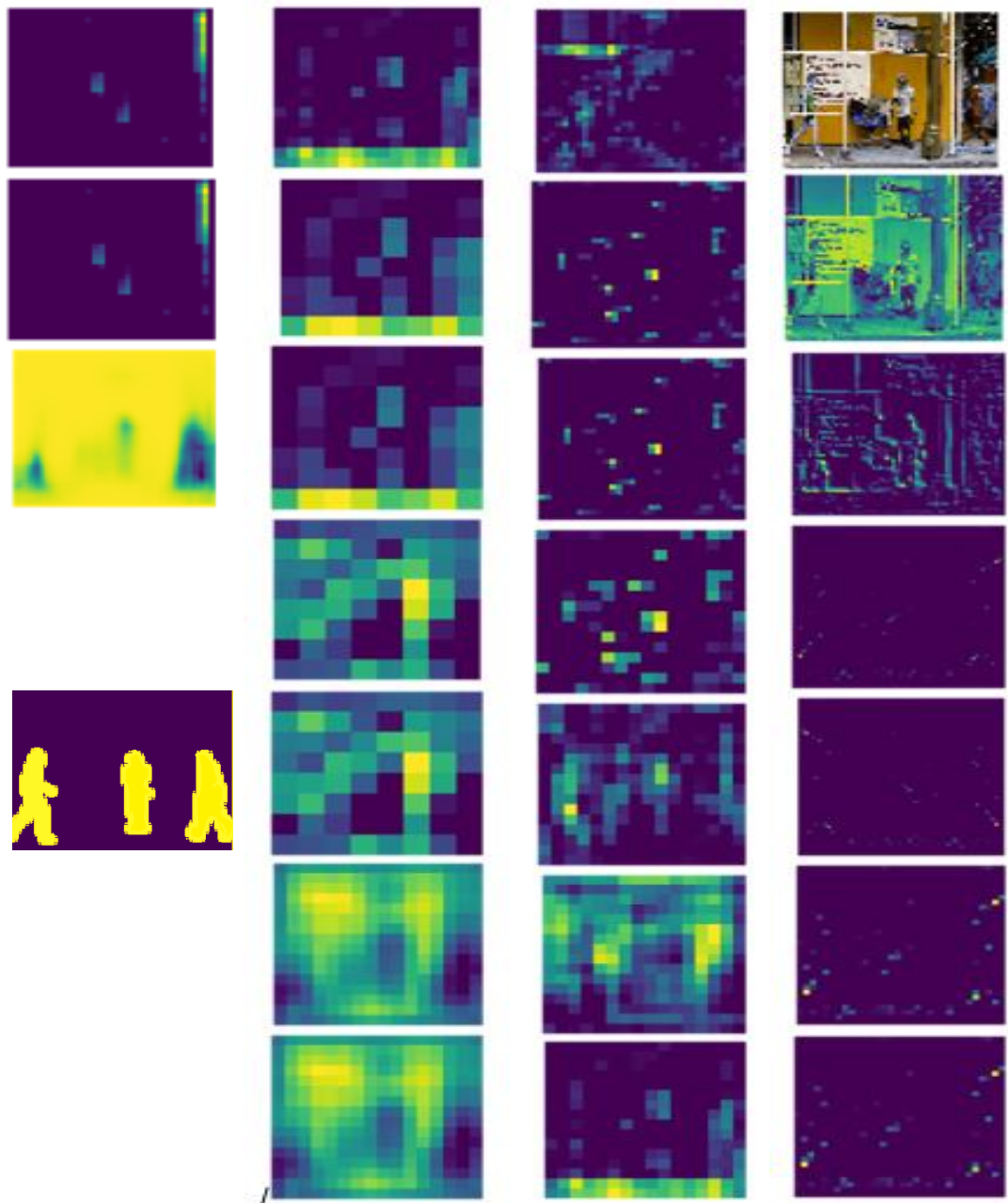


איור 32 - דוגמא לארכיטקטורת ה-*u-net*

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:



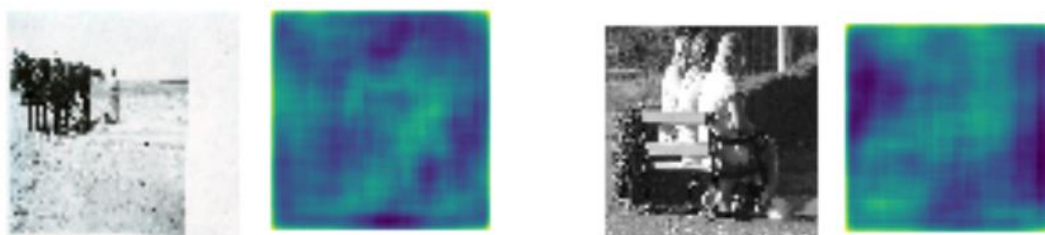
איור 33 - וויזואליזציה של טיפול הרשת בתמונה בהפרדה לשכבות

