

שמות: ואדים ליטבינוב, ירון גפן

תעודות זהות: 305641508, 314552365

## שיטות מתקדמות בעיבוד תמונה רפואי - דו"ח מסכם

### מבוא

נושא הפרויקט ומטרת העל שלו הינה הרכבת מודל אשר יקבל כקלט צילומי ריאות ויסווג אותם ל-4 מחלקות:

- שלילי לדלקת ריאות
- דלקת ריאות טיפוסית לקורונה
- דלקת ריאות לא טיפוסית לקורונה
- לא ניתן לקבוע.

מאגר הנתונים נלקח מ־Kaggle [1] ונוצר על ידי SIIM (Society for Imaging Informatics in Medicine), חברה אשר מטרתה הנגשת מידע ונתונים רפואיים על מנת לעזור באתגרים שונים במחקר ובחדשנות. הפורמט של הדוגמאות במאגר שלנו נקרא DICOM (Digital Imaging and Communications in Medicine) והוא מהווה סטנדרט לדימות רפואי במאגרי מידע.

הקורונה (שם מדעי: SARS CoV-2) הינה נגיף נשימתי אשר התגלה בדצמבר 2019 בסין ורוצף בהמשך [2]. מאז, הוא חולל שמות בכל העולם [3], כאשר כמעט בכל המדינות אנשים רבים מתו ממחלה כתוצאה מהנגיף (COVID19) או חלו במצב קשה ממנה. הנגיף הינו נגיף נשימתי ונכנס לגוף מדרכי הנשימה לרוב, וכך הוא גם מופץ ויוצר הידבקות נרחבות באוכלוסייה. בהיותו נגיף נשימתי, תסמין נרחב שרואים בחולי קורונה קשים ובינוניים הוא דלקת ריאות [4]. במהלך הזמן, רופאים וחוקרים הבחינו כי דלקת הריאות אשר נגרמת מקורונה היא בעלת מאפיינים טיפוסיים ושניתן לבצע אבחנה מבדלת על פי צילומי CT [5].

### סקירת ספרות ועבודות קשורות

כאשר התקבעה העמדה בספרות כי דלקת הריאות שנגרמת מקורונה היא בעלת מאפיינים שונים מדלקת ריאות רגילה, החלו להשתמש בלמידה חישובית בתחום. היתרון הגדול ביותר של שימוש בלמידת מכונה על מנת לסווג אדם כחולה קורונה הוא חיסכון בזמן האבחנה לעומת שיטת הבדיקה הקלאסית של RT-PCR שיכולה להימשך 24 שעות ומעלה. בנוסף, סיווג על ידי למידת מכונה יכול להיות לעזר לרופאים כאשר הם מנסים לסווג צילומי CT ריאות בעזרת אבחנה מבדלת.

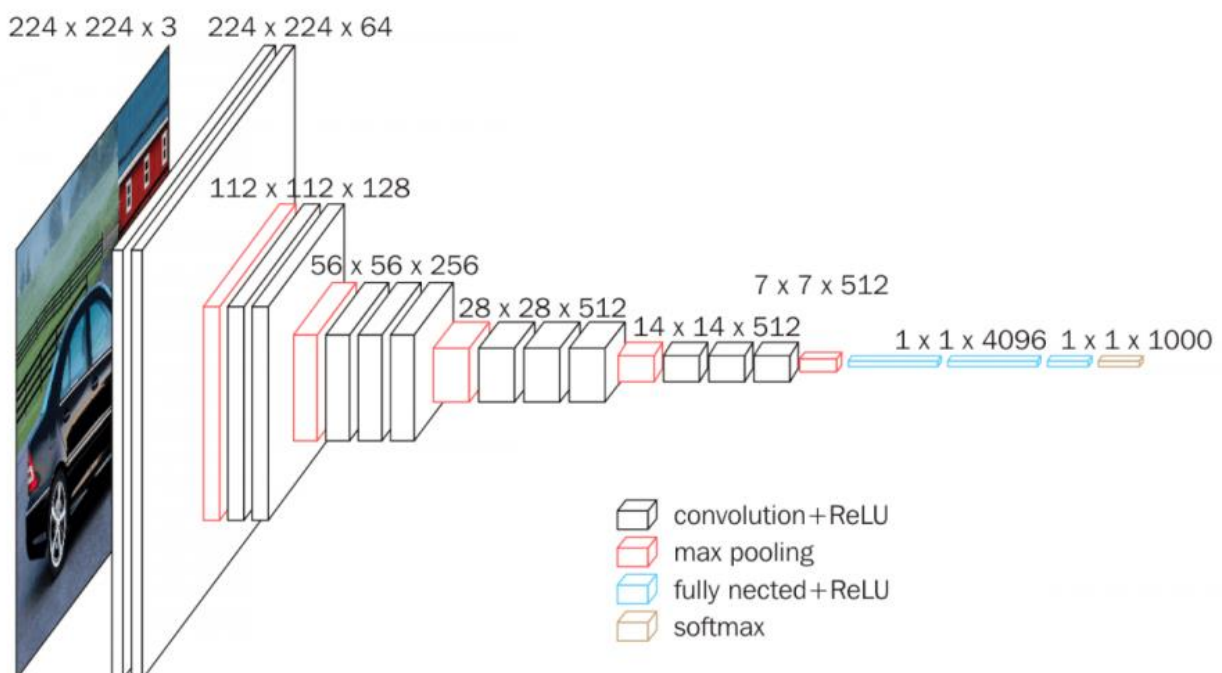
במאמר "The clinical classification of patients with COVID-19 pneumonia was predicted by Radiomics using chest CT" [6] החוקרים ניסו למצוא מאפיינים מרכזיים בסיווג בין צילומי ריאות של חולים קשים ובינוניים. במשימה זו הם נעזרו באלגוריתם גנטי אשר בעזרת שינויים קטנים ("מוטציות") בסוג האלגוריתם, הפרמטרים שלו, המאפיינים שעל פיהם מסווגים ועוד, מצא את המודל המוצלח ביותר לסיווג ואת המאפיינים בעלי החשיבות הרבה ביותר שעל פיהם ניתן לסווג. המודל השיג AUC של קרוב ל-1 והיה מוצלח בהחלט. לבסוף, נערכו ניתוחים סטטיסטיים על המאפיינים ולעתים ראו הבדלים מובהקים בין הקבוצות ולעתים לא. בעיות עיקריות אשר נצפו היו שההכללה של המודל לא הייתה מוצלחת במיוחד ובנוסף נצפתה התאמת-יתר מסוימת, ככל הנראה כתוצאה ממספר מועט של דוגמאות.

