

המחלקה להנדסת תוכנה

למידה וניתוח של תמונות רפואיות

Machine Learning to Medical Data

– Analyze CT Images

חיבור זה מהווה חלק מהדרישות לקבלת תואר בהנדסה

מאת:

אלון צליק שמילוביץ

סתיו ברזני

יוני 2017

סיון – תמוז ה'תשע"ז

המחלקה להנדסת תוכנה

למידה וניתוח של תמונות רפואיות

Machine Learning to Medical Data – Analyze CT Images

חיבור זה מהווה חלק מהדרישות לקבלת תואר בהנדסה

מאת:

אלון צליק שמילוביץ

סתיו ברזני

מנחה אקדמי: דר' שפנייר אסף ב. אישור: תאריך: 21.6.2017

רכז הפרויקטים: דר' יגל ראובן אישור: תאריך: 21.6.2017

מערכות ניהול הפרויקט:

מיקום	מערכת	
https://github.com/alonshmilo/MedicalData_jce	מאגר קוד	
http://alonshmilo.wixsite.com/jce-finalproject	יומן	
https://huboard.com/alonshmilo/MedicalData_jce	ניהול פרויקט	
https://github.com/alonshmilo/MedicalData_jce/wiki	אתר ויקי	
https://www.youtube.com/watch?v=OwxQbqJyb2Q&t=1s	סרטון	

תקציר

הפרויקט עוסק בניתוח תמונות רפואיות – תמונות CT, תוך שימוש בעיבוד תמונה ממוחשב ומערכת לומדת. במרכז הפרויקט עומדות שתי מערכות שעוסקות בניתוח תמונות רפואיות, אחת תלת מימדית ואחת דו מימדית. המערכות מבצעות ניתוח של התמונה הנתונה- תקבע על גביה את מיקום הצלעות ותסמן אותן. פלט המערכת יהיה אותה סריקת ה-CT, אך עם סימוני הצלעות, כל מערכת מבצעת את המוטל עליה בצורה שונה. נציין כי זוהי פעולה שאורכת כיום זמן רב - נעשית ע"י הרופא בצורה ידנית. המערכת הכוללת שנבנה היא מערכת חדשנית, שלא קיימת כיום כלל וכלל.

הצהרה

פרויקט זה בוצע בהנחייתו של דר' שפנייר אסף ב., במחלקה להנדסת תוכנה בעזריאלי – המכללה האקדמית להנדסה ירושלים.

הפרויקט המוצג הינו עבודתנו האישית ומהווה כחלק מהדרישות עבור תואר בוגר במדעים בהנדסת תוכנה.

הפרויקט הינו פרויקט זוגי מלא ובוצע בעבודה מאסיבית של שנינו יחד בעזרה הדדית.

הכרת תודה

נבקש להודות לכל האנשים שעזרו לנו במהלך ביצוע פרויקט הגמר, ובמיוחד –

לדרי' אסף ב. שפנייר על האמון, העצמאות והתמיכה שניתנה לנו במהלך ביצוע עבודת הגמר. עבודת גמר זו דרשה מאתנו ריכוז ומאמץ של שעות מרובות, והעלתה לנו שאלות רבות, עליהם השכיל אסף לענות בכל זמן ובכל פלטפורמה בצורה מלאה. בכל פנייה שלנו, אסף נאות להיפגש עמנו ולעזור היכן שיכל.

למר משה שפירו, מנהל המעבדה להנדסת תוכנה במכללה, על העזרה הרבה בהכנת פלטפורמות עבודה במעבדה. בזכותו, הייתה לנו האפשרות להריץ את מערכות הפרויקט בסביבה מתאימה ונעימה. משה הגיע אלינו בכל פעם שביקשנו וישב אתנו לפתור כל בעיה טכנית שהתעוררה.

לדרי' ראובן יגל על התמיכה וריכוז הפרויקט מבחינת לוח זמנים, הגשות תקופתיות, דגשים וכל פרט שרצינו לדעת – ראובן עמד לרשותנו.

בנוסף, אנו מעוניינים להודות האחד לשנייה, על העבודה המשותפת וההדדית, ההקשבה, העזרה, ברגעים הטובים יותר וברגעים הטובים פחות.

על הכל, תודה.

תוכן עניינים

8.....	1 מבוא
8.....	1.1 מילון מונחים, סימנים וקיצורים
9.....	1.2 רקע
9.....	1.2.1 CT – טומוגרפיה ממוחשבת
10.....	1.2.2 נוירונים
10.....	1.2.3 Machine Learning
11.....	1.2.4 למידה עמוקה – Deep Learning
12.....	2 מסגרת ומטרת הפרויקט
12.....	2.1 כלים בהם השתמשנו בביצוע הפרויקט
13.....	3 הבעיה
13.....	3.1 דרישות ואפיון הבעיה
16.....	4 תיאור המערכת שמומשה
16.....	4.1 מבנה המערכת
17.....	4.1.1 אלגוריתמי למידה
19.....	4.2 ארכיטקטורת תתי מערכות
19.....	4.2.1 מערכת DeepMedic הרשת המלאה
20.....	4.2.2 מערכת DeepMedic – הרשת הקטנה tinyCnn
21.....	4.2.3 מערכת Yolo
24.....	4.3 שימוש בתתי המערכות
24.....	4.3.1 שימוש במערכת Deep Medic
26.....	4.3.2 שימוש מערכת Yolo
28.....	5 ניסויים ובדיקות
28.....	5.1 מערך ניסויים
28.....	5.1.1 DeepMedic
30.....	5.1.2 Yolo
31.....	6 תוצאות ומסקנות
31.....	6.1 תוצאות הניסויים, המדידות והפעלת המערכת
31.....	6.1.1 תוצאות מערכת DeepMedic
34.....	6.1.2 תוצאות מערכת Yolo
39.....	6.2 מסקנות הפרויקט
40.....	6.3 סיכום
40.....	7 ספרות
40.....	7.1 מאמרים

41	8 רשימות
41	8.1 רשימת גרפים
41	8.2 רשימת תרשימים
41	8.3 רשימת תמונות
42	8.4 רשימת טבלאות
42	9 מקורות ספרותיים

1 מבוא

1.1 מילון מונחים, סימנים וקיצורים

- משתמש - רופא או איש רפואה שיש לו עניין בניתוח התמונות הרפואיות.
- תמונות CT - תמונות אלו מתקבלות לאחר ביצוע בדיקת טומוגרפיה ממוחשבת - Computed Tomography. בדיקה זו הינה סוג של צילום לא פולשני שנועד להעניק תמונה תלת מימדית של פנים הגוף. על ידי קרני רנטגן ואותות אלקטרוניים נוצרת הדמיה תלת מימדית של הגוף. מתוך הדמיה זו ניתן לייצר תמונות של חתכים שונים.
- חתכים - סדרה של צילומי רנטגן מזוויות שונות שבהמשך מעובדות באמצעות המחשב לתמונות.
- תמונת Grayscale – תמונה בגווני אפור – בסקאלה שבין שחור ולבן, ולכן ערכי המטריצה שלהם נעים בין 0 ל-1.
- תמונה בינארית – תמונה שערכי המטריצה שלה הם 0 ו-1 בלבד. כלומר תמונה בצבעי שחור ולבן.
- Classification – סיווג של יישויות למספר סגור של קבוצות מוגדרות.
- תמונת סגמנטציה – חלוקה של תמונה דיגיטלית למטעים (סגמנטים). הכוונה לפישוט של התמונה לאוסף של אובייקטים בעלי משמעות, כמו מציאת גבולות, קווים והבלטה של חלקים מסוימים רצויים.
- Ground Truth Labels – תוויות המהוות סימון של חלק מבוקש בתמונה.
- ROI MASK – אזור עניין. ROI – Range Of Interest. התמונה היא תמונה בינארית, כאשר באזור העניין עצמו – ערך המטריצה הוא 1. אחרת – 0.
- Dice Coefficient – מדד דיוק הנמדד על שתי תמונות ומצביע על ההבדלים ביניהם.
- Voxel – מייצג ערך של גריד במודל תלת מימדי מרחבי. הערך נקבע על פי מיקומו יחסית לשאר הווקסלים, ובכך נבדלים מפיסקלים. בתמונות רפואיות, משמשים להגדרת רזולוציה. המילה ווקסל מורכב מ"ווליום" ו"פיקסל".
- Dropout – שיטה בה מבטלים חלק מהקשרים בין שכבה לשכבה ברשת נוירונים המיועדת ללמידה, ובכך מצד אחד מחלישים למידה על סיווגים מסוימים, אך מחזקים את המיקוד והלמידה בסיווגים אחרים.
- Overfitting – מצב בו האלגוריתם מבצע שינון יתר של המידע.

- Feature Map – (בהקשר של שכבה קונבולוציונית) הפלט של פילטר אחד ביחד לשכבה הקודמת. בעצם מדובר בחלוקה של התמונה לעומק, וכל קבוצה היא Feature Map.
- רגרסיה – חיזוי של ערך על בסיס נתונים קיימים.

1.2 רקע

CT 1.2.1 – טומוגרפיה ממוחשבת

בדיקת CT הינה סריקה ממוחשבת, הנקראת "טומוגרפיה ממוחשבת", שנעשית על ידי טכניקת קרני רנטגן, המאפשרת הפקת תמונות ברמת דיוק גבוהה מאוד, לצורך הדגמת האיברים הפנימיים של הגוף.

לאחר עיבוד ממוחשב של החתכים מתקבלת הדגמה תלת ממדית של הגוף שניתן לצפות בה על ידי סדרת חתכים אופקיים או בצורה אנכית משלימה. זאת להבדיל מטכניקת צילום הרנטגן הרגילה שבה התמונה המתקבלת היא תמונה דו ממדית, בדומה לסרט צילום רגיל.

סורק CT הינו סורק מסתובב, אליו מחובר מחשב רב עוצמה, אשר מעבד את התמונה הכוללת. בדיקה זו מאפשרת לגלות האם נגרם נזק לאיבר או האם מתחולל תהליך פנימי כלשהו באחד מאיברי הגוף, וכן על מנת לזהות את המבנה האנטומי המדויק של החולה, למשל לפני ניתוח פולשני.

בדיקת CT עושים במגוון רב מאוד של מצבים, על מנת לזהות האם קיים שינוי כלשהו באנטומיה התקינה של איברי הגוף, זאת ברמת האיבר והרקמה. הבדיקה יכולה לאתר נזק לאיברים פנימיים עקב תהליכים שונים כגון – טראומה, שברים, דלקות, זיהומים, גידולים ותהליכים תופסי מקום אחרים, שלא ניתן לראות בעין או על ידי צילום פשוט.

לעיתים מוסיפים לבדיקה חומר ניגוד, הניתן בשתייה, או המוזרק דרך הוריד על מנת להדגים טוב יותר דימומים, זרימת דם מוגברת או חוסר אספקת הדם לאזור מסוים, למשל בעקבות קריש דם חוסם או גידול לוחץ. אינדיקציה נוספת לביצוע הבדיקה היא כאמור לפני ביצוע פעולה חודרנית, למשל ניתוח או ביופסיה.

בעת ביצוע הבדיקה, מקור קרינה שנוע מסביב לחולה מעביר קרני רנטגן דרך הגוף בזוויות שונות. גלאים קולטים את הקרניים לאחר שעברו דרך הגוף ושולחים אותות אלקטרוניים למחשב. זה מעבד את הנתונים ויוצר הדמיה תלת ממדית של הגוף או תמונות דו ממדיות של חתכים ספציפיים. דיוקה של תוצאת הסריקה גדל ככל שמספר זוויות השיקוף גדל.

1.2.2 נוירונים

המוח מורכב משני סוגים של תאים: תאי עצב (נוירונים) ותאי גליה. הנוירונים מסוגלים להעביר מידע לנוירונים אחרים באמצעות תהליכים אלקטרוכימיים.