תוצאות



איור 34 - וויזואליזציה של סגמנטציית הולכי רגל מוצלחות

דוגמאות למקרי קצה שלא צלחו



איור 35 - וויזואליזציה של סגמנטציית הולכי רגל לא מוצלחות (עקב תנאי תאורה קשים, גודל דמות שונה והסתרות)

3. סיכום

כמו שציינתי בהתחלה, המטרה הייתה ליצור רשת CNN בארכיטקטורה חדשנית שמסוגלת 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 לזיכרון כך שיהיה מסודר בצורה הגיונית (מבחינת פורמט, רזולוציה, כמות ערוצים וכו') גם בכניסה למודל.

3.1 הצעות להמשך עבודה בנושא ויישול 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 שבנית. 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 ה API של COCO ניתן גם לשערך את ה POSE 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 אבנה מודל נפרד. בהינתן תמונה שהקטגוריה שלה ידועה, אשתמש במודל שהוגדר ע"י אותה קטגוריה.
- 6) כמובן שגם את ארכיטקטורת ומבנה הרשת ניתן לשפר - תהליך שדורש הרבה זמן מחקר:
 - א. אפשר לנסות כלי אופטימיזציה שונים אני השתמשתי ב"אדם" אבל לפי מחקרים ומאמרים עדכניים ישנם עוד אלגוריתמים מהירים יותר ומדויקים יותר שניתן להשתמש בהם,
 - ב. הרשת שבנית דיי קטנה מבחינת כמות שכבות - אפשר להוסיף עוד שכבות וכמובן להגדיל את רזולוציית התמונות.
 - ג. במידה והמבנה החדש יגיע למצב של התאמת יתר יהיה אפשר להתגבר על over Fitting בעזרת שיטות שלא השתמשתי כמו data 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 0-07-042807-7](#), p.2.
- [2] ["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"](#). Wall Street Journal. [ISSN 0099-9660](#). Retrieved 2020-06-21.
- [4] ["PRELIMINARY REPORT – HIGHWAY – HWY18MH010"](#) (PDF) . [National Transportation Safety Board](#). May 24, 2018.
- [5] ["Kaggle"](#). Kaggle competitions, Kaggle Inc.
- [6] ["Google achieves AI 'breakthrough' by beating Go champion"](#). 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"](#). *Neural Networks* 61: 85–117. 1 January 2015. [ISSN 0893-6080](#). [arXiv:1404.7828](#). [doi:10.1016/j.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](#), 1999, page 155
- [11] Stuart J. Russell, Peter Norvig (2010) [Artificial Intelligence: A Modern Approach](#), Third Edition, Prentice Hall [ISBN 9780136042594](#).
- [12] LeCun, Yann. ["LeNet-5, convolutional neural networks"](#).
- [13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press
- [14] Yoshua Bengio , Xavier Glorot , DIRO, Universite de Montr ´ eal, Montr ´ eal, Qu ´ ebec, Canada, "Understanding the difficulty of training deep feedforward neural networks"
- [15] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), pages 1:886–893, 2005
- [16] 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
- [17] Coco, Common Objects in Context, <http://cocodataset.org/#download>
- [18] ["TensorFlow Release"](#). Retrieved Aug 09, 2020
- [19] ["Keras Documentation"](#). *keras.io*. Retrieved Aug 09, 2020.
- [20] "Project Jupyter - About Us". 2018-04-20. Retrieved 2018-05-03.
- [21] Ramachandran, Prajit; Barret, Zoph; Quoc, V. Le (October 16, 2017). "Searching for Activation Functions". [arXiv:1710.05941 \[cs.NE\]](#).
- [22] Ng, A. (2011) Machine Learning. <https://www.coursera.org/course/ml>
- [23] Claesen, Marc; Bart De Moor (2015). "Hyperparameter Search in Machine Learning". [arXiv:1502.02127 \[cs.LG\]](#).
- [24] AWS Machine Learning Foundations Course <https://classroom.udacity.com/courses/ud090>
- [25] CS231n: Convolutional Neural Networks for Visual Recognition <http://cs231n.stanford.edu/>