מאמר נוסף שעוסק בתחום המשיק לנושא הפרויקט נקרא "Classification of COVID-19 Chest CT Images Based on Ensemble Deep Learning" [7]. בעיית הסיווג של המאמר דומה לבעיה שלנו כאשר שלושת מחלקות הסיווג הן:

- דלקת ריאות טיפוסית ל-COVID19
- דלקת ריאות רגילה
- ללא דלקת ריאות (קבוצת ביקורת)

במאמר נעשה שימוש בלמידה עמוקה, ובפרט ברשת קונבולוציה ידועה בשם VGG16.

להלן ארכיטקטורת הרשת המדוברת:



בנוסף, המאמר השתמש בשיטה בשם Stacked Generalization ensemble learning. בשיטה זו מבצעים חלוקה של קבוצת האימון ל- $m$  קבוצות קטנות יותר שלא חופפות ביניהן, ויתבצע אימון של הרשת על כל קבוצת אימון בנפרד.

הגדרה פורמאלית של חלוקת הקבוצות עבור  $U$  קבוצת האימון הכוללת ו- $|U| = N$  הינה:

$$\begin{aligned} U &= U_1 \cup U_2 \cup U_3 \cdots U_m, \\ \phi &= U_1 \cap U_2 \cap U_3 \cdots U_m, \\ \frac{N}{m} &= \text{Num}(U_i), \quad i = 1, 2, 3, \dots, m. \end{aligned}$$

בשלב ההיסק, כל אחת מ- $m$  הרשתות יקבלו את הדוגמה, והסיווג יתבצע בעזרת הכרעה מסוימת. אחוזי הדיוק שמודל זה הניב היו גבוהים למדי, וכך גם לגבי אחוזי ה-sensitivity, specificity וה-precision.

## שיטות

אלגוריתם ה-base line שנבחר לבעיית סיווג זו הוא הפרספטרום. הפרספטרום הוא אלגוריתם ותיק אשר היה בין הראשונים להשתמש ברעיון של למשקל משתנים שונים (לעומת K-Nearest-Neighbors לדוגמה שמתבונן על כל המימדים באופן שווה). הפרספטרום הקלאסי עסק בבעיית סיווג דו מחלקתית, אך ניתן להרחיב אותו גם לבעיית סיווג רב מחלקתית ולכן היה להשתמש בו בבעיה שלנו.

הפרמטרים של הפרספטרום שנבחרו הם:

- ללא רגולריזציה, כיוון שאין חשש רב שמסווג לינארי יצליח לזכור דוגמאות מורכבות ממימד בגודל 50,176 (224x224) ולהגיע להתאמת יתר
- הפרספטרום למד גם את ה-bias (ה- $b_0$  שקובע היכן מישור ההחלטה חותך את ציר ה- $\gamma$  כאשר כל המשתנים מאופסים)
- מספר ה-epochs המקסימלי הוגדר ל-1000. הפרספטרום מגיע למספר זה רק כאשר ה-loss הוא לא יציב ומתקשה להגיע להתכנסות בזמן מוקדם יותר.
- מקדם הלמידה היה מקדם ברירת המחדל שהוא 1.
- מישקול המחלקות הוגדר למאוזן, על מנת להמנע מהטיה של חוסר איזון בין התיוגים (בוצע גם לפני שימוש ברשת נוירונים בהמשך)

המסווג הלינארי פרספטרון הגיע לדיוק של 38.6%, כאשר בהמשך נצפה ממודל למידה עמוקה עם סט היפותזות לא לינארי להגיע לביצועים יותר גבוהים.