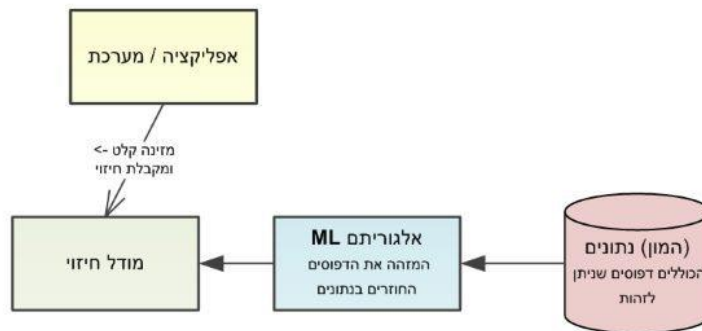
כדי להמיר את תכונות הנוירון הביולוגי למלאכותי (הממוחשב), מתבצעת הפשטה של מונח הנוירון. הנוירון הממוחשב הוא יחידת עיבוד פשוטה המקבלת קלט, מבצעת עיבוד של הנתון, ומספקת פלט. הקלט הוא מערך של פלטים של נוירונים אחרים, הפלט הוא אות בינארי בעוצמה קבועה. עוצמת הקשר בין פלט כלשהו לבין נקודת קלט הינה בעלת "משקל" ומתארת את מידת השפעת תוצאת החישוב של נוירון אחד על נוירון אחר המקבל קלט ממנו. על מנת להפעיל את רשת הנוירונים, יש להגדיר את המשקולות בקשרים בין כל נוירון לשכנו. כאשר נוירון מחובר לשכנו במשקל 0, הדבר מהווה בעצם נתק ביניהם. המודל הפשוט, מדבר על נוירון כמבצע פעולת סכימה פשוטה המייצר קלט על פי נקודת סף מוגדרת.

Machine Learning 1.2.3

Machine Learning הינו כלי שמאפשר לפתור בעיות תוכנה, שלא ניתן לפתור בדרך הרגילה, כלומר בעזרת קוד עם הפעולות המוכרות הבסיסיות. כמובן שפעולות אלו קיימות בקוד, אך מהות הקוד היא שיטה מתמטית וסטטיסטית המאפשרת לחשוב בפרדיגמה שונה של תכנות על הבעיות הללו.

עולם Machine Learning נחלק לשני קטגוריות:

- למידה מפקחת – Supervised Learning. במקרה זה יש לנו מאגר של נתונים ותשובות שהושגו בצורות שונות, ועל המחשב לבצע למידה שלהם. במקרה זה, הקטגוריות ידועות, ויש לנו Classes מוגדרים. באופן עקרוני, בוחרים אלגוריתם של למידה ומפעילים על רוב הנתונים. את הבדיקה מבצעים על שאר הנתונים, וכך יודעים האם הלמידה בוצעה כראוי.
- למידה לא מפקחת – Unsupervised Learning. בקטגוריה זו הנתונים שבידינו אינם מסודרים, ואין לנו סיווגים מוכנים, אלא תוך כדי הלמידה מבצעים חלוקה לסיווגים.



תרשים 1 – תהליך למידה

השלבים בתהליך :

- הגדרת הבעיה וכיוון הפתרון – מהי הבעיה, מה רוצים ללמוד.
- השגת הנתונים – הבאת נתונים מוצלחים, טובים, נכונים, שניתן להסתמך עליהם. כמות הנתונים מאוד חשובה – ככל שיהיו יותר נתונים, כך הלמידה תהיה יותר חזקה ותכלול יותר מקרים ואפשרויות.
- נורמליזציה והתאמה של הנתונים – הבאת כל הנתונים להיות באותו הפורמט, השלמת נתונים חסרים ותיקון אי התאמות.
- בחירת אלגוריתם – בלמידה, ישנם עשרות אלגוריתמים, וכמה שרלוונטיים לכל סוג של בעיה. הבחירה באלגוריתם תשפיע על רמת הדיוק של התוצאה, על הרגישות לכמות ואיכות הנתונים ועל ההתאמה לבעיה הספציפית וסוג הנתונים שיש לנו.
- אימון המודל – יישום אלגוריתם הלמידה, הזרמת הנתונים למחשב ובעצם למידה של הנתונים באופן מלא.
- בחינת המודל – הרצת קבוצת הביקורת שבנתונים על מנת לראות את החיזוי שנוצר – האם החיזוי תואם לתוצאות המקוריות.

1.2.4 למידה עמוקה – Deep Learning

למידה עמוקה היא מודל חישובי המכיל קבוצת אלגוריתמים ונשען על תשתית של רשתות נוירונים המחקים את תהליך הלמידה במח האנושי. התחום מייצר פתרונות לבעיות ותהליכים שגרתיים ויומיומיים : מזיהוי דיבור בסמארטפונים, ועד לשדרוג של חלקים מעולם הרפואה וההתקדמות לעבר תחבורה אוטונומית, תעשייה יעילה וחכמה יותר.

2 מסגרת ומטרת הפרויקט

הפרויקט בוצע במסגרת המחלקה להנדסת תוכנה, בעזריאלי – המכללה האקדמית להנדסה ירושלים.

עיתוי ביצוע הפרויקט הינו שנה ד' כחלק מלימודי תואר ראשון בוגר במדעים, בהנדסת תוכנה.

תחום המחקר שנבחר הוא עיבוד תמונות רפואיות, תוך שימוש בטכניקות למידה – אימון ובדיקה. תחום זה לאחרונה חווה התפתחות משמעותית בהופעתן של טכניקות חדשות ומעבדים גרפיים חדשים המאפשרים למידה ברמה גבוהה והגעה לאחוזי דיוק טובים.

מטרת הפרויקט היא עיבוד סריקות CT תלת-מימדיות וביצוע סימון עצמות הצלעות על גבי חתכים דו-מימדיים שלהם.

2.1 כלים בהם השתמשנו בביצוע הפרויקט

במהלך הפרויקט עשינו שימוש במספר כלים:

- תוכנת ITK-SNAP – תוכנה זו נועדה להצגה ועריכה של תמונות מסוג NIFTI – nii שהן בעצם התמונות הרפואיות בהם עשינו שימוש. בתוכנה זו ניתן גם להכין קבצי סגמנטציה בהם ייעשה שימוש במערכת DeepMedic.
- כלי סימון תמונות BBox Label Tool – כלי שבעזרתו ניתן לסמן על גבי התמונה קופסאות סימון שמציירות פריט. בסיום הסימון, נשמר קובץ txt עם הקואורדינטות שסומנו. כלי זה מטפל רק בקבצים מסוג jpeg.
- מחשב בעל מעבד גרפי GTX980 – המחשב נמצא במעבדה וסודר על מנת שנוכל לבצע הרצות של אימונים בשתי המערכות בהם אנו עושים שימוש. מחשב זה היה פעיל יום ולילה, וזאת משום שאימונים אלו לוקחים זמן. מערכת ההפעלה היא Kubuntu.
- מחשב iMac במעבדה – מחשב בו נעזרנו להכנת datan עבור הריצה של מערכת Yolo. בעזרת מחשב זה היה קל יותר לבצע זאת בשל המסך הגדול ונתוניו.
- מחשבים ניידים שלנו MacBook בעלי מערכת ההפעלה macOS, בעזרתם ביצענו ריצות ראשונות של הרשתות המבוקשות ובדיקות כאלו ואחרות.
- שפות התכנות – בוצע שימוש בשפות Python, c, Matlab. עבור השפות האלו בוצע שימוש בתוכנות עריכה תואמות, כולל תוכנת Matlab מלאה.

- עורך שפת פייתון pyCharm.
- תוכנת MATLAB.

3 הבעיה

3.1 דרישות ואפיון הבעיה

בעת קבלת החתכים ישנו צורך לזהות בקלות את הצלעות ולמספר אותן, לשם זיהוי האיברים האחרים - ניתן ע"י מספור הצלעות לזהות איברים גדולים/קטנים מהרגיל ולאתר מחלות שונות וע"י כך לסייע למטופלים רבים.

המציאות כיום היא שרופא מקבל את תוצאות בדיקת הCT, ונאלץ באופן ידני לזהות את הצלעות, לספור אותן, למדוד את המרחקים ביניהם ולהסיק את המסקנות הרפואיות מכך. לדוגמא איבר גדול או קטן מהנורמה ישפיע על גודל הצלעות המוצגות בתמונה. הדרישה היא למערכת שתעזור לרופא לדעת את מיקום הצלעות וכך לנתח את תמונת הCT באופן יותר מהיר ולהסיק מסקנות בנוגע למחלות אפשריות.

על המערכת לתמוך בתמונות רפואיות היוצאות כפלט ממכשירי הבדיקה המוחזקים בבתי חולים, בהם נעשה שימוש כאשר הנבדקים נשלחים על ידי רופאיהם. המערכת נועדה:

1. להקל על עבודתו של הרופא.
2. לחסוך בזמן רופא שהינו זמן יקר מאוד שעלול להגיע למאות שקלים לשעה.
3. למקד את הרופא בקביעת מסקנות רפואיות ולמנוע עיסוק מיותר בספירה של צלעות וזיהוי מרחקים בעייתיים.
4. קידום הטכנולוגיה בעבודת הרופא.

כאשר מטופל עובר צילום CT מתקבלת תמונה תלת מימדית של הגוף. מעין קופסא תלת מימדית שמדמה את כל האיברים שבגוף. מתוך הצילום הזה, יש צורך להבליט את עצמות הצלעות, שמהווים בעצם את כלוב בית החזה.

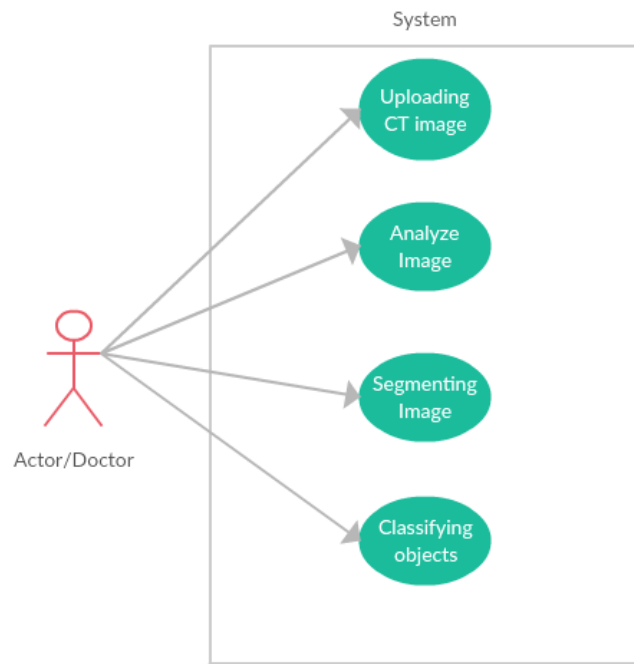
הבעיה היא, בהינתן תמונות דו מימדיות של חתכים של אותה הסריקה – סימון ויזואלי של עצמות הצלעות על גבי התמונה. בעיה זו, נובעת מכך שהרופא עושה פעולה זו באופן ידני, מה שמקשה עליו בביצוע עבודתו. וזאת, על מנת שיוכל להתמצא בסריקת הצלעות. למעשה, הסריקה נותנת לרופא נקודת ציון. לדוגמא, כבד בריא נמצא ליד חוליה כלשהי, ולכן אם הרופא מבחין בכבד שהוא גדול יותר, זוהי אינדיקציה לבעיה רפואית.

יש לייצר מודל תלת ממדי שמדמה את עצמות הצלעות בצורה ברורה ומובנת לעין, על מנת שהרופא יוכל להשתמש במידע זה ולבצע את הבדיקות שברצונו לבצע.

על המערכת לבצע את שתי הפעולות האלו, ובעצם ללמוד כיצד נראות הצלעות ולהוציאן מתמונות חדשות. הדרישה היא שהמערכת תבצע זאת באופן אוטומטי, ובצורה כמה שיותר מדויקת, על מנת שהרופא יוכל להתרכז בעבודתו ובזיהוי הבעיות, ולא בפעולות מקדימות שאינן קשורות לאבחנה.

בתרשים 3.2 מתוארים מקרי השימוש (Use-cases) הנדרשים.