5. נספחים

5.1 רשימת נוסחאות:

$$(1) \quad \varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

$$(2) \quad \varphi(v) = g(v) = \frac{1}{(1 + \exp(-av))}$$

$$(3) \quad g(v) = \max(0, v)$$

$$(4) \quad \varphi(v) = \begin{cases} v & \text{if } v > 0 \\ 0.01v & \text{if } v \leq 0 \end{cases}$$

$$(5) \quad \varphi(v) = \begin{cases} v & \text{if } v > 0 \\ av & \text{if } v \leq 0 \end{cases}$$

$$(6) \quad \sigma(z)_j = f \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad , \quad \text{for } j=1, \dots, K$$

$$(7) \quad \mathbf{f}: \mathbf{X} \rightarrow \mathbb{R}$$

$$(8) \quad \vec{x} \rightarrow \mathbf{y}$$

$$(9) \quad I[f] = \int_{X \times Y} V(f(\vec{x}), y) p(\vec{x}, y) d\vec{x} dy$$

$$(10) \quad np.random.randn([\text{matrix shape}]) * \sqrt{\frac{2}{([\text{input layer size}] + ([\text{output layer size}]))}}$$

$$(11) \quad Z^1, Z^2, \dots, Z^m \rightarrow Z^{L(i)}$$

$$(12) \quad \mu^2 = \frac{1}{m} \sum_i (Z^i)$$

$$(13) \quad \sigma^2 = \frac{1}{m} \sum_i (Z_i - \mu)^2$$

$$(14) \quad Z_{norm}^i = \frac{Z^i - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$$(15) \quad \tilde{z}^i = \gamma * Z_{norm}^i + \beta$$

$$(16) \quad \gamma = \sqrt{\sigma^2 + \varepsilon}; \quad \beta = \mu \Rightarrow \tilde{z}^i = z^i$$

5.2 ראשי תיבות

AI - Artificial intelligence

ANN - Artificial Neural Network

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

5.3 רשימת איורים:

- איור 1 -** אתגרי זיהוי הולכי רגל , דוגמאות :
- איור 2 -** אבני דרך בציר הזמן של רשתות נוירונים ולמידת מכונה
- איור 3 -** השוואה בין רשת בגוף האדם לרשת נוירונים מלאכותית
- איור 4 -** רשת נוירונים כתת מערכת בעולם הבינה המלאכותית.
- איור 5 -** השוואה בין הביצועים בין השיטות השונות אל מול כמות המידע.
- איור 6 -** מודל של נוירון
- איור 7 -** גרף פונקציית סף
- איור 8 -** גרף פונקציית סיגמוייד
- איור 9 -** גרף פונקציית המיישר
- איור 10 -** גרף פונקציית מיישר דולפת
- איור 11 -** גרף פונקציית מיישר פרמטרית
- איור 12 -** רשת חד-שכבתית ללא משוב
- איור 13 -** רשת רב-שכבתית ללא משוב
- איור 14 -** רשתות חוזרות
- איור 15 -** רשת רב שכבתית לסיווג מרובה מחלקות
- איור 16 -** למדיה ממוקחת לעומת למידה ללא פיקוח
- איור 17 -** רשת קונבולוציה פולי קונקטד.
- איור 18 -** פירוט שכבות הקונבולוציה
- איור 19 -** משקל והטיה
- איור 20 -** הבדלים בסוגי זיהוי אובייקטים.
- איור 21 -** חמשת הדרגות לטכנולוגיית נהיגה אוטונומית אוטונומית.
- איור 22 -** זיהוי אובייקטים מבוסס חלקים
- איור 23 -** זיהוי אובייקטים מבוסס תנועה
- איור 24 -** זיהוי אובייקטים מבוסס מספר מצלמות
- איור 25 -** דוגמאות לארכיטקטורת רשת מקודד-מפענח (UNET)
- איור 26 -** זמן אימון רשת בעזרת GPU מסוג TESLA לעומת זמן אימון רשת בעזרת CPU
- איור 27 -** ארכיטקטורת הרשת (מקודד-מפענח – UNET)
- איור 28 -** רשת סטנדרטית (a) לעומת רשת לאחר dropout (b)
- איור 29 -** הדגמת מספרית של הבדל בתהליכי ה pooling בחלון של 2X2 על מטריצה בגודל 4X4
- איור 30 -** סיווג חיוביים האמיתיים (TP) מול סיווג החיוביים הכוזבים (FP) והשליליים
- איור 31 -** גרף ה ROC עבור זיהוי הולך רגל.
- איור 32 -** דוגמא לארכיטקטורת ה u-net
- איור 33 -** וויזואליזציה של טיפול הרשת בתמונה בהפרדה לשכבות
- איור 34 -** וויזואליזציה של סגמנטציית הולכי רגל מוצלחות
- איור 35 -** וויזואליזציה של סגמנטציית הולכי רגל לא מוצלחות (עקב תנאי תאורה קשים , גודל דמות שונה והסתרות)