### גישת למידה עמוקה מתקדמת

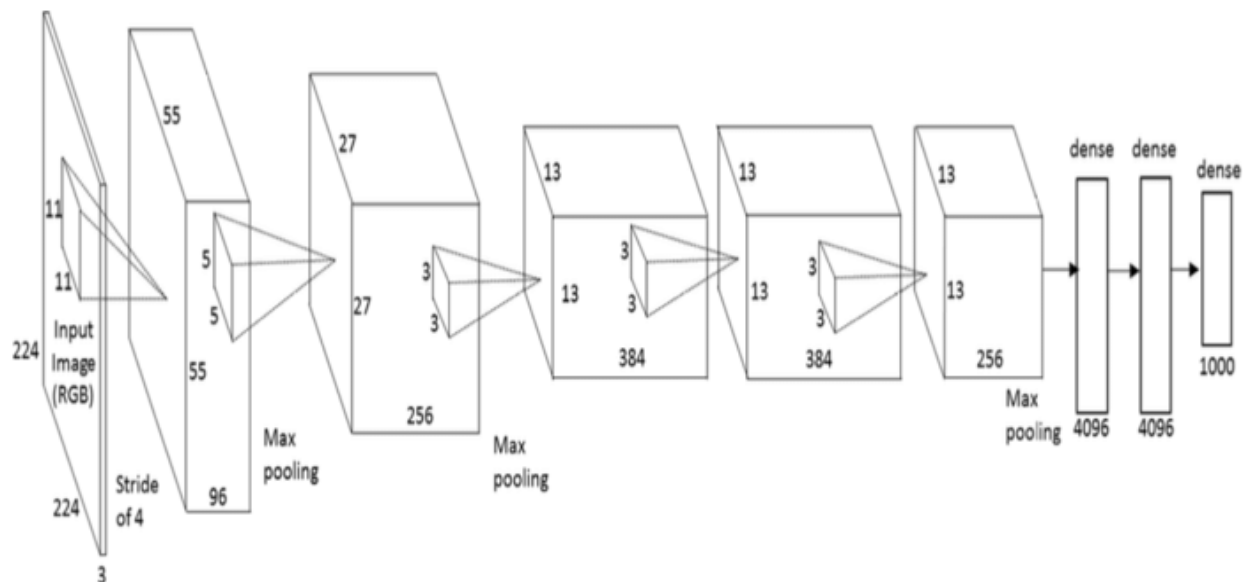
העיבוד המקדים היחיד אשר בוצע הוא איחוד של המחלקות המתויגות "דלקת ריאות לא טיפוסית ל-COVID19" ו-"לא ניתן לקבוע". עיבוד זה בא לטפל בבעיית המספרים הנמוכים של הדוגמאות המתויגות כך (7.6%-17.5% בהתאמה). בנוסף, ההפרדה הראשית והקלה יותר שעל המודל ללמוד היא ההפרדה בין האם יש או אין דלקת ריאות. לאחר מכן, בתוך "יש דלקת ריאות" עליו ללמוד הפרדה עדינה יותר בין דלקת ריאות טיפוסית ל-COVID19 לבין לא טיפוסית. הרגשנו כי הפרדה נוספת בין דלקת ריאות לא טיפוסית לבין דלקת ריאות שלא ניתן לקבוע לגביה היא לא יעילה ורק תקשה על ביצוע המשימה הראשית שהיא זיהוי חולים ב-COVID19 מסריקות הדימות שלהם.

כיוון שהדוגמאות הינן תמונות שחור לבן, והמשתנים הם מפיקסלים שמוגדרים בטווח אחיד של  $[0, 256]$ , לא הרגשנו צורך לבצע נירמול נוסף ולהעבירם לטווח  $[0, 1]$  או לתקן אותם כך שיתפלגו כמו גאוסיאן  $X \sim N(0,1)$ . לכן גם למודלים יש צורך ב-bias (bias), כיוון שהממוצע של כל משתנה אינו שווה ל-0 ועקב כך גם הדוגמאות עצמן אינן ממורכזות סביב ה-0.

את בעיית חוסר האיזון בתיוגים (דלקת ריאות טיפוסית עם ~47% מהתיוגים, כאשר שתי המחלקות האחרות פחות או יותר שוות) פתרנו בעזרת מישקול מחדש לדוגמאות המתויגות. המשקל המקורי של כל מחלקה נלקח, נורמל, והוחלף ב- $1/\text{original\_weights}$ , כך שכלל שהמחלקה יותר שכיחה כך משקל כל דוגמה ששייכת לה יהיה נמוך יותר. המשקלים הסופיים אשר התקבלו הם 0.3647 ל-0 (ללא דלקת ריאות), 0.2106 ל-1 (דלקת ריאות טיפוסית) ו-0.4247 ל-2 (דלקת ריאות לא טיפוסית + לא ניתן לקבוע) ווקטור משקולות זה ניתן לפונקציית ה-loss. דרך פעולה זו ביטלה את החשש לרשת שתעדיף לסווג לתיוג השכיח ביותר וכך להגיע לביצועים גבוהים יחסית ללא למידה.

בתחילה היה ניסיון לבנות רשת מאפס, אך כאשר התברר כי כמות הפרמטרים שאפשר לכייל היא גבוהה מדי, החלטנו להתחיל עם רשת ידועה בשם AlexNet [8] ולכייל אותה בהתאם לצרכי הבעיה שלנו.