- Uploading CT image – טעינת תמונה CT בפורמט הנכון, שהינו סוג (NIFTI) files) והסוג המכוון – .gz. וידוא שהסוג תואם.
- Analyze image – ניתוח התמונה על ידי אימון ויצירת משקלים לניתוח תמונות אחרות.
- Segmenting image – ביצוע סגמנטציות עבור תמונות וסריקות חדשות.
- Classifying image – ביצוע סיווג על גבי התמונות : מציאת העצמות וסימונן.



תרשים 2 – Use cases

3.2 תיאור הבעיה מבחינת הנדסת תוכנה – הבעיה האלגוריתמית

הפרויקט הינו פרויקט מחקרי-מדעי.

הפרויקט מספק אתגרים בעיבוד תמונה רפואית ובלמידה של תמונות רפואיות, למציאת הצלעות וסימון.

מבחינה אלגוריתמית - לזהות את הצלעות מבלי לפספס, אלגוריתמי עיבוד תמונה, אלגוריתמי מערכות לומדות -כדי ללמוד היכן נמצאות הצלעות השונות בחתכים השונים.

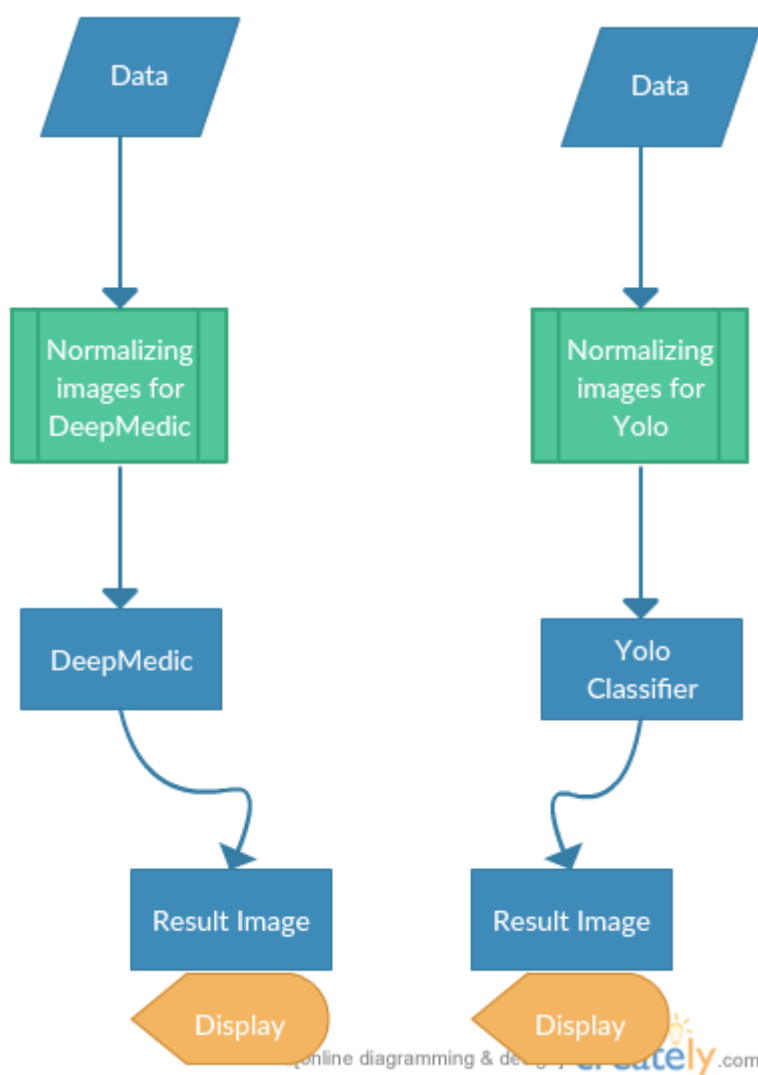
4 תיאור המערכת שמומשה

4.1 מבנה המערכת

המערכת שלנו בנויה משתי מערכות עיקריות:

- מערכת DeepMedic
- מערכת Yolo – Real Time Object Detection

אופן פעולת המערכת הכללית מתוארת בתרשים הבא:



תרשים 3 - אופן פעולת המערכת

4.1.1 אלגוריתמי למידה

תתי המערכות שמרכיבות את המערכת שלנו עושות שימוש CNN – Convolutional Neural Network. רשתות אלו נועדו ללימוד וסיווג של תמונות ובעיקרן עומדת פעולת קונבולוציה של תמונות.

קונבולוציה של תמונות הינה פעולה מתמטית בין תמונה לבין פילטר מיוחד בעל תפקיד מסוים. למשל, פילטר שתפקידו לזהות שינויי צבע בתמונה המקורית או הבלטה של צבעים בהירים ועוד. תמונת התוצאה הינה תמונה בממדים קטנים יותר שניתנים לחישוב מראש ותלויים בגודל הפילטר, עומקו וגודל התמונה המקורית ומהווה הפשטה של התמונה המקורית. פעולת הקונבולוציה מתבצעת כך :

השמה של הפילטר כמטריצה מול התמונה, וביצוע הכפלה של כל תא במטריצת הפילטר עם התא של התמונה שממול. לבסוף, מבצעים סכימה של הערכים המוכפלים, והערך שיוצא הינו הערך שיחליף את כל האזור הזה במטריצה החדשה – תמונת התוצאה. לאחר מכן, זזים "ימינה" מספר צעדים מוגדר מראש (strides), וביצוע הפעולה מחדש. כך נוצרת מטריצה חדשה בעלת ערכים חדשים, ובעלת ממד קטן יותר.

את גודל הפלט ניתן לחשב על פי הנוסחה :

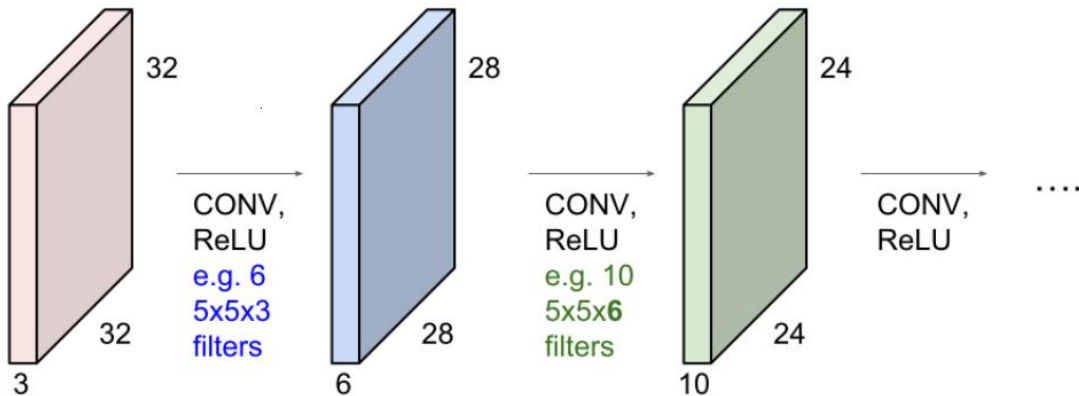
$$\text{Output size} = ((N-F)/\text{strides}) + 1$$

כאשר N היא גודל המטריצה (התמונה), F הוא גודל הפילטר ו-strides הוא גודל הקפיצות.

במידה והקפיצות לא מאפשרות לנו תזוזה מלאה לרוחב המטריצה, יש לרפד את המטריצה מסביב באפסים במספר הבא : $(F - 1) / 2$.

עומק התוצאה נקבע על ידי עומק הפילטר.

בתרשים 4.3 מתוארת פעולת הקונבולוציה.



תרשים 4- קונבולוציה

לפני עידן רשתות ה-CNN, הפילטרים הללו היו מהונדסים ידנית. רשתות ה-CNN למעשה לומדות את הפילטרים הללו באופן אוטומטי.

רשתות קונבולוציוניות בנויות משכבות:

- שכבת קונבולוציה – ביצוע פעולת הקונבולוציה, עם פילטרים מתאימים שתפקידם לבצע הפשטה של התמונה, והבלטה של אזורי עניין.
 - שכבת Pooling – שמירה על הנתונים החשובים ואיבוד של המידע שאינו חשוב לנו. השכבה מקטינה את גודל המרחב ומקטינה את הפרמטרים והחישובים ברשת. עושה זאת על ידי פילטרים של 2×2 בקפיצות של 2, ללא עומק.
 - שכבת פונקציה ReLU – בשכבה זו מבוצעת פונקציית האקטיבציה, כמו $\max(0, x)$ שגובלת באפס. בשכבה זו לא מתבצע שינוי של גודל הרשת.
 - שכבת Fully Connected – שכבה אחרונה שמבצעת חיבור של כל התוצאות וסיווג התוצאה – מבצעת חישוב של scores שכל class קיבל בסיום חישובי הרשת והחלטה על ה-class. כל נירון בשכבה זו מחובר לכל המספרים מהשכבה הקודמת.
- רשת קונבולוציונית CNN בנויה משכבות. לכל שכבה יש API פשוט: היא הופכת קלט תלת ממדי לפלט תלת ממדי תוך שימוש בפונקציות שונות, עם וללא פרמטרים. הפרמטרים הם בעצם המשקלים של כל נירון.

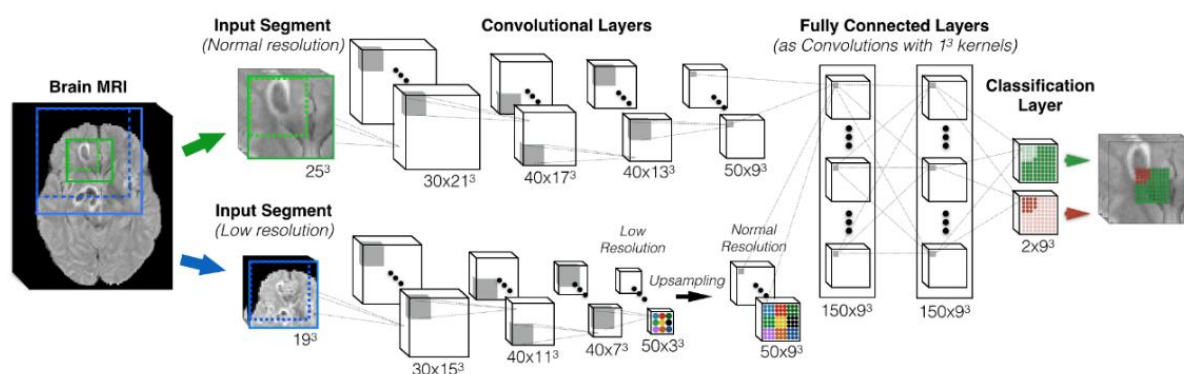
4.2 ארכיטקטורת תתי מערכות

4.2.1 מערכת DeepMedic הרשת המלאה

מערכת זו פותחה על ידי Kontantinos Kamnitsas, שגם פרסם מאמר אקדמי בנושא [1]

יחד עם צוותו. תפקידה של מערכת זו הוא לזהות גידולים סרטניים במח, בהינתן סריקות MRI של המח. חקרנו את אופן פעולת המערכת לעומק והסבנו אותה לזיהוי וגילוי של עצמות הצלעות.

בתרשים 4 ניתן לראות את התהליך שעוברת התמונה מרגע הכנסתה למערכת ועד לפלט הסיווג. התרשים הוא עבור הרשת המלאה:



תרשים 5- רשת Deep Medic המקורית המלאה

השכבות במערכת מפורטות בטבלה הבאה בשורה מס' 4:

3D Networks	#Pathways: FMs/Layer	FMs/Hidd.	Seg.Norm.	Seg.Low	B.S.	DSC(%)
Shallow(+)	1: 30,40,40,50	-	25x25x25	-	10	60.2(61.7)
Deep(+)	1: 30,30,40,40,40,40,50,50	-	25x25x25	-	10	60.0(64.9)
BigDeep+	1: 60,60,80,80,80,80,100,100	150,150	25x25x25	-	10	65.2
DeepMedic	2: 30,30,40,40,40,40,50,50	150,150	25x25x25	19x19x19	10	66.6
2D Networks	#Pathways: FMs/Layer	FMs/Hidd.	Seg.Norm.	Seg.Low	B.S.	DSC(%)
Dm2dPatch*	2: 30,30,40,40,40,40,50,50	150,150	17x17x1	17x17x1	540	58.8
Dm2dSeg	2: 30,30,40,40,40,40,50,50	150,150	25x25x1	19x19x1	250	60.9
Wider2dSeg	2: 60,60,80,80,80,80,100,100	200,200	25x25x1	19x19x1	100	61.3
Deeper2dSeg	2: 16 layers, linearly 30 to 50	150,150	41x41x1	35x35x1	100	61.5
Large2dSeg	2: 12 layers, linearly 45 to 80	200,200	33x33x1	27x27x1	100	61.3

טבלה 1 - שכבות ברשת deepmedic

ארכיטקטורת הרשת המלאה:

הרשת מריצה תהליכים תלת ממדיים סביב כל ווקסל. האימון עוסק בכל ווקסל בנפרד ומתמקד בווקסל המרכזי שבמרחב הנבדק. במהלך האימון, הפרמטרים של הקרנלים עוברים

אופטימיזציה תוך שימוש ב-Gradient Descent (שיטה למציאת מינימום), תוך חתירה למציאת מינימום של שגיאה בין החיזוי לבין המצב במציאות. בשיטה זו, מתקדמים לאט לכיוון המינימום על ידי ביצוע גרדיאנט למסלול עליו אנו נעים, כאשר הצעד שמבוצע אינו מלא, אלא מוגדר מראש בין 0 ל-1, וזאת על מנת לא לפספס את המינימום. אחת ההגבלות בטכניקה זו היא שהסגמנטציה של כל ווקסל מיוצג על ידי כמות קטנה של מידע ויזואלי סביבו. ובאופן אינטואיטיבי, ניתן להסיק שככל שקיימת כמות גדולה יותר של מידע, הסיכוי להגיע לתוצאות טובות יותר גדל. אבל, הגדלת כמות המידע המעובד עלולה לדרוש כוח חישוב גדול מדי. הפתרון המוצע על ידי המערכת, הוא חלוקה לשני נתיבים קונבולוציוניים מקבילים, כאשר שניהם מטפלות באותו הכמות של המידע. הקלט של הנתיב השני הוא בעצם דגימה של התמונה מהנתיב הראשון ובכך מאפשר הרחבה של כמות המידע שמעובד.

רכיב חשוב נוסף בארכיטקטורה היא סביבה קונבולוציונית מלאה, שמייצר יעילות על כל מרחב התמונה.

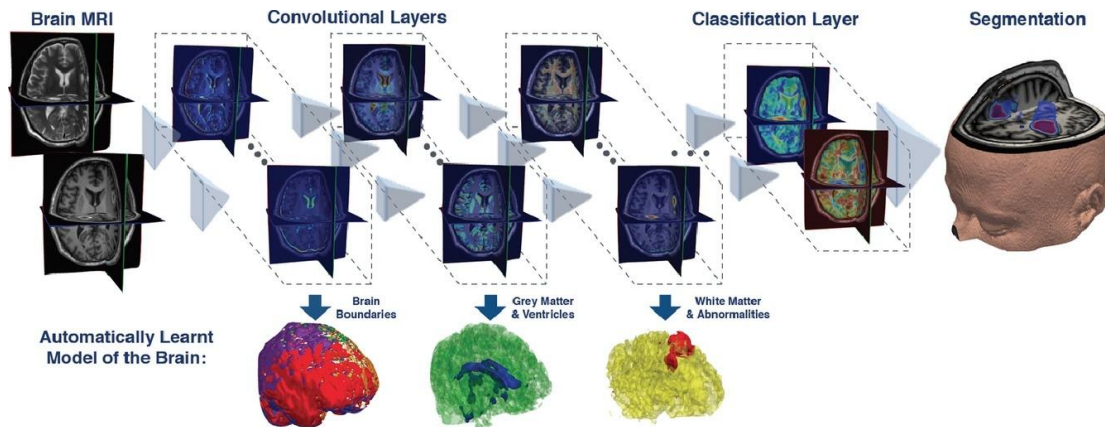
הרשת בעצם מורכבת משכבות שכל אחת מבצעת קונבולוציה בין תמונת הקלט לבין קרנל (פילטר) בגודל 5^3 , או קונבולוציה לקרנל בגודל 3^3 , ועוד שכבת קונבולוציה אחרונה לפני הסיווג עצמו. הרשת מבצעת גם Dropout בין השכבה התשיעית לעשירית בשיעור של 50%. קצב הלמידה נקבע ל-0.01 ועולה תוך כדי האימון ומגיע עד 0.6 בהתאם לפונקציית ReLU שמבצעת את הפעלת הסיווג.

המערכת המלאה בנויה מ-8 שכבות. הווקטור [30,30,40,40,40,40,50,50] מייצג את מספר ה-Feature Maps ב-8 השכבות בהתאמה (ניתן לראות לפי תרשים 5), כאשר גודל הקרנל שפועל על כל Feature Map הוא [3,3,3].

זמן האימון תוך שימוש ב-NVIDIA GTX Titan Black GPU מגיע ליום בעוד ביצוע בדיקה מגיע לכ-3 דקות.

4.2.2 DeepMedic – הרשת הקטנה tinyCnn

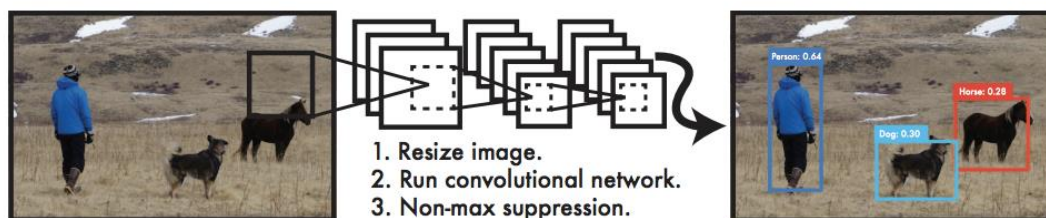
המערכת בנויה מ-3 שכבות. הווקטור [4,5,6] מייצג את מספר ה-Feature Maps בשלושת השכבות בהתאמה, כאשר הגודל של הקרנל שפועל על כל Feature Map הוא [3,3,3]. למערכת יש גם שני ערוצים בה היא פועלת בהתאם לרשת המלאה.



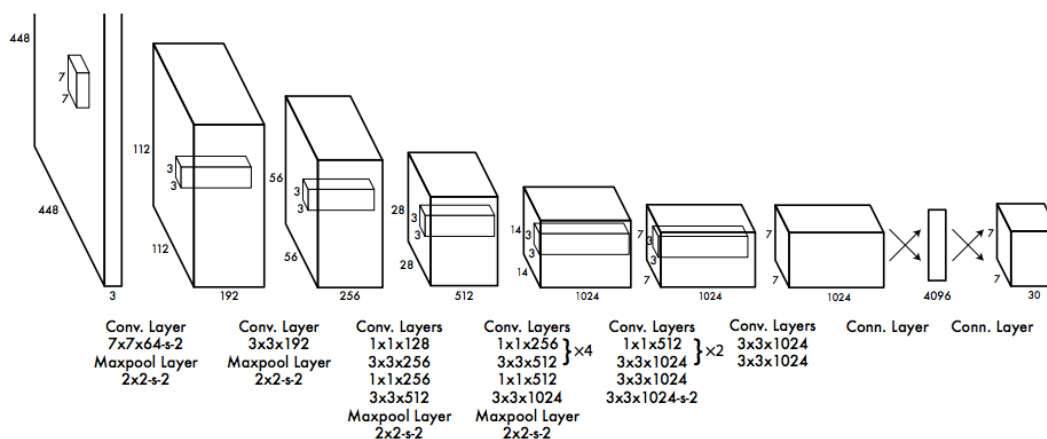
תרשים 6 - רשת tinyCnn

4.2.3 מערכת Yolo

מערכת זו עוסקת בזיהוי פריטים מתוך תמונה וסימונם. המערכת מבוססת אף היא על רשתות קונבולוציוניות. המערכת פותחה על ידי Joseph Redmon, Ali Farhadi ומוצגת במאמר [2].



תרשים 7 - פעולת ה-yolo



תרשים 8 - ארכיטקטורת Yolo

ארכיטקטורת המערכת :

המערכת עושה שימוש באלגוריתם k-means. אלגוריתם זה הינו שיטה עבור ניתוח אשכולות וסיווגים. מטרתו לחלק את התצפיות ל-k אשכולות לפי מרכזי כובד כאשר כל תצפית משויכת לאחד ממרכזי הכובד. על ידי בחירה נכונה של מרכזי כובד, ניתן לאתר את הקבוצות השונות. זהו מודל סטטיסטי. המודל מנסה למצוא אשכולות בעלי מרחבי מידה הניתנים להשוואה. המודל שואף לחלק את התצפיות למספר אשכולות על מנת למזער את סכום המרחקים בין התצפיות בתוך האשכול.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

טבלה 2 - מבנה רשת ה-CNN של מערכת Yolo

בנוסף לאלגוריתם זה, מתבצע שימוש בשכבות קונבולוציה, יחד עם שכבות Maxpool, שתפקידן לפשט את התמונה ולמצוא את האשכול, הסיווג, אליו שייכים הפריטים שבתמונה. תפקיד הקונבולוציה הוא לפשט את התמונה ובכך גם למצוא ולהפריד את הפריטים בתמונה. Maxpool בוחרת את המקסימום, זורקת את כל מה שלא מעניין, ועוזרת לבצע את הסיווג. בסופו של דבר, מתבצע סיווג סופי וחיבור של כל השכבות (על פי עקרון ה-Fully connected שהוצג ברקע). לאחר ביצוע הסיווג, מסירים את השכבות הקונבולוציוניות, ושמים את קופסאות הסיווג במקומות שזוהו, עם הסיווג המתאים משמות ה-classes. בטבלה 2 ניתן לראות את בניית המערכת בחלוקה לשכבות.

האימון עבור הסיווג מתבצע בשימוש באלגוריתם Stochastic Gradient Descent שמשמש למציאת מינימום. הוא מבצע פעולה דומה ל-Gradient Descent, אך בשינוי פעולות מתמטיות מורכבות, והתוצאה יוצאת דומה ופחות יקרה מבחינת זמני ריצה. קצב הלימוד ההתחלתי הוא 0.1, כוח חישוב פולינומיאלי 4, ומומנטום תזוזה והתקרבות למינימום של 0.9. פרמטרי האימון משתנים בהתאם לרזולוציה איתם המערכת מתמודדת.

האימון עבור הזיהוי מתבצע בהפיכת שכבת הקונבולוציה האחרונה לשכבה שמשמשת בקרנל של 1×1 עם מספר הפלטים שאנו צריכים עבור הזיהוי. קצב הלמידה ההתחלתי הוא 10^{-3} וחלוקה לקבוצות של 10, 60 ו-90 קבוצות. מומנטום תזוזה לקראת המינימום – 0.9.

4.3 שימוש בתתי המערכות

4.3.1 שימוש במערכת Deep Medic

שימוש מלא במערכת מחולק ל-3 חלקים:

- בניית מודל ראשוני – בשלב זה מוגדר שם המודל והגדרות ראשוניות שיהוו בסיס לריצת המערכת.
- ביצוע אימון – שלב ה-Training. שלב זה מהווה את הלמידה עצמה. בשלב הזה המערכת לומדת כיצד נראה כלוב החזה – עצמות הצלעות, כיצד לחלץ את המודל מתוך סריקה תלת מימדית. ניתן להמשיל את התהליך הזה ללמידה אנושית בכך שמלמדים את הלומד כיצד נראה פריט מסוים - מסמנים אותו בתוך ערימה של פריטים אחרים. על מנת לבצע את הלמידה, מספקים למערכת 3 סוגים של תמונות:
 - סריקה תלת מימדית. תמונה זו היא תמונת Grayscale.
 - תמונת Ground-Truth Labels – תמונה בינארית שמסמנת את הצלעות בלבד. הצלעות מסומנות בלבן (ערך מטריצה – 1) וכל שאר התמונה שחור (ערך מטריצה – 0). תמונה זו, שגם היא תלת מימדית, מהווה בעצם את הצורה שאנו מבקשים למצוא מתוך הסריקה. תמונה זו אמורה להתאים לסריקה אותה הגשנו לאימון. כלומר, כאשר נעלה את הסריקה בתוכנת ITK-SNAP ומעליה נשים את תמונת ה-GT, הסימונים של הסגמנטציה יתאימו לסריקה ויווצר מודל של הצלעות – מה שאנו מבקשים.
 - תמונת ROI MASK – תמונת בינארית, גם כן תלת-מימדית, שהחלק הלבן שלה מסמן את כל האזור שבתוכו יכולים להתגלות עצמות הצלעות בסריקה המקורית. גם תמונה זו אמורה להתאים לסריקה הנתונה, באותו האופן של תמונת ה-GT.