## ארכיטקטורה:



כפי שניתן לראות, הקלט הוא תמונות ברזולוציה דומה לשלנו (224x224) אך צבעוניות (בעומק 3 לעומת 1 שלנו). לאחר מכן הארכיטקטורה היא: שכבת קונבולוציה עם פילטר בגודל 11x11 והיסט של 4, Max-pooling עם פילטר 3x3 והיסט של 2, שכבת קונבולוציה עם פילטר 5x5 והיסט של 2 (עם ריפוד באפסים בשביל לדאוג שהקונבולוציה תהיה מוגדרת מבחינת מימדים), Max-Pooling עם פילטר 3x3 והיסט של 2, שלוש שכבות קונבולוציה רצופות עם פילטר בגודל 3x3 והיסט של 1 כולל ריפוד באפסים, שכבת Max-pooling עם פילטר בגודל 3x3 והיסט של 2. פונקציית האקטיבציה בין השכבות היא Relu. לאחר מכן מבצעים השטחה של כל ה-channels ומכניסים את זה כקלט לרשת נוירונים Fully connected ששכבתה הראשונה והשניה הן בגודל 4096 נוירונים, ושכבת הפלט בגודל 1000 כיוון שיש 1000 מחלקות (במקרה שלנו שכבת הפלט תהיה בגודל 3 כי זה מספר המחלקות). בין שכבות הנוירונים גם כן יש פונקציית אקטיבציה Relu, וכאן יש גם שימוש ב-Dropout ברנולי עם הסתברות של 50% לאפס משקולת וכך להקשות על הרשת לזכור וליצור הכללה טובה יותר. קבוע הלמידה של הרשת היה 0.01, גודל ה-Batch 128, Optimizer SGD (שמהווה רגולריזציה לא ישירה).

על בסיס זה ביצענו שינויים בהיפר-פרמטרים ובארכיטקטורה והתבוננו כיצד משתנים הביצועים. השינויים בהיפר-פרמטרים והביצועים מפורטים בטבלה מטה כאשר 1 הוא AlexNet המקורי.

בסופו של דבר נבחרה הרשת הבאה, שהיא רשת #8 בטבלה למטה:

Layer (type:depth-idx)	Param #
AlexNet	--
└Sequential: 1-1	--
└Conv2d: 2-1	3,904
└ReLU: 2-2	--
└MaxPool2d: 2-3	--
└Conv2d: 2-4	51,264
└ReLU: 2-5	--
└MaxPool2d: 2-6	--
└Conv2d: 2-7	73,856
└ReLU: 2-8	--
└Conv2d: 2-9	295,168
└ReLU: 2-10	--
└Conv2d: 2-11	295,040
└ReLU: 2-12	--
└MaxPool2d: 2-13	--
└AdaptiveAvgPool2d: 1-2	--
└Sequential: 1-3	--
└Dropout: 2-14	--
└Linear: 2-15	18,878,464
└ReLU: 2-16	--
└Dropout: 2-17	--
└Linear: 2-18	16,781,312
└ReLU: 2-19	--
└Linear: 2-20	12,291
=====	
Total params: 36,391,299	
Trainable params: 36,391,299	
Non-trainable params: 0	
=====	

```
AlexNet(  
  (features): Sequential(  
    (0): Conv2d(1, 32, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))  
    (1): ReLU(inplace=True)  
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (3): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))  
    (4): ReLU(inplace=True)  
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (6): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (7): ReLU(inplace=True)  
    (8): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (9): ReLU(inplace=True)  
    (10): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (11): ReLU(inplace=True)  
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))  
  (classifier): Sequential(  
    (0): Dropout(p=0.5, inplace=False)  
    (1): Linear(in_features=4608, out_features=4096, bias=True)  
    (2): ReLU(inplace=True)  
    (3): Dropout(p=0.5, inplace=False)  
    (4): Linear(in_features=4096, out_features=4096, bias=True)  
    (5): ReLU(inplace=True)  
    (6): Linear(in_features=4096, out_features=3, bias=True)  
  )  
)
```

ההיפר פרמטרים של הרשת הם:

קבוע למידה: 0.001

epochs 18 ללמידה

גודל Batch: 16

שיטת אופטימיזציה: stochastic gradient descent

פונקציית מחיר המתאימה לפלט הסתברותי בבעיה רב מחלקתית: Cross entropy

$$CE = - \sum_i^C t_i \log(f(s)_i)$$

כאשר C זה מספר המחלקות (3 אצלנו),  $t_i$  זה התיג של המחלקה ה-i בדוגמה הנוכחית, ו- $f(s)_i$  זה סיווג ההיפותזה שלנו שקיבלה את הדוגמה ה-i.

מפאת זמן הרצה רב של הרשתות, לא בוצע grid search וניסיון של הרבה מאוד היפר פרמטרים שונים ושילובים שלהם, אלא בוצעו מספר ניסיונות ידנית של היפר פרמטרים שונים שנראו סבירים. לדוגמה בכוון קבוע הלמידה, קבוע הלמידה הבא שנוסה חולק ב-10, וכך כמה פעמים ברצף.

#### אנליזה

	#1 (AlexNet – Baseline)	#2 (AlexNet – hyperparameter tuning)	#3 (AlexNet – hyperparameter tuning)	#4 (AlexNet – hyperparameter tuning)
Input size	224	224	224	224
Learning rate	0.01	0.001	0.001	0.001
Epochs	40	30	40	18
Batch size	128	128	32	16
Preprocess	None	None	None	None
Optimizer	SGD	SGD	SGD	SGD
Parameters	57M	57M	57M	57M
Fold train time	2.2 minutes	1.6 minutes	3.1 minutes	2 minutes
Train acc	0.6	0.56	0.69	0.58
Val acc	0.47	0.49	0.54	0.56
Train loss	0.8	0.92	0.68	0.9
Val loss	1.06	0.97	1.04	0.97

	#5 (AlexNet – hyperparameter tuning)	#6 (Modified AlexNet architecture_v1)	#7 (Modified AlexNet architecture_v1)	#8 (Modified AlexNet architecture_v2)
Input size	224	224	224	224
Learning rate	0.001	0.001	0.001	0.001
Epochs	14	18	18	18
Batch size	8	16	16	16
Preprocess	None	None	None	None
Optimizer	SGD	SGD	SGD	SGD
Parameters	57M	55M	55M	36M
Fold train time	3 minutes	1.7 minutes	1.7 minutes	1.2 minutes
Train acc	0.56	0.58	0.58	0.57
Val acc	0.54	0.54	0.54	0.56
Train loss	0.94	0.9	0.9	0.91
Val loss	0.98	0.98	0.97	0.96