לאחר ביצוע שלב האימון התקבלו קבצי משקלים שנוצרו במהלך האימון. לאחר כל Epoch נוצר קובץ משקלים רלוונטי אליו, ובסוף האימון התקבל קובץ משקלים סופי שהוגדר כ-Final. קובץ משקלים זה מהווה סיכום של כל שלב האימון ומועבר לשלב הבא, של הבדיקה, ויהיה את הבסיס ממנו ייקבעו הסגמנטציות על הסריקות החדשות שיינתנו בשלב הבא.

שלב האימון מחולק לקבוצות אימון (Epochs) ותת-קבוצות אימון (Subepochs). מספר הקבוצות ותת-הקבוצות נקבע בקובץ הקונפיגורציה, כאשר ברירת המחדל הוא :

Epochs = 35, Subepochs=20.

החלוקה הזו היא בעצם חלוקה של המידע למקטעים ולתת-מקטעים עליהם מתבצע האימון.

- ביצוע בדיקה – שלב Testn – שלב זה הינו שלב התוצאות, השלב בו אנו משתמשים במה שהמערכת למדה, דהיינו בקובץ המשקלים הסופי, ומחילים את הידע על סריקה חדשה. תפקיד השלב הזה הוא בעצם להתבונן על סריקה חדשה ולזהות מתוכה את עצמות הצלעות – כלוב החזה. שלב זה משול לשלב התוצאה בתהליך למידה אנושי – לאחר שהמוח למד כיצד נראה פריט מסוים, הוא מתבונן בתמונה אחרת ומזהה את אותו הפריט.
ישנן שתי רשתות עיקריות :

- tinyCnn – הרשת הקטנה והמהירה. מבצעת למידה ובדיקה מהירים על התמונות המוצעות, מבלי להיכנס לעומקים ולניתוח עמוק של התמונות. רשת זו מתאמנת ובודקת תוך מספר דקות בריצה על מחשבי mac הניידים בשימוש ב-CPU רגיל.
- Full DeepMedic – הרשת המלאה, שמחולקת ל-Epochs ול-Subepochs, מבצעת ניתוח עמוק של התמונות – הסריקה, ה-GT וה-ROI MASK. רשת זו דורשת מחשב בעל GPU חזק, וכאשר הורצה על מחשב המעבדה, לקח לתהליך האימון למעלה מיממה, ולתהליך הבדיקה כ-10 דקות.

המערכת ממומשת בשפת Python שמריצה את כל הפונקציות שנועדו לבצע את בניית המודל, אימון ובדיקה. המערכת מופעלת בטרמינל ועובדת בצורה טובה בעיקר בפלטפורמת לינוקס. לצד קבצי הפיתוח שאחראים על ריצת הפונקציות, לכל שלב יש להכין קבצי קונפיגורציה, שם נמצאים כל הפרמטרים הנחוצים ללמידה, וניתוב למיקומי התמונות עליהם מבצעים את האימון או הבדיקה על המחשב. שימוש בפונקציות המערכת היא לפי פקודות ודגלים והם מפורטים במאגר הקוד של הפרויקט.

מניפולציות על תמונות בוצעו באמצעות תוכנת Matlab והסקריפטים בהם נעשה שימוש
נמצאים במאגר הקוד.

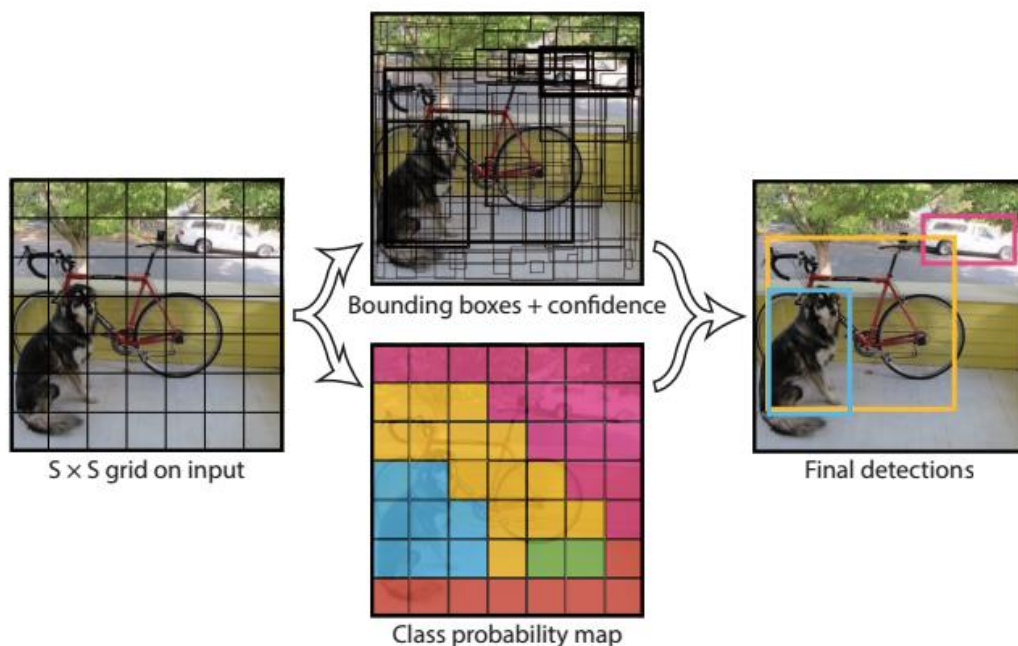
4.3.2 שימוש מערכת Yolo

המערכת מקבלת קובץ משקלים מאומן ותמונה, והפלט של המערכת הוא אותה
התמונה, אך על גביה בוצעו סימונים של פריטים שהמערכת מצאה לפי קובץ המשקלים.

קובץ המשקלים נוצר לאחר תהליך ארוך של הכנת נתונים מתאימים ואימון. תהליך
הכנת הנתונים :

- הכנת כמות מספקת של תמונות בפורמט jpeg.. כמות סבירה של תמונות
הנחוצה לביצוע אימון טוב היא כ-300 תמונות. הכנת התמונות מבצעת באמצעות כלי שנקרא
BBox Label Tool. כלי זה הינו כלי סימון. את התמונות מעלים לכלי, ועל כל תמונה מבצעים
סימון של הפריט הנדרש לזיהוי. לאחר ביצוע הסימונים על כל התמונות נוצר עבור כל תמונה
קובץ txt שמכיל : בשורה הראשונה את מספר הסימונים בתמונה, והחל מהשורה השנייה, את
הקואורדינטות של כל סימון על גבי התמונה.
- לאחר ביצוע הסימון, יש להעביר את הקבצים תהליך Convert שהופך את
המספרים שבקבצי txt לפורמט של המערכת. לשם כך השתמשנו בסקריפט מיוחד שנמצא
במאגר הקוד.
- לאחר שכל קבצי הטסט נמצאים בפורמט המסודר, יש לחלק את הנתונים ל-2
סטים : אחד עבור האימון והני עבור הבדיקה. לשם כך בוצע שימוש בסקריפט שמאפשר בחירה
של אחוז הנתונים שיוקצה לאימון. החלוקה מתבצעת באופן אוטומטי על פי סקריפט זה ויוצר
שני קבצים, אחד לאימון ואחד לבדיקה, כאשר שניהם בעצם קבצי טקסט שמכילים רשימה של
התמונות – כל קובץ והתמונות ששייכות לשלב זה.
- לאחר מכן, יש להכין 3 קבצי קונפיגורציה : קובץ cfg שמכיל פרטים על האימון –
חלוקה לקבוצות, כמות השכבות קונבולוציוניות ועוד. קובץ names שמכיל את כל השמות של
הפריטים, הקלאסים, שאנו רוצים שהמערכת תזהה. אלו בעצם Labels שיסומנו על גבי תמונת
התוצאה. במקרה שלנו יש רק שורה אחת : bone. והקובץ השלישי הינו קובץ datan שמפרט
ניתובים data שלו שהם בעצם התמונות והיכן למקם את קבצי המשקלים שנוצרים. קובץ זה
גם מכיל הפנייה לקובץ names.

- כשכל האמור לעיל מוכן, מריצים את שלב האימון. יחד עם קובץ משקלים שהוא pre-trained, ויחד עם קבצי הקונפיגורציה, מריצים את האימון. מספר האיטרציות המלא הוא 100k, זהו מספר שניתן לשנות בקונפיגורציה, אך ההמלצה היא לתת למערכת לרוץ על האימון כמה שיותר. לאחר כל איטרציה מופיעה פלט על המסך שמראה בין השאר את loss של הנתונים באימון. ערך זה יורד אקספוננציאלית. כאשר loss מגיע נמוך (בערך ל-3) ונשאר יציב, ניתן לעצור את המערכת. כל כ-1k איטרציות נוצר קובץ משקלים מעודכן בו ניתן להשתמש, ובסיום עבודת המערכת הקובץ הזה הוא הכי מעודכן.
 - כעת מריצים יחד עם קובץ המשקלים את תהליך הזיהוי של התמונה, שבסופו מתקבלת תמונת התוצאה. בנוסף לתמונת התוצאה, מופיע על גבי הטרמינל כל הקלאסים שזוהו, יחד עם ההסתברות שזה הם.
- מערכת זו פועלת על תמונות דו-מימד, ועל תמונות בפורמט jpeg בלבד. לשם המרת התמונות מפורמט nii שהתקבל לאחר ריצת DeepMedicn, השתמשנו בסקריפט המרה שנמצא במאגר הקוד.



תרשים 9 - פעולת זיהוי האובייקטים במערכת Yolo

כפי שניתן לראות בתמונה, המערכת מתייחסת לזיהוי כבעיית הרגרסיה. היא מחלקת את התמונה לגריד בגודל $S \times S$, ועבור כל תא בגריד היא חוזרת B פריטים מהתמונה, ו-C קלאסים אפשריים ברמה ההסתברותית. יחד, הם יוצרים את הזיהוי המלא – כמה פריטים יש בתמונה וכל אחד מהם – מה סיווגו (לפי הscoren ההסתברותי המקסימלי שקיבל).

המערכת ממומשת באמצעות שפת C, ודורשת התקנות מסוימות. לאחר הורדת החבילה של המערכת, יש לבצע make ונוצר קובץ הרצה, לו יש שימוש שמפורט במאגר הקוד.

זמן ריצת שלב האימון של קובץ המשקלים הוא כ-4 ימים. ההערכה היא שהזמן המלא הוא כ-8 ימים, הריצה הופסקה באופן יזום באיטרציה 51.2k מתוך 100k, משום שהloss נע סביב ה-3 וכבר לא ירד מזה זמן רב. מה שאומר שרמת הדיוק הגיע למקסימום האפשרי בהינתן הנתונים הללו.

5 ניסויים ובדיקות

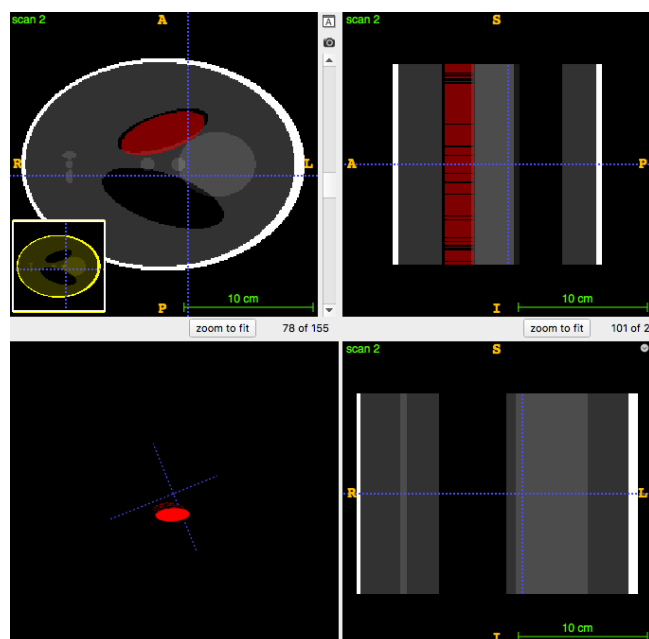
5.1 מערך ניסויים

המערכת שלנו כוללת שתי תתי מערכות, ולכן לכל מערכת נבצע בדיקות בנפרד.

DeepMedic 5.1.1

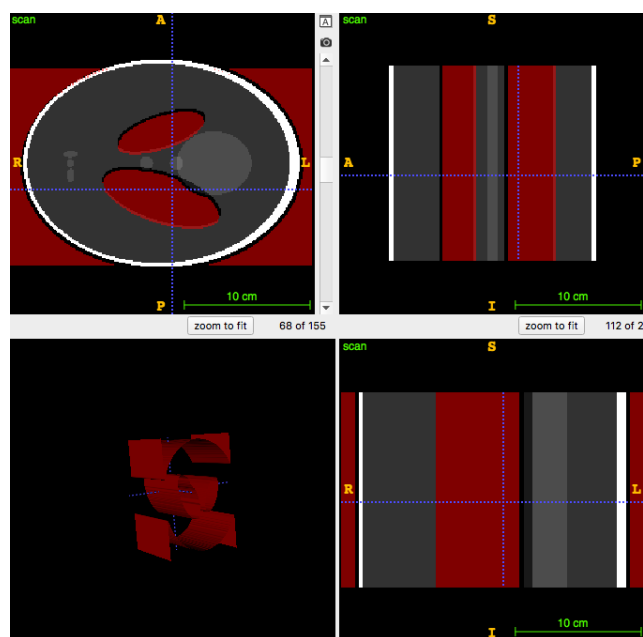
בדיקות המערכות המפורטות מטה בוצעו בכל הרצה של הרשת הקטנה (tinyCnn) ושל הרשת המלאה. בצורה כזו וידאנו שקבצי הקונפיגורציה תואמים את דרישות המערכת, הנתונים הוכנו במלואם והם אכן קיימים, ומבוצע שימוש נכון בפקודה ובדגלים בטרמינל.

- בדיקת הרשת tinyCnn עם דאטה סינטטי, באמצעות פונקציית Phantom בתוכנת Matlab. פונקציה זו מייצרת צורות ב-Grayscale ואפשרות לבצע סגמנטציות של חלק מהצורות. בדיקה זו נועדה על מנת לחוש את הרשת הקטנה, כיצד היא פועלת, והאם היא פועלת בצורה מיטבית על נתונים פשוטים. המערכת זיהתה היטב את התמונות שנוצרו. תמונה 1 מתארת תמונת Grayscale עם סימון של אליפסה. תמונת ה-Grayscale היא בעצם הסריקה, והסימון הוא תמונת ה-GT.



תמונה 1 - דאטה סינטטי, סריקה עם סגמנטציה

ובתמונה 2 מתוארת התוצאה, כאשר המערכת זיהתה את האליפסה השנייה שהיא באותו הצבע, ואת הרקע שמחוץ לאליפסה הגדולה שגם הוא בצבע שחור.



תמונה 2 - תוצאת ריצה של דאטה סינטטי

- בהרצת המערכת קיימת בדיקה של הנתיב שניתן בשימוש בפקודה.
- בדיקה של מימדי התמונה – שהמימדים תואמים (שורה 106).
- בדיקות שימוש מבחינת דגלים בטרמינל.
- בדיקות Accuracy ב- accuracyMonitor.py :
- דיוק ממוצע : $(TruePos + TrueNeg) / \text{All predicted voxels}$.
- רגישות ממוצעת : $TruePos / RealPos$.
- בדיקת ייחודיות : $TrueNeg / RealNeg$.
- בדיקת Dice ממוצעת : $[2 * (\text{All common between TruePos and RealPos}) / (TruePos + RealPos)]$
- ב- cnn3d.py :
- בדיקות דיוק שונות בסיום כל Epoch.
- בדיקות קצב למידה :
- בדיקות מימדים.
- בדיקת דגלים.

Yolo 5.1.2

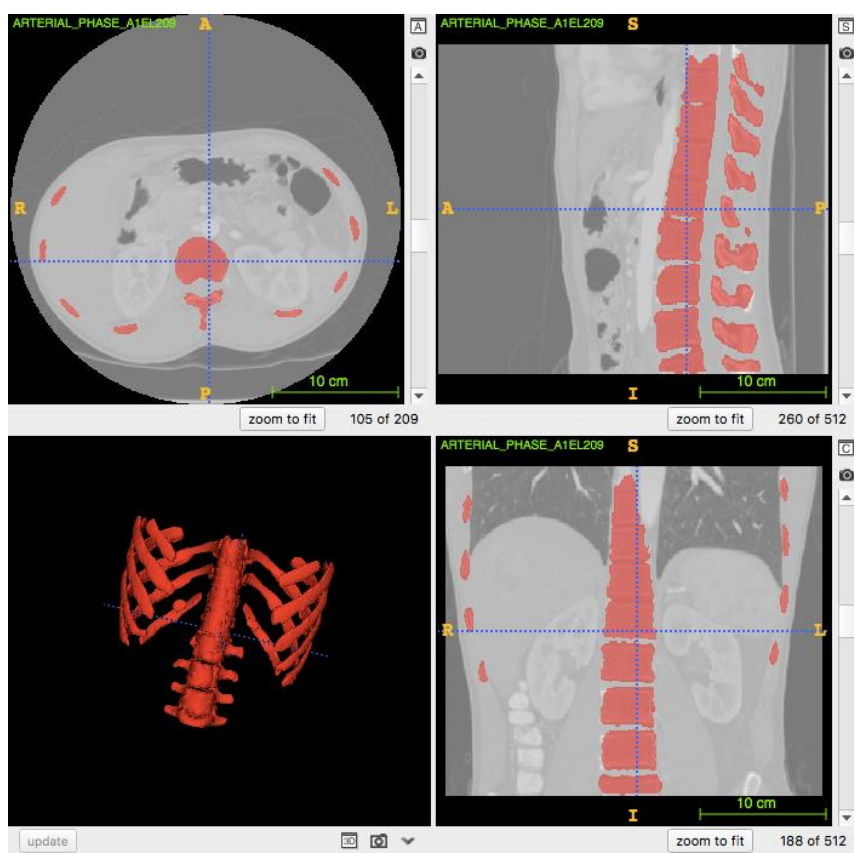
- במערכת Yolo בוצעו בדיקות של תמונות אל מול קבצי משקלים.
- בוצעו מספר בדיקות :
- תמונת GT אל מול המשקלים המקוריים שסופקו על ידי המפתחים.
 - תמונת סריקה Scan אל מול המשקלים המקוריים שסופקו על ידי המפתחים.
 - תמונת סריקה Scan אל מול קבצי משקלים שעברו אימון על תמונות שלנו ואמורים לזהות את העצמות.

6 תוצאות ומסקנות

6.1 תוצאות הניסויים, המדידות והפעלת המערכת
שתי תתי-המערכות המרכיבות את המערכת שלנו הורצו מספר פעמים.

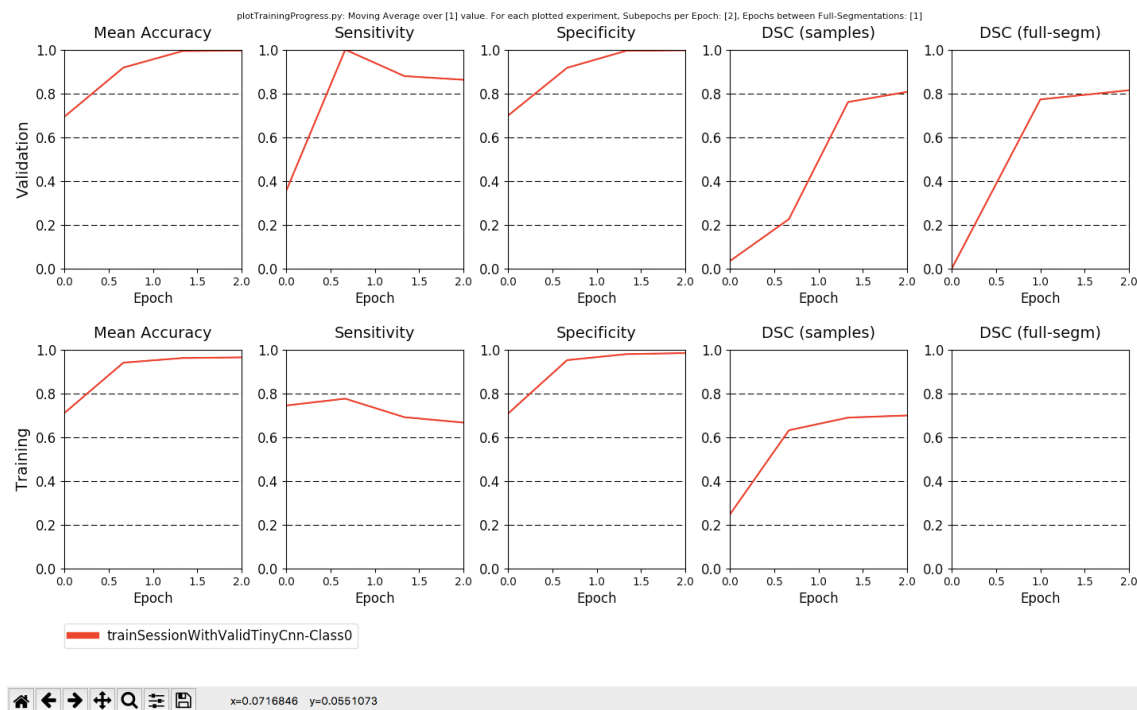
6.1.1 תוצאות מערכת DeepMedicn

דוגמא לתמונת סגמנטציה שהוכנה עבור סריקה בממדים 512x512x209 :



תמונה 3 - תמונת סגמנטציה 512x512x209

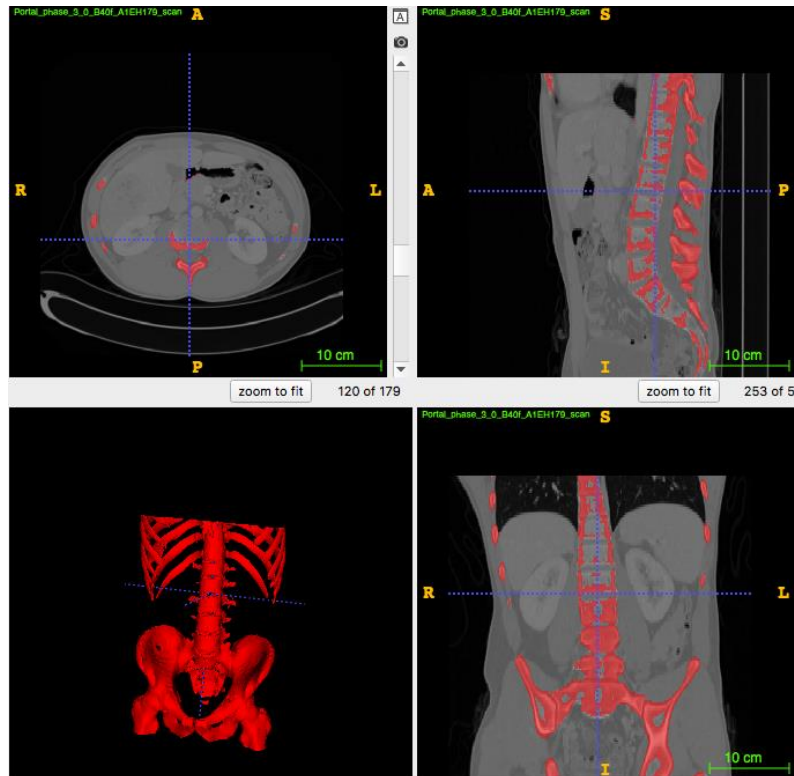
בהרצת אימון תוך שימוש ברשת הקטנה tinyCnn התקבל גרף 1 :