נציין כי במהלך העבודה התמודדנו עם בעיות רבות של סיבוכיות זמן ומקום. בגלל סיבוכיות מקום גדולה במהלך הלמידה, פעמים רבות הזיכרון הוירטואלי של המחשב (RAM) התמלא והתהליך קרס. לכן בנוסף לניסיון לפתור את בעיית הלמידה התחשבנו גם בגודל הרשת והשתדלנו להקטינה כמה שניתן עם פגיעה מזערית בביצועים. ניתן לראות כי הביצועים של מודל #8 נותנים פחות או יותר אותם ביצועים, אך יש ירידה של 21 מיליון פרמטרים מהרשת המקורית (#1) מה שגזר גם ירידה בזמן האימון של Fold בדקה שלמה.

ארכיטקטורת רשתות מספר 1-5 לא שונו ונשארו כמו שמתואר לעיל. ברשת #6 ו-#7, הפחתנו את מספר הצ'אנלים בשכבות הקונבולוציה ושינינו אותם ל-פי 2 בכל שכבה, כאשר בשכבה הראשונה ישנו מעבר מצ'אנל 1 ל-32 ובשכבה האחרונה מספר הצ'אנלים נשאר אותו הדבר (256). מעבר זה הפחית משמעותית את מספר הפרמטרים והקל על סיבוכיות המקום והזמן והביצועים כמעט ולא נפגעו. ברשת #8 בשכבה האחרונה הורדנו את מספר הצ'אנלים בחזרה לגודל הכניסה של השכבה הקודמת בכדי להקטין את מספר הנוירונים בשכבת הכניסה לחלק ה-Fully Connected של הרשת.

על מנת להעריך טיב מודל ולהשוות ביצועים בין מודלים היה שימוש ב-K-Fold-Cross-Validation אשר מחלק את מאגר הנתונים ל-K חלקים (במקרה שלנו K=5), משתמש בחלק אחד בתור קבוצת אימות\מבחן, ועל שאר הנתונים הוא מתאמן. דיוק ממוצע של K ההרצות מחושב ומשומש כמדד להערכה. בנוסף, על מנת להתמודד עם הטבע האקראי של K-Fold והחשש מדוגמאות מעטות שיבחרו להיות ב-Fold מסוים, היה שימוש ב-Stratification שוידא שבכל Fold יהיה שיעור הדוגמאות מכל תיגו כמו שיעורם במאגר הכללי.

מטריצות ה-confusion הן ממימד 3X3 כי בעיית הסיווג כבר לא בינארית. הטורים מייצגים את הסיווגים של הרשת והשורות את התיגו האמיתי. המטריצות עבור ההרצות:



1#

```
[[210  51  52]
 [116 298 130]
 [ 96 103  92]]
```

2#

```
[[228  38  47]
 [117 312 119]
 [110  93  84]]
```

3#

```
[[228  38  47]
 [117 312 119]
 [110  93  84]]
```

4#

```
[[229  42  71]
 [ 97 348 147]
 [100  97 114]]
```

5#

```
[[236  60  49]
 [105 402  92]
 [103 136  77]]
```

6#

```
[[225  67  54]
 [ 86 374 139]
 [ 90 134  92]]
```

7#

```
[[245  54  46]
 [123 353 123]
 [111 115  90]]
```

8#

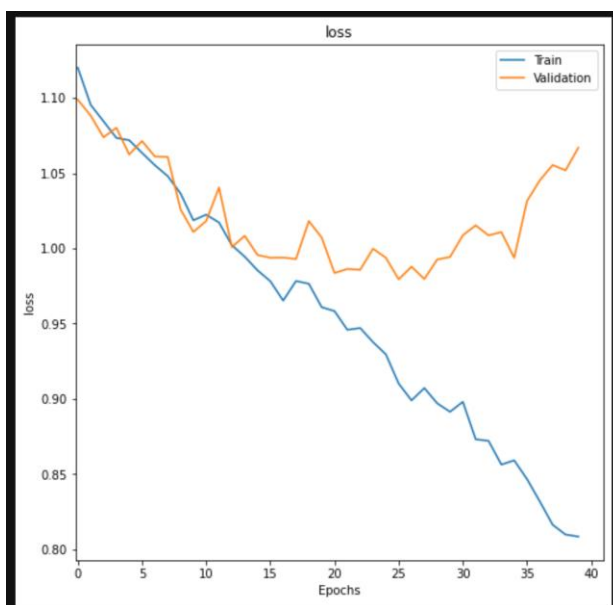
```
[[245  49  51]
 [125 352 121]
 [115 110  92]]
```

אחד הדברים העיקריים שניתן לראות מהתבוננות במטריצות זה שהחיזוי 2 הוא קשה לרשת ביותר ברוב המקרים. מצב זה נלקח בחשבון לפני תחילת העבודה, שכן סיווג שמורכב משני סיווגים שונים (דלקת ריאות לא טיפוסית ולא ניתן לקבוע) יהיה ככל הנראה בעייתי למציאת יצוג מיטבי על ידי הקונבולוציה וללמידה. המחלקה של "דלקת ריאות טיפוסית" מוצלחת יותר וקל יותר היה ללמוד את מישור ההפרדה שלה לבין המחלקות האחרות. המחלקה אשר כל המודלים הצליחו בה הכי הרבה היא ה"שלילי לדלקת ריאות", כאשר הפספוסים (FN) היו דו ספרתיים בלבד בכל המודלים. ניתן להסביר זאת גם אינטואיטיבית, כיוון שלזהות אם יש או אין דלקת ריאות זו משימה פשוטה יותר, ותלויה בהאם יש אזורים לבנים בריאות בתמונת הדימות בלבד.