גרף 1 - ריצת אימון על *tinyCnn*

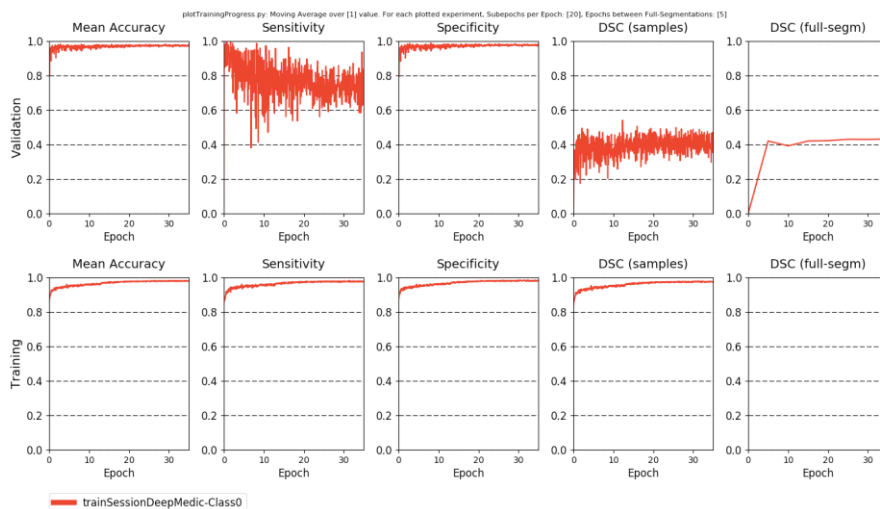
על סמך הגרף ניתן לומר שבשימוש במעט Epochs, אפשר לראות דיוק ממוצע יותר טוב מאשר באימון ברשת המלאה.

בריצת test התקבלה הסגמנטציה :



תמונה 4 - סגמנטציה 512x512x179

בגרף 2 הריצה מצביעה על דיוק ששואף ל-1 בכל Epochs שבשלב האימון (הגרפים בשורה התחתונה), Dice בינוני בריצה על הוולידציה :

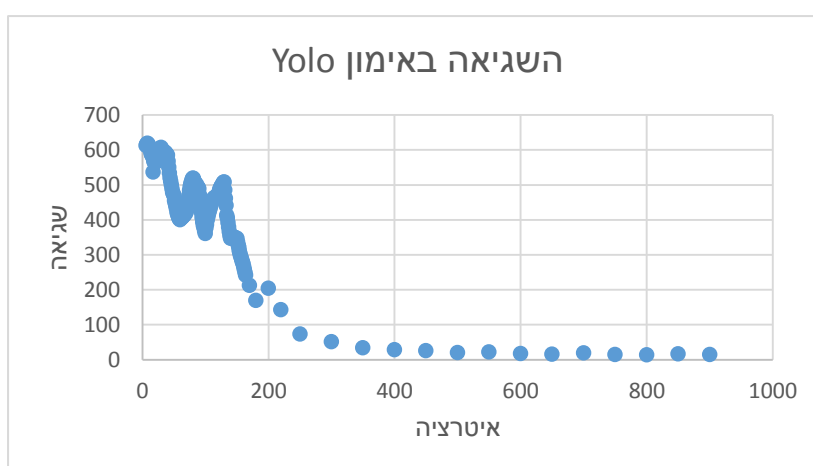


גרף 2 - ריצת אימון על הרשת המלאה

6.1.2 תוצאות מערכת Yolo

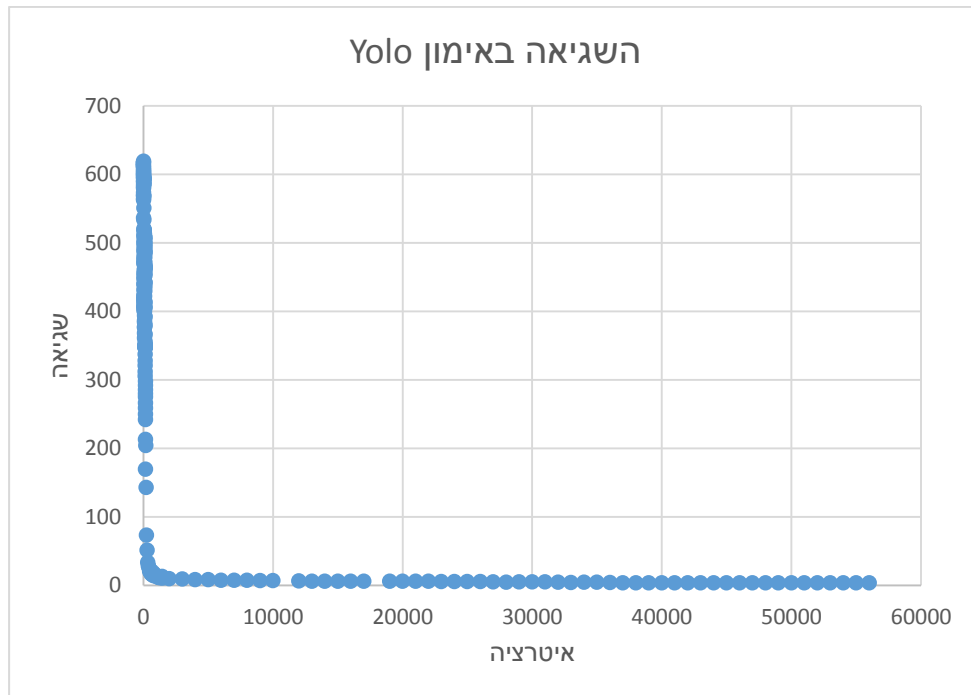
בוצע אימון שהתבסס על 209 תמונות מסומנות. האימון עבר מעל ל-50k איטרציות, החל משגיאת loss נתונים של 612~ והגיע עד ל3 באופן אקספוננציאלי.

האימון החל עם שגיאה מאוד גדולה של מעל 600 וככל שהתקדם באיטרציות, המגמה הייתה מגמת ירידה כפי שניתן לראות בגרף 3. ניתן גם לראות את הקפיצות בשגיאה. הקפיצות הודגשו ב-200 האיטרציות הראשונות ולכן מוצג גרף 3 כגרף מצומצם – רק 100 איטרציות ראשונות:



גרף 3 - שגיאה באימון Yolo לאיטרציה

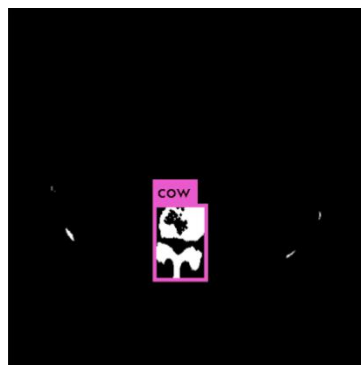
בגרף 4 מוצגת השגיאה לכל אורך האימון, עד שהופסק. ניתן לראות בבירור את הירידה המהירה ואת ההתייצבות סביב ערכים נמוכים. הקווים האדומים מצביעים על ימי האימון, כאשר האימון הופסק במהלך היום הרביעי. כבר ביום הראשון לאימון, השגיאה ירדה באופן דרסטי והגיעה לערכים של 15-17, תוך המשך ירידה. וביום הרביעי השגיאה התייצבה סביב ה-3 ומכאן כבר לא נרשמה ירידה כלל ולכן האימון הופסק.



גרף 4 - השגיאה באימון Yolo - הגרף המלא

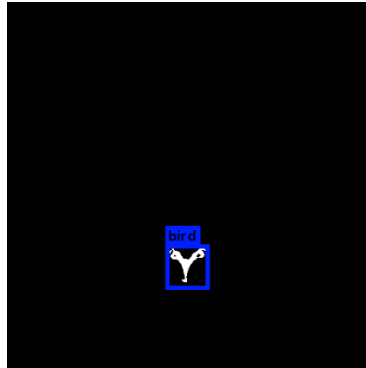
- בהינתן תמונת GT, בדיקת סיווג אל מול קובץ המשקלים המקורי שסופק על ידי המפתחים:

- בראשונה התקבל סיווג cow של 70%:



תמונה 5 - סימון Yolo על גבי תמונת GT

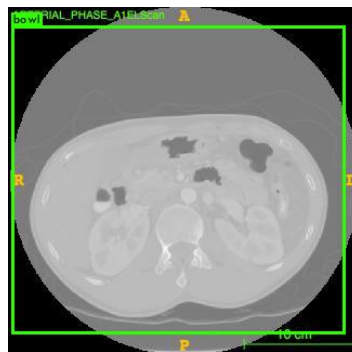
- בשנייה התקבל סיווג bird של 61% (התקבל סימון אחד):



תמונה 6 - סימון Yolo על גבי תמונת GT ניסיון נוסף

- בהינתן סריקה, בדיקת סיווג אל מול קובץ המשקלים המקורי, התקבל סיווג של

33%-ב bowl



תמונה 7 - סימון Yolo על גבי סריקה

- בהינתן סריקה, בדיקת סיווג אל מול קובץ משקלים ראשון שעבר אימון על נתונים תואמים Labels תואמים, התקבלה תמונה 6 - סיווג bone עם 10 סימונים. לכל סימון הוצמד אחוז שמהווה את scoren שקיבל הסיווג, והוא מפורט להלן:

bone: 76%

bone: 76%

bone: 85%

bone: 76%

bone: 86%

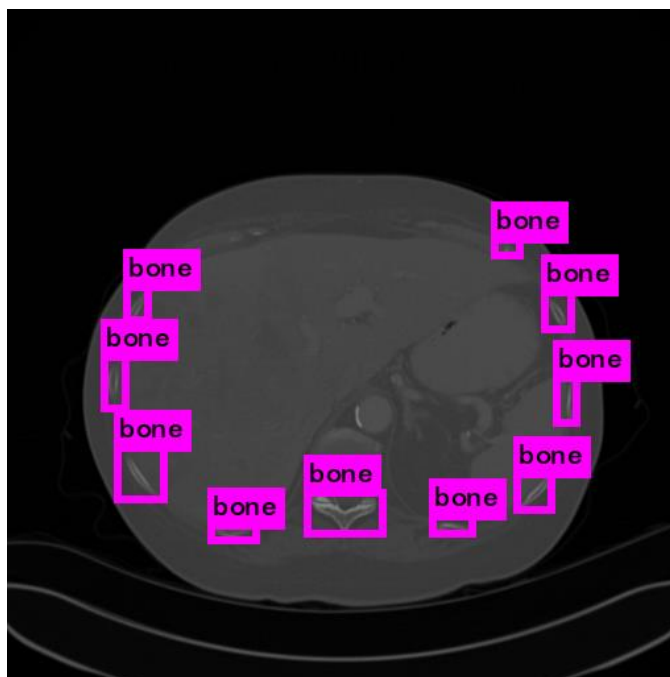
bone: 86%

bone: 83%

bone: 74%

bone: 53%

bone: 53%

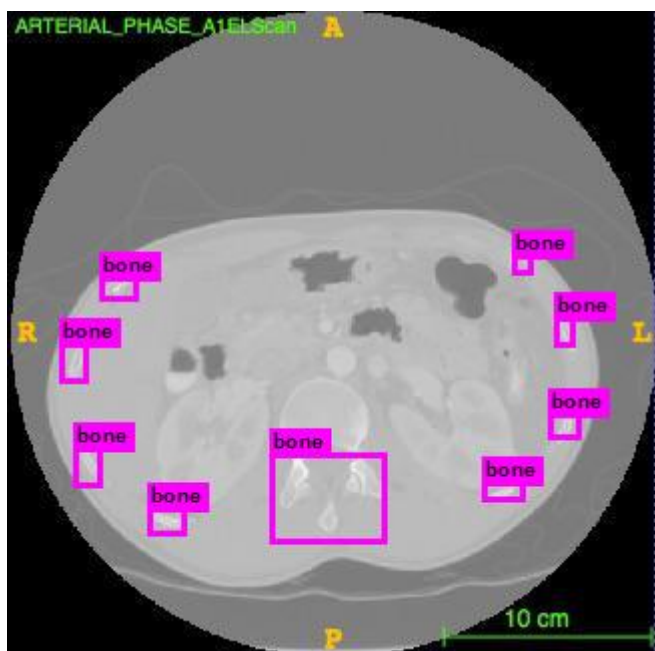


תמונה 8 - תוצאת זיהוי yolo

ניתן לשים לב כי המערכת זיהתה באופן יפה את העצמות גם כאשר הזוויות הן שונות. למשל העצם הימנית העליונה נמצאת בזווית אחרת לחלוטין מהעצם השמאלית העליונה ועדיין זוהתה. כמו כן, הסימון המרכזי – הצורה של העצם שונה לחלוטין מהשאר, ועדיין העצם זוהתה.

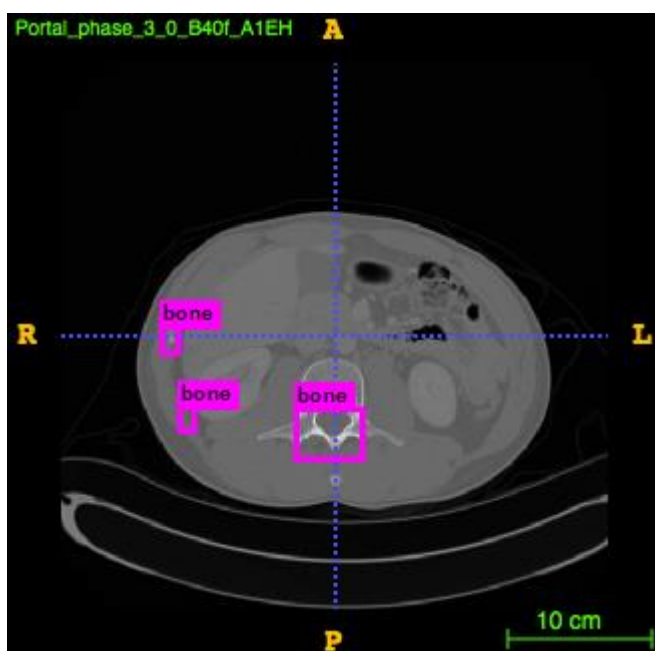
מה שמחוזה שהמערכת עובדת כראוי והאימון סיפק את כל הצורות האפשריות שהוכנסו
בסימון התמונות לאימון.

תמונה נוספת, הפעם מסריקה אחרת :



תמונה 9- זיהוי של Yolo על סריקה אחרת

ועוד תוצאה מתמונה נמוכה, שלא כוללת את כל העצמות :



תמונה 10 - זיהוי של Yolo על תמונה נמוכה

סה"כ – הזיהוי בוצע בצורה יפה מאוד, על תמונות שונות, בממדים שונים, וממיקומים בחתכים השונים. מדובר על אימון שלקח 4 ימים על GPU ולכן התוצאות מעידות על אימון נכון וטוב.

6.2 מסקנות הפרויקט

- הכנה טובה של נתונים (data), דיוק בתשובות, חשוב מאוד ללמידה ולהכנה של משקלים שישמשו את תהליך הבדיקה ותקינות המערכת לאחר האימון.
- הכנסת "רעשים" למערכת – קצת נתונים שגויים, עוזרים בהפקה של תוצאות אפילו יותר מדויקות, המערכת יודעת לזהות פריטים שדומים.
- ככל שהאימון מדויק יותר, תמונות פגומות או פריטים פגומים בתמונות, בעלי סבירות גבוהה יותר להיות מזוהים ומסומנים, וזאת בשל האיכות הגבוהה של הלמידה.
- כמות רבה יותר של נתונים עם סימונים נכונים, בצורות מסוימות, תורם ליצירת משקלים נכונים. למשל, במערכת Yolo, מעבר לצורה הסטנדרטית של העצם בצורת אליפסה חלולה, ניתנו גם צורות אחרות, למשל של האגן, וגם אלו זוהו כעצמות, באותו הסיווג. זאת בשל אימון איכותי ותמונות מסומנות היטב עם מגוון רחב של אפשרויות עבור סיווג אחד ויחיד.
- רשתות קונבולוציוניות הן כיום הבחירה הטובה ביותר ללמידה עמוקה של תמונות – גם בתלת מימד וגם בדו מימד. שתי המערכות שלנו ביצעו למידה בעזרת אלגוריתם זה, והתקבלו תוצאות מעניינות ונכונות. מצד אחד נבנה מודל תלת מימד ומצד שני בוצעו סימונים דו מימדיים עם התמונות המבוקשות.
- נראה שרשת עמוקה לאו דווקא עדיפה על רשת קטנה – DeepMedic התקבלו תוצאות מדויקות ונקיות יותר כאשר בוצע אימון ובדיקה על הרשת הקטנה מאשר על הרשת הגדולה.
- שימוש ב-GPU אכן מגביר באופן משמעותי את כח החישוב, ועדיין התקבלו תוצאות יפות ומספקות.
- ניתן עוד לפתח את המערכות ולהגיע לדיוקים גבוהים יותר. ניתן לספק לרופא כלים חדשניים יותר תוך שימוש בטכנולוגיות הכי חדשות ומתקדמות על מנת להקל עליו/ה בעבודה, ולעזור להגיע לאבחנות טובות יותר ומקצועיות יותר.

6.3 סיכום

העבודה על פרויקט זה הייתה מאתגרת במיוחד ומעניינת.

התמודדנו עם שלל נושאים חדשים שטרם ראינו, כמו עיבוד תמונה, מערכות למידה, עבודה מול GPU, עבודה עם מערכות קיימות תוך שינוי ייעודן.

במיוחד מאתגרת הייתה העבודה מול המערכות הקיימות ותפעולן על נתונים שסופקו על מנת לעמוד בדרישות ובבעיה של פרויקט זה.

כאשר הנתונים הסינטטיים הוכנסו לתוך מערכת DeepMedic, הייתה הבנה של דרך פעולת המערכת, מה כל פרמטר אומר, מה משמעותה של כל תמונה ותפקידה. ומאותו הרגע הפרויקט החל לצמוח ולתפוס גבהים ותוצאות. הכנת תמונות הסגמנטציה היו אתגר גם כן והיוו אבן דרך בשכלול הפרויקט והבאתו לדיוק טוב יותר והבנה טובה יותר של מושג הסגמנטציה.

במערכת Yolo האתגרים היו דומים, כמו טיפול בסקריפטים וטיפול בנתונים באופן ידני – מעבר על כל תמונה ותמונה וביצוע סימון על גביהן. כאן התווסף אתגר של הצורך לבצע בדיקת תקינות, אך הבדיקה מתבצעת רק לאחר ששלב האימון מסתיים, וזה, כאמור, לקח כ-4 ימים. כך נוצר מצב שבקובץ המשקלים הראשון שהוכן, קרתה טעות בניית קובץ השמות, והסיווגים כולם קיבלו ערכים דיפולטיביים person במקום bone שהתבקש.

7 ספרות

7.1 מאמרים

מתחילת הפרויקט בוצע סקר ספרות שהעלה:

- מאמרם של Kontantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, Ben Glocker משנת 2016. מציעים רשת תלת מימדית קונבולוציונית, בעלת עומק של 11 שכבות לסגמנטציה של פגיעות במוח. הארכיטקטורה המוצעת היא תוצאה של ניתוח של גבולות של רשתות דומות באפליקציות קיימות. במאמר הם מבצעים ניתוח עמוק של הרשתות, ומציעים דרכי חישוב אחרות, מחולקות יותר, על מנת להתמודד עם מורכבות החישוב ברשתות מסוג אלו. מציעים חלוקה של תמונות הקלט לחלקים. בנוסף, מציעים דרכים להתמודדות עם מצבי false-negative.

<http://www.sciencedirect.com/science/article/pii/S1361841516301839>

- מאמרם של Joseph Redmon, Ali Farhadi משנת 2016. בו מציגים מערכת זמן אמת לזיהוי אובייקטים, שיכולה לזהות מעל 9000 קטגוריות בתמונה. מציעים שיטה ללמידה מהירה ומדויקת של אובייקטים באמצעות רשתות קונבולוציוניות. המאמר נכתב על נתונים מסוג coco ו-pascal, אך השיטה ניתנת להתאמה לנתונים אחרים, כאשר מספקים אותם בצורה מסודרת, באיכות וכמות מספקים.

<https://arxiv.org/pdf/1612.08242.pdf>

- מאמרם של Konstantinos Kamnitsas, Liang Chen, Christian Ledig, Daniel Rucckert and Ben Glocker משנת 2015. במאמר מתואר בקצרה רשת נוירונים קונבולוציונית בעלת 11 שכבות, מפותלת לשני נתיבים ותלת מימדית. הטענה במאמר היא שניתן לאמן מערכת שכזו לזיהוי גידולים ופגיעות במוח שמבוסס על סטי-נתונים בעלי 28 מקרים. התוצאות המובטחות הן בעלות דיוק Dice של 64%.

<http://www.isles-challenge.org/ISLES2015/articles/kamnk1.pdf>

8 רשימות

8.1 רשימת גרפים

- גרף 1 - ריצת אימון על tinyCnn 32
גרף 2 - ריצת אימון על הרשת המלאה 34
גרף 3 - שגיאה באימון Yolo לאיטרציה 34
גרף 4 - השגיאה באימון Yolo - הגרף המלא 35

8.2 רשימת תרשימים

- תרשים 1 – תהליך למידה 11
תרשים 2 – Use cases 14
תרשים 3 - אופן פעולת המערכת 16
תרשים 4- קונבולוציה 18
תרשים 5- רשת Deep Medic המקורית המלאה 19
תרשים 6 - רשת tinyCnn 21
תרשים 7 - פעולת yolon 21
תרשים 8 - ארכיטקטורת Yolo 22
תרשים 9 - פעולת זיהוי האובייקטים במערכת Yolo 27

8.3 רשימת תמונות

- תמונה 1 - דאטה סינטטי, סריקה עם סגמנטציה 29
תמונה 2 - תוצאת ריצה של דאטה סינטטי 29

31.....	תמונה 3 - תמונת סגמנטציה 512x512x209
33.....	תמונה 4 - סגמנטציה 512x512x179
35.....	תמונה 5 - סימון Yolo על גבי תמונת GT
36.....	תמונה 6 - סימון Yolo על גבי תמונת GT ניסיון נוסף
36.....	תמונה 7 - סימון Yolo על גבי סריקה
37.....	תמונה 8 - תוצאת זיהוי solo
38.....	תמונה 9 - זיהוי של Yolo על סריקה אחרת
38.....	תמונה 10 - זיהוי של Yolo על תמונה נמוכה

8.4 רשימת טבלאות

19.....	טבלה 1 - שכבות ברשת deepmedic
23.....	טבלה 2 - מבנה רשת ה-CNN של מערכת Yolo

9 מקורות ספרותיים

-
- [1] <http://www.sciencedirect.com/science/article/pii/S1361841516301839>
- [2] <https://arxiv.org/pdf/1612.08242.pdf>
- [3] http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
- [4] <http://www.infomed.co.il/examination-310/>

Abstract

The project deals with analyzing medical images – CT images, by using programmed image processing and machine learning. In the center of the project there are two systems that deals with analyzing medical images, the first is 3D and the second is 2D. The systems analyze the given image and will mark the rib's bones upon it. The system's output will be the same CT scan, but the bones marking – every system does her job in a different way. This action takes a lot of time while being made by the doctor. The whole system that we introduce does not exist today, and being an innovative.

Software Engineering Department

Machine Learning to Medical Data

– Analyze CT Images

By

Alon Tsalik Shmilovich

Stav Barazani

Academic Supervisor: Dr. Spanier Assaf B.



Software Engineering Department

Machine Learning to Medical Data

– Analyze CT Images

By

Alon Tsalik Shmilovich

Stav Barazani

June 2017

Sivan-Tamuz 5777