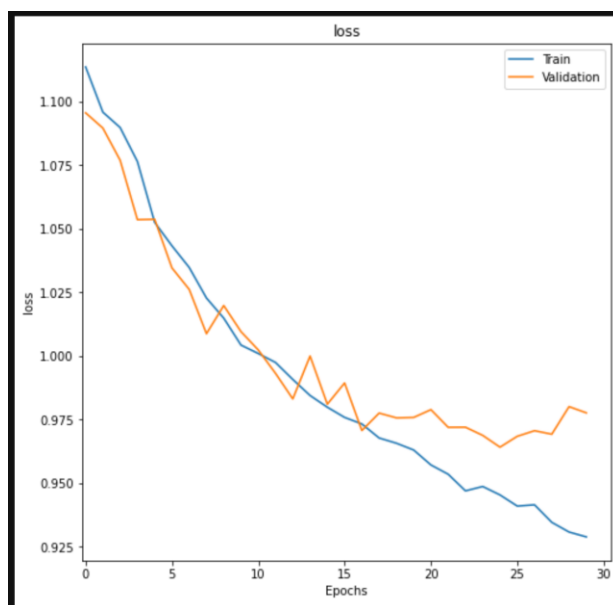
## הכללה

גרפים של עקומת הלימוד של שמונת המודלים:

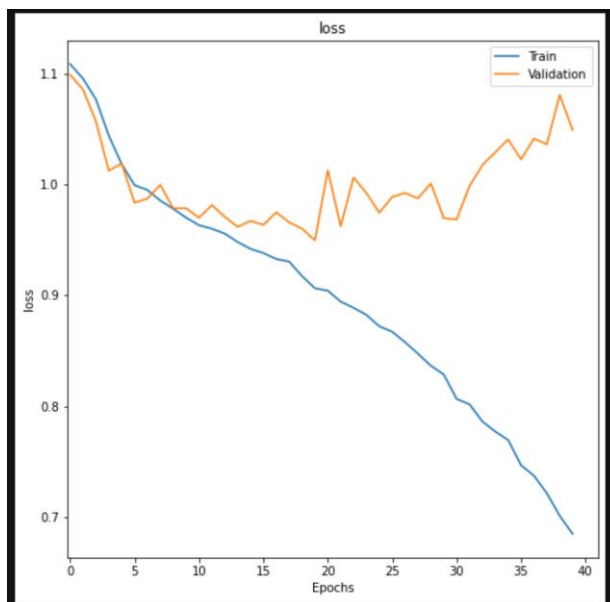
#1



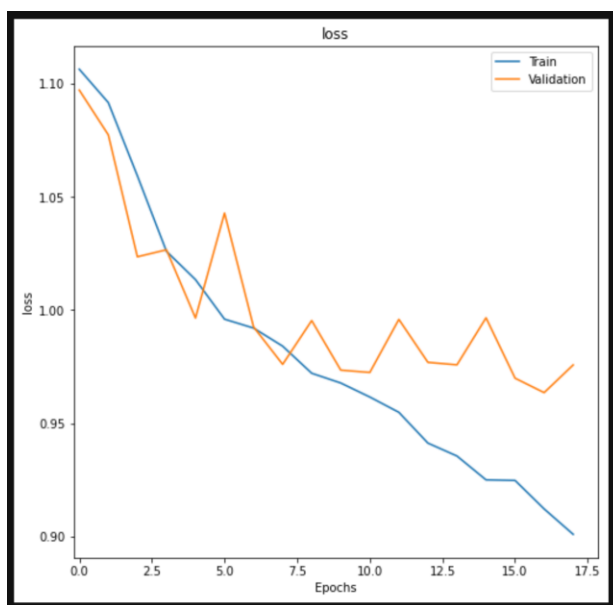
#2



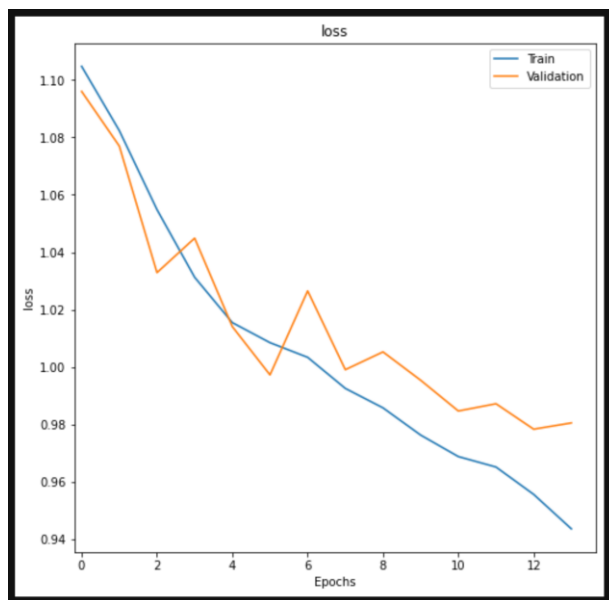
#3



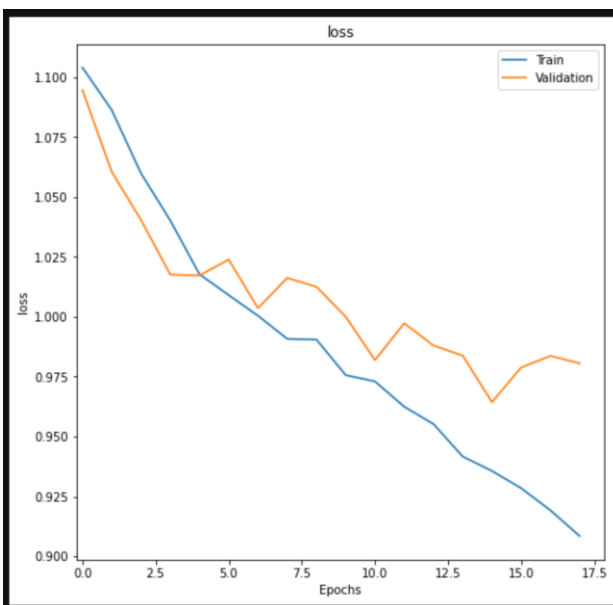
#4



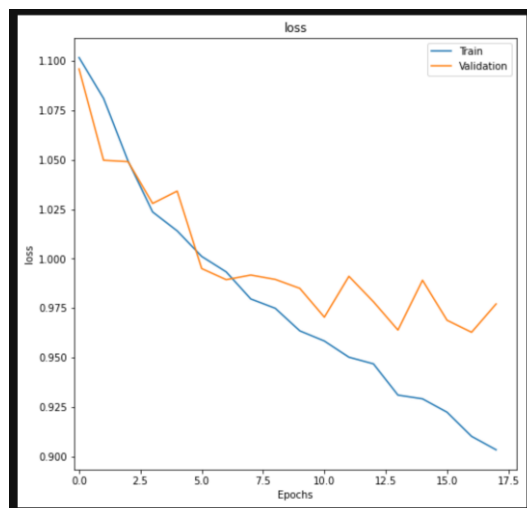
#5



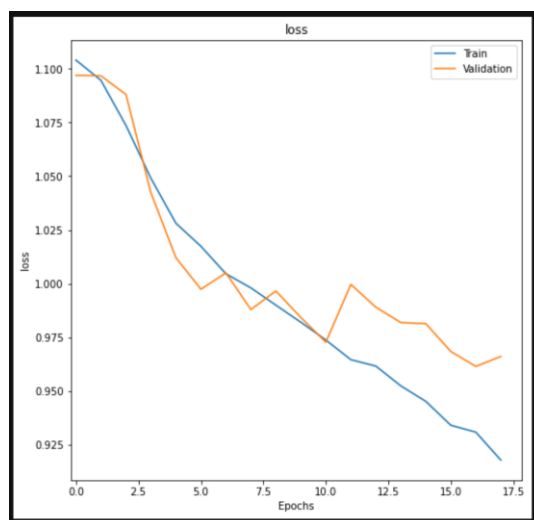
#6



#7



#8



נציין כי הופקו גם גרפים של אחוזי דיוק כתלות ב-epoch, אך כיוון שהמגמות שלהם דומות מאוד למגמות ה-loss בחרנו להביא את הנ"ל.

ניתן לראות כי בעיית התאמת-יתר קיימת בחלק מהרשתות, לדוגמה ברשתות מספר 1 ו-3 אנו רואים פער הכללה שהולך וגדל, מה שיכול להעיד על כך שהרשת על עשרות מיליוני פרמטרים הצליחה לשנן את התמונות בקבוצת האימון אך לא למדה כלום ולא הצליחה להכליל לעולם האמיתי. לדוגמה, ברשת #3, נפתור בעיה זו באמצעות Early stopping, שהוא כידוע סוג של רגולריזציה ונעצור את האימון לאחר 10 epochs בלבד. רשתות אחרות (#2, #5) התאפיינו בפער הכללה קטן למדי כאשר גם לאחר epochs רבים לא נצפה עליה דרמטית ב-loss בקבוצת האימון/המבחן. כדי לפתור בעיה זו מבעוד מועד ישנו Dropout ברנולי עם 50% לכיבוי משקולת בין שכבות

הנזירות, אך אולי זה לא מספיק והקונבולוציה היא זו שמספקת את יכולת השינון, וצריך בעתיד לנסות Dropout גם בשכבות הקונבולוציה.

## ROC

לאחר התלבטות האם להשתמש גם בעקומת ROC למדידת ביצועים הגענו למסקנה שבבעיית סיווג רב המחלקתית שלנו הדבר לא יועיל יותר מדי משתי סיבות: האחת, ROC קלאסי בבעיית סיווג בינארית מתייחס לכל ערכי הסף האפשריים להחלטה, ועל פי העקומה אפשר לבחור את ערך הסף אשר יספק את מירב ה-True Positive Rate וימזער את ה-False Positive Rate. במקרה שלנו החלטת הסיווג נקבעת על ידי Softmax ואין ערך סף קלאסי שניתן לבחור ולשנות בכך סיווגים ותוצאות (במקרה ושכבת הפלט בגודל 1 ופונקציית ההחלטה היא סיגמואיד לדוגמה). סיבה שניה היא שכדי לייצר ROC לבעיית סיווג רב מחלקתית צריך להפוך אותה לבינארית ובעצם לעשות ROC לכל מחלקה בנפרד בכל מודל. על מנת להעריך את טיב המודל בכללותו צריך לשקלל את כל העקומות וזה יכול קצת לחטוא את המטרה של להוות דרך פשוטה וחזותית להעריך טיב מודל. בנוסף, אם העקומה המתקבלת היא לינארית למקוטעין ובעלת שני קטעים ("שפיץ" אחד) אז הייתה התייחסות רק לערך סף אחד וזה מידע שאנו יכולים להפיק גם ממטריצת ה-confusion ואין סיבה לעשות ROC.

יתרון נוסף של ROC זה שהוא לא תלוי באיזון של התיוגים (כיוון שהצירים שלו הם RATE, יחס) אך גם כאן זה לא מעלה ולא מוריד כיוון שכבר טיפלנו בבעיית חוסר האיזון עם המישיקול מחדש של הדוגמאות, כמפורט בעמוד 4.

באופן כללי, כאשר התבוננו על ביצועי הרשתות לעומת ביצועי הפרספטרון ניתן לראות הפרש גדול. הסיבה לכך היא שהבעיה מורכבת מדי בשביל מסווג לינארי פשוט ושצריך הרכבה של פונקציות לא לינאריות על מנת למצוא הפרדה סבירה. בנוסף, הפרספטרון מקבל כקלט וקטור שטוח אשר מאבד את רוב תכונותיו המרחביות, תכונות אשר חשובות מאוד להבנה של מה נמצא בתמונה וביצירת ייצוג נכון שלו.

רשת #8 הייתה המוצלחת ביותר בגלל מספר סיבות. ישנו הדיוק, אשר היה 56% על קבוצת האימון כאשר זה גדול שווה לכל ביצוע אחר של רשת. סיבה נוספת היא שהוא הכליל בצורה לא רעה ולא הגיע להתאמת-יתר כמו רשתות אחרות. ובפן שלא נוגע לביצועים עצמם אך חשוב גם כן, הרשת הרבה יותר רזה מהרשתות האחרות מבחינת מספר המשקולות שלה וכתוצאה מכך האלגוריתם רץ בזמן מהיר הרבה יותר.

## דיון

בהתבסס על עבודתנו ובעיקר על הביצועים הנמוכים יחסית, ניתן להסיק שבעיית סיווג זו היא קשה למדי. גם רשת ידועה שהצליחה מאוד על מאגר נתוני תמונות התקשתה מאוד, וקל וחומר שדבר דומה יקרה גם באלגוריתמי למידה לא עמוקה. ראיה לכך שהבעיה היא קשה ניתן לראות בכך שגם רדיולוגים מומחים לעיתים לא מסווגים נכונה דלקת ריאות שנגרמת מקורונה.

עבודה נוספת שנדרשת היא בעיקר מעבר יותר שיטתי על ההיפר פרמטרים בעזרת Grid search, וזה גם מה שהיינו עושים בהנחת יותר זמן. אם מחליטים לא להתבסס על AlexNet כרשת המקור, ניתן להשתמש ברשתות ידועות אחרות אשר הניבו ביצועים טובים על מאגרים גדולים של תמונות רפואיות וצילומי רנטגן או CT.

יש להניח כי גם עם המשך המגפה ניתן יהיה להגדיל את המאגר עוד וכך לתת בסיס מוצק יותר ללמידה של רשתות גדולות שמסוגלות לשנן ולהכנס להתאמת-יתר בקלות עם מאגרי נתונים יחסית קטנים כמו שלנו.

המסקנה העיקרית שיצאנו ממנה היא הקושי של הבעיה הנתונה. מבחינת תובנות על רשת, חזינו במו עינינו כי ישנה יתירות רבה בפרמטרים של רשתות, וכי רשתות רזות בהרבה יכולות לתפקד באופן דומה מבחינת איכות הלמידה והביצועים.

## ביבליוגרפיה

- [1] <https://www.kaggle.com/shanmukh05/siim-covid19-dataset-256px-jpg>
- [2] Qu, Y. M., Kang, E. M., & Cong, H. Y. (2020). Positive result of Sars-Cov-2 in sputum from a cured patient with COVID-19. *Travel medicine and infectious disease*, 34, 101619.
- [3] Sohrabi, C. Alsafib. Z., O'Neill, N., Khanb, M., Kerwanc, A., Al-Jabirc, A., Iosifidis, C., & Aghad, 71-76.
- [4] Chan, J. F., Lau, S. K., To, K. K., Cheng, V. C., Woo, P. C., & Yuen, K. Y. (2015). Middle East respiratory syndrome coronavirus: another zoonotic betacoronavirus causing SARS-like disease. *Clinical microbiology reviews*, 28(2), 465-522.
- [5] Hani, C., Trieu, N. H., Saab, I., Dangeard, S., Bennani, S., Chassagnon, G., & Revel, M. P. (2020). COVID-19 pneumonia: a review of typical CT findings and differential diagnosis. *Diagnostic and interventional imaging*, 101(5), 263-268.
- [6] Xiong, F., Wang, Y., You, T., han Li, H., ting Fu, T., Tan, H., ... & Jiang, Y. (2021). The clinical classification of patients with COVID-19 pneumonia was predicted by Radiomics using chest CT. *Medicine*, 100(12), e25307.



[7] Li, X., Tan, W., Liu, P., Zhou, Q., & Yang, J. (2021). Classification of COVID-19 Chest CT Images Based on Ensemble Deep Learning. *Journal of Healthcare Engineering*, 2021.

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.